Spatio-Temporal Deep Learning for Medical Image Sequences

Vom Promotionsausschuss der Technischen Universität Hamburg

zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

von
Marcel Bengs

aus
Neumünster

2023

1. Gutachter: Prof. Dr.-Ing. Alexander Schlaefer
2. Gutachter: Prof. Dr.-Ing. Rolf-Rainer Grigat

Tag der mündlichen Prüfung: 10.11.2023

Acknowledgments

I could not have taken this journey without the valuable support and feedback of numerous other people. I would like to thank my advisor, Prof. Dr.-Ing. Alexander Schlaefer. I am very thankful for his guidance, numerous insightful discussions, and the opportunity to work on various scientific projects. Moreover, thanks go to Prof. Dr.-Ing. Rolf-Rainer Grigat for his valuable support and feedback. Thanks should also go to my great colleagues, Sarah Latus, Sarah Grube, Johanna Sprenger, Martin Gromniak, Finn Behrendt, Robin Mieling, Matthias Schlüter, Debayan Bhattacharya, Stefan Gerlach, Lennart Holstein, and Maximilian Neidhardt for the support and valuable discussions. Special thanks go to Maximilian Neidhardt for the insightful discussions regarding the impact of deep learning on elastography and how these fields can be connected. Furthermore, I would like to express my deepest gratitude to my colleague and close friend, Nils Gessert, for years of friendship, insightful research projects, and inspirational conference trips. Our brainstorming sessions were not only the most productive but also the most enjoyable. Thanks also go to Katrin Rausch, Thore Saathoff, Martin Fischer, and Michael Freude, who helped me whenever there were any technical or organizational difficulties. Many thanks also go to my research collaborators Julia Krüger, Dennis Eggert, Michael Bockmayr, and Roland Opfer. Major thanks should also go to my close friend Felix Gessert for his support and encouraging feedback during challenging times.
I would like to express my deepest appreciation to my parents and my sister for always being supportive and believing in me. Words cannot express my gratitude to my wife for her endless support and patience. Without her, I would never have had the endurance to finish this work. Finally, I would like to thank my family and friends.

Abstract

Observing and analyzing changes over time are at the heart of many clinical applications such as motion analysis, disease progression, or elastography. To this end, different medical imaging modalities can be used that allow for acquiring sequences of images at different spatial and temporal resolutions. Recently, fast volumetric imaging modalities have also gained traction, which allow capturing a dynamic process in its entirety both spatially and temporally. However, analyzing such spatio-temporal data and addressing different clinical problems remains challenging and typically requires several steps, inducing manual feature extraction and model building. Recently, deep learning methods have shown promising results for processing medical image data in various clinical applications, including first pioneering results for processing high-dimensional and multi-dimensional medical image data. These methods bring the advantage of end-to-end data processing. In particular, processing entire sequences of volumetric images end-to-end with deep learning is an interesting direction to improve and simplify otherwise complex data processing pipelines. We study and present spatio-temporal deep learning methods for analyzing medical image sequences. Spatio-temporal feature learning with deep learning is a difficult task with several architecture choices to learn effectively and efficiently from sequences of image data. We study whether complex spatio-temporal relationships can be learned effectively directly from sequences of medical image data, including sequences of volumetric images. We systematically study network types and operations, how the temporal information can be processed, how the complex spatio-temporal relationships can be learned, and how short-term and long-term image sequences can be processed. To this end, we present and study a compact and unified spatio-temporal deep learning concept for analyzing medical image sequences across a range of experiments. We combine this concept with different methods for end-to-end spatio-temporal feature learning ranging from pair-wise image processing to entire long-term 4D spatio-temporal data processing. We present architecture concepts such as multi-path Siamese networks, hybrid approaches of convolutional neural networks, and recurrent neural networks at different scales. We focus on two application scenarios, motion analysis and dynamic elastography, using optical coherence tomography and ultrasound as imaging modalities. Our results show that with our approach, high performance with inference times in the range of a few milliseconds can be achieved, outperforming previous approaches. Our results regarding architectures and concepts for learning from medical image sequences indicate that promising results can be achieved by using entire image sequences and learning joint spatio-temporal features in an end-to-end fashion. Our results highlight that 3D/4D spatio-temporal convolutions and convolutional-recurrent modules at different feature scales are well suited to this end. Overall, our findings demonstrate that spatio-temporal deep learning can effectively address end-to-end processing of sequences of medical image data, including sequences of volumetric images.

Contents

Chapter 1

Introduction

1.1 Sequences of Medical Image Data

Medical imaging advancements have changed patient care and treatment dramatically [135, 250]. In particular, sequences of medical image data for recognizing and analyzing dynamic processes over time have become an essential part. Within the field of medical imaging, different modalities have been considered to this end, including ultrasound (US), optical coherence tomography (OCT), functional magnetic resonance imaging (fMRI), or computed tomography (CT). These modalities allow to quickly acquire images over time and come with varying field of views as well as different spatial and temporal resolutions. Data acquired over space and time is typically referred to as spatio-temporal data, indicating information from space and time [2]. Recent advancements in image acquisition even allow for capturing sequences of volumetric images (3D+t) in real-time. This leads to 4D spatio-temporal data and allows for observation of dynamic processes in their entirety, both spatially and temporally. For example, such fast volumetric imaging can be achieved with US [361], or OCT [496]. These modalities also bring the advantage of being non-invasive and non-ionizing. Thereby, dynamic processes during clinical applications can be imaged and visualized with minimal impact on the patient. This makes sequences of medical image data, particularly non-invasive and non-ionizing modalities, a valuable tool for image-guided interventions and computer-assisted diagnosis. Examples include determining organ size, shape, or position as a function of time or functional analysis, i.e., behavior or activity of organs over time [430]. More specifically, sequences of images can be used to obtain information on surgical tool poses or tissue motion and location over time to adapt treatments accordingly, e.g., during radiation therapy [46], or minimally invasive surgery [55]. Sequences of images can also be used to visualize tissue displacement in response to a defined excitation force, which then allows for conclusions regarding tissue properties such as elasticity [7, 156, 389]. Such a technique is commonly referred to as elastography and can be used, e.g., for discriminating different liver fibrosis stages [148, 309], or for early tumor, detection [306, 540]. Moreover, sequences of images can also be used for functional analysis of the brain over time, e.g., to study Alzheimer's disease [105], or autism spectrum disorder [356].

Whereas sequences of medical image data allow to observe and analyze various dynamic processes, addressing the underlying clinical problems after image acquisition typically requires several steps and is a challenging task. Visual inspection of such multi-dimensional and rich data is difficult for medical experts, and semi-automated or fully-automated assessment pipelines are required to assist the clinicians. Traditional processing pipelines for medical image data analysis include pre-processing, feature selection, model selection, and post-processing [281]. Each of these steps is a challenging problem, can attribute as a potential source of error, and requires several assumptions [14]. In particular, defining and extracting discriminant features from images requires substantial domain knowledge and task-specific adaptation performed by human experts [281]. Combined with the hardware requirements for processing high-dimensional 3D (2D+t) and 4D (3D+t) spatio-temporal data, this requires notable adaption and manual tuning of data processing pipelines and often results in substantial computation times up to several seconds [98, 341]. This becomes critical in application scenarios where real-time processing is required, e.g., to quickly compensate a patient's motion to ensure the efficacy and safety during a treatment [46]. As a result, different clinical problems and imaging modalities typically require highly specialized processing pipelines, and addressing clinical problems with sequences of medical image data remains an ongoing research problem. Thus, a flexible and efficient end-to-end data processing approach for medical image sequences could overcome many of the aforementioned challenges. Learning and utilizing spatio-temporal features directly from sequences of image data without the requirement of handcrafting features and subsequent model building could not only simplify the data processing process but could also improve the performance by using more abstract and robust spatio-temporal features that are beyond human interpretation [29]. Deep learning has recently shown promising results for efficient end-to-end processing of medical image data in various clinical applications, including first pioneering results for processing high-dimensional and multi-dimensional medical image data [14, 33, 162, 281].

1.2 Deep Learning for Medical Image Analysis

In the last years, deep learning has shown promising results for numerous computer vision tasks across a wide range of industries and applications [473]. Deep learning for medical image analysis has grown rapidly since 2015 [281] and has demonstrated numerous success stories for various tasks [312], such as image classification [60, 139], image segmentation [285], image reconstruction [528], and object detection [23]. Hardware advancements, as well as the availability of open source software packages such as Tensorflow [1], or PyTorch [349] contributed to the growing interest and to this large number of success stories [281].

A fundamental idea of deep learning is to learn end-to-end relationships directly between data and the task at hand. Such an approach is also referred to as representation learning, meaning that the patterns and feature representations needed for a task are discovered automatically from data [29, 258]. Thereby, traditional

machine learning pipelines have been disrupted, and steps such as designing and selecting handcrafted features have been challenged [9, 414]. An advantage that results from the end-to-end processing of the data with deep learning is that the entire data processing pipeline can be simplified, and also, it can reduce performance limitations due to incorrect assumptions made during feature and model selection [258]. The automatic feature learning process can be achieved by combining concepts of computer science, statistics, and optimization [314]. A famous deep learning approach is a convolutional neural network (CNN) [259, 260] that has been designed and developed to learn discriminative features from annotated image datasets in an end-to-end fashion, e.g., to perform image classification [368].

In the last years, a plethora of medical image analysis tasks have been studied with deep learning, e.g., a recent survey paper included over 300 publications to consider the variety of methods and clinical applications [281]. Recently, it has been found that the diagnostic performance of deep learning using medical image data can already be on par with medical experts for various diseases, e.g., breast cancer, dermatological cancer, lung cancer, or thyroid cancer [286]. Furthermore, encouraging results have also been demonstrated for the analysis of medical image sequences for applications such as fetal heart analysis using 2D US videos [350], surgical tool tracking in 2D endoscopic videos [538], characterization of parkinsonian gait from 2D videos [185], cardiac left ventricle quantification using sequences of 2D magnetic resonance imaging (MRI) data [467], or tissue landmark tracking using 2D US videos [98]. Moreover, first promising results have also been achieved for efficient high-dimensional, and multi-dimensional data processing, e.g., for 4D fMRI data [273, 537], or for 4D CT processing [87, 263, 264, 320].
Overall, the recent success stories make deep learning a promising approach to analyzing medical image data, including sequences of medical image data. However, several open challenges regarding deep learning and medical image analysis still need to be addressed, including many aspects related to data and deep learning architectures [14, 281, 410, 435, 475]. Especially analyzing high-dimensional and multi-dimensional medical image data, such as sequences of images, brings up many unsolved questions. Ultimately, this leads to the research questions of this work.

1.3 Research Questions

In some cases, concepts from the natural image domain can directly be transferred to address medical image analysis tasks, e.g., for 2D image classification [35,41,165,281]. Here, the task is similar to those widely studied in the natural image domain, and successful deep learning concepts can readily be used and transferred with minimal adaption while achieving impressive performance [139]. However, the medical imaging domain contains many other challenges that are usually not present in the natural image domain, including the analysis of volumetric image data, anisotropic voxel sizes, the absence of large-scale datasets, and the requirement of expert knowledge for data annotation [281]. As a result, general concepts can be transferred to medical image analysis, however, architecture

design and development remains a persistent problem [14, 162, 281, 410, 435, 475]. In particular, analyzing entire sequences of medical images combines and amplifies these challenges due to the rich and complex data structure.

Until recently, before the availability of increased computational hardware and large-scale public datasets, deep learning demonstrated limited performance regarding video analysis tasks in the natural image domain [69, 192, 231]. Today, many deep learning methods have been presented to leverage information from image sequences. These methods can also be referred to as spatio-temporal deep learning, indicating that spatio-temporal data is processed. In this context, a typical application scenario is video classification [231], e.g., to perform human action recognition [438]. However, despite the substantial efforts, spatio-temporal feature learning with deep learning remains a difficult task with several choices regarding the architecture and the data to learn effectively and efficiently from sequences of images [503]. Similar, it has been pointed out that the spatial and temporal nature of videos requires tailored deep learning approaches to process such rich and complex data [369]. Analyzing sequences of images is not only challenging due to the substantial memory requirements but also due to the high-dimensionality of the resulting feature space. In such a scenario, it quickly becomes difficult to represent possible feature configurations with data, commonly referred to as the curse of dimensionality [176]. Considering medical image sequences, some modalities even allow for sequences of volumetric images, as outlined previously, making this problem even more severe.
Compared to processing a single image, processing entire sequences of images also brings several conceptual questions. Sequence information can be utilized using only image pairs, multiple images, or the entire image sequence. Using more than two images may promise improved performance and consistency but typically results in substantially increased computational requirements and notable run-times. In particular, the latter becomes problematic for clinical tasks such as motion analysis, where real-time processing is beneficial or even required [62, 132, 134, 360]. Also, using entire sequences increases the input dimension and the resulting feature space, which brings up the curse of dimensionality. As a result, analyzing sequences of medical images is often performed with specialized deep learning approaches, and typically only little temporal information or only parts of the entire available spatio-temporal data is considered [98, 127, 163, 168, 171, 175, 225, 254, 371, 464, 537]. In particular, processing entire sequence of volumetric images end-to-end with deep learning is an interesting direction for further research.

While deep learning removes the task of feature engineering, it brings up the task of network design engineering [527]. Searching for customized deep learning solutions explicitly tailored to each problem is neither desirable nor feasible, given the substantial computational requirements to train such approaches in the context of image sequence processing. Although similar approaches are typically used for 2D image classification, as outlined previously, very little work studies processing of medical image sequences with a similar concept across a range of applications

and modalities. Given the substantial challenges of medical image sequences, a spatio-temporal deep learning approach is typically developed for a particular task combined with a specific imaging modality. Thus, deep learning concepts for medical image sequence analysis typically differ substantially, e.g., comparing deep learning methods for motion analysis [47], or for dynamic elastography [3]. Notably, systematically studying a similar spatio-temporal deep learning concept for end-to-end representation learning across different applications and modalities is rarely considered. Overall, this leads to our two fundamental research questions of this work regarding spatio-temporal deep learning and medical image sequences.

- **Research Question 1:** *Can sequences of medical image data be processed effectively with spatio-temporal deep learning?*
- **Research Question 2:** *How can spatio-temporal feature learning be performed with deep learning?*

The first research question considers whether complex spatio-temporal relationships can be learned directly from sequences of medical image data, including sequences of volumetric images. We consider an approach effective when real-time constraints can be met, robust performance can be achieved, the task can be addressed end-to-end, and a similar concept can be shared across different tasks and imaging modalities. Moreover, we consider an approach effective when it actually simplifies otherwise complex and nested data processing pipelines.
The second research question focuses on the deep learning aspects. It includes which network types and operations can be used, how the temporal information can be processed, how the complex spatio-temporal relationships can be learned to achieve robust and high performance, and how short-term and long-term image sequences can be processed. Also, we address how to cope with the substantial computational requirements resulting from medical image sequences. We also focus on processing sequences of volumetric images end-to-end. Both of our research questions go hand in hand, the first focuses on efficacy for applications, and the second focuses on methodological aspects.
Addressing both research questions requires conceptualization of the entire learning tasks, from data collection and annotation to deep learning model design and development of training strategies. In this work, we address these aspects, focusing on two application scenarios, motion analysis, and dynamic elastography, using OCT and US as imaging modalities. To this end, we present and study a compact and unified spatio-temporal deep learning concept for the analysis of medical image sequences across a range of experiments.

1.4 Contribution and Outlook

In the following sections, we provide an outlook of our contributions regarding our application scenarios and spatio-temporal deep learning. Afterwards, we outline the structure of this thesis.

1.4.1 Processing Sequences of Medical Image Data

To study our research questions regarding spatio-temporal deep learning, we consider motion analysis and dynamic elastography using OCT and US as imaging modalities. Both motion analysis and dynamic elastography require processing sequences of images and analyzing complex spatio-temporal features of the imaged process. Motion analysis is a relevant task for the localization of instruments and targets [13, 55, 76], and for motion compensation during interventions, e.g., during radiotherapy [328, 334, 339]. In this application scenario, using sequences of volumetric images is preferable or even required due to the three-dimensional nature of targets and motion. Dynamic elastography is an approach that allows for quantitative estimation of tissue elasticity, i.e., stiffness of organs and tissues [7, 156, 389]. This is relevant, e.g., for diagnosis [182, 510] as well as surgical planning [67, 262, 351]. This can be achieved by inducing waves of displacements into tissue and by analyzing the resulting wave propagation using high-frequency imaging, i.e., sequences of images [415].

Motion Analysis. We demonstrate OCT-based object pose estimation and markerless tissue motion estimation using sequences of volumetric OCT images and spatio-temporal deep learning. Moreover, we study markerless motion analysis of tissue using long-term sequences of volumetric US images. We also show end-to-end motion forecasting directly from sequences of volumetric images. Despite using such multi-dimensional data, we demonstrate robust motion analysis in real-time with inference time in the range of a few milliseconds using our spatio-temporal deep learning approach.

Dynamic Elastography. We demonstrate that local elastic properties can be estimated directly from US shear wave data with spatio-temporal deep learning. Our spatio-temporal deep learning approach is designed to identify temporal patterns in an end-to-end fashion without any explicit feature extraction or physical model of the wave propagation. We show end-to-end elasticity estimation using sequences of 2D and 3D OCT image data. For both imaging modalities, we demonstrate that with our spatio-temporal deep learning approach, accurate estimation of elastic tissue properties can be performed with robust performance w.r.t. the measuring position relative to the wave excitation point.

1.4.2 Spatio-Temporal Deep Learning Approaches

In this work, we study and present several spatio-temporal deep learning approaches for medical image sequence processing, including long-term sequences of volumetric images. This includes the development of new architectures and training strategies. We present and compare methods that perform analysis of image sequences based on image pairs up to entire sequences ranging over several hundreds of images. We present and design 3D/4D spatio-temporal CNNs, multi-path Siamese CNNs, and hybrid approaches of CNNs and recurrent neural networks (RNNs). In this context, we also present an efficient 4D architecture to process long-term 4D image sequences in real-time. Also, we address the aspects

of training with such data that results in substantial computational requirements. To reduce the challenge of hand-crafting architectures for specific problems, we incorporated all these concepts into a compact and unified CNN architecture concept for end-to-end regression from sequences of medical image data. Our approach can be used to perform estimations for each image of a sequence or to perform a single estimation for an entire sequence. A conceptualized visualization of our approach is given in Figure 1.1.

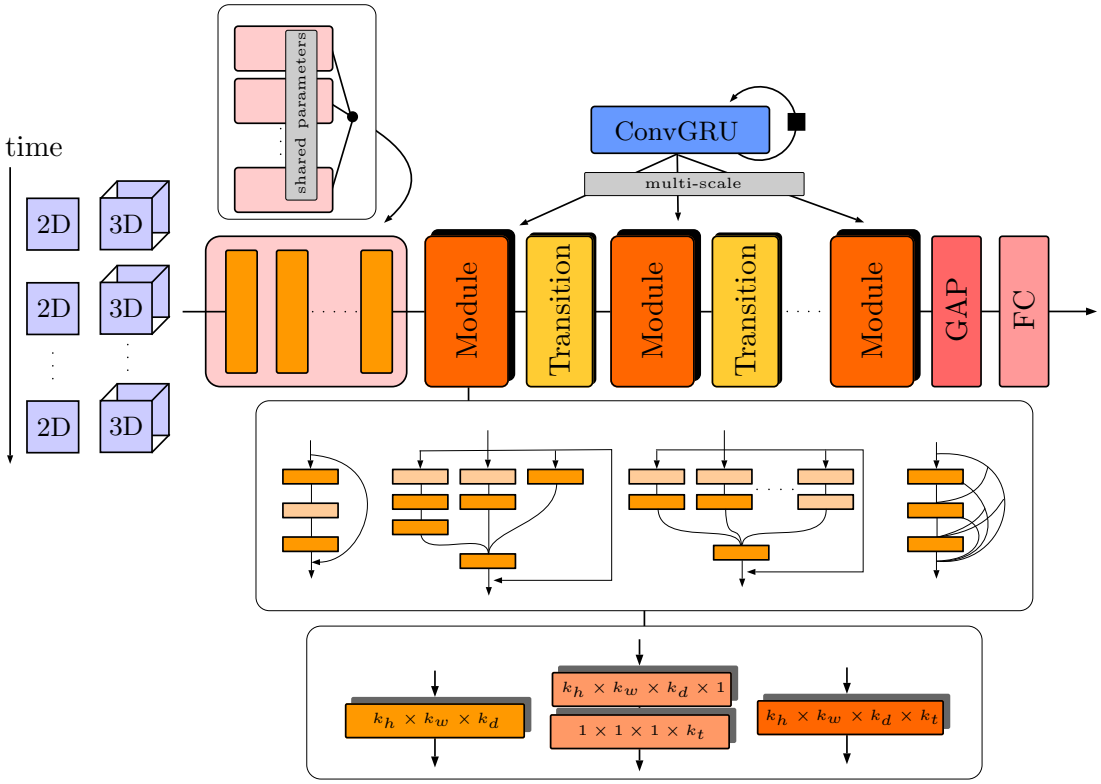

Figure 1.1: Our proposed spatio-temporal deep learning concept for end-to-end regression from sequences of 2D/3D images. We present and combine several spatio-temporal feature learning concepts into a compact and unified CNN concept. This includes multi-path Siamese CNNs with shared parameters, spatio-temporal CNNs in different fashions, and multi-scale spatio-temporal recurrence.

1.5 Organization

Chapter 2: Spatio-Temporal Medical Imaging In this chapter, we introduce how spatial changes over time can be observed with sequences of medical image data from a conceptual level and how such data can be acquired. We describe the principles of OCT and US ranging from 1D up to 4D imaging. Therein, we describe how image sequences can be acquired with both a high spatial and temporal resolution leading to highly resolved spatio-temporal data. Lastly, we introduce fundamental image processing concepts, which are closely related to concepts used for deep learning methods.

Chapter 3: Deep Learning In this chapter, we introduce the foundations of deep learning with a focus on supervised machine learning tasks. We explain the concepts of CNNs for image analysis and highlight the concept of RNNs for sequence analysis. In this context, we also explain how deep learning approaches can be optimized and evaluated, i.e., how learning from data can be performed.

Chapter 4: Spatio-Temporal Deep Learning Methods In this chapter, we describe and develop our deep learning methods for medical image sequence analysis. We build our approaches based on the two widely used architecture concepts for video analysis in the natural image domain, CNNs, and RNNs. We introduce the relevant background for each of our methods, and present our approaches and methodical developments.

Chapter 5: Application Scenarios and Related Work In this chapter, we describe the application scenarios that are considered for studying the two main research questions of this work in more detail. We highlight the background of our application scenarios, summarize related work, and demonstrate open challenges with a focus on OCT and US as imaging modalities.

Chapter 6: Experiments and Evaluations In this chapter, we present a range of experiments considering both motion analysis and dynamic elastography using sequences of OCT and US image data. For each application scenario, we highlight the challenges, how the setting affects the learning task, and how these can be addressed by our methods.

Chapter 7: Discussion In this chapter, we discuss our results and the findings of this work from a more general perspective in the context of our research questions also highlight interesting directions for future work.

Chapter 8: Conclusion In this chapter, we summarize the findings of our work and report the conclusion of this dissertation.

Chapter 2

Spatio-Temporal Medical Imaging

In this chapter, we introduce how spatial changes over time can be observed with sequences of medical image data from a conceptual level and how such data can be acquired. In this context, we introduce the concept of a spatio-temporal data representation. Next, we introduce two medical imaging modalities, US and OCT, which allow for non-invasive and non-ionizing spatio-temporal imaging. We describe the principles of these modalities ranging from 1D up to 4D imaging. We present how image sequences can be acquired with both a high spatial and temporal resolution leading to highly resolved spatio-temporal data. Lastly, we introduce fundamental image processing concepts, which are relevant for analyzing changes in image sequences and closely related to concepts used for deep learning methods.

2.1 Spatio-Temporal Data

In this section, we highlight the concept of analyzing dynamic processes with image sequence and motivate the concept of spatio-temporal data from a general perspective. This section is based on [222]. Typically, we associate dynamic processes with change over time, and analyzing such a process requires observations, i.e., data at multiple time points. Such data can be obtained with subsequent image acquisition over time, which yields an image sequence, $x_t = [x^{[1]}, x^{[2]}, \ldots, x^{[n_t]}]$. For 2D images, this results in $x^{[i]} \in \mathbb{R}^{n_h \times n_w \times n_c}$ where n_h and n_w denote the number of pixels along the height and width direction of an image, respectively. The number of color channels is given by n_c, and the sequence length is given by n_t. In our thesis, we also consider image sequences of volumetric images, this results in $x^{[i]} \in \mathbb{R}^{n_h \times n_w \times n_d \times n_c}$ with n_d for the number of voxels along the depths dimension. Throughout this work, we use superscripts with square parentheses to indicate the different time points similar to [176]. We focus on changes between images that result from motion, detecting other changes, such as appearance changes over time, is not part of this thesis.

Dynamic processes can lead to changes in the intensity values of an image. Similar, finding corresponding pixels or pixel formations from one image $x^{[1]}$ in another image $x^{[\tau]}$ at a time point τ can be used, e.g., to describe motion. However, a fundamental challenge is that typically corresponding points in two images can not be unambiguously determined. Physical correspondence might not lead to

identical visual correspondence in images, e.g., in the case of a physical motion where an object has no distinct visual landmarks. Visual correspondence can be present without actual physical correspondence, e.g., due to brightness changes that result from illumination changes. Fundamentally, it is difficult to directly relate actual dynamic changes such as motion to visual changes, especially only given two images.
While change can be estimated based on an image pair from two time points, it only provides snap-shot information of a dynamic process. Such snap-shot information makes it difficult to predict how a process might continue over time and makes analysis sensitive to noise and occlusion. This motivates using multiple images for analysis, which can be achieved by acquiring and analyzing an entire sequence of images with a high temporal frequency. Such an image sequence allows for a representation that spans over space and time, which leads to 3D spatio-temporal data $x_t \in \mathbb{R}^{n_h \times n_w \times n_t \times n_c}$ for sequences of 2D images, and to 4D spatio-temporal data $x_t \in \mathbb{R}^{n_h \times n_w \times n_d \times n_t \times n_c}$ for sequences of 3D images. Visualizing such data can also be referred to as space-time images, where the temporal axis is considered an additional image dimension. The concept of a spatio-temporal data representation is visualized in Figure 2.1.
One advantage of a spatio-temporal data representation is that spatio-temporal features and patterns can be present that can be used to analyze dynamic processes. For example, the velocity of an object constantly moving over time can be related to the orientation in space-time images as shown in Figure 2.1. In particular, assuming a fast temporal acquisition rate relative to the observed dynamics, i.e., the Nyquist-Shannon sampling theorem [406] is fulfilled, continuous features and patterns in space-time images can be expected. That is, the sampling frequency is at least twice the highest frequency contained in the signal. Although spatio-temporal data has appealing advantages compared to only using two images, note that it increases the size of the data and thus makes data analysis more computationally complex and time-consuming.
In summary, dynamic processes such as motion can be captured with image sequences and can be analyzed by considering spatial and temporal changes in the pixel intensity values. Moreover, acquiring an entire sequence of images results in spatio-temporal data that encodes features about the imaged process. However, addressing a clinical task requires an effective imaging modality and a robust method for data analysis. In the following sections, we describe two imaging modalities that can perform volumetric imaging with a high temporal resolution: US and OCT. Afterwards, we focus on the data processing aspect using deep learning methods.

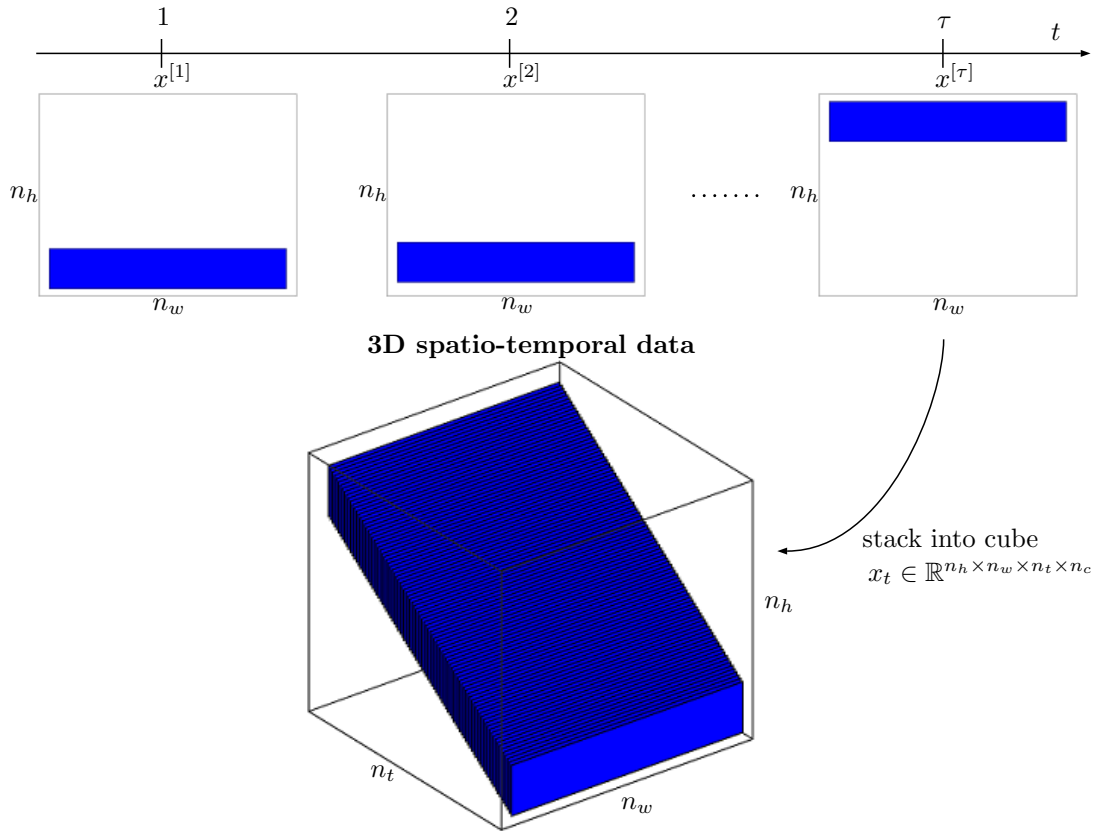

Figure 2.1: An image sequence of a blue rectangle that moves with a constant velocity into the vertical direction. (Top) Images of the blue rectangle at different time points. (Bottom) 3D space-time representation of the sequence where the temporal axis is considered as an additional data dimension. The space-time representation of x_t demonstrates that the movement with constant velocity leads to a constant orientation in the space-time image. Figure based on [2].

2.2 Image Sequence Acquisition

2.2.1 Medical Ultrasound Imaging

In this section, we describe the background of how spatio-temporal data can be acquired with US imaging. US is a non-invasive and non-ionizing imaging modality with a high-temporal resolution, allowing 1D up to 4D data acquisition in real-time [217]. In general, US has a long history in medical imaging [361], and today it is widely established with various applications such as breast cancer screening [456] or cardiac examinations [10].

The following paragraph is based on [434]. The principle of ultrasound imaging is based on sending ultrasound pulses modulated with a chosen center frequency and recording the echo time and amplitude of the reflected signal afterwards. In this way, different reflection properties of tissue can be measured, and a corresponding image can be generated. Ultrasound wave generation and detection are

typically performed with piezoelectric crystals that turn electrical energy into mechanical energy and vice versa. For example, when a sinusoidal electrical signal is applied to a piezoelectric crystal, the surface of the piezoelectric crystal vibrates with the same frequency, and compression waves are transmitted. Similar, when a compression wave reaches the piezoelectric crystal, a corresponding electrical signal is generated. This property makes piezoelectric crystals suitable for transmitting and receiving during ultrasound imaging. For a detailed explanation of piezoelectric crystals, we refer the interested reader to [199].
The device containing piezoelectric crystals is typically called a transducer. Different spatial arrangements of the piezoelectric crystals exist, e.g., the piezoelectric crystals can be arranged in a one-dimensional or two-dimensional pattern across a straight or convex surface. In particular, to address the different needs during clinical practice, various different transducer types have been developed. An overview of different transducer types w.r.t. application scenarios are given, e.g., in [440].
Ultrasound imaging is based on progressive longitudinal compression waves, i.e., the displacement is parallel to the propagation direction. The frequencies of the waves are typically higher than 20 kHz. When sound waves are transmitted into a tissue, several acoustic phenomena occur, such as reflection, transmission, and attenuation. These acoustic phenomena occur when a wave propagates from one medium to another with varying acoustic impedances. For plane progressing waves, the acoustic impedance is defined by

$$Z = \rho \cdot v_a \tag{2.1}$$

with ρ for the mass density and v_a for the acoustic wave velocity. Acoustic wave velocities in human tissue are in the range of $v_a = 1540\,\mathrm{m\,s^{-1}}$ close to the acoustic velocity in water $v_a = 1493\,\mathrm{m\,s^{-1}}$. In contrast, the acoustic velocity in bone is much higher, i.e., in the range of $v_a = 4000\,\mathrm{m\,s^{-1}}$. It is important to note that most of the wave is reflected when there is a great difference in the impedance of the two media and that most of the wave is transmitted when there is a small difference in the impedances. That is, given two media with high variance in impedance, most of the wave is reflected, and only a few parts of the wave are transmitted. Thus, it becomes difficult to image subsequent tissue structures. Commonly this is referred to as shadowing artifacts. The transmitted part of the wave can be different in direction compared to the incident wave depending on the acoustic velocities of the two media and hence is also called a refracted wave. Similar, also the amplitude of the transmitted wave might change compared to the incident wave. Lastly, a propagating sound wave is affected by energy loss over the propagation distance, typically called attenuation. For medical ultrasound imaging, it can be shown that attenuation is directly related to the wave frequency, where lower frequencies yield lower attenuation [68, 88]. Hence, to increase the penetration depths during imaging, lower frequencies are required.

The following paragraph is based on [101, 190]. In order to perform the imaging process, several ultrasound beam forming and receive techniques exist with individual advantages and disadvantages w.r.t. frame rate and image quality.

A fundamental assumption of ultrasound imaging is a constant sound velocity throughout the tissue, which allows direct conversion of time to distance. Thereby, geometric assignment of the received signal over time can be performed, and depth scans of the acoustic properties can be acquired.
One approach is linear beam forming, where imaging is based on depth scans acquired line-by-line. To this end, multiple piezoelectric elements of a transducer are grouped as a sub-aperture and are used to transmit and receive the ultrasound signal, see Figure 2.2. A larger subset of elements can also be used for the receive phase.

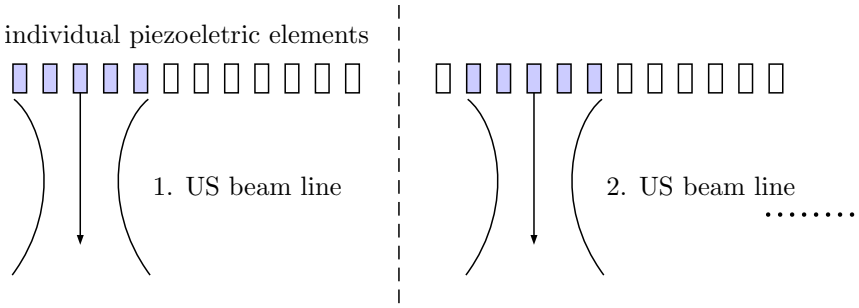

Figure 2.2: Electronic beam scanning for a linear-array transducer. Multiple piezoelectric elements of a transducer are grouped and are used to transmit and receive the ultrasound signal. By shifting the sub-aperture, imaging can be performed line-by-line. Figure based on [190].

By shifting the sub-aperture over the transducer array, multiple scans can be acquired, line-by-line, and thereby a two-dimensional scan can be performed in a raster pattern, see Figure 2.3. Notably, the piezoelectric elements of a transducer can also be arranged in a two-dimensional pattern, as mentioned previously, then the sub-aperture can be shifted across two lateral directions, and volumetric imaging can be performed. A single depths scan is typically called an A-scans, and a two-dimensional and three-dimensional scan is called B-scan or C-scan, respectively [88]. Another approach is phased array beam forming, where the entire array of the transducer is used as an aperture. Here, the signals exciting the piezoelectric elements are temporally shifted to steer the beam with a specific angle relative to the direction normal to the array aperture. By steering the beam with different angles, multiple A-scans can be acquired, line by line, to obtain a B-scan or C-scan.

The following paragraph is based on [101, 190]. To improve the temporal resolution of the image acquisition, the piezoelectric elements during a transmission event can be exited all at once or temporally shifted to generate an unfocused or focused beam, respectively. A larger area can be imaged at once with an unfocused beam, thus improving the temporal resolution compared to the focused case, which might require multiple transmissions for the same area. However, focusing can improve the penetration depths and image quality compared to the unfocused case due to higher pressures and smaller beams resulting from focused beams.

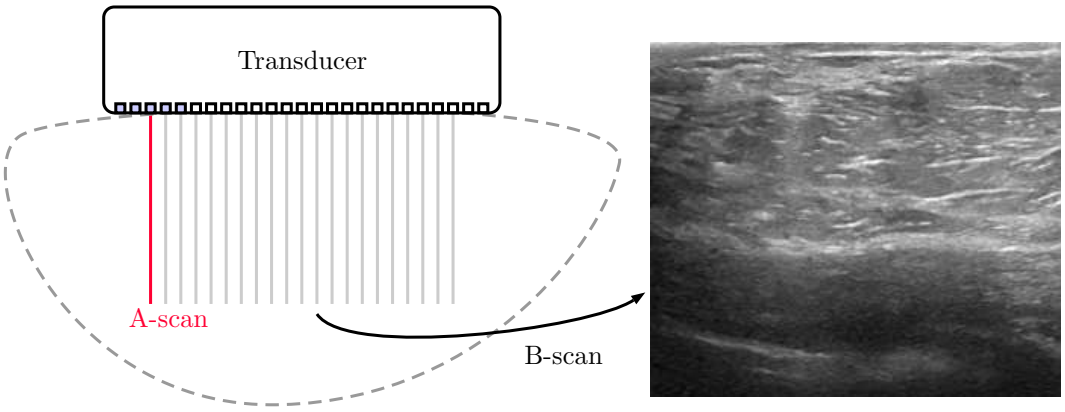

Figure 2.3: Example process of ultrasound imaging with linear beam forming. A US image is generated line-by-line by acquiring subsequent A-scans. US image data taken from public Breast Ultrasound Images Dataset [5]. Figure based on [190].

The receive phase of the imaging process can further be defined by how the received signals are combined. In particular, multiple A-scans can be reconstructed in parallel for a single transmission event to improve the temporal resolution. This approach is called multi-line acquisition and can be achieved by assuming multiple, narrower beams during receive and summing and delaying the received signals of the piezoelectric elements accordingly [101]. An efficient and widespread technique is delay-and-sum (DAS) [353]. While multi-line acquisition improves the temporal resolution compared to single-line acquisition, typically, this approach requires a wider beam during transmission, reducing the lateral resolution. Moreover, also multiple beams can be used in the transmission phase to improve the temporal resolution, called multi-line transmission beam forming [101, 102].
In general, the recorded signal at the piezoelectric elements is typically referred to as radio-frequency data (RF-data). After post-processing, the received amplitude signals are visualized as grayscale values, and corresponding 2D brightness image or 3D brightness image volume can be generated. Post-processing steps typically include filtering and envelope detection of the high-frequency RF-data and attenuation correction, log-compression, and scan conversion [434].

The following paragraph is based on [101]. To maximize the temporal resolution, plane wave imaging has been proposed, where only a single transmission is used for image generation [101,385–387]. Such an approach is commonly used for shear wave imaging [267,268,415,449,450]. Here, only a single wide and homogeneous beam is used during transmission by exiting all piezoelectric elements of the transducer with the same signal. Afterwards, multi-line acquisition is performed for the entire image during receive, i.e., all lines forming the entire image are constructed in parallel. Notably, the temporal resolution of this approach is only limited by the sound velocity of the media that is imaged and the imaging depths, as well as the processing time for image reconstruction. While this approach maximizes the temporal resolution, it comes at the cost of reduced spatial resolution and contrast [101,385–387]. To counteract this problem, plane wave image com-

pounding can be used [315]. Here, multiple images acquired with transmissions steered at varying angles are averaged. Naturally, such an approach reduces the frame rate by the number of transmission angles, but the image quality can be improved, and the trade-off can be adjusted.

The following paragraph is based on [300]. The spatial resolution of ultrasound imaging is mostly defined by the properties of the transducer. The resolution along the direction of wave propagation typically referred to as axial resolution, is determined by the transmitted pulse length. The transmitted pulse length is defined by the number of cycles and the wavelength

$$\lambda = \frac{v_a}{f_q} \tag{2.2}$$

of the pulse with v_a for the wave velocity f_q for the frequency. This demonstrates that the resolution improves with increasing frequency. However, recall that the penetration depth decreases with increasing frequency due to attenuation. Hence, ultrasound imaging suffers from a trade-off between penetration depths and spatial resolution. Considering example values for diagnostic ultrasound, for frequencies in the range of $f_q = 2\,\mathrm{MH}z$ to $f_q = 15\,\mathrm{MH}z$, the penetration depths are in the range of 3 cm up to 25 cm with an axial resolution in the range of 0.15 mm and 0.8 mm respectively [391]. The resolution perpendicular to the axial direction, typically referred to as lateral resolution, is mostly dependent on the wavelength and the width of the transmitted beam and is typically an order of magnitude worse than the resolution in the axial direction. Lastly, as outlined before, the temporal resolution depends on the ultrasound beam forming and the receive technique. For example, with plane wave imaging, frame rates up to $\sim 15\,\mathrm{kHz}$ can be achieved [44]. Notably, using such high frame rates during imaging also requires sufficient hardware for processing, which becomes feasible through the recent developments of GPU technologies [101, 422]. For linear imaging, the frame rate depends on the number of A-scans, the imaging depths, and the acoustic wave velocity.

The following paragraph is based on [95, 434]. Recall that the received amplitudes are dependent on the impedance differences of the tissue. While tissue is typically inhomogeneous with local deviations of density and compressibility, reflections are not only present at tissue boundaries but also at small inhomogeneities inside and outside of the desired sample/image volume. This results in constructive and destructive interference of the backscattered waves that impact the imaging. These scatter reflections appear as a random granular pattern in the US image and are usually considered scatter noise. Notably, under the exact same imaging circumstances, two images show the exact same speckle noise, i.e., speckle can be considered a deterministic artifact [59]. That is, while these scatter reflections can be considered as a source of noise that contributes to the signal, they are dependent on the tissue structure. They hence can also be considered a source of information that allows distinguishing different tissue types [94]. However, speckle noise notably impacts the image quality and thus impacts human interpretation [373]. Thus, several speckle reduction techniques have been

developed for image enhancement, including deep learning methods [230, 345]. Moreover, while a constant acoustic wave velocity is assumed throughout the tissue for image generation, this assumption rarely holds in practice, and the ultrasound beam travels with different velocities through different tissue layers. Therefore, when assuming a constant velocity for image generation, geometrical correctness can not be assumed.

In summary, US is a non-invasive imaging modality that allows for volumetric imaging with high spatial and temporal resolution, and image generation is performed line-by-line, but also, an entire image can be acquired at once using plane-wave imaging. This imaging modality relies on backreflected sound waves from internal structures of tissue that are measured and visualized. Repeating this imaging procedure allows for spatio-temporal data acquisition of up to 4D data.

2.2.2 Optical Coherence Tomography

In this section, we describe the background of OCT from 1D imaging to 3D imaging over time, which is relevant to our work. In general, OCT is a non-invasive imaging modality based on optical backscattering or backreflection of light and was first introduced in 1991 [209]. In recent years, OCT has been adopted for various clinical applications such as ophthalmology [310, 380], cardiology [370, 470], endoscopy [245, 516], dermatology [91], and oncology [150, 476]. This section is based on [119] and focuses on the general imaging principles relevant to our work.

OCT imaging is performed based on the magnitude and echo time of backreflected light from internal tissue microstructures. OCT is similar to US except that OCT uses light instead of sound waves. Notably, light is substantially faster than sound, and hence direct electronic detection of echo time delays of light waves is impossible. Therefore, alternative measurement methods, such as interferometry, are used to determine the magnitude and echo time. Interferometry uses correlation between light that traveled a known distance and light that is backscattered from the tissue. This principle is based on the Michelson type interferometer shown in Figure 2.4.
An incident beam is dived into a reference beam, which travels a variable reference path, and into a signal beam which is backscattered from tissue. By combining the signal and reference beam and by varying the reference path length, interference signals will be generated over time. Whereas for a coherent light source, interference will generate over a wide range of path length differences, for a low-coherent light source, interference will only be generated when the path difference ΔL is within the coherence length. That is, adjusting the reference path length can be used to record interference for a particular axial sample distance, i.e., a specific layer of a sample. Using this property and by scanning the reference path length, a 1D depth scan can be performed, and the recorded intensities can be used for 1D imaging. This interferometric detection technique based on a low-coherent light source and based on scanning of a reference arm length is usually

called time domain detection. Note that a 1D depth scan represents differences in optical properties of a tissue and is typically referred to as A-scan, similar to 1D US imaging.
By scanning the beam transversely at different transverse or lateral positions, e.g., by reflecting the beam with controllable scanning mirrors, multiple A-scans in a raster pattern can be acquired. In this way, 2D cross-sectional imaging (B-scan) or 3D volumetric imaging (C-scan) of tissue can be performed. An example of a 2D OCT image is shown in Figure 2.5.

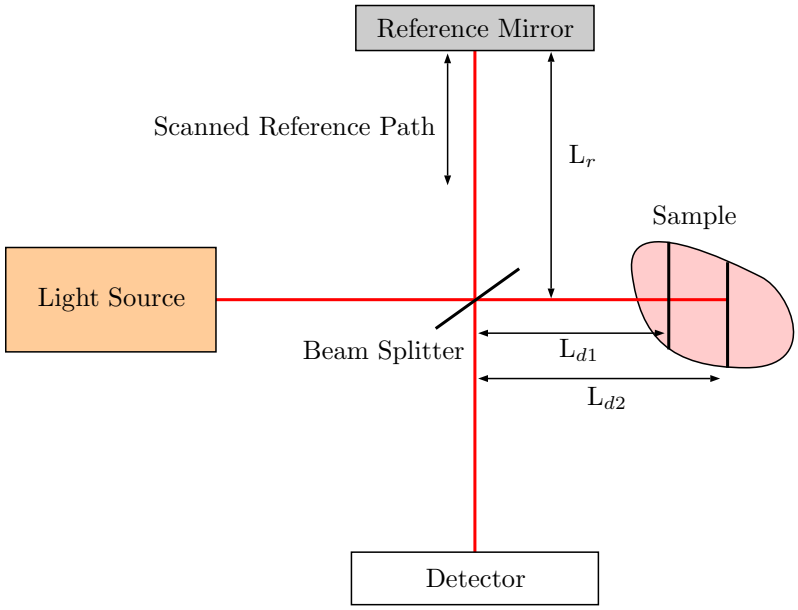

Figure 2.4: Schematic drawing of a Michelson interferometer. A beam is divided into a reference beam, which travels a variable reference path (L_r), and into a signal beam which is backscattered from tissue at reflective structures, e.g., at sample distance L_{d1} or L_{d2}. Using a low-coherent light source, an interference signal on the detector will only be generated when the path difference ΔL between the reference path and the sample distance is within the coherence length of the light source. Figure based on [119].

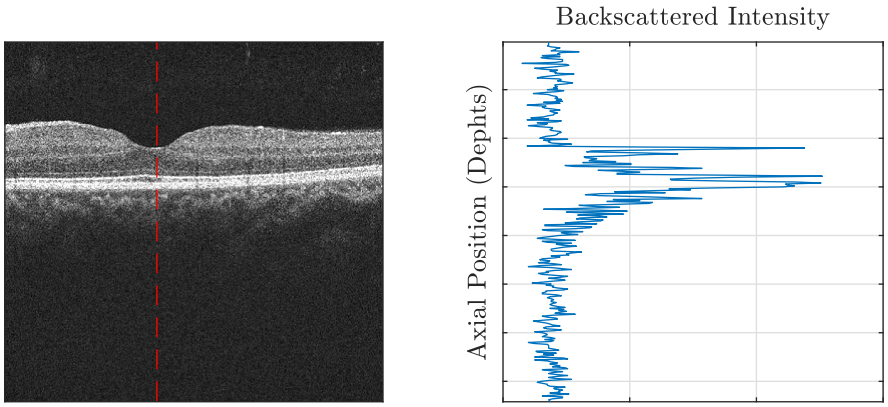

Figure 2.5: Example OCT image (B-scan) of a healthy retina. A single scan-line is highlighted in red, and the corresponding depths profile (A-scan) is shown on the right. The B-scan is generated by acquiring A-scans line-by-line. Figure based on [151]. OCT image data taken from the public Optical Coherence Tomography Image Retinal Database [172].

Considering the aforementioned described imaging principle of OCT, it becomes clear that the axial imaging resolution of OCT is determined by the coherence length of the light. Typically, OCT has an axial resolution of 1 µm to 15 µm. However, attenuation from scattering limits the image penetration depths typically to only a few hundred micrometers, i.e., ∼2 mm. The lateral resolution of 2D or 3D OCT depends mostly on the optics.

While cross-section or volumetric imaging with OCT is performed based on multiple A-scans in a raster pattern, the temporal imaging resolution of OCT can be related to the A-scan rate. Early time-domain OCT systems demonstrated limited A-scan acquisition speed in the range of ∼2 kHz [492]. With recent hardware advancements in the field of OCT, higher A-scan rates can be achieved with Fourier domain detection. This technique leads to improved sensitivity compared to time-domain detection and does not require scanning of a reference arm length, i.e., does not require a mechanical scanning mechanism which limits the temporal resolution [79, 265]. Two approaches are typically used for Fourier domain detection. The first approach is spectral domain OCT. This approach uses a broad-bandwidth light source, a spectrometer, and a line-scan camera [65, 143, 497, 498]. Detection of the backscattered light of the entire scan depths is performed at once based on Fourier transforming the measured interference signal. The second approach is swept-source OCT. This approach uses a narrow-bandwidth light source, and a single photodiode detector and the frequency of the narrow bandwidth light source is swept as a function of time [74, 79, 522]. Using a Fourier transformation of the combined signal of the reference signal and backreflected signal then allows relating the measured interference signal to depths. Additional explanations can be found, e.g., in [119].
Overall, considering spatio-temporal data acquisition, Fourier domain detection significantly increases the temporal acquisition rate compared to time domain de-

tection by acquiring data for a depth scan all at once. In recent years, acquisition rates have improved substantially, and A-scan rates, even up to the MHz-range, have been achieved using Fourier domain detection [246, 404, 492].

Usually, we use the amplitude information of the complex-valued data after the Fourier transformation for imaging with Fourier domain detection [482]. However, phase information is also utilized, e.g., to detect small displacements during shear wave elastography [236, 547]. Relying on the property of OCT that the phase signal is generally random but temporally invariant for a stationary sample [198], the axial displacement between two scans can be related to the change in the OCT phase signal. In fact, for small displacements, the axial displacement of a pixel and the phase difference $\Delta\varphi$ between two consecutive scans can be related with a linear relationship [426, 482, 483]. The phase difference of two subsequent measurements can therefore provide a direct measure of axial displacement between the two measurements. This allows for displacement estimation smaller than the wavelength of the OCT laser beam, i.e., sub-wavelength displacements in the range of nanometers [236, 485, 547]. Thus, by acquiring OCT phase data over time, an image sequence of pixel-wise displacement information can be obtained [547]. However, while the phase data allows for high sensitivity, it is also affected by various sources of noise, and achieving phase stability remains a challenging problem [73, 274, 547]. In addition to that, the measured phase difference is wrapped within $(-\pi, \pi)$ radians, and estimating the actual phase difference requires phase-unwrapping, which can become problematic when $|\Delta\varphi|$ is greater than π [73, 547].

While OCT technology has improved substantially in recent years, speckle remains and is inherent to the principles of OCT, similar to US. OCT speckle varies with properties and motion of the sample and is influenced by the light source [400]. Considering image quality, speckle reduces image contrast, and signal-to-noise ratio [400]. This reduced image quality can impact diagnosis and can make it difficult to distinguish small tissue structure boundaries [299]. To counteract speckle noise and thereby to improve the image quality, substantial efforts have been made to reduce this particular type of noise directly during image acquisition or after image acquisition as a post-processing step [179, 400]. Also, deep learning methods have recently gained traction for OCT speckle denoising [71, 106, 179, 299]. While speckle can be considered a source of noise, it is influenced by the properties of the sample, and hence speckle carries information about the properties of the sample. In this regard, it can also be considered a source of information similar to speckle noise in US images [400, 416]. It has been shown that OCT speckle can provide information to, e.g., to differentiate tissue types [178, 232, 279].

In summary, OCT is a non-invasive imaging modality that allows for volumetric imaging with high spatial and temporal resolution, and image generation is performed line-by-line. This imaging modality relies on backreflected light from internal microstructures of tissue that is measured and visualized.

2.3 Image Filtering

In the last sections, we demonstrated how spatio-temporal data can be acquired with OCT and US. In this section, we address aspects of image processing and analysis and explain concepts, such as correlation and convolution, that have a high relevance also for deep learning methods, especially CNNs that we introduce in the following Chapter 3. This section is based on [444]. To estimate changes between an image pair, we often search and compare distinct features or similarity in general. For simplicity, we assume gray-scale images and omit a color channel dimension at this point. In this context, correlation, and convolution are often used. The two-dimensional discrete correlation is given by

$$g_{cr}(o_1, o_2) = \sum_{i=1}^{k_h} \sum_{j=1}^{k_w} x_{o_1+i-1, o_2+j-1} \mathrm{K}_{i,j} \tag{2.3}$$

with $x \in \mathbb{R}^{n_h \times n_w}$ and $\mathrm{K} \in \mathbb{R}^{k_h \times k_w}$. The -1 in Equation 2.3 is used since algebraic notations start with 1. Usually, we refer to K and K_{ij} as the kernel and the filter coefficients, respectively. Thus, at any point $o = (o_1, o_2)$ for an image x we perform a neighborhood operator or local operator, which performs a weighted summation of a collection of pixels. Another important concept is convolution, and the two-dimensional discrete convolution is given by

$$g_{cv}(o_1, o_2) = \sum_{i=1}^{k_h} \sum_{j=1}^{k_w} x_{o_1-i+1, o_2-j+1} \mathrm{K}_{i,j} \tag{2.4}$$

Notably, correlation and convolution are related and are both spatial filters. The only difference between the two is that the kernel is flipped for the convolution and that, thereby, commutative property is achieved. Usually, this property is not relevant for neural networks, and both terms correlation and convolution are used interchangeably in the deep learning community [176]. For image processing, spatial filters can be used to enhance images or to highlight specific features such as edges. This requires selecting the filter coefficients $\mathrm{K}_{i,j}$ of the kernel and applying the spatial filter to every pixel of an image x afterwards. For example, a simple filter is to simply average the values of a neighborhood, often called box filter, which can be used for image smoothing. This can be achieved by setting all elements of the kernel $\mathrm{K} \in \mathbb{R}^{k_h \times k_w}$ to $\mathrm{K}_{i,j} = \frac{1}{k_h \cdot k_w}$. Another example is the Sobel filter given by

$$\mathrm{K}_{Sw} = \frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{2.5}$$

and

$$\mathrm{K}_{Sh} = \frac{1}{8} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{2.6}$$

to detect edges in the horizontal (image width) and vertical (image height) direction, respectively. Note that with edges, we refer to sharp changes in the pixel

intensity values of an image. The concept of correlation can also be used to find a template image $I \in \mathbb{R}^{k_h \times k_w}$ from a first image $x^{[1]} \in \mathbb{R}^{n_h \times n_w}$ in a second image $x^{[2]} \in \mathbb{R}^{n_h \times n_w}$, e.g., for translational alignment or to estimate the translation of a specific region using an image pair. Here, we define the kernel K based on a template region in a first image, i.e., $\mathrm{K} = I \subset x^{[1]}$ and then perform correlation with a second image $x^{[2]}$. To account for bright regions in the second image, commonly a normalized version is used called normalized cross-correlation (NCC) given by

$$\mathrm{NCC}_o = \frac{\sum_{i=1}^{k_h} \sum_{j=1}^{k_w} \left(x^{[2]}_{o_1+i-1,o_2+j-1} - \mu_x \right) \left(\mathrm{K}_{i,j} - \mu_\mathrm{K} \right)}{\sqrt{\sum_{i=1}^{k_h} \sum_{j=1}^{k_w} \left(x^{[2]}_{o_1+i-1,o_2+j-1} - \mu_x \right)^2} \sqrt{\sum_{i=1}^{k_h} \sum_{j=1}^{k_w} \left(\mathrm{K}_{i,j} - \mu_\mathrm{K} \right)^2}} \tag{2.7}$$

with

$$\mu_x = \frac{1}{k_h \cdot k_w} \sum_{i=1}^{k_h} \sum_{j=1}^{k_w} x^{[2]}_{o_1+i-1,o_2+j-1} \tag{2.8}$$

$$\mu_\mathrm{K} = \frac{1}{k_h \cdot k_w} \sum_{i=1}^{k_h} \sum_{j=1}^{k_w} \mathrm{K}_{i,j} \tag{2.9}$$

for the corresponding mean values [266]. Now, by using Equation 2.7 we can compare the template image with multiple or even all possible patches from the second image and by finding the maximum value with the corresponding position $o = (o_1, o_2)$ and we can, e.g., estimate the translation of the template between two time points. Note that NCC is always in the range [-1,1].
In summary, correlation and convolution can be used for image processing to highlight specific features such as edges and can also be used to search for a template image in a second image. Notably, image processing and feature extraction with correlation and convolution require the selection of the parameters of the kernel K, and thus involves manual tuning and adaption w.r.t. task. In our next Chapter 3 we introduce CNNs that still rely on the concepts introduced in this section but remove the burden of manual tuning and adaption of the filter coefficients. We extend this concept to spatio-temporal data in Chapter 4.

2.4 Summary

In this section, we motivated the concept of analyzing dynamic processes with sequences of images and described the idea of a spatio-temporal data representation. This representation considers the temporal axis as an additional data dimension leading to 3D and 4D spatio-temporal data for sequences of 2D and 3D images, respectively. Next, we described how such data can be acquired with two imaging modalities, US and OCT. In this context, we explained how imaging with a high temporal and spatial resolution can be performed ranging from 1D to 4D imaging. Moreover, both imaging modalities are non-invasive and non-ionizing with minimal harm for a patient compared to other imaging modalities such as, e.g., CT [207]. US is an imaging modality that is already widely established in clinical practice, and imaging is based on sending ultrasound wave pulses and

measuring the echo time, and amplitude of reflected sound waves [434]. Thereby, different acoustical properties of tissue can be measured and used for real-time imaging. OCT is an imaging modality that is based on optical backscattering of light, and different optical properties are measured and used for real-time imaging [119]. OCT has recently been adopted for various clinical applications, e.g., ophthalmology [310, 380], or oncology [150, 476]. Both modalities even allow for 4D spatio-temporal data, thereby allowing for observations of dynamic processes in their entirety, spatially and temporally. Lastly, we introduced fundamental concepts of image filtering, such as the concept of correlation and convolution, which have high relevance for deep learning methods that are the focus of this work.

Chapter 3

Deep Learning

In this chapter, we introduce the foundations of deep learning with a focus on supervised machine learning tasks. First, we highlight relevant concepts of machine learning for our study. Then, we describe the concepts of fully-connected neural networks, which are a core concept for understanding other more advanced architectures. In this context, we also explain how deep learning approaches can be optimized, i.e., how learning from data can be performed. Next, we explain the concepts of CNNs for image analysis and highlight the concept of RNNs for sequence analysis. Moreover, we describe the concept of regularization and hyperparameters, and lastly, we present important evaluation metrics for study. Overall, these concepts are fundamental for our deep learning methods to analyze medical image sequences.

3.1 Foundations

3.1.1 Supervised Machine Learning

In this section, we summarize concepts of machine learning, which are fundamental for our study on deep learning methods for spatio-temporal data processing. This section is based on [176], and for a detailed explanation of machine learning methods in general, we refer the interested reader to [52, 318]. A concise and established definition of machine learning is given by:

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." Mitchel (1997) [313, p. 2]

The task T of a learning algorithm is the actual task that should be addressed, e.g., classification or regression. Hence, the process of learning is not the task of a learning algorithm itself. Instead, it is the approach that is taken to address a particular task T. By learning from data, the advantage is that also tasks can be addressed that are difficult or even too difficult to solve with hand-designed fixed programs and rules. This includes various different tasks, e.g., classification, regression, transcription, machine translation, anomaly detection, or denoising. The task T can also be related to how an input example or data point x should be processed by a learning algorithm. In our work, we focus on regression tasks

where we estimate a continuous value from the input example. An example defines a collective set of features and can be written as a vector of features $x \in \mathbb{R}^{n_x}$ where n_x denotes the number of individual features. Considering an image as an example, each individual pixel value can be considered a feature. The collective set of features can also be referred to as feature space. For an image with n_c color channels and with size n_w and n_h for the width and heights, respectively, yields $x \in \mathbb{R}^{n_h \cdot n_w \cdot n_c}$ for the feature vector and keeping the spatial structure of the image this leads to a feature tensor $x \in \mathbb{R}^{n_h \times n_w \times n_c}$. In our work, we consider up to 4D spatio-temporal datapoints, which leads to feature tensors $x_t \in \mathbb{R}^{n_h \times n_w \times n_d \times n_t \times n_c}$.

Based on the given experience E provided to a learning algorithm, it can typically be categorized into unsupervised, supervised, or reinforcement learning. In this work, we focus on supervised learning algorithms and refer the interested reader to [176, 314] for the other categories. In brief, for unsupervised learning algorithms, the goal is to find a meaningful structure of the dataset. Reinforcement learning algorithms interact with a dynamic environment and the goal is to learn a particular behavior through trial-and-error interactions. In general, a collection of multiple (m) examples or data points is typically called a dataset and can be considered as the experience E provided to a learning algorithm. For supervised learning, each example of the dataset is annotated with a target or label $y \in \mathbb{R}^{n_y}$, where n_y denotes the number of continuous target values or distinct categories for regression and classification, respectively. The complete dataset is given by pairs of examples and the corresponding targets, i.e., a dataset is given by $\mathcal{D} = \{(x^{\{1\}}, y^{\{1\}}), (x^{\{2\}}, y^{\{2\}}), , \ldots, (x^{\{m\}}, y^{\{m\}})\}$. The annotation process is usually performed by an expert or instructor, hence the term supervised learning. Given a set of annotated examples, the task T of supervised learning is to predict or estimate the target y from the input x. More formally, a supervised learning algorithm can be used to approximate a function

$$f^* : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y},\ x \mapsto f^*(x).$$

by a model $\widetilde{y} = f_{ML}(x, \mathrm{w})$ with parameters w of the machine learning algorithm, which are optimized based on data. The estimated outputs of a machine learning algorithm are given by $\widetilde{y}$. Lastly, the performance measure P is used to evaluate the ability of the learning algorithm to address a particular task. Hence, P is typically specific to T. For example, a widely adopted performance measure P for classification tasks is accuracy, i.e., the proportion of correctly classified outputs. For regression tasks, distance measures are typically used for P, e.g., the mean absolute error between the actual target y and the predicted output $\widetilde{y}$. We explain different performance measures for regression tasks in more depth in Section 3.6.

A key concept of machine learning is generalization, which defines the ability to perform well on new, previously unseen examples. In this context, unseen refers to examples that were not used during training, i.e., the optimization process of the learnable parameters. In fact, the goal of a machine learning algorithm is not to simply memorize a given dataset. Instead, the goal is to learn the underlying concepts of the data. Two important terms, underfitting and

overfitting, related to generalization are relevant to our study. For evaluation, a dataset is typically split into three non-overlapping subsets used for training, testing, and validation of the algorithm. The first subset, the training dataset $\mathcal{D}_{train} = \{(x^{\{i\}}, y^{\{i\}})\}_{i=1}^{m_{tr}}$ is used to optimize the learning algorithm by minimizing a training error, which is a performance measure of the algorithm on the training dataset. The number of training examples is given by m_{tr}. Afterwards, the test dataset $\mathcal{D}_{test} = \{(x^{\{i\}}, y^{\{i\}})\}_{i=1}^{m_t}$ is used to evaluate the algorithm's performance on new, previously unseen examples that are not present during training. The number of test examples is given by m_t. The performance measure on the test examples is typically called the test error or generalization error. The validation dataset $\mathcal{D}_{val} = \{(x^{\{i\}}, y^{\{i\}})\}_{i=1}^{m_{val}}$ is used to select parameters of the learning algorithm that cannot be directly learned from the data. These parameters are typically called hyperparameters and usually involve manual tuning and selection by a human expert. Note that a distinct validation dataset is required for this process, as selecting hyperparameters based on the test dataset would falsify the observed performance on new, unseen data. The number of validation examples is given by m_{val}. We explain hyperparameter and the selection process in more depth in Section 3.5.

Considering the training dataset and the optimization procedure, a model might not be able to achieve a sufficiently low error on the training data. This is considered underfitting. In contrast to that, overfitting refers to a model that leads to a low error on the training dataset, but the error on the test dataset is high, i.e., the model generalizes poorly, and the gap between training and the test error is notably large. In this context, the terms bias and variance of a learning algorithm are also used. Bias refers to the expected deviation between the predicted values and the actual target values resulting from assumptions about a learning algorithm, and variance refers to how much the predicted values of a learning algorithm would differ if a different training dataset was used. To control these aspects, the capacity of a model is typically adjusted. While capacity is no completely formal term, it is usually considered a model's ability to represent a wide variety of functions, and a low and high capacity is associated with underfitting and overfitting, respectively. Figure 3.1 visualizes the concept of overfitting and underfitting of a model w.r.t. capacity, and Figure 3.2 shows an example of fitting data with a model using different capacities. Notably, capacity is an important aspect of learning algorithms and requires adjustment. Thus, constraints on the set of functions a model can use as a solution are typically made. Considering the capacity of a learning algorithm and the amount of available training data, the gap between training error and test error increases with the increasing capacity of a model and decreases with an increasing amount of training data [53]. In practice, the capacity of a learning algorithm is not only affected by the set of functions that can be used as a solution but also by the imperfection of the optimization algorithm.

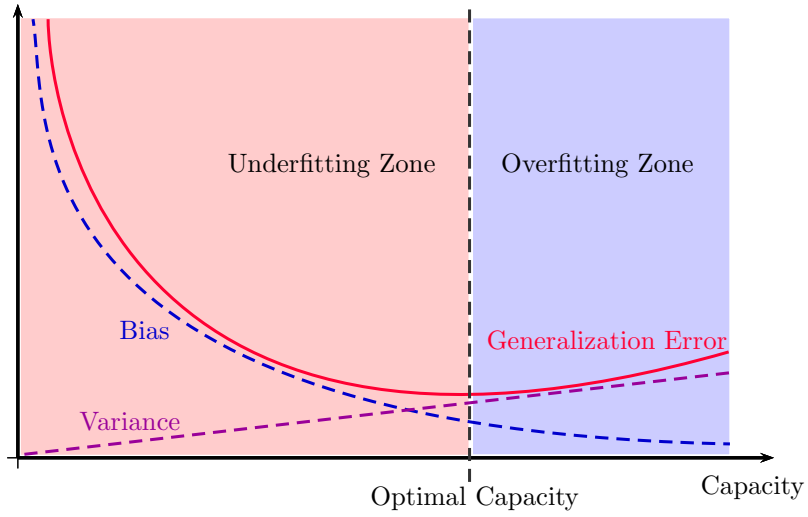

Figure 3.1: Typical relationship between capacity of a machine learning model and the resulting generalization error. Increasing the capacity typically decreases bias and increases variance of a model. A capacity lower than the optimal capacity leads to underfitting, and a higher capacity leads to overfitting. Figure based on [176].

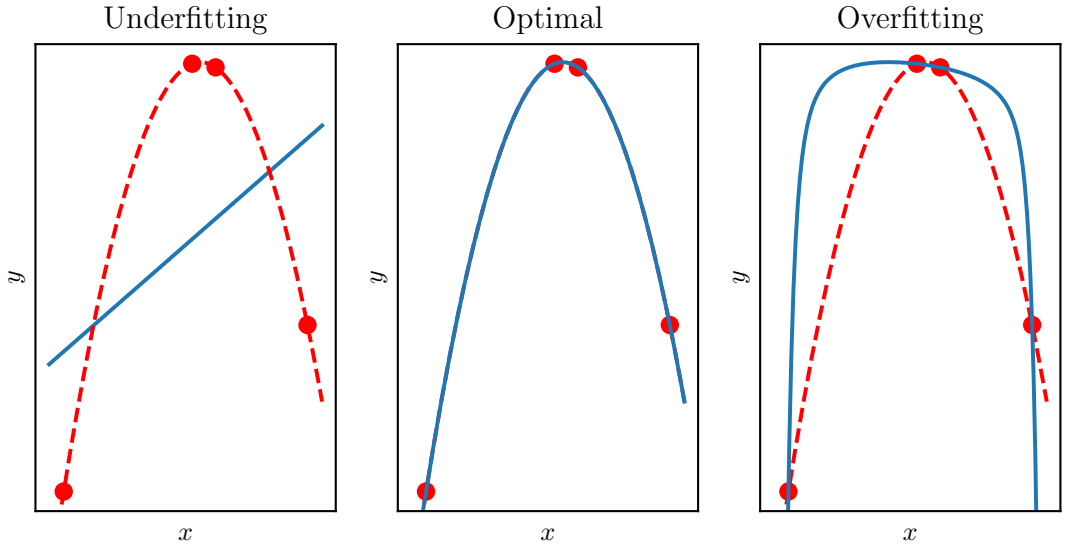

Figure 3.2: Example illustration of fitting data with models using different capacities. We use a quadratic function (red) and sample four data points (dots) for fitting a model. (Left) We use a linear function to fit the data, and we fail to represent the underlying data due to underfitting. (Center) We use a polynomial of degree 2, i.e., a quadratic function, to fit the data, and the underlying data is well represented. (Right) We use a polynomial of degree 30 to fit the data, and we fail to represent the underlying data. Figure based on [176].

3.1.2 Fully-Connected Neural Network

The foundations of deep learning have a long history, starting with a mathematical formulation of an artificial neuron in 1943 by Warren McCulloch and Walter Pitts [305], followed by the perceptron learning algorithm in 1958 by Frank Rosenblatt [374]. Today, the mathematical formulation of the artificial neuron is still one of the core concepts of deep learning, particularly for the fully-connected neural network, which can be considered a fundamental deep learning concept [176]. In fact, these early development inspired by the function of the human brain have established the term "artificial neural networks" to this day [52]. In this section, we introduce the concept of fully-connected neural networks. This section is based on [52, 176], if not indicated otherwise.

A fully-connected neural network is based on a composition of artificial neurons. In general, a single artificial neuron can be described by

$$\widetilde{y}(x, \mathrm{w}) = h_{out}\left(\sum_{i=1}^{n_x} \mathrm{w}_i \cdot x_i + \mathrm{w}_0\right) \tag{3.1}$$

with h_{out} for the output activation function, $\widetilde{y}$ for the output / estimated label, and w for the learnable parameters. Typically, w_0 is called the bias-term and $\mathrm{w}_{i\neq 0}$ are called the weights. A single artificial neuron can be used to perform linear regression or logistic regression, i.e., classification by choosing h_{out} as the identity or as a nonlinear activation function, respectively. However, no complex nonlinear functions can be represented with such an approach. This can be achieved by using a transformed version of the input $\phi_i(x)$, and combining this with Equation 3.1 yields

$$\widetilde{y}(x, \mathrm{w}) = h_{out}\left(\sum_{i=1}^{n_x} \mathrm{w_i}\phi_i(x_i) + \mathrm{w}_0\right). \tag{3.2}$$

The traditional approach is to manually engineer $\phi(x)$ as a feature extractor, e.g., that transforms the raw data pixel values of an image into a meaningful feature vector for the task at hand [258]. This quickly becomes a difficult task, is often highly problem-specific and requires substantial domain expertise. A fundamental concept of deep learning is to circumvent manual engineering of $\phi(x)$, by learning $\phi(x)$ directly from data. This concept is also referred to as representational learning, where a method processes raw data in an end-to-end fashion and automatically finds suitable features for a task [258]. For deep learning algorithms, $\phi(x)$ is formulated as a nonlinear transformation dependent on parameters, which are optimized based on a dataset. Thereby, highly complex nonlinear relationships can be approximated and learned end-to-end from data. Formulating $\phi(x)$ as a series of transformations similar to Equation 3.1, leads to the concept of a feedforward fully-connected neural network. Figure 3.3 shows an example feedforward fully-connected neural network.

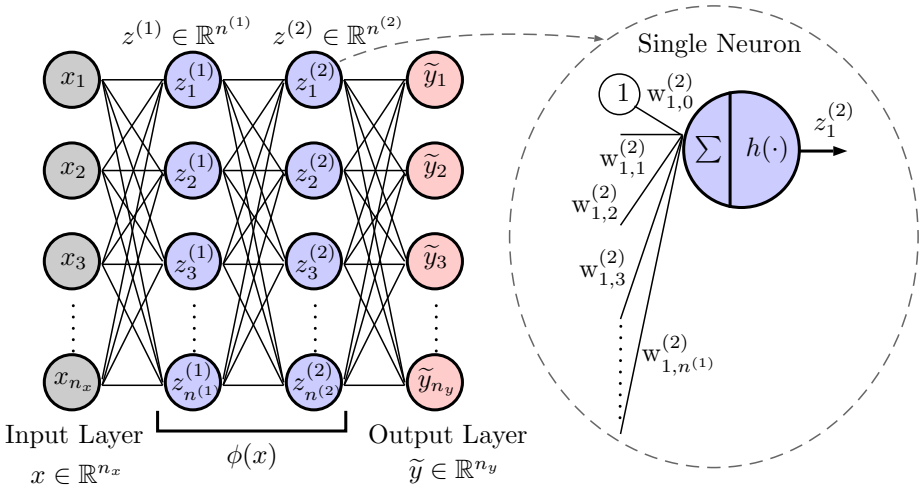

Figure 3.3: Example of a feedforward fully-connected neural network that maps an input $x \in \mathbb{R}^{n_x}$ to an output $\widetilde{y} \in \mathbb{R}^{n_y}$. Multiple artificial neurons are arranged in layers and connected in a feedforward fashion. Each neuron in a layer l is connected to all the neurons from the previous layer $l-1$. The output of a neuron is denoted by $z_i^{(l)}$and the weights are given by $\mathrm{w}_{i,j}^{(l)}$. A nonlinear transformation of the input dependent on the parameters of the neurons is given by $\phi(x)$. Figure based on [176].

The concept of a feedforward fully-connected neural network is to arrange multiple neurons similar to Equation 3.1 in layers and to connect the layers in a feedforward fashion. Hence, a single neuron is also called a node or unit of a neural network. We define $l \in \{1, ..., L\}$ as the layer number and L as the total number of layers, and denote the number of neurons per layer with $n^{(l)} \in \mathbb{N}$. For each layer l the output of a neuron in that layer is given by

$$z_j^{(l)} = h\left(\sum_{i=1}^{n^{(l-1)}} \mathrm{w}_{ji}^{(l)} z_i^{(l-1)} + \mathrm{w}_{j0}^{(l)}\right). \tag{3.3}$$

That is, each neuron receives the outputs from the previous layer as input and performs a vector-to-scalar function. Typically, the first layer is called the input layer, and the last layer is called the output layer, and it holds $z^{(0)} = x$ and $z^{(L)} = \widetilde{y}$. Intermediate layers are usually called hidden layers of a network. We denote the layer number with the upper index. The number of neurons in the output layer defines the number of outputs, hence $n^{(L)} = n_y$. The number of layers and the number of neurons per layer are also referred to as the depths and width of a network, respectively, and the name deep learning can be related to this terminology [176]. Another important aspect is the non-linear activation function $h(\cdot)$ of a neuron. While there are various non-linear activation functions, such as the sigmoid function or hyperbolic tangent activation function, that have been used for a long time, rectifying nonlinearities (ReLu) [174] demonstrated superior performance and are now widely adopted across various deep learning architectures and applications. For a recent survey on activation functions, we refer the interested reader to [120]. The three different activation functions that

we use during our work are visualized in Figure 3.4. A sigmoid activation function is given by

$$h_\sigma(a) = \frac{1}{1 + e^{-a}} \tag{3.4}$$

and a hyperbolic tangent activation function is given by

$$\tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}} \tag{3.5}$$

Lastly, a ReLu activation function is given by

$$h_{RL}(a) = \max\{0, a\}, \tag{3.6}$$

where a defines the activation value of a neuron, i.e., the weighted summation of all inputs.

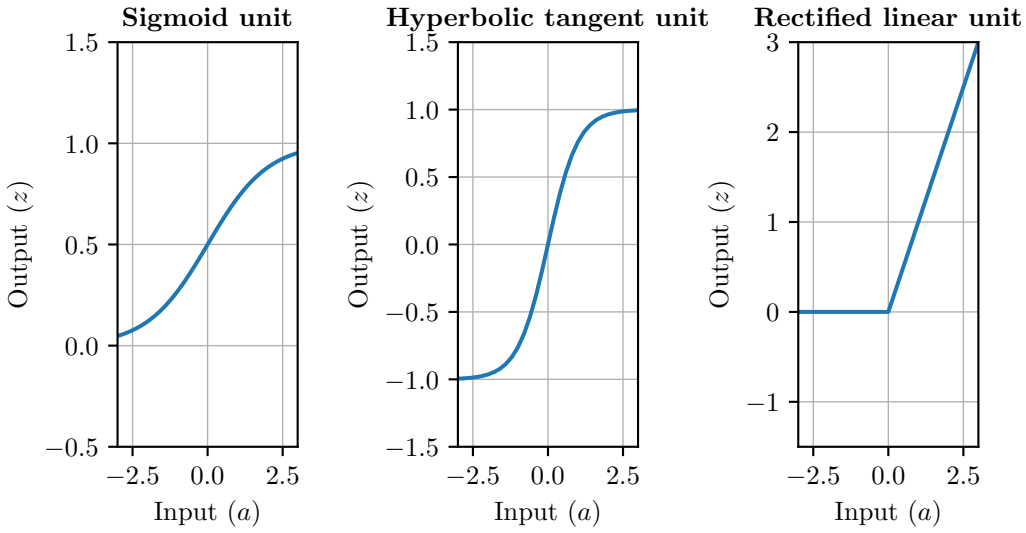

Figure 3.4: Example of three different activation functions that can be used for hidden neurons of a network. The activation value of a neuron is denoted by a, which defines the weighted summation of all inputs of that neuron. Note that the y-axis have different scales.

The advantage of the ReLu activation function compared to the sigmoid function or hyperbolic tangent function is that the derivative of the functions remains large and consistent whenever the function is active [176]. This property is beneficial for the optimization of the parameters of the network that involves gradient estimation. We provide more information on the optimization process in the following Section 3.1.3. For the output layer, the activation function h_{out} is the corresponding output function for the task that is addressed, e.g., the identity function for regression tasks, as also outlined previously.
An important property of fully-connected neural networks is that a network with at least one hidden layer, a sufficiently large number of hidden units can be considered universal approximators [205, 206]. Whereas this is a promising property,

in practice learning a particular function might still fail, e.g., due to the imperfection of the optimization or overfitting of the training data [176]. In the next section, we explain how the parameters w of a network can be optimized.

3.1.3 Neural Network Training

This section is based on [176]. Given a supervised training dataset $\mathcal{D}_{train} = \{(x^{\{i\}}, y^{\{i\}})\}_{i=1}^{m_{tr}}$, we minimize the deviation between the ground truth label y and the output of a neural network $\widetilde{y}$ with the goal to achieve generalization, i.e., to learn the underlying principles of the dataset. To this end, we typically formulate a loss function

$$\mathcal{L}(\mathrm{w}) = \frac{1}{m_{tr}} \sum_{i=1}^{m_{tr}} \mathscr{L}(f(x^{\{i\}}; \mathrm{w}), y^{\{i\}}) \tag{3.7}$$

with a per-example loss $\mathscr{L}(\cdot)$ that is a measure of the deviation between the estimated and ground truth label. Considering regression tasks, typical loss functions are the mean squared error (MSE) loss function

$$\mathcal{L}(\mathrm{w}) = \frac{1}{m_{tr}} \sum_{i=1}^{m_{tr}} \parallel f(x^{\{i\}}; \mathrm{w}) - y^{\{i\}} \parallel^2 \tag{3.8}$$

or the mean absolute error (MAE) loss function

$$\mathcal{L}(\mathrm{w}) = \frac{1}{m_{tr}} \sum_{i=1}^{m_{tr}} \left| f(x^{\{i\}}; \mathrm{w}) - y^{\{i\}} \right|. \tag{3.9}$$

While the loss function 3.7 is minimized using the training dataset this is not the ultimate goal of the optimization process, which is in contrast to typical optimization problems. Moreover, it should be noted that learning the parameters w is typically a non-convex optimization problem given the nonlinearity of deep learning algorithms. That is, there are local minima that are not globally optimal, and in practice, the optimal solution is rarely found. However, this is not considered a practical problem, as the final goal is generalization and not optimal performance on the training dataset [177].
Given the non-convex loss functions involved during the training of deep learning algorithm, we typically optimize the parameters w by reducing the value of a loss function using iterative, gradient-based optimizers. One approach is gradient descent optimization which iteratively updates the parameters w in the direction of the steepest descent of $\mathcal{L}(\mathrm{w})$ by using the direction of the negative gradient w.r.t. parameters w. In each iteration, the gradient is given by

$$\nabla_{\mathrm{w}} \mathcal{L}(\mathrm{w}) = \frac{1}{m_{tr}} \sum_{i=1}^{m_{tr}} \nabla_{\mathrm{w}} \mathscr{L}(f(x^{\{i\}}; \mathrm{w}), y^{\{i\}}) \tag{3.10}$$

and the parameters of a network are then updated iteratively using

$$\mathrm{w} \leftarrow \mathrm{w} - \mathrm{lr} \cdot \nabla_{\mathrm{w}} \mathcal{L}(\mathrm{w}). \tag{3.11}$$

An important hyperparameter of this update scheme is the learning rate $\text{lr} \in \mathbb{R}_+$. When the learning rate is selected too small, the training process, i.e., reducing the loss function, requires many iterations and thereby increases the computational efforts. Instead, if the learning rate is too large the training process can diverge, i.e., the loss function does not decrease during the training process. That is, choosing the learning rate represents a trade-off between fast and accurate convergence [28]. In practice, the learning rate is a hyperparameter that is usually manually tuned w.r.t. the task and the dataset. Also, different strategies exist to adapt the learning rate during training [96, 421], e.g., the learning rate is reduced linearly during the training process. Thereby, the trade-off between accurate and fast convergence can be adjusted.

To perform gradient descent optimization, we require gradient information. This can be addressed efficiently for deep learning methods with the backpropagation algorithm [379]. Recall that to produce an output $\widetilde{y}$ of a neural network, the input x is sent through the network, i.e., from the input layer to the output layer. The loss function 3.7 can be included in this process such that information from the input flows through the network and produces a scalar value of the loss function. This process is typically called forward propagation, and the entire process can be formulated as a computational graph. After forward propagation and given the value of the loss function, the concept of backpropagation is to propagate information of the loss function backwards through the network, i.e., computational graph to calculate the gradient of the loss function w.r.t. the parameters w of the network. This can be performed by recursively applying the chain rule of calculus starting from the loss function all the way to the input of the network [52, 176].

While optimization of neural networks is an iterative procedure, it requires initialization of the network parameter w as a first step of the training process. Here, different initialization strategies exist, and finding such strategies is an ongoing research field to improve the optimization process and generalization of neural networks [191]. For deep learning models, a common approach is to randomly initialize the weights by drawing values from a uniform, or Gaussian distribution [173, 196]. Another approach for weight initialization is to use the weights of a network of the exact same type that was trained ideally on a similar but also a different task. This approach is usually called transfer learning and can improve performance compared to random initialization [35, 41, 165, 447]. Moreover, today other methods such as batch normalization [216], which we introduce in Section 3.4, reduce the sensitivity w.r.t. weight initialization.

In practice, performing gradient descent optimization using the entire training dataset quickly becomes computationally expensive given the large-scale datasets that are used and often required for deep learning. Hence, stochastic gradient descent (SGD) and its variants are mostly used, where only a subset of the entire training dataset is used in each iteration of the gradient-based optimization. The subset is typically referred to as mini-batch $\mathcal{B} \subset \mathcal{D}_{train}$ with $m_b \ll m_{tr}$ and the

gradient calculation is given by

$$\nabla_{\mathrm{w}}\widetilde{\mathcal{L}}(\mathrm{w}) = \frac{1}{m_b}\sum_{i=1}^{m_b}\nabla_{\mathrm{w}}\mathscr{L}(f(x^{\{i\}};\mathrm{w}), y^{\{i\}}) \tag{3.12}$$

The mini-batch is randomly sampled from the training dataset, and m_b is usually called the batch size. Note, that this leads to a noisy estimate of the average gradient over all samples of a training dataset, hence the term stochastic [258]. For SGD, we then update the weights with

$$\mathrm{w} \leftarrow \mathrm{w} - \mathrm{lr}\cdot\nabla_{\mathrm{w}}\widetilde{\mathcal{L}}(\mathrm{w}). \tag{3.13}$$

Using a mini-batch is not only advantageous to reduce the computational requirements, but it can also improve generalization due to the additional noise in the learning process [495]. In practice, the batch size is a hyperparameter that is chosen based on the hardware constraints, the dataset type, and the task. Another important hyperparameter is the number of steps of the iterative optimization procedure, which also needs to be selected w.r.t. dataset, task, and generalization performance. With SGD, we randomly sample batches from the training data without replacement, and iterating over the entire dataset is typically called a single epoch. In this context, the maximum number of epochs represents an important hyperparameter and we explain the selection of hyperparameters in more depths in Section 3.5.

While SGD is still used and effective for deep learning optimization, today, numerous variations exist that can improve performance and accelerate learning by incorporating gradient information of previous iterations for an update step of the weights. One approach is to combine SGD with an exponentially decaying moving average of past gradients. This method is also referred to as SGD with momentum, inspired by the concept of momentum in physics [358]. Moreover, variants have been presented that also propose adaptive learning rates for each parameter of a network, such as AdaGrad [121], or Adam [242]. In this work, we use the latter, which is a popular optimization algorithm in the field of deep learning [242]. In addition to the exponential weighting of the gradients (first momentum), the Adam optimizer adjusts the learning rate of each parameter using the moving average of the squared gradients (second moment). The term "Adam" can be related to "adaptive moments" [242]. So far, there is no single best optimization algorithm for deep learning algorithms in general, and choosing an optimization algorithm can be considered an additional hyperparameter.

Other challenges of training neural networks are vanishing and exploding gradients during optimization, especially in the case of deep architectures that consist of multiple layers [32,173]. However, these challenges can be addressed today with layer normalization strategies [16, 93, 216, 500], and architecture design concepts that improve gradient flow [197, 203, 210]. We explain each of these concepts in the following sections of this chapter.

3.2 Convolutional Neural Network

In this section, we explain the concept of CNNs, which is an architecture concept specifically designed and developed for image analysis tasks [259]. CNNs were first presented and successfully trained in 1989 by LeCun et al. [259]. Later LeCun et al. [260] presented a first famous CNN architecture, LeNet in 1989, which achieved notable performance for the classification of handwritten characters from gray-scale images. Since then, numerous improvements have been proposed [11, 48, 50, 275]. Still, these concepts largely rely on the same fundamental principles described in this section. This section is based on [176], if not indicated otherwise.

Recall that the input of a fully-connected neural network is a feature vector $x \in \mathbb{R}^{n_x}$. For data that has a grid-like topology, this requires flattening the original data to a vector, e.g., for a 2D image with n_c color channels $x \in \mathbb{R}^{n_h \times n_w \times n_c}$ this leads to $x \in \mathbb{R}^{n_h \cdot n_w \cdot n_c}$. This has three major disadvantages.
First, the fully connected approach quickly leads to a substantial number of parameters of the network. In particular, for an image $x \in \mathbb{R}^{n_h \times n_w \times n_c}$ and a network with $n^{(1)}$ neurons in the first layer this leads to $n_h \times n_w \times n_c \times n^{(1)}$ parameters only in the first layer. Bringing this into perspective, for an image $x \in \mathbb{R}^{224 \times 224 \times 3}$ and $n^{(1)} = 100$ neurons in the first layers, this leads to 15 052 800 parameters in the first layer of the network. This increases the run-time notably and also increases the risk of overfitting.
Second, fully-connected neural network do not leverage information of the original topology of the data. Thereby, it is ignored that local groups of values are often highly correlated, e.g., for images, local groups of pixel values often form distinct local features such as edges or corners.
Third, fully-connected neural networks have per se no invariance w.r.t. shifts or perturbation of the input data, and this property must be learned from data. Note that an image, e.g., of a cat remains an image of a cat if the entire image is shifted by a few pixels in any spatial direction, and hence should be classified as such.
CNNs keep and leverage the original topology of the data and address the previously mentioned issues by adopting the concept of sparse interactions and parameter sharing. This can be achieved by adopting convolutions as the fundamental operation of the network and by learning the parameters of convolutions from data. Consider a 2D input image resulting in an input tensor $x \in \mathbb{R}^{n_h \times n_w \times n_c}$ where n_h and n_w define the height and width axis for which ordering matters and n_c defines the color channel axis which provides different views of the data. With sparse interactions, a neuron is only connected to a small subset of the input. Typically, we call this subset the local receptive field $V_o \subset x$, $V_o \in \mathbb{R}^{k_h \times k_w \times n_c}$ of a neuron z_o with $o = (o_1, o_2)$ for the indices of the output tensor, see Figure 3.5. The number of parameters are further reduced with parameter sharing and the same set of parameters is used for more than one neuron of a layer. In fact, the same set of parameters is used for all neurons of a layer, and typically we refer to the set of parameters as the kernel $\mathrm{K} \in \mathbb{R}^{k_h \times k_w \times n_c}$ and the bias term $\mathrm{w}_0 \in \mathbb{R}$.

Note that sparse interactions and parameter sharing reduce the number of parameters substantially compared to the fully-connected approach, where each neuron is connected to the entire input with individual weights.

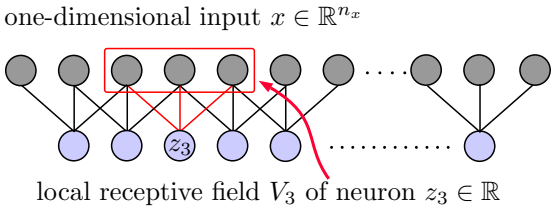

Figure 3.5: CNNs use the concept of sparse connection and only connect a neuron z_o to a small subset of the input called local receptive field V_o. The example shows a one-dimensional input x for simplicity. Figure based on [176].

Overall, parameter sharing and sparse interactions in combination can be interpreted as striding a kernel over an input to generate the output values, see Figure 3.6. Bringing these concepts together, the output of a neuron $z_o^{(l)}$ for a convolutional layer l is given by

$$z_o^{(l)} = h(a_o^{(l)}). \tag{3.14}$$

with

$$a_o^{(l)} = \mathrm{w}_0^{(l)} + \left(\mathrm{K}^{(l)} \otimes x^{(l-1)}\right)(o_1, o_2) \tag{3.15}$$

and

$$\left(\mathrm{K}^{(l)} \otimes x^{(l-1)}\right)(o_1, o_2) \coloneqq \sum_{i_1=1}^{k_h} \sum_{i_2=1}^{k_w} \sum_{i_3=1}^{n_c} \mathrm{K}_{i_1,i_2,i_3}^{(l)} \cdot x_{o_1+i_1-1,o_2+i_2-1,i_3}^{(l-1)}. \tag{3.16}$$

A nonlinear activation function is denoted by $h(\cdot)$. We call such a layer convolutional layer, whereas many machine learning libraries typically implement Equation 3.16 for simplicity, which describes a discrete cross-correlation. Note, we introduced both operations in Section 2.3 and highlighted that both operations are closely related with the difference that the kernel is flipped for the case of a convolution. However, while the parameters of the kernel are learned by the network, both operations can be used interchangeably, and the parameters of the kernels are learned accordingly.

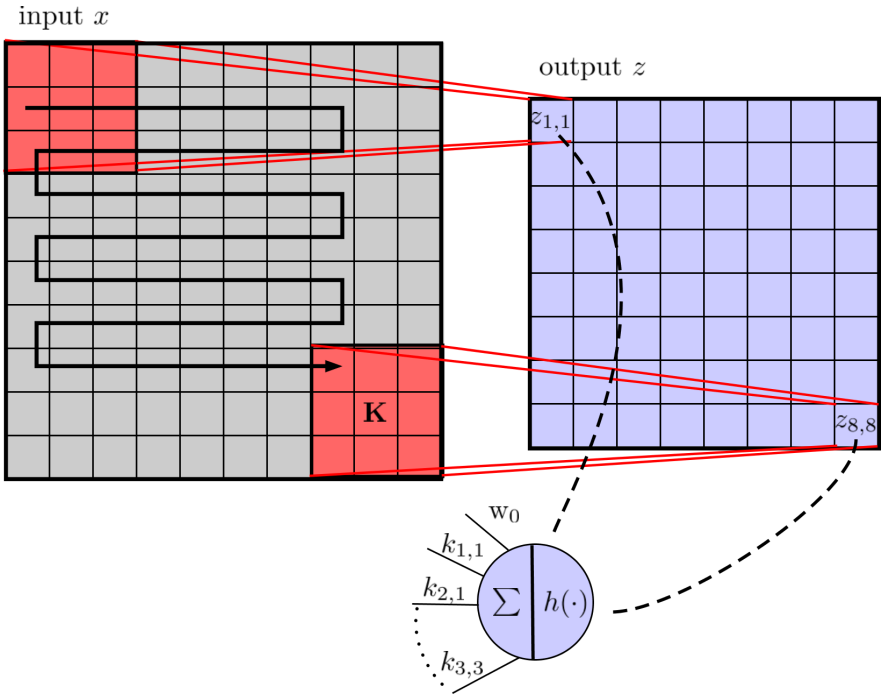

Figure 3.6: Sparse interaction combined with parameter sharing. Each neuron z_0 is only connected to a small subset of the input x and the same set of parameters is used for each neuron. This can be interpreted as striding a set of parameters $\mathrm{K} \in \mathbb{R}^{3\times3\times1}$, i.e., the kernel over the input x to estimate the output values of the neurons z_o. Figure based on [176].

Typically, multiple convolutional layers are used in parallel to extract various different features from the input. The number of convolutional layers in parallel is often referred to as the number of feature maps, and usually, the term convolutional layer in a network refers to a collection of multiple parallel convolutional layers, hence the overall number of parameters of a convolutional layer is given by $k_h \times k_w \times n_{in} \times n_{out}$ where n_{in} denotes the number of feature maps of the previous layer and n_{out} denotes the number of parallel convolutions, i.e., the number of feature maps of the current layer that can be estimated in parallel.

Similar to fully-connected neural networks, a CNN is composed based on successive layers to represent complex nonlinear functions. By using multiple successive convolutional layers, hierarchical features can be learned, and it also allows to indirectly learn features from larger portions of the input, see Figure 3.7. Regarding the size of the local receptive field, i.e., kernel size for a CNN, using a kernel size of $k_j = 3$ is a typical default choice, while larger kernel sizes can be represented by using multiple consecutive convolutional layers [418]. Moreover, also convolutional layers with a kernel size of $\mathrm{K} \in \mathbb{R}^{1\times1\times n_{in}\times n_{out}}$ with $n_{out} < n_{in}$ can be used for a CNN to downsample the feature dimension. Usually, such a layer is referred to as bottleneck layer [197]. To reduce computational costs and the size of the output as well as to quickly increase the local receptive field w.r.t. the input image, possible positions of the kernel can be skipped. Notably, this might come at the cost of missing information from the input. This can be performed

by using a kernel stride $\mathrm{s} > 1$ which leads to

$$\mathrm{conv2D}(\mathrm{K}^{(l)}, x^{(l-1)}, \mathrm{s})_o = \sum_{i_1=1}^{k_h} \sum_{i_2=1}^{k_w} \sum_{i_3=1}^{n_c} \mathrm{K}^{(l)}_{i_1,i_2,i_3} \cdot x^{(l-1)}_{(o_1-1)\cdot \mathrm{s}+i_1,(o_2-1)\cdot \mathrm{s}+i_2,i_3}. \tag{3.17}$$

The stride can be chosen individually along the different input dimensions, e.g., a stride larger than one can be used for the height axis of an input image, and a stride of one can be used for the width axis.
We introduced the concept of CNNs for 2D images for simplicity. However, the concept can also be extended to higher dimensions. For example, for a 3D input image, a 3D convolution

$$\mathrm{conv3D}(\mathrm{K}^{(l)}, x^{(l-1)}, \mathrm{s})_o = \sum_{i_1=1}^{k_h} \sum_{i_2=1}^{k_w} \sum_{i_3=1}^{k_d} \sum_{i_4=1}^{n_c} \mathrm{K}^{(l)}_{i_1,i_2,i_3,i_4} \cdot x^{(l-1)}_{(o_1-1)\cdot \mathrm{s}+i_1,(o_2-1)\cdot \mathrm{s}+i_2,(o_3-1)\cdot \mathrm{s}+i_3,i_4} \tag{3.18}$$

can be used. When considering 4D spatio-temporal data in Chapter 4, we also present the concept of performing convolutions across 3D space and the time leading to 4D spatio-temporal convolutions.

It stands out that applying a convolution layer to the input reduces the size, e.g., see Figure 3.6. In particular, even with a stride $\mathrm{s}_j = 1$ the size of the input is reduced with a convolutional layer, which limits the maximum number of convolutional layers that can be used consecutively and thereby limits abstract feature learning from the data. Following [122], the number of overlapping but non-equal local receptive fields along an axis j of the input is given by

$$1 \leq \mathrm{N}_{\mathrm{LRF}j} \leq \frac{n_j - k_j}{\mathrm{s}_j} + 1. \tag{3.19}$$

This can be addressed by padding the inputs with $\mathrm{p}_{\mathrm{z_j}}$ zeros at the start and end of an axis j of the input

$$1 \leq \mathrm{N}_{\mathrm{LRF}j} \leq \frac{n_j + 2\mathrm{p}_{\mathrm{z_j}} - k_j}{\mathrm{s}_j} + 1 \tag{3.20}$$

[122]. Thus, by zero-padding the input, the input size can be preserved and multiple consecutive layers can be used.

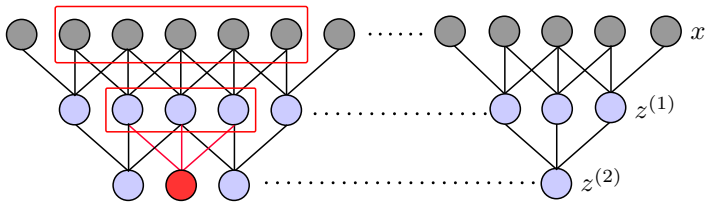

Figure 3.7: The receptive field w.r.t. the input of each neuron of a CNN increases with increasing depths of the network. That is, neurons in deeper layers are indirectly connected to large parts of the input or even the entire input. Figure based on [176].

Another layer type that is used for CNNs is a pooling layer. Instead of using a learnable set of parameters, here, each neuron statistically summarizes the values of the corresponding local receptive field. This is beneficial to achieve invariance to local translations of the input. One approach is maximum pooling [542]

$$z_o^{(l)} = \max(V_o^{(l-1)}),$$

which uses the maximum value from a local receptive field as output. Another approach is average pooling, which takes the average from a local receptive field of a neuron that yields

$$z_o^{(l)} = \text{avg}(V_o^{(l-1)}).$$

Moreover, pooling layers can also be used for downsampling the input by using a stride larger than one.

Lastly, the output layer of a CNN is usually a fully-connected layer for regression and classification tasks, where the number of neurons is equal to the number of outputs. Hence, this requires transforming the tensor structure of the last CNN layer to a feature vector. One approach is to simply flatten the tensor to a vector, i.e., let $z^{(L)} \in \mathbb{R}^{n_h^{(L)} \times n_w^{(L)} \times n_c^{(L)}}$ be the output of the last CNN layer, then we flatten the tensor to a vector $z^{(L)} \in \mathbb{R}^{n_h^{(L)} \cdot n_w^{(L)} \cdot n_c^{(L)}}$. However, this quickly results in a substantial amount of parameters. Another more parameter-effective approach is global average pooling (GAP), where the average is taken over an entire feature map [278]. Here, the output of the last CNN layer is transformed to a vector $\widetilde{z}^{(L)} \in \mathbb{R}^{n_c^{(L)}}$. In this way, correspondences between the feature maps and the output values of the network is enforced [278], and the number of parameters is reduced substantially. Finally, the network output $\widetilde{y} \in \mathbb{R}^{n_y}$ is given by

$$\widetilde{y}_j = h_{out}\left(\sum_{i=1}^{n_c^{(L)}} \mathrm{w}_{ji} \cdot \widetilde{z}_i^{(L)} + \mathrm{w}_{j0}\right) \quad for\ j = 1, 2, ..., n_y. \tag{3.21}$$

Overall, an example CNN can be constructed by subsequently using multiple convolutional and pooling layers followed by a GAP layer and a final output layer, see Figure 3.8.

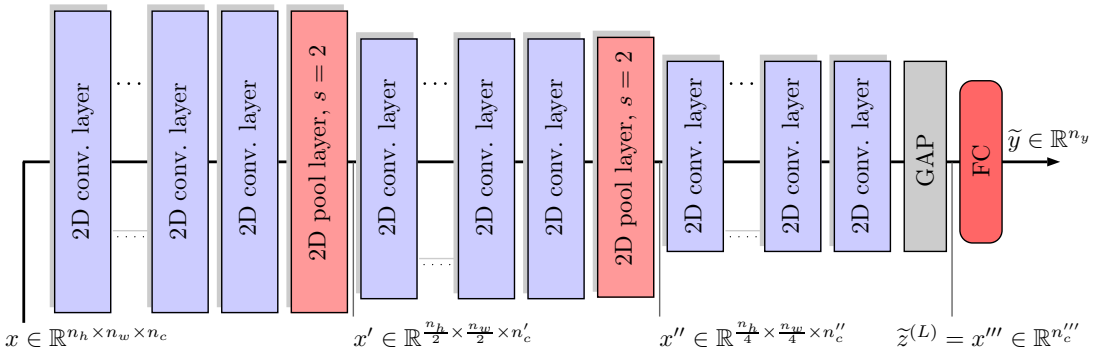

Figure 3.8: Example structure of a 2DCNN architecture that maps an input image $x \in \mathbb{R}^{n_h \times n_w \times n_c}$ to an output $\widetilde{y} \in \mathbb{R}^{n_y}$. The network consists of multiple subsequent 2D convolutional layers and 2D pooling layers. The pooling layers use a stride of two, and hence the input size is reduced by a factor of two. The last layer of the network is a fully connected network (FC) that maps the feature representation of the last layer $\widetilde{z}^{(L)} = x'''$ to the number of target outputs.

Within the last years, numerous improvements have been proposed for 2D image processing using classification benchmark datasets, e.g., ImageNet [103], including CNN architecture developments [11, 48, 50, 248, 275, 418]. For example, the Inception network and its variants have been presented [216, 442, 443]. The authors present the concept of making a CNN wider and using convolutional layers with kernels with different sizes in parallel to extract features at different scales. While there is a trend to use more layers for a CNN and deeper networks tend to outperform shallower ones, He et al. [197] highlighted that adding more layers can cause exploding or vanishing gradients, and thus simply adding more layers can saturate performance, or can even decrease performance on the training dataset. To address this problem, He et al. [197] presented Residual Networks (ResNets), which use the idea of skip connections or shortcut connections to improve gradient flow. Here, a convolutional layer receives a combined input that consists of the output from the previous layer combined with the output from a layer further ahead, see Figure 3.9. Building on this concept, Huang et al. [210] presented Densely Connected Convolutional Networks (DenseNet) that use the concept to connect a layer to all its preceding layers by combing the different layer outputs in the feature dimension, see Figure 3.9. Also, architectures that combine the previous concepts have been proposed [11, 441, 504]. For example, ResNeXt uses multiple convolutional layers in parallel combined with a skip connection [504]. Recently, Tan et al. [448] highlight that effective data processing with CNNs requires balancing network depth, width, and input resolution. In practice, the concept of the different architectures can be used as a backbone for custom CNN approaches, or entire architectures can be transferred between tasks. For the interested reader, a recent survey on CNN architectures is given in [48].

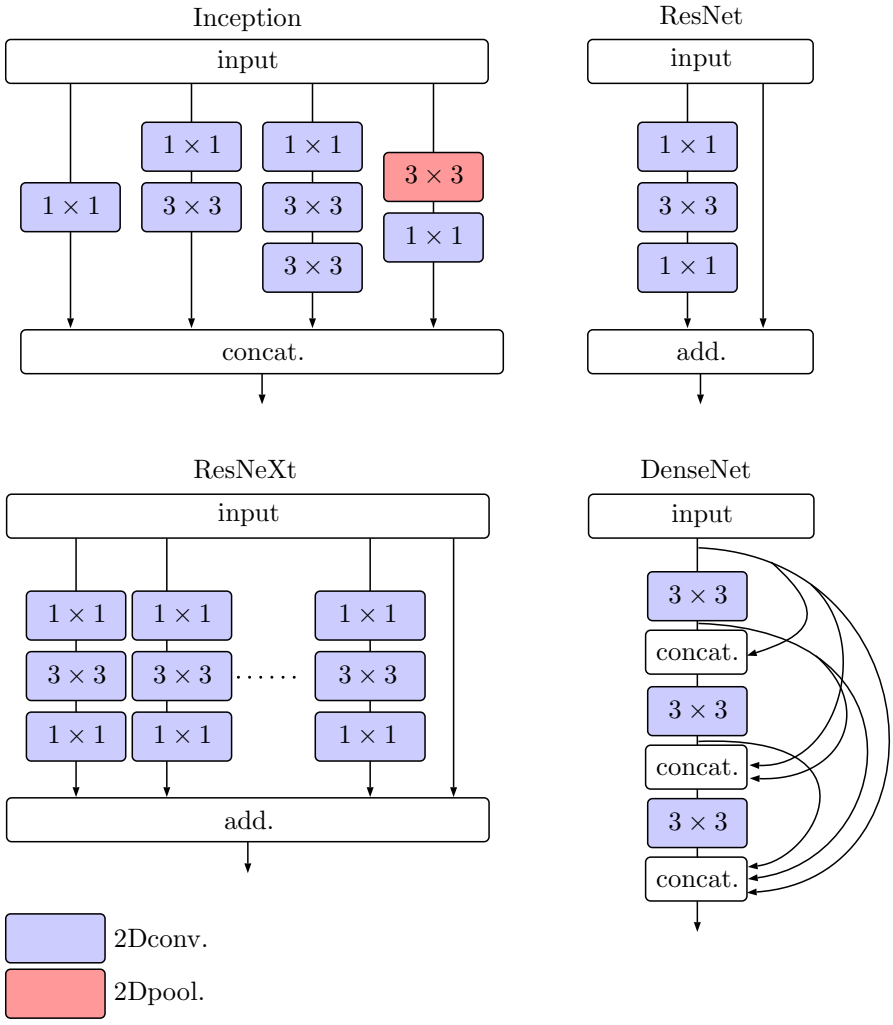

Figure 3.9: Examples of modern architecture concepts for CNNs [197, 210, 443, 504]. Inception uses multiple parallel layers and the outputs are concatenated along the feature dimension. ResNet uses a skip connection from the input to the output of a layer. ResNeXt uses multiple parallel layers in combination with a skip connection. DenseNet connects all layers to their preceding layers.

3.3 Recurrent Neural Network

In the last section, we introduced CNNs that have been designed and developed for data with grid-like topology and that rely on the concept of parameter sharing and spare connections. In this section, we introduce RNNs, an architecture concept that has been developed for the processing of sequential data. RNNs have shown promising results for applications such as time series prediction, natural language processing, image and video captioning, and video classification [381, 439]. In this work, we address spatio-temporal data and thus consider the sequence dimension exclusively as the temporal dimension. This section is based on [176], if not indicated otherwise.

Recall that we use superscripts with parentheses to indicate different time points. Given a sequence of inputs $x_t = [x^{[1]}, x^{[2]}, ..., x^{[n_t]}]$ with $x^{[i]} \in \mathbb{R}^{n_x}$ using a fully-connected neural network requires considering the entire input sequence as a single vector $x_t \in \mathbb{R}^{n_x \cdot n_t}$. This results in a substantial amount of parameters, ignores the order of the sequences, and only sequences with fixed lengths can be processed. To address these limitations, RNNs have been proposed that also rely on the concept of parameter sharing, similar to CNNs, but introducing parameter sharing across time in a different fashion with the motivation to address the aforementioned limitations. The fundamental concept of RNNs is to process the inputs sequentially and introduce cycles in the network's architectures such that information from previous time steps is combined with the current input. In its general form, the output of a layer with a recurrent connection at a time-point τ is given by

$$z^{[\tau]} = f(z^{[\tau-1]}, x^{[\tau]}; \mathrm{w}) \tag{3.22}$$

Hence, each output $z^{[\tau]} \in \mathbb{R}^{n_z}$ is a function of the previous output $z^{[\tau-1]} \in \mathbb{R}^{n_z}$ and the current input $x^{[\tau]}$. The number of neurons in the recurrent layer is given by $n_z \in \mathbb{N}$. More specifically, an example network with a recurrent connection is shown in Figure 3.10, and the output at each time point is given by

$$\widetilde{y}^{[\tau]} = h_{out}\left(\mathrm{V} z^{[\tau]} + \widetilde{\mathrm{w}}_0\right) \tag{3.23}$$

with

$$z^{[\tau]} = h(\mathrm{w}_0 + \mathrm{W} z^{[\tau-1]} + \mathrm{U} x^{[\tau]}) \tag{3.24}$$

with the bias terms $\widetilde{\mathrm{w}}_0 \in \mathbb{R}^{n_y}$,$\mathrm{w}_0 \in \mathbb{R}^{n_z}$ and the weight matrices $\mathrm{W} \in \mathbb{R}^{n_z \times n_z}$,$\mathrm{U} \in \mathbb{R}^{n_z \times n_x}$, $\mathrm{V} \in \mathbb{R}^{n_y \times n_z}$. The number of outputs / targets is given by $n_y \in \mathbb{N}$, and the number of input features is given by $n_x \in \mathbb{N}$. The computational graph of an RNN can be visualized as a model with recurrent connection or as an unfolded computational graph, see Figure 3.10. In particular, the latter visualizes the information flow forward in time.

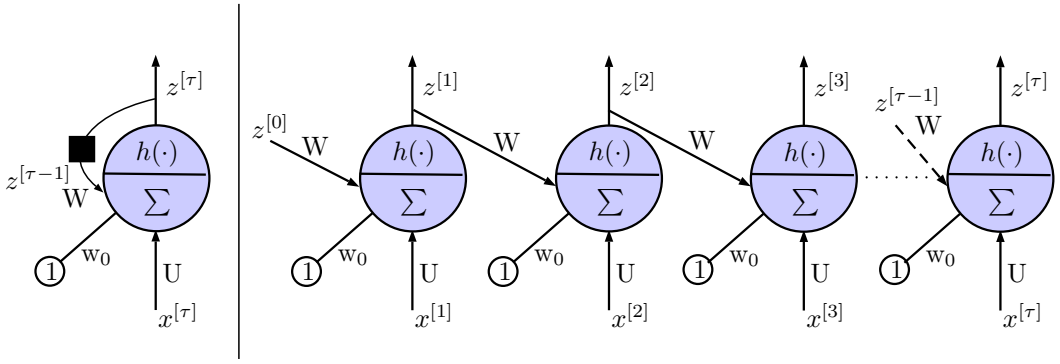

Figure 3.10: RNN unit where the previous output of the node $z^{[\tau-1]}$ is combined with the input $x^{[\tau]}$ to estimate the output $z^{[\tau]}$ of the node. (Left) Circuit diagram of an RNN unit. The bias term w_0 and weights of the input and of the recurrent connection are given by U, W respectively. (Right) unfolded representation of the RNN unit, where each node represents one-time point. We indicate a delay of a single time step with a black square. Figure based on [176].

Considering Figure 3.10 and Equation 3.24, an important property of an RNN is that we use the same set of parameters across different time points. Similarly, it stands out that Equation 3.24 maps a sequence with an arbitrary length to a fixed vector $z^{[\tau]} \in \mathbb{R}^{n_z}$. In this way, it provides a selective memory or lossy summary of the input sequence with the ability to capture and summarize long-term information. This brings the advantage of substantially reduced parameters, and the model can generalize to sequences with variable lengths. Notably, as an alternative, the recurrent connection can also be formulated based on the network output $\widetilde{y}^{[\tau]}$. However, in this case, $\widetilde{y}^{[\tau]}$ describes both the required output w.r.t. the task and the selective memory of the sequence and hence usually has less expressive power than using the output of a hidden layer.
Moreover, different input-to-output relationships can be formulated with RNNs. For example, in some scenarios, such as the classification of an entire sequence, only the last output after processing the entire sequence might be relevant. This is usually referred to as many-to-one behavior. Another approach is predicting an entire sequence from a sequence of inputs, e.g., during language processing [439]. This is typically called many-to-many behavior. Lastly, predicting a single output from a single input is called one-to-one behavior.

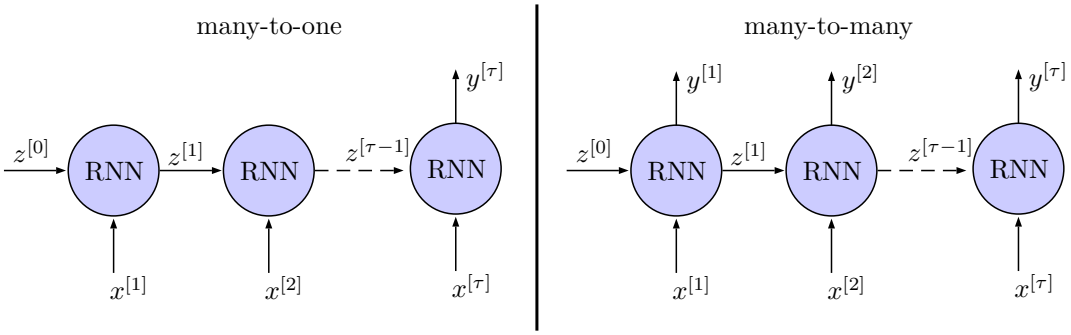

Figure 3.11: Two different input-output relationships of an RNN. (Left) Many-to-one, where a final output is estimated for the entire sequence. (Right) Many-to-many where an output is estimated for each input of a sequence. Figure based on [176].

Furthermore, RNNs can also be used in a bi-directional fashion [401]. Here, the outputs of two RNNs are combined. One RNN receives the input sequence forward in time and another RNN receives the input sequence backward in time and both outputs are combined. Hence, the combined output is not only dependent on the past input values but also dependent on the future input values. In practice, multiple RNN units can be used subsequently, or an RNN unit can also be incorporated into a network consisting of many fully-connected layers. So far, we assumed that the input is a feature vector $x^{[i]} \in \mathbb{R}^{n_x}$, however, RNNs can also be formulated for sequences of images by using convolutions for Equation 3.24 instead of linear transformations [506]. We provide more details on this variant when we consider 4D spatio-temporal data processing in Chapter 4.

In Section 3.1.3, we explained how the weights of a neural network can be learned

from data. This training procedure can also be applied to RNNs by performing forward and backward propagation through the unfolded computational graph. In this context, the gradient calculation required for optimization is also referred to as backpropagation through time (BPTT) [491]. Note that forward propagation of an RNN that follows Equation 3.24 is sequential and cannot be performed in parallel. Hence, training of an RNN quickly becomes an expensive operation with increasing sequence length. Another challenge that results from RNNs is that learning long-term relationships can become difficult due to exploding or vanishing gradients that result from the deep computational graph where we use the same set of weights at multiple time points [30, 32, 202]. These properties make RNNs usually more difficult to train than CNNs.

To reduce the aforementioned problems, leaky units [200,317] introduce weighted linear self-connections from the previous time step to accumulate the information from the past. The output of a leaky unit is given by

$$z^{[\tau]} = \xi \circ z^{[\tau-1]} + (\mathbf{1} - \xi) \circ h(\mathrm{w}_0 + \mathrm{W} z^{[\tau-1]} + \mathrm{U} x^{[\tau]}) \tag{3.25}$$

with $\mathbf{1}$ indicating a vector where all elements are one. An element-wise product is indicated as $\circ$. The parameter vector $\xi \in \mathbb{R}^{n_z}$ with elements $\xi_i \in [0, 1]$ allows controlling the information flow from the past and can either be learned from data or fixed in advance. However, a limitation of this approach is that the parameter ξ remains fixed for the entire sequence. Hence, a simple running average is performed without the ability to reset or filter information. Gated RNNs address these shortcomings. These networks increase the overall number of parameters compared to a simple RNN, but learning of long-term dependencies is substantially improved [77, 82, 84, 85, 203, 227].
One approach is a long short-term memory (LSTM) model that was proposed by Hochreiter and Schmidhuber in 1997 [203], and has been further refined by Gers et al. [161]. A key concept of the LSTM model is that the weight of the self-loop is not fixed. Instead, it is based on the context using the outputs of additional hidden units. LSTMs use state units $s^{[\tau]}$ with self-connections as additional components where the information flow is controlled by a forget and an internal input gate. An LSTM is described by the following equations [180]. The state units are given by

$$s^{[\tau]} = f^{[\tau]} \circ s^{[\tau-1]} + g^{[\tau]} \circ \tanh\left(\mathrm{w}_0 + \mathrm{W} z^{[\tau-1]} + \mathrm{U} x^{[\tau]}\right) \tag{3.26}$$

with the forget gate

$$f^{[\tau]} = h_\sigma\left(\mathrm{w}_0^f + \mathrm{W}^f z^{[\tau-1]} + \mathrm{U}^f x^{[\tau]}\right) \tag{3.27}$$

and the external input gate

$$g^{[\tau]} = h_\sigma\left(\mathrm{w}_0^g + \mathrm{W}^g z^{[\tau-1]} + \mathrm{U}^g x^{[\tau]}\right). \tag{3.28}$$

The final output of an LSTM unit is described by

$$z^{[\tau]} = \tanh(s^{[\tau]}) \circ q^{[\tau]} \tag{3.29}$$

with the additional output gate

$$q^{[\tau]} = h_\sigma \left(\mathrm{w}_0^q + \mathrm{W}^q z^{[\tau-1]} + \mathrm{U}^q x^{[\tau]} \right). \tag{3.30}$$

The weights are given by $\mathrm{W}, \mathrm{W}^g, \mathrm{W}^f, \mathrm{W}^q \in \mathbb{R}^{n_z \times n_z}$, $\mathrm{U}, \mathrm{U}^g, \mathrm{U}^f, \mathrm{U}^q \in \mathbb{R}^{n_z \times n_x}$ and the bias terms are given by $\mathrm{w}_0, \mathrm{w}_0^f, \mathrm{w}_0^g, \mathrm{w}_0^q \in \mathbb{R}^{n_z}$. Note the similarity of Equation 3.25 and Equation 3.26 with the difference that the weighting parameter ξ is now replaced with gates dependent on the context. The units of the forget gate $f_i^{[\tau]} \in [0, 1]$ allow to reset the internal state, and the units of the external input gate $g_i^{[\tau]} \in [0, 1]$ allow for updating the internal state. Figure 3.12 visualizes the concept of an LSTM model. Another variant is the gated recurrent unit (GRU) [77, 82, 84, 85, 227] that is more compact than the LSTM unit by using a single gate that controls both the forget and update mechanisms. A GRU is described by the following equations [227]. The output of a GRU is given by

$$z^{[\tau]} = u^{[\tau]} \circ z^{[\tau-1]} + (\mathbf{1} - u^{[\tau]}) \circ \tanh \left(\mathrm{w}_0 + \mathrm{W} \left(r^{[\tau]} \circ z^{[\tau-1]} \right) + \mathrm{U} x^{[\tau]} \right) \tag{3.31}$$

with the update gate

$$u^{[\tau]} = h_\sigma \left(\mathrm{w}_0^u + \mathrm{W}^u z^{[\tau-1]} + \mathrm{U}^u x^{[\tau]} \right) \tag{3.32}$$

and the reset gate

$$r^{[\tau]} = h_\sigma \left(\mathrm{w}_0^r + \mathrm{W}^r z^{[\tau-1]} + \mathrm{U}^r x^{[\tau]} \right). \tag{3.33}$$

The weights are given by $\mathrm{W}, \mathrm{W}^u, \mathrm{W}^r \in \mathbb{R}^{n_z \times n_z}$, $\mathrm{U}, \mathrm{U}^u, \mathrm{U}^r \in \mathbb{R}^{n_z \times n_x}$, and the bias terms are given by $\mathrm{w}_0, \mathrm{w}_0^u, \mathrm{w}_0^r \in \mathbb{R}^{n_z}$. In the context of a GRU, $z^{[\tau]}$ is also called the state vector. The elements of the update gate $u_i^{[\tau]} \in [0, 1]$ control the linear self-connection from the previous state similar to the parameter vector ξ of the leaky unit with the difference that the gating is now dependent on the context. The elements of the reset gate $r_i^{[\tau]} \in [0, 1]$ introduce an additional nonlinear effect and allow filtering of the state vector from the previous time-step for computation of the new state. A GRU model is shown in Figure 3.13. Notably, even more compact variants of a GRU exist, where the reset gate $r^{[\tau]}$ is removed [183, 520]. Similarly, various architecture variants can be developed by removing gates from the LSTM unit, and an extensive survey [183] on the different variants has been recently performed. Moreover, other nonlinear activation functions than the sigmoid or hyperbolic tangent function can also be used for an LSTM unit [140]. However, there is no clear winner between the different variants in general, and the architecture choice is dependent on the task and dataset. Overall, RNNs are a powerful approach for sequential data, but learning from extremely long sequences remains a persistent problem. Similarly, the sequential nature of RNNs limits parallelization.

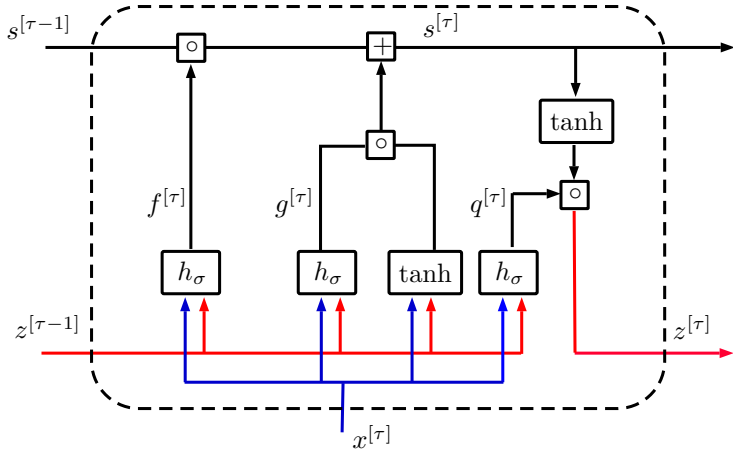

Figure 3.12: Illustration of an example LSTM cell [181,203]. The input is given by $x^{[\tau]}$, the output is given by $z^{[\tau]}$ and the state unit is given by $s^{[\tau]}$. The update gate, external input gate, and the output gate are denoted by $f^{[\tau]}$, $g^{[\tau]}$,$q^{[\tau]}$, respectively. Figure based on [176].

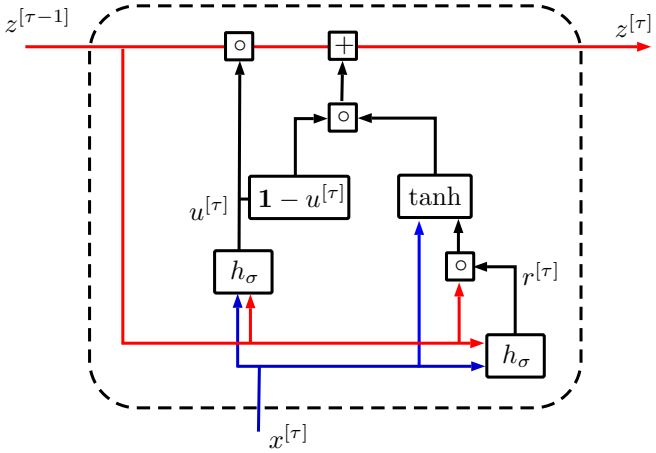

Figure 3.13: Illustration of an example GRU cell [77,227]. The input is given by $x^{[\tau]}$, the output is given by $z^{[\tau]}$. The output gate and the reset gate are denoted by $u^{[\tau]}$,$r^{[\tau]}$ respectively. Figure based on [176].

3.4 Regularization

This section is based on [176,316]. A fundamental goal of machine learning methods is to achieve generalization and, thereby, to achieve high performance on data that has not been used for the training process. Besides architecture developments that reduce the number of parameters, various regularization methods have been proposed to improve generalization. Regularization methods can address the architecture, the training process, and the training data. Considering the parameters of a model, the approach of parameter sharing and spare connections that we introduced in the previous sections could also be considered regularization methods. In general, according to Goodfellow et al. [176] regularization comprises all

methods that aim explicitly towards improving generalization. Usually, different regularization methods are combined and adapted for the task at hand.

A straightforward way to improve generalization of a deep learning approach is to collect more data. However, the amount of data is usually limited, especially in the medical domain. This quickly leads to a limited variation of the samples presented during training, which can result in overfitting and hence poor generalization. One way to address this problem is to transform the samples during training or to create artificial training samples. Such techniques are commonly referred to as data augmentation techniques. Considering image data, data augmentation techniques include flipping or rotating the image, injecting random noise, or translating the image a few pixels in each direction. Also, erasing areas of the image can be performed, e.g., by randomly setting pixel values to zero [107]. In general, numerous image transformations could be used as data augmentation approaches, as far as the transformation does not conflict with the assigned label of the image. In practice, hand-designed data augmentation strategies that incorporate domain knowledge of the task at hand are an effective way to improve generalization.

Another commonly used regularization approach is early stopping, which can also reduce the computational time efforts by reducing the overall number of training iterations [176]. Recall that the training process of a deep learning approach is an iterative procedure and that we need to define the maximum number of iterations as a hyperparameter. In particular, generalization performance improves during training but typically starts to decrease at some point due to overfitting of the training data. To reduce the burden to find the optimal number of iterations and thereby improve generalization, the performance on the validation dataset is periodically evaluated and tracked during the training process. Then, the training is stopped automatically when the performance on the validation set does not improve. Ideally, the performance on the validation dataset is evaluated every training epoch. However, this can result in notable additional computational efforts, and hence usually, the performance is evaluated less frequently.

Moreover, also regularization methods have been proposed that add noise to the weights of the networks. Srivastava et al. (2014) [433] have proposed dropout that aims towards learning features, which perform well in many situations and also provides an efficient approximation of ensemble learning. Ensemble learning refers to combining the predictions of several models. The concept of dropout is to randomly drop neurons of a network with a probability, i.e., dropout rate p_{drop}, during each iteration of the training process. A first advantage of this approach is that it enforces the neurons to learn more robust connections during training and to reduce the so-called co-adaptions between neurons. Moreover, applying dropout to a network with n_z neurons provides an efficient approximation of ensemble learning with 2^{n_z} networks that also share parameters. In particular, by randomly dropping neurons in each training iteration, a subnetwork is sampled from all possible 2^{n_z} subnetworks. Usually, not all possible 2^{n_z} subnetworks are

explicitly trained. However, this is sufficient due to the parameter-sharing scheme between the networks. After training, evaluating all 2^{n_z} networks during testing is computationally impractical. Hence an approximation is used called weight scaling inference. Weight scaling inference approximates the expected output of all 2^{n_z} subnetworks by taking the full network with all neurons and by scaling the weights of the neurons with the probability to keep that neuron.

Another famous approach that improves generalization and the training process in general is to incorporate normalization layers into a deep learning architecture [16, 93, 216, 295]. This concept has been highlighted by Ioffe and Szegedy (2015), who have proposed batch normalization [216]. The authors of this approach presented the approach with the motivation to address the challenge that the distribution of each neuron's inputs changes during training, which results from the parameter updates. The authors refer to this phenomenon as the internal covariate shift, and as a countermeasure, the authors propose to normalize the activation values of the neurons. In particular, with batch normalization, the activation of a neuron in a layer is normalized using

$$\hat{a}_i = \frac{a_i - \mu_i}{\sqrt{\sigma_i^2 - \epsilon_{BN}}} \tag{3.34}$$

with the mean

$$\mu_i = \frac{1}{m_b} \sum_{j=1}^{m_b} a_{ij} \tag{3.35}$$

and variance

$$\sigma_i^2 = \frac{1}{m_b} \sum_{j=1}^{m_b} (a_{ij} - \mu_i)^2 \tag{3.36}$$

estimated across the batch $\mathcal{B}$ in each training iteration with ϵ_{BN} as a small constant for numerical stability. While restricting the activation values to a standard normal distribution might change what a layer can represent, two additional learnable parameter $\mathrm{w}_{1_\mathrm{i}}, \mathrm{w}_{2_\mathrm{i}}$ are used per neuron that adjust the distribution using

$$\widetilde{a}_i = \mathrm{w}_{1_\mathrm{i}} \hat{a}_i + \mathrm{w}_{2_\mathrm{i}}. \tag{3.37}$$

In this way, the network could also learn to remove the effect of batch normalization if required, by setting $\mathrm{w}_{1_\mathrm{i}} = \mu_i$ and $\mathrm{w}_{2_\mathrm{i}} = \sqrt{\sigma_i^2 - \epsilon}$. Then, batch normalization represents an identity transformation. During testing, prediction is performed deterministically for a single sample, and varying batch statistics are not desired. Hence, a moving average of the mean and variance values over all batches of the training process are used during testing. The authors of the method recommend performing batch normalization before applying the nonlinear activation function. However, this has been challenged recently, and similarly, it has been challenged whether the performance improvements of batch normalization are really associated with internal covariate shift [388]. In particular, the underlying effect of batch normalization, and the actual reasons for the associated performance improvements are still under research [295, 298, 388]. An explanation

for the regularization effect of batch normalization is the noise that is added to the training process due to the varying batch statistics [216, 298]. Hence, using a smaller batch size in combination with batch normalization is associated with a stronger regularization effect. However, if the batch size is chosen too small, batch normalization can reduce generalization performance due to inaccurate estimation of the batch statistics.
Nowadays, batch normalization is used by default in most deep learning architectures [388]. Building on the concept of batch normalization, additional normalization methods have been proposed within the last years that perform the normalization independent of the batch dimension, such as layer normalization [16] or group normalization [500]. The first approach performs normalization based on the feature dimension, and the second approach divides the feature dimension into groups, and normalization is performed within each group.

3.5 Hyperparameters

In the previous sections, we described how training of a neural network can be performed, explained different deep learning architecture concepts, and highlighted regularization methods to improve generalization. In this context, we introduced several hyperparameters, which are parameters that can not be directly learned from data and need to be selected manually, semi-automatically, or automatically. This section is based on [176], if not indicated otherwise.

In general, choosing hyperparameters remains a persistent problem for deep learning methods and needs to be performed w.r.t. the task and dataset. Hyperparameters are selected based on the validation dataset performance and not directly based on the test dataset performance. Note that the test dataset is used to evaluate a model's performance during application on new, previously unseen data. The main goal of hyperparameter selection is to adjust the effective capacity of a model to achieve generalization by adjusting all involved aspects, such as the architecture and the training process. One approach to obtain the different datasets is to randomly split the entire data into fixed subsets, e.g., 70% of the data is for training, and 10% and 20% are used for validation and testing, respectively. While this approach can become problematic for small datasets due to the small test dataset size, cross-validation is typically performed. Here, the entire dataset is split into non-overlapping subsets, and training is repeatedly performed, leaving out data from one subset for testing. Afterwards, performance is averaged across the different folds. Notably, this increases computational efforts.

Choosing hyperparameters manually requires a strong intuition about the different behavior and impact of the different hyperparameters, the dataset, and the problem at hand. Instead, automatic hyperparameter selection can be performed by a grid or random search over the range of possible hyperparameter values. However, this quickly leads to substantial computational costs, and performing a search over all possible values of the hyperparameter space is usually practically impossible. Hence, a semi-automatic approach is typically performed where

the range of values is substantially in advance based on prior manual sections. Still, this approach is computationally demanding. Bringing this into perspective, when a single training process takes around one day, and we want to evaluate, e.g., five different learning rates, three different network configurations, three different regularization methods, and two different loss functions, this results in an overall training time of 90 days. Even more advanced approaches are model-based hyperparameter optimization, and neural architecture search [137]. Here, the selection process is defined as an additional optimization problem, which quickly results in substantial computational efforts. Hence, manual hyperparameter selection is usually required for most of the hyperparameters, given limited computational resources. Thus, hyperparameter selection typically requires substantial expertise. Another challenge for hyperparameter selection is that the available amount of data is limited, and hence splitting an additional validation dataset from the entire dataset reduces the amount of data available for training. One approach to address this challenge is to combine the validation and training dataset after hyperparameter selection and to train the model with the entire dataset and the selected hyperparameters. However, this increases the computational efforts even more. Overall, hyperparameter selection is critical for addressing a task with deep learning methods and usually involves notable computational efforts.

3.6 Performance Measures

After defining and training a machine learning model, we usually want to assess the performance of the method. Similarly, comparing the performance of deep learning models requires measurable performance metrics. In this work, we focus on multi-output regression tasks, where the output $\tilde{y} \in \mathbb{R}^{n_y}$ and target $y \in \mathbb{R}^{n_y}$ are vectors of scalar values. To this end, we summarize widely used and relevant regression metrics for our work in this section. This section is based on [54], if not indicated otherwise.

Given a supervised dataset $\mathcal{D} = \{(x^{\{i\}}, y^{\{i\}})\}_{i=1}^{m}$, one widely adopted regression metric is the MSE is given by

$$\mathrm{MSE} = \frac{1}{n_y} \sum_{j=1}^{n_y} \frac{1}{m} \sum_{i=1}^{m} \left(y_j^{\{i\}} - \widetilde{y}_j^{\{i\}} \right)^2 \tag{3.38}$$

or the MAE is given by

$$\mathrm{MAE} = \frac{1}{n_y} \sum_{j=1}^{n_y} \frac{1}{m} \sum_{i=1}^{m} \left| y_j^{\{i\}} - \widetilde{y}_j^{\{i\}} \right|. \tag{3.39}$$

Comparing both performance metrics, the MSE is more sensitive w.r.t. outlier values than the MAE. Another widely used relative performance metric is the

average correlation coefficient (aCC) given by

$$\mathrm{aCC} = \frac{1}{n_y}\sum_{j=1}^{n_y}\mathrm{PCC}_j = \frac{1}{n_y}\sum_{j=1}^{n_y}\frac{\sum_{i=1}^{m}\left(y_j^{\{i\}}-\bar{y}_j\right)\left(\widetilde{y}_j^{\{i\}}-\bar{\widetilde{y}}_j\right)}{\sqrt{\sum_{i=1}^{m}\left(y_j^{\{i\}}-\bar{y}_j\right)^2\sum_{i=1}^{m}\left(\widetilde{y}_j^{\{i\}}-\bar{\widetilde{y}}_j\right)^2}} \tag{3.40}$$

where $\bar{\widetilde{y}}_j$ and $\bar{y}_j$ denote the mean across the dataset for the outputs and targets, respectively. This metric is independent of the unit and measures the linear correlation between the target and output values. For a single output regression task ($n_y = 1$), this metric is equal to the Pearson correlation coefficient (PCC) between the target and output values. Also, relative performance metrics independent of the unit can be formulated by normalizing a metric [54]. One approach is to normalize a metric by the standard deviation of the target values, e.g., for the MAE this yields the relative MAE (rMAE) given by

$$\mathrm{rMAE} = \frac{1}{n_y}\sum_{j=1}^{n_y}\frac{1}{m}\sum_{i=1}^{m}\frac{\left|y_j^{\{i\}}-\widetilde{y}_j^{\{i\}}\right|}{\sigma_{y_j}} \tag{3.41}$$

with

$$\sigma_{y_j} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(y_j^{\{i\}}-\mu_{y_j}\right)^2} \tag{3.42}$$

and

$$\mu_{y_j} = \frac{1}{m}\sum_{i=1}^{m}y_j^{\{i\}}.$$

In the context of regression tasks related to motion analysis also, the tracking error (TE) is a commonly used metric defined by the Euclidean distance between the estimated position $\tilde{y}$ and the ground truth position y [97, 98, 282]. The TE is given by

$$\mathrm{TE} = \|y-\tilde{y}\|_2 = \sqrt{\sum_{j=1}^{n_y}(y_j-\tilde{y}_j)^2}. \tag{3.43}$$

Furthermore, in the context of elastography, we also evaluate the segmentation performance using deep learning models. To this end, a commonly used metric is the Sørensen–Dice coefficient [111, 429] defined by

$$\mathrm{Dice} = \frac{2\,|y\circ\widetilde{y}|}{|y|+|\widetilde{y}|} \tag{3.44}$$

where y and $\widetilde{y}$ are two binary vectors and with $\circ$ for the element-wise product.

Moreover, besides general regression performance metrics, other metrics include, e.g., training time and the inference time of a model. The training time is an important measure when it comes to hyperparameter selection and retraining of a model in general. Note that for a model with shorter training times, a variety of

hyperparameters can be evaluated compared to a model with notably increased training time. The inference time of a model is an important measure to assess real-time capabilities, which is essential in application scenarios such as motion compensation. We also consider the throughput of a model, which refers to the number of estimates within one second. To maximize throughput, we want to process as many inputs as possible in parallel.

3.7 Summary

In this chapter, we introduced the foundations of deep learning with a focus on supervised machine learning tasks. Deep learning methods learn end-to-end relationships directly from data, and supervised deep learning approaches are optimized to learn a defined input-output behavior. Thereby, the burden of manual feature engineering can be removed, and extracting relevant features can be learned from data by means of representational learning [176, 258]. In this context, the final goal of machine learning methods is to perform well for samples that have not been presented during the training process [52, 318]. This requires learning the underlying principles of the data, and typically, we refer to this ability as generalization [233]. As a first network architecture, we introduced fully-connected neural networks that consist of artificial neurons arranged in layers with parameters w that need to be learned from data in a training process. The training process is agnostic w.r.t. the network architecture and involves the definition of a loss function and the selection of an optimization algorithm, which is typically an iterative, gradient-based optimizer [242]. However, fully-connected neural networks bring many disadvantages when it comes to image analysis. For example, fully-connected neural networks ignore the topology of the data and quickly result in a substantial amount of parameters [176]. These disadvantages can be addressed with the idea of CNNs that have improved many medical image analysis tasks [14]. We explained this architecture concept that relies on convolutions as fundamental operations and highlighted the concept of sparse connections and parameter sharing [260]. While CNNs and fully-connected neural network process the input in a feedforward fashion, RNNs introduce the idea of feedback connections that bring additional advantages when it comes to sequence analysis, such as direct processing of sequences with variable length and adaption to different input-to-output behaviors [176]. Here, we also introduced the underlying fundamentals and modern approaches such as LSTMs [203], and GRUs [77, 82, 84, 85, 227]. Lastly, we highlighted the aspects of regularization, hyperparameter selection, and performance measures w.r.t. regression tasks.

Chapter 4

Spatio-Temporal Deep Learning Methods

In this chapter, we describe and develop our deep learning methods for multi-dimensional medical image sequence analysis. We build our approach based on the two widely used architecture concepts for video analysis in the natural image domain, CNNs, and RNNs. We introduce the relevant background for each of our methods and present our approaches and methodical developments. We outline the general data processing pipeline with deep learning and the steps involved in approaching a task based on sequences of medical image data. We present and explain different operations and architecture building blocks, which are flexible in nature and thus can be adapted to different imaging modalities, dimensions, and tasks. As a result, we develop and present a compact architecture concept for 3D and 4D spatio-temporal data that allows for integrating different network operations. Our methods range from pair-wise processing to entire long-term sequences and can be used for image-level and sequence-level predictions. Lastly, we address the challenge of training with medical image sequences. In this context, we present custom loss functions, regularization strategies, and a specific training approach that addresses training with sequences of hundreds of volumetric images.

4.1 Spatio-Temporal CNNs

4.1.1 Background

Motivated by the success of deep learning for image analysis, it is natural to raise the question of how CNNs can be extended to image sequence analysis. Deep learning for image sequence analysis gained traction in the natural image domain around 2014 due to the availability of increased computational hardware and large-scale public datasets [69, 231]. Today, only a few years later, a plethora of CNN-based methods have been presented for 2D image sequence analysis from the natural image domain. In this context, a typical application scenario is video classification [231] with human action recognition [438] as a widely studied example. Recent reviews of CNNs for video analysis are given in [69, 369, 438]. In the following paragraphs, we highlight relevant developments and advancements of CNNs for video analysis in the context of our work. These considerations

are fundamental for our spatio-temporal deep learning concepts for analyzing multi-dimensional medical image sequences. Early developments of deep learning for video analysis mainly used CNNs, typically pre-trained, as an image-wise feature extractor, followed by a classification model such as support vector machine [369, 508, 526]. These approaches rely on the combination and aggregation of extracted image-wise features. Whereas manual feature engineering efforts can be reduced, no end-to-end processing of the data and no joint-spatio-temporal feature learning are performed [457]. This requires and motivates an end-to-end data processing pipeline with deep learning.

One of the first approaches to this end, typically called two-stream CNNs [142, 417], uses two data processing streams, one spatial and one temporal stream. The spatial stream processes a static image, and the temporal stream receives optical flow information. Afterwards, both streams are fused, and the target output is estimated. As fusion operations, e.g., concatenation, summation, or multiplication of the feature representations can be used [369]. While initially only short-term image sequences of a few images have been processed with such an approach, long-term video classification can also be achieved [479]. To this end, the long-term video is divided into segments, each processed with a two-stream CNN. Afterwards, the classification results of the segments are fused. Such an approach is usually called temporal segment networks (TSN). However, the two-stream approach is memory intensive with substantial computational costs. Recently, it has been shown that additional optical flow information is not necessarily beneficial for video classification and, most importantly, that the high computational requirements of optical flow estimation limit two stream networks for real-time applications [369]. This motivates end-to-end processing of the sequence without any additional feature extraction.

A simple approach to extend a CNN from image processing to end-to-end processing of image sequences is to consider the channel dimension of the input as the temporal dimension [231, 354]. More formally, for a sequence of 2D images, $x_t = [x^{[0]}, x^{[2]}, \ldots, x^{[n]}]$ with $x^{[i]} \in \mathbb{R}^{n_h \times n_w \times n_c}$ all images of the sequences are stacked into the channel (color) dimension, which yields $x_t \in \mathbb{R}^{n_h \times n_w \times n_c \cdot n_t}$ for the network input. This input can be considered as a regular 2D image with multiple color channels ($n_c \cdot n_t$) and hence can be processed with a 2DCNN. Note that this allows for a similar architecture compared to single 2D image processing, although an entire sequence is processed. However, it quickly leads to a substantial amount of parameters in the first convolutional layer of the network, i.e., the input layer. Recall, that the parameters of a 2D convolutional layer are given by $k_h \cdot k_w \cdot n_{in} \cdot n_{out}$ with $n_{in} = n_c \cdot n_t$. Hence, the network can be considered as fully-connected along the temporal dimension. Another limitation is that the temporal processing is only performed in the first convolutional layer at the input pixel-level. Hence, no complex spatio-temporal features can be used and learned [457]. Such an approach can also be referred to as early fusion, indicating that all the temporal information of the sequences is combined at an early stage of an architecture [231]. Overall, a CNN with a time-channel is compact and easy to adapt and only requires minor changes compared to an architecture that is used for single image processing. However, such an approach only demonstrated minor performance

improvements compared to single image processing for video classification [231]. Alternative approaches are late and slow fusion. Late fusion combines feature representations of the images of the sequence at the output layer, and slow fusion combines feature representations of the images throughout the network [231].

A natural but more complex extension of a CNN to a sequence of images is to perform convolutions across space and time, i.e., to use spatio-temporal convolutions [17, 224, 256, 453, 457]. Thereby, joint-spatio-temporal feature learning throughout the network can be performed. This approach considers the spatio-temporal structure of the input $x_t \in \mathbb{R}^{n_h \times n_w \times n_t \times n_c}$, and then performs convolutions across space and time. Thereby, it preserves the original data structure. Figure 4.1 visualizes the input-output behavior for different kinds of convolutions combined with sequences of images. By using convolutions across space and time, the advantages of a CNN can be taken to spatio-temporal data analysis, i.e., invariance w.r.t. shifts of the input data and leveraging of the original data topology [457]. Recall that motion can be considered as an orientation in space-time images [2], and with spatio-temporal convolutions, these spatio-temporal features can be learned from data w.r.t. a given task. By stacking multiple spatio-temporal convolutional layers, a spatio-temporal CNN can be designed and trained in an end-to-end fashion to learn joint features from space and time. In this way, complex and abstract spatio-temporal relationships can be learned jointly. The concept of spatio-temporal convolutions can be combined with different modern backbone 2DCNN concepts such as the Inception concept [442], ResNet concept [197], or DenseNet concept [210], and various adoptions have been presented [63, 109, 110, 192, 463]. Also, a two stream concept has been presented that uses spatio-temporal information at different temporal resolutions [141]. The architecture consists of two streams, one lightweight spatio-temporal CNN receives frames at high temporal, and another spatio-temporal CNN receives frames at a low temporal resolution. Both streams are fused across the architecture depths.

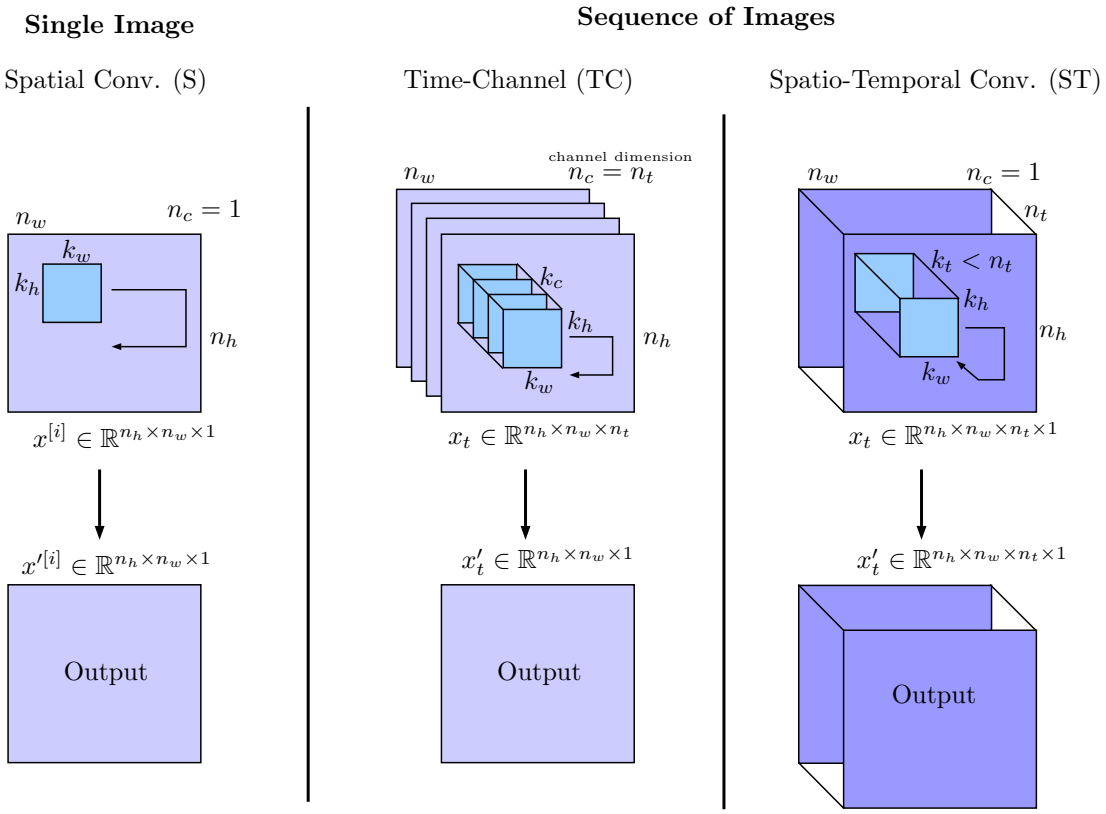

Figure 4.1: Convolutions and image sequences. (Left) A 2D convolution applied to a single image of a sequence results in a 2D output. (Middle) Stacking multiple images into the channel dimension and applying 2D convolutions results in a 2D output. (Right) Considering the input as a spatio-temporal tensor and applying a 3D convolution leads to 3D output and preserves the spatio-temporal structure. Figure based on [457].

However, spatio-temporal convolutions notably increase the number of parameters and computational efforts compared to single image analysis. Note that spatio-temporal convolutions extend the kernel by an additional dimension. That is, we use a 3D kernel compared to a 2D kernel that is used for single 2D image analysis [457]. One approach to address the increased model complexity is disentangling spatial and temporal processing. To this end, factorized spatio-temporal convolutions and their variants have been proposed for video analysis [297, 364, 437, 459, 505]. Here, the underlying concept is to split a spatio-temporal convolution with a kernel $k_h \times k_w \times k_t$ into two subsequent convolutions, one spatial convolution with a kernel $k_h \times k_w \times 1$ and one temporal convolution with a kernel $1 \times 1 \times k_t$. Thereby, the parameters are reduced compared to the full spatio-temporal convolution. Different variants can be designed using this concept, e.g., performing the spatial convolution first, performing the temporal convolution first, or performing the convolutions in parallel [297, 364]. These variants can be mixed across an architecture, and also only a few of the convolutions can be replaced with factorized spatio-temporal convolutions [505]. Also, an even more drastic variant has been presented, where the full spatio-temporal convolution is

factorized, i.e., a 3D convolution is replaced by three subsequent one-dimensional convolutions [513]. Recently, additional ways to replace the costly spatio-temporal convolutions have been presented, such as channel-separated convolutional networks [458] or channel tensorization [270]. The former separates channel interaction and spatio-temporal interaction of a 3D convolution into two separate convolutions. The latter divides the channel dimension into sub-dimensions and subsequently performs a convolution on each sub-dimension to reduce the overall complexity. Another way to reduce the model complexity is to combine a 2DCNN and a 3DCNN for sequences of 2D images. This concept can be used by first learning a spatial representation of individual images of a sequence using a 2DCNN, which are then processed jointly by a 3DCNN [505, 545]. Note that the spatial representation of the individual images is smaller in size than the original image size, which reduces the computational efforts of the joint processing. Such an approach is similar to multi-path architecture concepts that we introduce in Section 4.2.

Typically, video analysis in the natural image domain involves large-scale video datasets with millions of videos ranging over several minutes [69, 369, 408]. To address this challenging data processing task, additional concepts have been proposed to improve efficiency and to reduce model the complexity of 3DCNNs, which learn temporal dependencies between frames with a 2DCNN across the architecture based on the channel dimension [277, 297, 478, 490]. Disentangled data processing within a CNN can be used as a form of regularization and to improve efficiency. However, this comes at the cost that joint feature learning can not directly be performed, or only partly be performed, which limits the discriminative power and ultimately may limit the performance. Hence, architecture choices can be made based on a trade-off between performance and efficiency [69,369,438].

Summarized, spatio-temporal feature learning with CNNs can be performed based on an aggregation of image-wise extracted features [369, 508, 526], based on channel interactions [231, 354], or based on spatio-temporal kernels [17, 224, 256, 453, 457]. Moreover, architecture choices can be made that lead to a trade-off between performance and efficiency [69, 277, 297, 478, 490]. Overall, considering the different methods and the task of video classification, there is no single best method that outperforms all other methods across all datasets [69,369,438]. Moreover, recent efforts in the natural image domain tend to focus on improving the efficiency of spatio-temporal deep learning for video classification rather than accuracy [69] due to the availability of large-scale training datasets that require learning from millions of videos. It stands out that the complexity of spatio-temporal feature learning and the increased computational efforts of video data processing require tailored networks to learn effectively and efficiently [503]. This makes architecture design and method development for spatio-temporal data processing an active research area with several architecture choices, with new methods constantly being developed. Spatio-temporal CNNs have been studied extensively in the natural image domain and are also recently gaining traction in the medical domain. However, there are only a limited number of applications so far. In

particular, 4D multi-dimensional medical image sequence processing has hardly been studied. Considering medical applications, early adopters that used spatio-temporal CNNs within the last years presented, e.g., fetal heart analysis from 2D US videos [350], automatic 2D US video description [407], surgical tool tracking in 2D endoscopic videos [538], characterization of parkinsonian gait from 2D videos [185], classification of Alzheimer's disease using fMRI data [348], or cardiac left ventricle quantification using sequences of 2D MRI images [467]. These studies show promising results regarding the ability of spatio-temporal CNNs to learn complex features end-to-end from sequences of medical image data.

4.1.2 Our Approaches

Parts of this section have been published in our studies presented in [37–40, 42, 164]. Building on previous work for video processing with deep learning from the natural domain, we introduce a compact and modular spatio-temporal CNN approach for multi-dimensional medical image sequences. This single-stream approach can be scaled to different dimensions and adapted to different tasks. We develop and present our methods for end-to-end regression considering our two application scenarios of this work, i.e., motion analysis and dynamic elastography. In this context, we evaluate and present different concepts for spatio-temporal data processing that can be integrated and exchanged using the same uniform architecture concept. In particular, we present 4D spatio-temporal CNNs that can be used to process sequences of volumetric images in an end-to-end fashion. Instead of using a fixed architecture that has been proposed for 2D image analysis such as AlexNet [248] or variants of ResNet [197] that would lead to substantial computational requirements when scaled directly to 3D/4D spatio-temporal data processing, we only build on these state-of-the art concepts and design a custom spatio-temporal CNN architecture.

Let $x_t = [x^{[0]}, x^{[1]}, \ldots, x^{[n]}]$ with $x^{[i]} \in \mathbb{R}^{n_h \times n_w \times n_d \times n_c}$ be an image sequence our goal is to estimate the corresponding sequence label $y \in \mathbb{R}^{n_y}$. We denote the current time point with $t = n$. Such a scenario is relevant when we assign a sequence-level label, e.g., during dynamic elastography, where we try to estimate a relevant tissue parameter based on an image sequence. Our baseline architecture concept to this end is shown in Figure 4.2. Inspired by image and video classification architectures from the natural image domain [197, 210, 216, 442, 443, 504], it consists of an initial convolutional head with one up to multiple convolutional layers. To reduce the computational requirements and to quickly increase the receptive field of the CNN, the last convolutional layer of this convolutional head can be used with a kernel stride larger than one. Then, we use architecture modules that represent subsequent architecture blocks. These architecture blocks consist of multiple convolutional layers and are state-of-the-art backbone concepts such as ResNet [197], Inception [216, 442, 443], DenseNet [210], or ResNeXt [504] that we introduced in Section 3.2. Between the architecture blocks, we use transition layers that are used for downsampling of the temporal and spatial dimensions. These transition layers are average pooling layers and allow for flexible downsam-

pling of the input sequence, i.e., only the spatial dimensions can be downsampled while keeping the size of the temporal dimension or vice versa. We use batch normalization [216] for all convolutional layers and employ the rectified linear function [174] as the activation function except for our output layer and use zero padding for the inputs. Lastly, we use a GAP layer that transforms the tensor representation into a feature vector, which is then fed to the output layer. The output layer is a fully-connected layer with single or multiple outputs $\widetilde{y} \in \mathbb{R}^{n_y}$. Depending on the output activation function, it can be adapted to regression or classification tasks. In our work, we focus on regression tasks and hence use the identity function as the output activation function.
The width (number of feature maps) and depth (number of blocks and layers) of our baseline framework can be adjusted to address different input sizes and complexities of learning tasks. These specific architecture choices can be considered hyperparameters and selected based on the validation set performance.
An advantage of our spatio-temporal CNN framework is that a similar architecture can be used for several types of inputs and spatio-temporal data processing strategies, as outlined in the following paragraphs. We use this baseline architecture for 3D and 4D spatio-temporal medical image data by combining the architecture concept with several different types of convolutions for 3D and 4D spatio-temporal data processing. That is, we use the same architecture but only replace the operations used for the layers. We outline the different operations in the following paragraphs, which results in our different spatio-temporal CNN approaches.

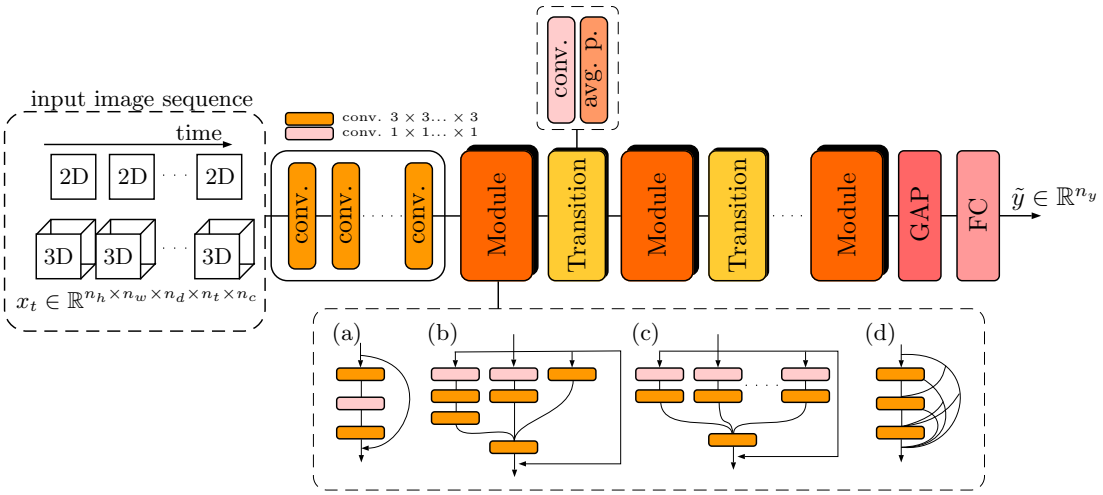

Figure 4.2: Our spatio-temporal CNN approach for multi-dimensional medical image sequences. Our custom architecture consists of an initial part with subsequent convolutional layers, followed by architecture modules representing architecture blocks. The different backbone architecture concepts are (a) ResNet, (b) Inception (c), ResNeXt, and (d) Densenet. After the last module, we use a GAP layer, and the output is connected to a fully connected output layer (FC). Figure adapted from [38].

TC-3D/4DCNN. As a baseline approach, we use the concept of using the channel dimension as a temporal dimension [231, 354]. In this way, we try to learn spatio-temporal relationships at the pixel-level of the input using channel-wise interactions. With this early fusion strategy, the temporal dimension is processed only in the first layer of the network. As outlined in the previous section, no joint spatio-temporal feature learning can be performed with this approach. The advantage of this approach is that it leads to a similar architecture compared to single image processing with similar computational costs. We refer to this approach as time-channel with the abbreviation TC. Combing this concept with our baseline architecture leads to 3D and 2D network operations for 4D and 3D spatio-temporal data, respectively.

ST-3D/4DCNN. To learn joint features from 4D data in an end-to-end fashion, we adopt the approach of 3D spatio-temporal convolutions to 4D spatio-temporal data [17, 224, 256, 453, 457]. Notably, learning joint features from space and time using 4D data requires 4D operations for a network. This is largely unexplored in the field of deep learning and an interesting research direction to address end-to-end learning from 4D data [81]. A 4D spatio-temporal convolution can be considered and implemented as multiple time-shifted 3D convolutions [533]. By striding the time-shifted 3D convolutions across time, we can perform a convolution across 3D space and time, see Figure 4.3. Following our notation from Section 3.2, and assuming zero padding and a stride s = 1, it can be written as

$$\text{conv4D}(\mathrm{K}, x)_o = \sum_{i_t=1}^{k_t} \text{conv3D}(\mathrm{K}^{[i_t]}, x^{[o_4+i_t-1]})_{o_1,o_2,o_3} \tag{4.1}$$

The weights of the 3D convolutions are given by $\mathrm{K}^{[i_t]} \in \mathbb{R}^{k_h \times k_w \times k_d \times n_c}$. To compute the final output of the layer we typically add the bias term and apply a non-linear activation function afterwards, as outlined in Section 3.2. We refer to the approach of spatio-temporal convolutions as ST. Combining 4D spatio-temporal convolutions and pooling operations with our CNN architecture results in a 4D spatio-temporal CNN.

F-ST-3D/4DCNN. As a parameter-efficient version of spatio-temporal convolutions, we also consider factorized spatio-temporal convolutions [364] and split a full spatio-temporal convolution into a temporal and a spatial convolution. We refer to this approach as *F-ST* for factorized spatio-temporal convolutions.

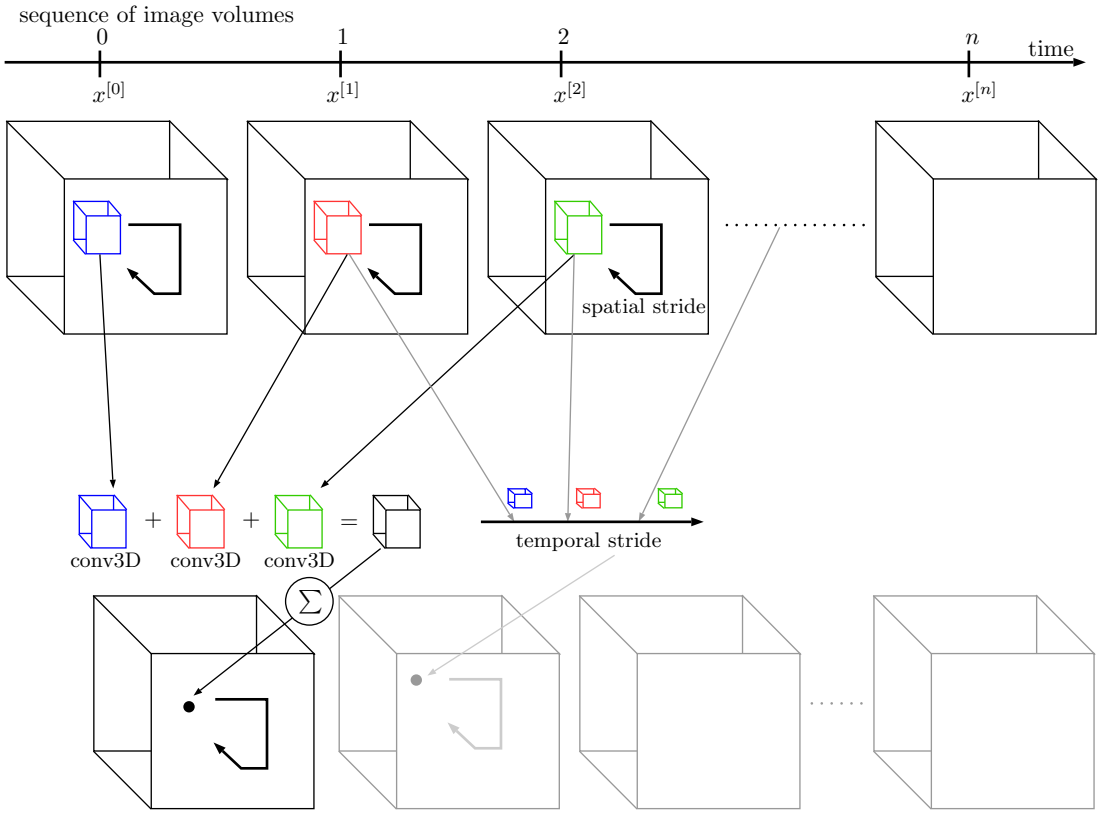

Figure 4.3: 4D spatio-temporal convolution. A 4D spatio-temporal convolution can process a sequence of volumetric images. It can be formulated based on multiple time-shifted 3D convolutions.

In summary, we use three different concepts to process spatio-temporal medical image data with a CNN: spatio-temporal feature learning based on channel-wise interactions, spatio-temporal kernels, and disentangled spatial and temporal processing. Our different types of convolutions for spatio-temporal data processing are visualized in Figure 4.4.
To address the increased model complexity of spatio-temporal CNNs that may result in overfitting, transfer learning methods have been presented for video analysis tasks [63, 110]. Transfer learning can be performed by pre-training on large-scale public video datasets before fine-tuning the task at hand or by transferring weights from a 2DCNN to a 3D spatio-temporal CNN [63]. The latter can be performed by inflating the 2D kernels $\mathrm{K} \in \mathbb{R}^{k_h \times k_w \times n_{in} \times n_{out}}$ of a 2DCNN, pre-trained on an image processing task, to a 3DCNN by copying the pre-trained weights along the additional temporal dimension. We also evaluate this aspect in our work and study to transfer weights from a 3DCNN trained on single volumes to a 4D spatio-temporal CNN that processes the entire sequence. This can be performed seamlessly with our unified architecture concept. We perform this step to address the increased computational requirements of 4DCNNs during training compared to 3DCNNs. Our approach is shown in Figure 4.5. First, we use our architecture concept with 3D operations and train the network end-to-end with single volumetric images. Then, we inflate the kernels along the temporal dimension by copying the weights and fine-tuning the resulting 4D network

with entire image sequences afterwards. We present and evaluate this transfer learning concept for position estimation of a marker object from sequences of OCT volumes in our experiments in Chapter 6.1. Overall, we present a compact spatio-temporal CNN approach for end-to-end regression using multi-dimensional medical image sequences. Our modular architecture can be used flexibly for different imaging modalities and tasks. Our architecture allows comparing different methods for spatio-temporal data processing with deep learning, which is relevant to address the question of how spatio-temporal feature learning from sequences of multi-dimensional medical image data can be performed. In our study, we use our architecture for motion analysis and elastography using OCT and US image data. Our architecture is designed and developed to map an entire sequence of images to a single output, i.e., a sequence-level label. This is relevant when a single output or label is assigned to an entire sequence. In our study, we evaluate the concept of spatio-temporal convolutions across various experiments ranging from position estimation to end-to-end elasticity estimation from sequences of OCT and US images. We report the corresponding results in Chapter 6.

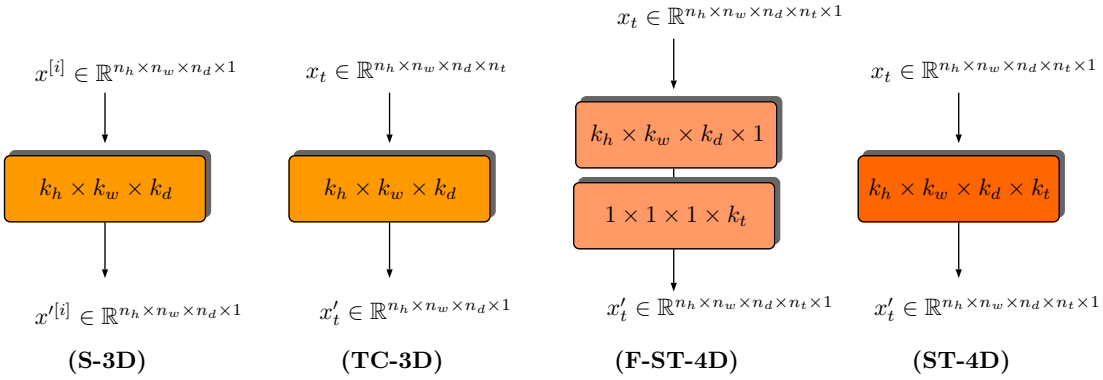

Figure 4.4: Our different types of convolutions for 4D spatio-temporal data processing: (S-3D) 3D spatial convolution that processes a single volume; (TC-3D) 3D convolution with temporal information stacked into the channel dimension; (F-ST-4D) factorized 4D spatio-temporal convolution; (ST-4D) 4D spatio-temporal convolution. Figure adapted from [38].

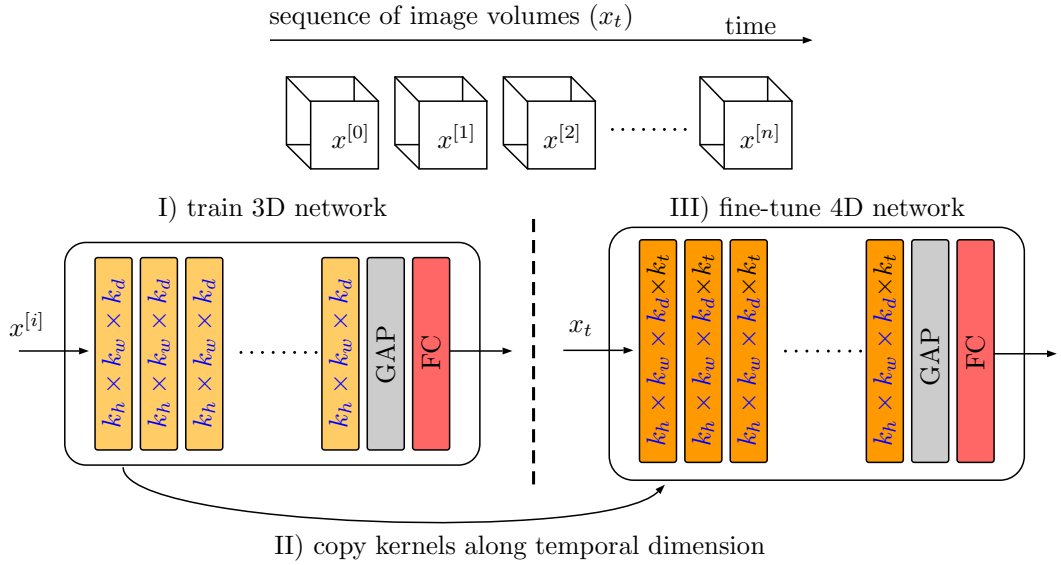

Figure 4.5: Transfer learning from a 3DCNN to a 4D spatio-temporal CNN. First, a 3D network is trained based on single image volumes. Second, the weights of the 3D network are transferred to a corresponding 4D version of the network by copying the weights along the temporal dimension. Lastly, the 4D network is trained based on the entire image sequence.

4.2 Multi-Path Concepts

4.2.1 Background

In scenarios such as video classification or dynamic elastography, we typically assign and predict target for an entire image sequence. In other scenarios, such as motion analysis or disease progression [271, 272], we are usually interested in determining the change between an initial state and different states. In these scenarios, we are interested between changes or similarity of individual time points, which typically requires image/frame-level predictions.

To address such a learning task, Siamese neural networks are widely adopted [72, 321]. Considering 2D images from two time points, $x^{[1]} \in \mathbb{R}^{n_h \times n_w \times c}$, $x^{[2]} \in \mathbb{R}^{n_h \times n_w \times c}$, the concept of Siamese neural networks is to create two separate paths in a CNN architecture that process both images individually, which are combined afterwards to perform joint processing, see Figure 4.6. This approach first learns and extracts encodings of two images, which are then processed jointly to determine the similarity or difference of the encodings [58, 80]. While processing the frames individually, both paths can be identical by using the same set of weights. Thereby, the number of parameters can be reduced, the same set of features is extracted from both images and this step can be considered an image pre-processing step that is learned from data [519]. Inspired by the concept that parameters are shared, such an approach is referred to as Siamese architecture [58, 80]. The final

output $\widetilde{y} \in \mathbb{R}^{n_y}$ of such an approach is given by

$$\widetilde{y} = f_2\left(f_1(x^{[1]}, \mathrm{w}_1), f_1(x^{[2]}, \mathrm{w}_1), \mathrm{w}_2\right) \tag{4.2}$$

where $f_1(\cdot)$ denotes the feature extraction part of the Siamese part with parameters w_1, and $f_2(\cdot)$ denotes the subsequent processing with parameters w_2. Note that $f_1(x^{[1]}; \mathrm{w}_1)$ and $f_1(x^{[2]}; \mathrm{w}_1)$ represent the learned encodings of the input images. Moreover, different fusion strategies can be used to combine the two feature representations before combined processing [369]. A common approach is to stack both feature representations into the channel dimension. More formally, let $x'^{[1]} \in \mathbb{R}^{n'_h \times n'_w \times c'}$, $x'^{[2]} \in \mathbb{R}^{n'_h \times n'_w \times c'}$ be the two corresponding encodings of the input images, then the input for the combined processing is given by $x'_t \in \mathbb{R}^{n'_h \times n'_w \times 2 \cdot c'}$. Other fusion strategies are adding or multiplying both representations element-wise [369]. Another architecture choice is when to fuse the two representations, i.e., at the beginning of a network or close to the final output layer. As also outlined previously, fusing both representations early in the network architecture is typically referred to as early fusion, and fusing both representations close to the network output is usually referred to as late fusion [142, 144]. Comparing these two options, late fusion leads to a more abstract feature representation of the individual images before the combined processing [231].

Siamese architectures can be used to predict the similarity of two images [80, 83, 524], but also to determine changes between two images [47, 118, 257]. One of the first applications of a Siamese approach was signature verification [58]. Recently, it has been shown that with such an approach, pixel-wise optical flow can be estimated between 2D images [118], and that object tracking can be performed using image pairs [47, 70, 338, 355]. Considering medical image data and applications, e.g., it has been shown that with Siamese architectures, motion estimation for compensation can be performed using two volumetric OCT images [168]. Moreover, Siamese architectures have been demonstrated for the evaluation of disease severity and change over time using image pairs [271, 272]. In the following section, we present our adaptions, which extend the concept of Siamese architectures to entire sequence processing, which allows to incorporate past or intermediate information of a sequence. Considering our research questions of this work, we hypothesize that more consistent and improved performance can be achieved by using additional temporal information.

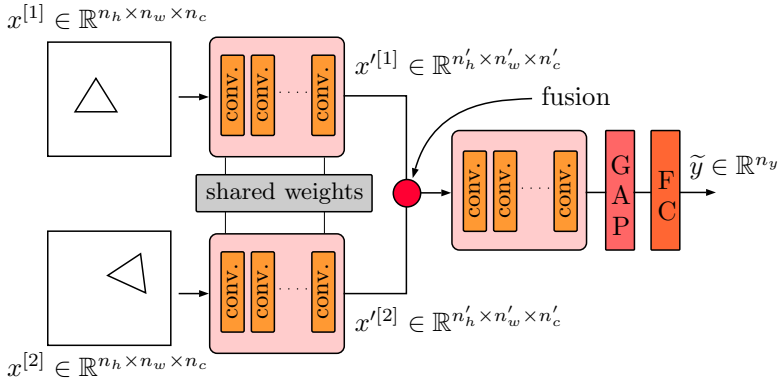

Figure 4.6: Siamese CNN architecture. Two input images $x^{[1]}$ and $x^{[2]}$ are first processed individually, afterwards the representations of both images $(x'^{[1]}, x'^{[2]})$ are fused and processed jointly to obtain the final output $\widetilde{y}$. The final output can be, e.g., a similarity score or distance vector between the images. Figure based on [80].

4.2.2 Our Approaches

Parts of this section have been published in our studies presented in [38–40, 42]. Building on the concept of Siamese architectures [58] and mixed 2D/3D spatio-temporal CNNs [505, 545], we adapt our spatio-temporal CNN framework introduced earlier, to estimate changes between two-time points. We focus our approach on the application scenario of motion analysis using a sequence of volumetric images. Our different methods are shown in Figure 4.7, and we explain each of our variants in detail in the following paragraphs.

Let $x_t = [x^{[0]}, x^{[1]}, \ldots, x^{[n]}]$ with $x^{[i]} \in \mathbb{R}^{n_h \times n_w \times n_d \times n_c}$ be an image sequence with corresponding labels $y_t = [y^{[0]}, y^{[1]}, \ldots, y^{[n]}]$ with $y^{[i]} \in \mathbb{R}^{n_y}$, our goal is to estimate the relative changes between the initial state $x^{[0]}$ and the current state $x^{[n]}$, or to estimate changes between volume pairs in general, e.g., between $x^{[i-1]}, x^{[i]}$. The labels could refer, e.g., to a relative motion of a target shown in the volumes, i.e., $y^{[n]} = T_{0,n} \in \mathbb{R}^3$ between $t = 0$ and $t = n$, or between two consecutive time points, hence $y^{[i]} = T_{i-1,i} \in \mathbb{R}^3$. Note that we use the lower index to indicate the two-time points.

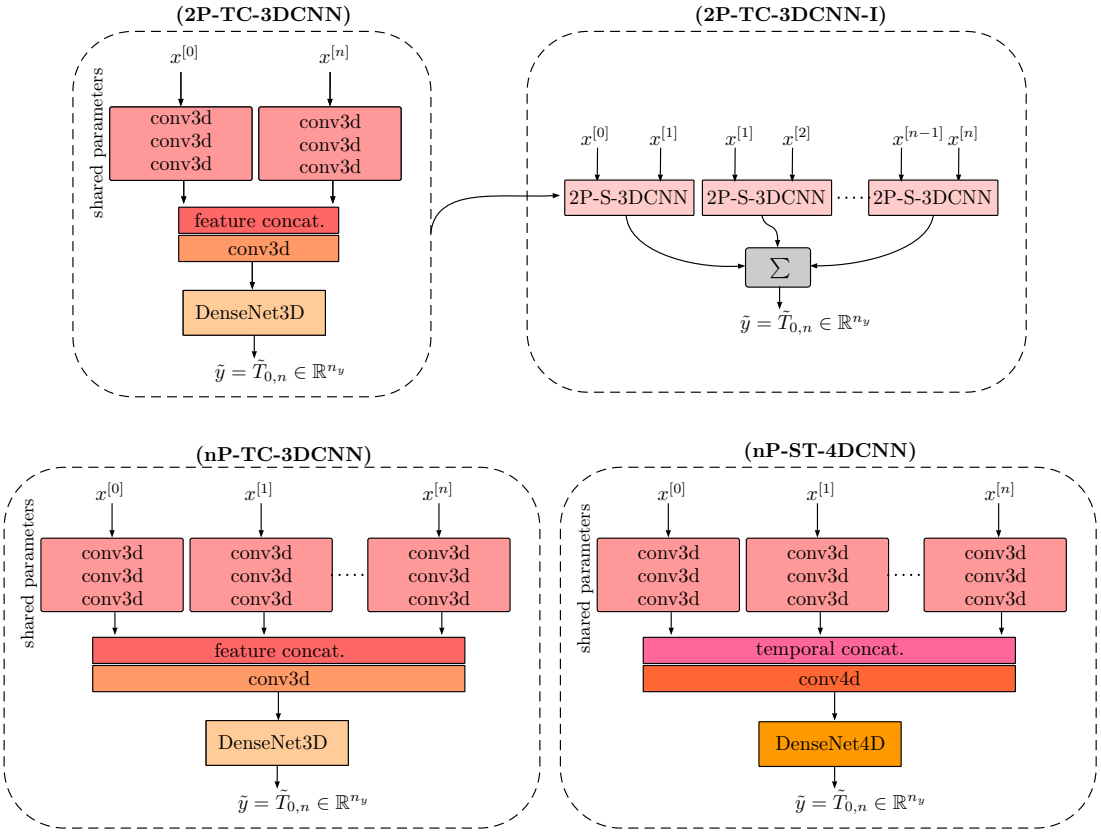

Figure 4.7: Our multi-path 3D/4D architecture concepts. The networks receive volumes from a sequence $x_t = [x^{[0]}, x^{[1]}, \ldots, x^{[n]}]$ to estimate, e.g., the translation vector $\tilde{T}_{0,n}$ between the volumes $x^{[0]}$ and $x^{[n]}$. For processing after the multi-path part of the architecture, we use a custom DenseNet3D/4D architecture to estimate the final output $\tilde{y}$. Figure adapted from [40].

2P-TC-3DCNN. First, as a baseline, we adapt the concept of a Siamese architecture with two paths, where two input volumes are first processed individually, followed by a combined processing [167]. Using our CNN framework, we use the initial head of our architecture that consists of several convolutional layers as Siamese part with shared parameters to process the two images. Then, we combine both representations of the two images at a concatenation point by stacking both representations into the channel dimension. Given two input volumes, $x^{[0]}, x^{[1]} \in \mathbb{R}^{n_h \times n_w \times n_d \times n_c}$ and the corresponding feature representations $x'^{[0]}, x'^{[1]} \in \mathbb{R}^{n'_h \times n'_w \times n'_d \times n'_c}$, this yields $x'_t \in \mathbb{R}^{n'_h \times n'_w \times n'_d \times 2 \times n'_c}$. Subsequently, we process the combined input x'_t jointly with the architecture modules of our CNN architecture. We refer to this architecture as 2P-TC-3DCNN.

2P-TC-3DCNN-I. Second, we extend the two-path concept to entire sequences by dividing a sequence into pairs, which are processed individually, and the results are combined afterwards. Notably, this approach shows similarities to TSN for video-based action recognition [479]. Given an input sequence $x_t = [x^{[0]}, x^{[1]}, \ldots, x^{[n]}]$, the two-path network receives input pairs $[x^{[0]}, x^{[1]}]$, $[x^{[1]}, x^{[2]}]$,..., $[x^{[n-1]}, x^{[n]}]$ and outputs are estimated for each image pair, which are then added

to obtain the final network output $\widetilde{y}$. This can be considered as late fusion of the information from the image pairs. Note that we share the same architecture and parameters for all input pairs. In this way, the network receives information from an entire sequence while still using the simple approach of two-path architecture. To estimate a difference or change between $x^{[0]}$ and $x^{[n]}$, the network can be trained in an end-to-end fashion based on the corresponding label, or the network can be trained based on labels for the individual image pairs. The latter approach only requires estimating the pair-wise labels during training and hence is less computationally expensive and can be used to learn from long-term sequences of several hundreds of volumes. We refer to this architecture as 2P-TC-3DCNN-I. We use the abbreviation I to indicate that the final estimation is based on multiple incremental (I) estimations.

nP-TC-3DCNN. Third, we extend the two-path architecture to a multi-path architecture, where the number of paths is equal to the number of volumes n_t of a sequence that are used as an input. Hence, each image volume $x^{[i]}$ of a sequence x_t is first processed individually with shared parameters, and we learn a feature representation of each volume of a sequence $x'^{[i]} = f_1(x^{[i]}, \mathrm{w}_1)$. Afterwards, we concatenate the representations along the feature dimension, this yields $x'_t \in \mathbb{R}^{n'_h \times n'_w \times n'_d \times n_t \cdot n'_c}$. Similar to the two-path approach, we then perform combined processing with a 3DCNN. Hence, the output of the network is given by

$$\widetilde{y} = f_2\left(f_1(x^{[0]}, w_1), f_1(x^{[2]}, w_1), \ldots, f_1(x^{[n]}, \mathrm{w}_1), \mathrm{w}_2\right) \tag{4.3}$$

Thereby, also intermediate or past information can be used now in an early or mid-fusion fashion. Note that we can use the same architecture and approach compared to the two-path approach, the only difference is that we use multiple paths which are fused at the concatenation point. We refer to this architecture as nP-TC-3DCNN.

nP-ST-4DCNN. Fourth, we combine the concepts of 4D spatio-temporal CNNs and multi-path Siamese architectures. At first, we use a multi-path Siamese 3DCNN to process each volume of the sequence. Afterwards, we concatenate the outputs along the temporal dimension, this yields $x'_t \in \mathbb{R}^{n'_h \times n'_w \times n'_d \times n_t \times n'_c}$ after the fusion point. In this way, we first learn a spatial representation of each input volume of a sequence. In a next step, we perform joint spatio-temporal feature learning by means of a 4D spatio-temporal CNN. Thereby, much more complex spatio-temporal relationships can be learned compared to the approach where the feature or channel dimension is used as temporal dimension. However, using 4D operations for the combined processing comes at increased computational costs compared to the previous approaches. Moreover, this approach is similar to using a 4D spatio-temporal CNN applied directly to a sequence. However, the mixed 3D/4D approach reduces parameters by sharing parameters for the Siamese part. In addition to that, the computational efforts can be reduced compared to a full 4DCNN by downsampling the image volumes in the Siamese part such that joint-processing is performed on smaller volumes afterwards. We refer to this approach as nP-ST-4DCNN. The abbreviation ST refers to the fact that spatio-temporal

convolutions are performed. Such an approach can also be considered a mixed 3D/4D spatio-temporal CNN similar to mixed 2D/3D spatio-temporal CNNs that have been proposed for video analysis [505, 545].

Summarized, we present various different multi-path architecture concepts for 4D spatio-temporal data that are suitable to estimate changes between two-time points. These concepts allow to integrate additional temporal information to improve performance and consistency. These architectures differ in complexity and use different concepts for processing the spatio-temporal data ranging from channel interactions to feature learning by means of spatio-temporal kernels. In our study, we use these architecture concepts for motion analysis from short-term OCT sequences and long-term US sequences. We report the corresponding results in Section 6.1.

4.3 ConvRNNs and Hybrid Approaches

4.3.1 Background

RNNs are another deep learning concept that can be used to learn spatio-temporal relations. From a conceptual level, one advantage of such an approach is that different input-to-output relationships can directly be addressed such as many-to-one or many-to-many [176]. In the first case, a single output is generated for a sequence of inputs, and in the second case, an output is generated for multiple or each input element of a sequence. Hence, the output is also a sequence. Another advantage of RNNs is that sequences with variable lengths can be directly addressed, as outlined in Section 3.3.
A regular RNN, as introduced in Section 3.3, processes input vectors $x^{[i]} \in \mathbb{R}^{n_x}$ in a sequential fashion. Hence, given a sequence of images x_t, it requires transforming each image to a feature vector as a first step. One approach is to simply flatten each image $x^{[i]} \in \mathbb{R}^{n_h \times n_w \times n_c}$ of the sequence into a one-dimensional vector $x'^{[i]} \in \mathbb{R}^{n_h \cdot n_w \cdot c}$ [369]. However, this results in substantial amounts of parameters for the RNN architecture. A more efficient way can be achieved with deep learning methods by using a CNN as a feature extractor. To extract the image-wise feature encodings, a 2DCNN can process the images of a sequence individually, or also 3D spatio-temporal CNNs can be used [17, 19, 438, 487, 488, 501]. Afterwards, an RNN can be used to process the sequence of extracted feature vectors in a recurrent fashion and the combined architecture of a CNN and RNN can be trained end-to-end, or the CNN can be used only as a fixed feature extractor [8, 115, 369, 502, 521]. Note that the latter is less computationally demanding. As an RNN architecture, an LSTM or GRU unit is typically used [438], which we introduced in Section 3.3.
Another approach is to replace the linear transformations of RNN units with convolutions [413, 506]. This brings the advantage that we can apply RNNs directly to images, learn localized spatial features, and replace the parameter intensive fully-connected layers in the input-to-state and state-to-state transitions. This also removes the requirement to learn and extract a feature vector from an image

before applying an RNN, and the original data topology is retained. Comparing the two types of RNNs, a recent survey paper on deep learning for video classification concluded that ConvRNNs perform better than regular RNNs [369]. Similar to LSTM units that we introduced in Section 3.3, a ConvLSTM unit [506] is described by

$$z^{[\tau]} = \tanh(s^{[\tau]}) \circ q^{[\tau]} \tag{4.4}$$

with the output gate

$$q^{[\tau]} = h_\sigma \left(\mathrm{w}_0^q + \mathrm{W}^q * z^{[\tau-1]} + \mathrm{U}^q * x^{[\tau]} \right) \tag{4.5}$$

the internal state

$$s^{[\tau]} = f^{[\tau]} \circ s^{[\tau-1]} + g^{[\tau]} \circ \tanh \left(\mathrm{w}_0 + \mathrm{W} * z^{[\tau-1]} + \mathrm{U} * x^{[\tau]} \right) \tag{4.6}$$

with the forget gate

$$f^{[\tau]} = h_\sigma \left(\mathrm{w}_0^f + \mathrm{W}^f * z^{[\tau-1]} + \mathrm{U}^f * x^{[\tau]} \right) \tag{4.7}$$

and the external input gate

$$g^{[\tau]} = h_\sigma \left(\mathrm{w}_0^g + \mathrm{W}^g * z^{[\tau-1]} + \mathrm{U}^g * x^{[\tau]} \right). \tag{4.8}$$

Note that $*$ denotes a convolution. For a 2D image, the kernels of the convolutions are given by $\mathrm{W}, \mathrm{W}^g, \mathrm{W}^f \in \mathbb{R}^{k_h \times k_w \times n_z \times n_z}$, $\mathrm{U}, \mathrm{U}^g, \mathrm{U}^f \in \mathbb{R}^{k_h \times k_w \times n_z \times n_x}$ and the bias terms are given by $\mathrm{w}_0, \mathrm{w}_0^f, \mathrm{w}_0^g \in \mathbb{R}^{n_z}$. Here, n_z denotes the number of output feature channels of the convolution, i.e., the number of parallel convolutions and n_x denotes the number of input channels. An element-wise product is indicated as $\circ$, and h_σ refers to the sigmoid function. Following the same concept, a ConvGRU unit [413] is described by

$$z^{[\tau]} = u^{[\tau]} \circ z^{[\tau-1]} + (\mathbf{1} - u^{[\tau]}) \circ \tanh \left(\mathrm{w}_0 + \mathrm{W} * \left(r^{[\tau]} \circ z^{[\tau-1]} \right) + \mathrm{U} * x^{[\tau]} \right) \tag{4.9}$$

with the update gate

$$u^{[\tau]} = h_\sigma \left(\mathrm{w}_0^u + \mathrm{W}^u * z^{[\tau-1]} + \mathrm{U}^u * x^{[\tau]} \right) \tag{4.10}$$

and the reset gate

$$r^{[\tau]} = h_\sigma \left(\mathrm{w}_0^r + \mathrm{W}^r * z^{[\tau-1]} + \mathrm{U}^r * x^{[\tau]} \right). \tag{4.11}$$

Assuming a 2D input image, the kernels of the convolutions are given by $\mathrm{W}, \mathrm{W}^u, \mathrm{W}^r \in \mathbb{R}^{k_h \times k_w \times n_z \times n_z}$, $\mathrm{U}, \mathrm{U}^u, \mathrm{U}^r \in \mathbb{R}^{k_h \times k_w \times n_z \times n_x}$, and the bias terms are given by $\mathrm{w}_0, \mathrm{w}_0^u, \mathrm{w}_0^r \in \mathbb{R}^{n_z}$. $\mathbf{1}$ indicates a vector where all elements are one.
A direct advantage of a ConvRNN is that the data structure can be preserved, and thereby the concept of a ConvRNN can be combined with a CNN in various ways [131, 211, 531]. Similar to the approach described previously, one widely adopted concept is first to use a CNN to learn an image feature representation for each image of a sequence [115, 486]. Afterwards, the feature representations are processed using a ConvRNN. Usually, the feature representation is smaller

in size than the original image, which reduces the computational efforts compared to applying a ConvRNN directly to the images. Another approach is to use the ConvRNN unit at the beginning of the architecture, followed by a CNN architecture. In this way, a compact spatial representation of the entire image sequence can be learned, and afterwards a CNN can be applied to the learned spatial representation of the entire sequence [169, 170]. Such an approach can be beneficial in scenarios where redundant information is present in the sequence [36] or where the relevant information for the task is mostly encoded in the spatial information [164]. Also, approaches have been presented where ConvRNNs are incorporated in the middle of an architecture, e.g., for encoder-decoder architectures [15, 291, 472].

Introducing both RNNs and CNNs for image sequence processing raises the question of the major differences and advantages of the approaches. A recent comparison of the two concepts concluded that both approaches lead to competitive performance with the following core differences [18]. First, using convolutions instead of recurrent units does not require sequential data processing and hence allows for parallel data processing. Second, using a CNN allows changing the receptive field size and can be used to downsample the input dimensions effectively using different kernel sizes and strides. This leads to a flexible approach that allows to adjust the memory size and complexity of the data processing. Third, RNNs are associated with exploding and vanishing gradients, as outlined previously, and hence are typically more difficult to train [30, 32, 202]. Fourth, the memory units of RNNs require larger memory during training than CNNs. Overall, a simple convolutional architecture can be more effective across diverse sequence modeling tasks than recurrent architectures [18]. Considering RNNs, one distinct advantage is that this approach is highly efficient during evaluation when an output is required for each image of a sequence. This comes from the fact that only the current image needs to be processed to generate the output and the information from the history of the sequence is encoded in the hidden states of the memory units. Another advantage of RNNs is that does not rely on the concept of a receptive field across the time dimension. Hence it can be transferred across tasks and domains that require different amounts of temporal history. Considering medical image data, the concept of ConvRNNs has been used and adapted for applications such as motion analysis from 2D US data [211], hyperspectral image data analysis [36], autism spectrum disorder classification from fMRI [131, 532], force estimation from sequences of OCT images [169, 170], or for tumor growth prediction [531].

4.3.2 Our Approaches

Parts of this section have been published in our studies presented in [36–40, 42]. Typically, sequences of images are processed in a recurrent fashion at the front, or the end of an architecture [115, 164, 211, 293, 319, 461, 486]. The first approach learns spatio-temporal relationships at the level of the original input data. The second approach learns spatio-temporal relationships at the level of abstract fea-

tures. Both concepts bring their individual advantages regarding medical image analysis, raising the question of how spatio-temporal feature learning should be performed. We study spatio-temporal recurrence at different feature levels to address scenarios where spatio-temporal features can be expected at different spatial and temporal scales. For example, during motion compensated radiotherapy, different scales of breathing motion and different target appearances are present [98].

DenseConvGRU. To develop our multi-scale approach, we integrate the concept of ConvRNNs at different feature levels into our unified spatio-temporal CNN framework, see Figure 4.8. Thereby, we can still use the same architecture concept but now perform spatio-temporal data processing using ConvRNNs. In our work, we rely on ConvGRUs, which are simpler and lead to fewer parameters than ConvLSTM. To address multi-scale feature learning, we use ConvGRU modules at the front, middle, and end of the architecture, as indicated in Figure 4.8. Similar to our previous considerations, the first ConvGRU learns spatio-temporal relationships at a level close to the input resolution. In contrast, the latter learns spatio-temporal relationships at the level of abstract features. Between the ConvGRUs, we use our CNN modules, consisting of several convolutional and pooling layers. Note that due to the recurrent processing of the sequence, the CNN part of the architecture shares parameters across the different images of a sequence. Thereby abstract feature representation can be learned between the ConvGRU modules, the image size can be reduced, and the receptive field of the ConvGRUs w.r.t. the input image increases. The ConvGRU module at the end of the architecture has a large global receptive field at a low spatial resolution, and the ConvGRU module at the front has a small localized receptive field at a high spatial resolution. In particular, by using different placements in a unified architecture, we can also perform a systematic study w.r.t. the performance. Our architecture concept can be used for 3D and 4D spatio-temporal data, and the architecture depths can be scaled, and additional ConvGRU modules can be used. Next, we describe the concept of our architecture more formally.

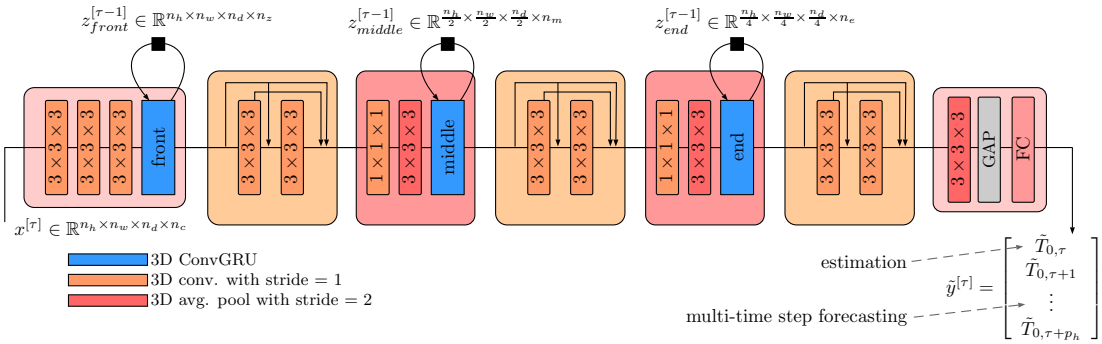

Figure 4.8: Our architecture DenseConvGRU-all: We perform multi-scale spatio-temporal processing by incorporating ConvGRU module at different features scales, denoted by front, middle, and end. For our CNN modules, we use DenseNet blocks. We perform end-to-end estimation and multi-time step forecasting with our approach. We indicate a delay of a single time step with a black square. Figure adapted from [42] (©2023 IEEE).

Let $x^{[\tau]} \in \mathbb{R}^{n_h \times n_w \times n_d \times n_c}$ be the network input at a time point $t = \tau$, first subsequent convolutional layers are applied to the input volume. Then, the first ConvGRU module (front) receives a feature representation $x'^{[\tau]} \in \mathbb{R}^{n_h \times n_w \times n_d \times n'_c}$ and the previous output of the ConvGRU module $z_{front}^{[\tau-1]} \in \mathbb{R}^{n_h \times n_w \times n_d \times n_z}$ to generate the output $z_{front}^{[\tau]} \in \mathbb{R}^{n_h \times n_w \times n_d \times n_f}$ according to Equation 4.9. The number of feature maps of the ConvGRU module at the front is denoted by n_f. Afterwards, multiple convolutional layers and a pooling layer are applied, leading to $x''^{[\tau]} \in \mathbb{R}^{\frac{n_h}{2} \times \frac{n_w}{2} \times \frac{n_d}{2} \times n''_c}$ which is processed by the second ConvGRU module that is placed in the middle of the architecture and that outputs $z_{middle}^{[\tau]} \in \mathbb{R}^{\frac{n_h}{2} \times \frac{n_w}{2} \times \frac{n_d}{2} \times n_m}$. The third module receives $x'''^{[\tau]} \in \mathbb{R}^{\frac{n_h}{4} \times \frac{n_w}{4} \times \frac{n_d}{4} \times n'''_c}$ as input and outputs $z_{end}^{[\tau]} \in \mathbb{R}^{\frac{n_h}{4} \times \frac{n_w}{4} \times \frac{n_d}{4} \times n_e}$, respectively. Summarized, the outputs of the different ConvGRU modules are denoted as $z_{front}^{[\tau]}$, $z_{middle}^{[\tau]}$, $z_{end}^{[\tau]}$. The last layers of our architecture are a GAP layer followed by a regression layer to estimate the target values $\widetilde{y} \in \mathbb{R}^{n_y}$. We refer to this architecture concept as DenseConvGRU-all.
Our DenseConvGRU approach can be applied to a sequence of medical image data in a many-to-one or many-to-many fashion and hence can be adapted to different tasks and output requirements, see Figure 4.9. Considering applications that require an output for each image of a sequence, e.g., motion analysis, our approach maps an input sequence of image volumes to an output sequence of motion estimates and predictions in a many-to-many fashion. Motion outputs are given by the network's output $\tilde{y}^{[\tau]}$ in one shot for each input volume $x^{[\tau]}$, and motion analysis can be performed for an ongoing input sequence. Note that this is efficient for tracking tasks. To perform motion estimation only the new input volume $x^{[n]}$ at the current time point $t = n$ needs to be processed by the network, and the history of the inputs is encoded at different scales in the previous outputs of the ConvGRU modules, i.e., $z_{front}^{[n-1]}$, $z_{middle}^{[n-1]}$, $z_{end}^{[n-1]}$. Considering the case where a single output is estimated for an entire sequence, e.g., during shear wave elastography, the entire sequence is processed with our architecture, and a single output is generated afterwards.

Summarized, we present DenseConvGRU, a hybrid approach of ConvGRU modules and a CNN built based on state-of-the-art architecture modules. This approach performs spatio-temporal processing in a recurrent fashion. A key concept of our approach is spatio-temporal recurrence at different feature scales that allow learning localized and global spatial-temporal relationships. In addition to that, combined with our architecture concept, it allows for a systematic evaluation of different ConvGRU module placements. However, a critical aspect of DenseConvGRU is training with long-term sequences. Hence, we present a specific training approach in our next Section 4.4. In this thesis, we use our concept for motion analysis from long-term 4D US sequences in the context of radiotherapy. We report the corresponding results in Section 6.1.3. Moreover, we evaluate this architecture concept for dynamic elastography using 4D OCT data and report our results in Section 6.2.4.

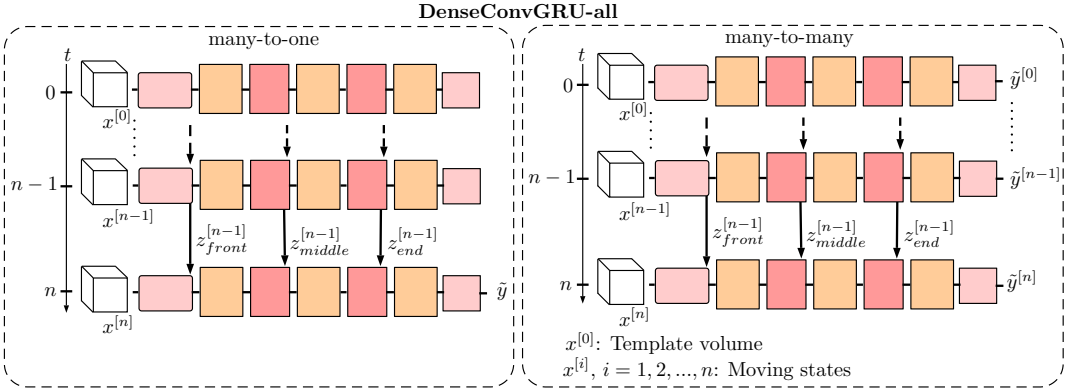

Figure 4.9: (Left) Many-to-one fashion for dynamic elastography, where an output is estimated after processing an entire sequence, i.e., a sequence level estimation is performed. (Right) Many-to-many fashion for motion analysis, where an output is generated for each input of a sequence, i.e., image-level estimations are performed. At the current time step $t = n$, only the current input volume $x^{[n]}$ needs to be processed and the motion history is encoded in the previous outputs of the ConvGRU modules at different scales ($z^{[n-1]}_{front}$, $z^{[n-1]}_{middle}$, $z^{[n-1]}_{end}$). Figure adapted from [42] (©2023 IEEE)).

4.4 Training Strategies

4.4.1 Multi-Task Learning and Loss-Regularization

In scenarios where multiple tasks share a same input x, multi-task learning can be used for regularization or to address multiple tasks simultaneously [34, 64]. The following paragraph is based on [378]. The idea of multi-task learning is to use a single approach for different tasks and to share parameters across the different tasks. Thereby, a model is enforced to learn more robust features that generalize well in different situations. For regularization, an auxiliary task is defined only for training, and the output of the auxiliary task is not required during testing. With multi-task learning, the loss function during training combines the individual loss functions of the different tasks. From an architecture perspective, one strict approach to performing multi-task learning is to share parameters of an architecture across different tasks until the final output layers of the different tasks. This is usually called hard parameter sharing and only requires to increase the number of output units. Another approach is soft parameter sharing, where only a part of an architecture is shared across the tasks, and afterwards, the architecture is split into different branches for the different tasks. Thereby, shared and individual features are learned to address the different tasks. Naturally, hard parameter sharing leads to a stronger regularization effect, a simpler model, and fewer parameters. However, usually multi-task learning increases the labeling efforts for each input as labels w.r.t. various tasks need to be assigned. Also, a larger model might be required to address multiple tasks simultaneously, increasing the computational efforts compared to addressing a single task with a smaller model.

Parts of this section have been published in our studies presented in [38–40, 42]. We consider and present multi-task learning combined with loss-regularization for motion analysis from sequences of multi-dimensional medical image data. At first, we study whether additional temporal information can also be beneficial at the model output by formulating an auxiliary task for regularization. Typically, we consider the MSE or MAE loss function between the estimated output $\widetilde{y} = \widetilde{T}_{0,n} \in \mathbb{R}^3$ and the ground truth $y = T_{0,n} \in \mathbb{R}^3$ during motion analysis. Recall that $T_{0,n}$ refers to a translation vector or motion of a target between the time points $t = 0$ and $t = n$. Using a MSE loss function, this leads to

$$\mathcal{L} = \frac{1}{m_b} \sum_{j=1}^{m_b} \left\| T_{0,n}^{\{j\}} - \widetilde{T}_{0,n}^{\{j\}} \right\|^2 \tag{4.12}$$

with m_b for the number of samples in a training batch. Often, we perform motion estimation for each volume of a sequence, i.e., we perform motion estimation for a continuous stream of volumes. Note that the label of an individual volume is, e.g., the corresponding position or motion vector of a target shown in the volume. Considering this, we propose and evaluate a custom loss function that incorporates past labels of a sequence into the training process for regularization by formulating an auxiliary task of estimating past motions in parallel. We hypothesize that the additional motion information could improve the performance and consistency. To this end, we extend the output of the model to also include past estimates and incorporate past label information into the training loss function, which leads to

$$\mathcal{L} = \frac{1}{m_b} \sum_{j=1}^{m_b} \frac{1}{p_y} \sum_{i=0}^{p_y} \theta_i \left\| T_{0,n-i}^{\{j\}} - \widetilde{T}_{0,n-i}^{\{j\}} \right\|^2 \tag{4.13}$$

with parameters $\theta_{i=0} = 1$ and $\theta_{i\neq 0} \in [0, 1]$ for weighting of the loss of the past estimates. Note that during inference, the additional outputs are not required. We define the number of past labels as p_y . Note that in this scenario the combined network output is extended to $\widetilde{y} \in \mathbb{R}^{3 \cdot p_y}$.

An advantage of multi-task learning for multi-dimensional medical image sequences may not only improve generalization but also that multiple tasks can be addressed at the same time, as outlined previously. For our application scenario of motion analysis, we use entire image sequences as input that encode the spatio-temporal information of a target's motion. We hypothesize that such data can be used to address both motion estimation and forecasting with a single model. To use a simple single-stream architecture, we perform hard parameter sharing. That is, it only requires adapting the training loss function and changing the number the model's outputs accordingly. Using future labels and extending the model output this yields

$$\mathcal{L} = \frac{1}{m_b} \sum_{j=1}^{m_b} \frac{1}{p_h} \sum_{i=0}^{p_h} \left\| T_{0,n+i}^{\{j\}} - \widetilde{T}_{0,n+i}^{\{j\}} \right\|^2 \tag{4.14}$$

for the loss function with p_h for the prediction horizon. We study these concepts for motion analysis using 4D OCT and US data in our experiment Section 6.1.

4.4.2 Training with Long-Term 4D Data

Parts of this section have been published in our study presented in [42]. We presented our approach DenseConvGRU which can be used to process multi-dimensional medical images in a recurrent fashion, in Section 4.3.2. This architecture is beneficial for scenarios where an output is required for each input volume in real-time, e.g., during motion analysis. We highlighted that such an architecture is efficient during testing, but training involves forward and backward propagation through the entire sequence, generally referred to as BPTT [491], and as also outlined in Chapter 3. This can quickly lead to substantial computational and memory requirements, which make training impractical, especially for long-term sequences of volumetric images. Also, training a recurrent approach with such data quickly becomes extremely time-consuming, with training times reaching several weeks, even for small datasets. To still address end-to-end data processing of sequences of medical image data, we address the aforementioned limitation, and we present a customized training approach by adopting and combining truncated BPTT [136,221,446,493], and curriculum learning [31]. We show the concept of our training approach in Figure 4.10 for our application scenario of motion analysis.

The idea of truncated BPTT is to perform forward propagation through the sequence, but backward propagation is only performed for a shorter sub-sequence. Thereby, the memory requirements and the time to perform a training step can be reduced substantially. However, learning long-term dependencies might become a difficult task, and convergence might be limited. Therefore, we adopt the idea of curriculum learning [31]. The concept of curriculum learning is to gradually increase the complexity of a learning task and thereby to improve convergence and generalization. Using this concept, we gradually increase the sequence length during training. Thereby training of the network first considers the simpler task of learning short-term spatio-temporal relationships. In a next step, more complex long-term spatio-temporal dependencies are learned. To this end, we divide our training into two phases.

For the first phase of the training strategy, we use a limited short-term sequence that is dependent on the available hardware and dataset, and we perform forward propagation and backward propagation through an entire sequence. The short-term sequence is sampled from the long-term sequence, e.g., only the first few images of a sequence are considered. In this way, the network is trained to learn short-term spatio-temporal relationships. Also, training on short-term sequences is substantially faster than training on long-term sequences. The first phase can also be considered a pre-training step to find suitable weights for learning long-term spatio-temporal relationships. In the second training phase, we increase the forward propagation sequence length but keep the backward propagation sequence length fixed. We define r and b as the sequence length that are used for forward and backward propagation, respectively. In the first phase of training, we set $r = b = \widetilde{n}_t$ and $\widetilde{n}_t < n_t$ where n_t describes the entire sequence length of the input and $\widetilde{n}_t$ describes the length of the sub-sequence. In

the second phase, we gradually increase $\widetilde{n}_t$ over the training epochs and we draw $r \sim \mathcal{U}(b, \widetilde{n}_t)$ from a discrete uniform distribution in each training iteration. Note that we keep b fixed, hence in our second training phase it holds $b < \widetilde{n}_t$, i.e., we perform truncated BPTT. The hyperparameter b and the schedule to increase $\widetilde{n}_t$ are chosen w.r.t. the dataset and hardware constraints. Considering our second research question of this work, how spatio-temporal feature learning can be performed, our approach raises the question of whether generalization from shorter sequences during training to longer sequences of volumetric image data during testing can be performed. We also address this question and study our DenseConvGRU architecture concept combined with our training approach in the context of motion analysis during radiotherapy in our experiment Section 6.1.3.

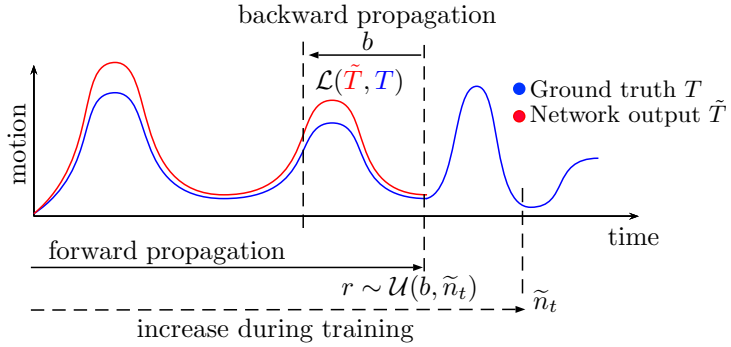

Figure 4.10: Training with long-term sequences for motion analysis. During training, we gradually increase the sequence length $\widetilde{n}_t$ that is used for forward propagation and combine this approach with truncated BPTT. Figure adapted from [42] (©2023 IEEE).

4.5 Summary

In this chapter, we presented our spatio-temporal deep learning methods for multi-dimensional medical image sequence analysis. At first, we introduced how CNNs can be used for image sequence processing and summarized relevant advancements in the field. We highlighted how such an approach could be used to perform sequence-level estimations, i.e., where a final label is assigned to an entire sequence. Findings from the natural image domain demonstrate that video analysis and the resulting complexity of the spatio-temporal feature learning require well-designed networks [369]. While such spatio-temporal CNNs have been extensively studied in the natural image domain, there is limited work considering medical image analysis. In particular, end-to-end processing of volumetric images over time has hardly been investigated. To address the different tasks and dimensionalities arising from medical image analysis, we present a compact and uniform spatio-temporal CNN architecture based on state-of-the-art backbone concepts. We systematically combine this architecture with different strategies for learning spatio-temporal features, ranging from channel-wise interactions to spatio-temporal kernels. Next, we address how changes between images of individual time points can be estimated with spatio-temporal deep learning and high-

light the concept of Siamese CNN architectures. Typically, these approaches only use two images to estimate changes or similarities between two-time points. We hypothesize that more consistent and improved performance could be achieved by using additional temporal information. Therefore, using our unified architecture concept, we present several concepts that allow the integration of intermediate or past temporal information. Lastly, we address long-term sequence analysis of several hundred images from an architecture and training perspective. To this end, we highlight the concept of ConvRNNs and hybrid approaches with a CNN. While spatio-temporal recurrence is typically performed at a single level, using our unified architecture concept, we present the concept of spatio-temporal recurrence at multiple feature levels to learn localized and global spatio-temporal relationships. We hypothesize that this is beneficial for tasks where features are relevant at different scales, such as during motion analysis, where both small and large motions typically need to be estimated. Moreover, exploiting spatio-temporal recurrence makes our approach highly efficient during inference, where image-wise estimations are required for an ongoing sequence of image volumes. Our different methods for 3D/4D spatio-temporal feature learning are summarized in Figure 4.11.

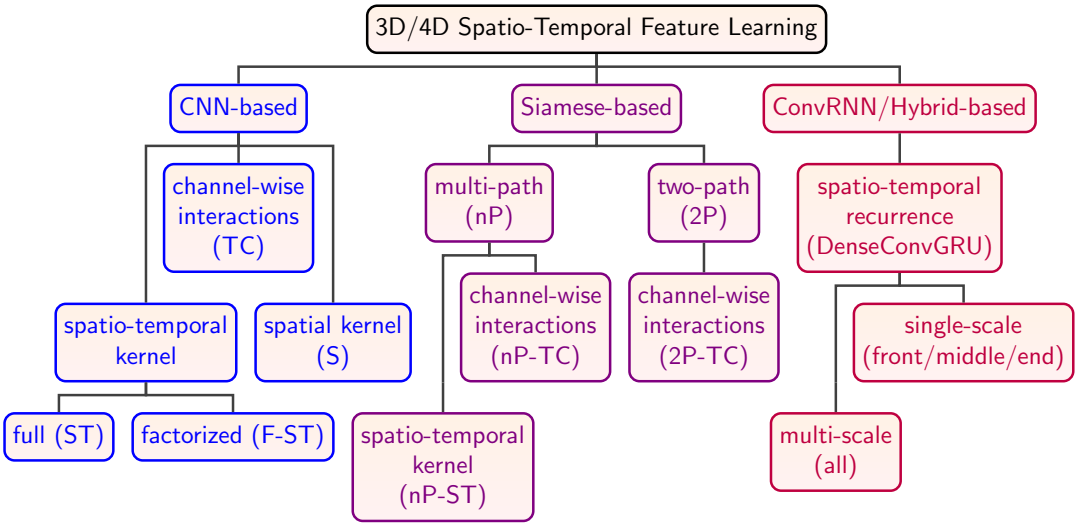

Figure 4.11: Overview of our methods for spatio-temporal feature learning from multi-dimensional medical image sequences. We integrate all these different methods into a unified CNN architecture. (c.f. [369])

We also present different training strategies that allow to integrate additional temporal information for regularization, to address multiple tasks simultaneously, and to learn from hundreds of volumetric images in an end-to-end fashion. As a result of this chapter, we present a unified architecture concept for both 3D and 4D spatio-temporal data that allows for the integration of different network operations and frame/image-level as well as sequence-level estimation. With this concept, only small architecture adaptations are required to address different clinical problems, and imaging modalities and network operations can be exchanged systematically. Ultimately, this raises the question of how the spatio-temporal

feature learning can and should be performed, i.e., whether our systematic approach is able to capture the complex spatio-temporal relationships underlying multi-dimensional medical image sequences. Similar, it raises the question of how the spatio-temporal information influences predictive performance in general. An overview of our different methods in relation to our experiments and application scenarios is given in Figure 4.12. In the next chapter, we present the application scenarios of our methods in more detail and summarize related work.

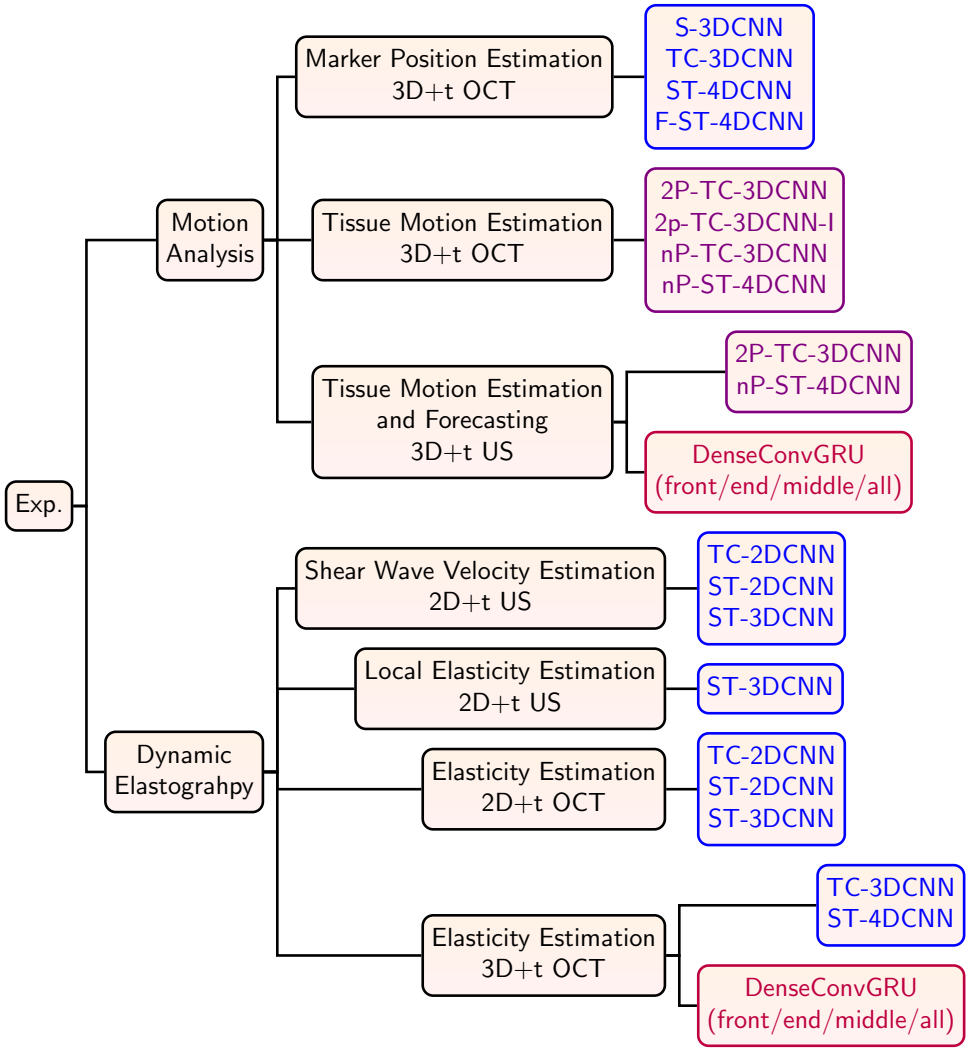

Figure 4.12: Overview of our methods and our application scenarios for our experiments (Exp.).

Chapter 5

Application Scenarios and Related Work

In this chapter, we describe the application scenarios that are considered for studying the two main research questions of this work. This first application scenario is motion analysis of a target from sequences of volumetric medical image data. The second application scenario is dynamic elastography for tissue characterization. Here, we focus on shear wave elastography. For both our application scenarios, we focus on OCT and US as imaging modalities. We highlight the background of our application scenarios, summarize related work, and demonstrate open challenges. Finally, we summarize both application scenarios in the context of our two research questions.

5.1 Position Estimation and Motion Compensation

In this section, we first introduce the relevant background related to motion analysis from volumetric image sequences to perform end-to-end motion estimation of a target. First, we describe the relevant background to estimate the position and motion of a target from image sequences. Second, we consider motion analysis and sequences of US data focusing on motion compensation during radiotherapy and summarize related work. Third, we consider sequences of OCT data and motion analysis related to markerless tissue motion analysis and object tracking. Overall, we highlight recent developments, challenges, and shortcomings related to end-to-end motion analysis from sequences of volumetric US and OCT data.

5.1.1 Background

In this section, we focus on the fundamental background of motion analysis from image sequences with a focus on tracking before going into more depth in the next two sections that focus on our two main imaging modalities and our studied application scenarios. In general, determining motion from image sequences is a fundamental and broad task in computer vision and image processing. A key concept of vision-based motion analysis is to estimate the position and orientation of a target using an image sequence [90]. Performing this step for each image of a sequence allows following the trajectory of a target [518]. Considering the natural image domain, examples include motion analysis of human body

parts [529], autonomous driving [187], or motion analysis in sport [90, 249]. Considering the medical image domain, motion analysis is relevant in various clinical applications such as minimally invasive surgery [13, 55, 76], motion compensation during radiotherapy [328,334,339], analysis of cardiovascular diseases [27], or general motion compensation for visualization systems [6, 337, 544]. Overall, motion analysis from image sequences can be considered a core task of image processing and medical image analysis, and the entire research field is vast and rapidly growing. In particular, vision-based methods bring the advantage that motion analysis and tracking can be performed without major modifications of already complex treatment systems and surgical instruments [55].

Tremendous efforts have been made, and various conventional image processing methods have been proposed to address the task of motion analysis and tracking using image sequences [222, 444, 469, 512, 544]. These methods can be divided into feature-based and area-based methods and aim towards finding correspondence between images that can be used for, e.g., object tracking or image registration [55, 337, 544]. The following paragraph is based on [544]. Feature-based approaches find correspondence between defined and hand-crafted image features, e.g., corners, line intersections, texture, or edges, and do not rely on direct image intensity values [518]. These approaches require feature definition, extraction, and subsequent detection based on a similarity measure. One approach to performing the detection step is brute-force comparison of extracted features using a distance metric [223]. Also, conventional machine learning algorithms such as support vector machines or decision trees can be used in a sliding window fashion on image regions using extracted features to search for the target of interest, e.g., to perform surgical tool detection and tracking [55]. An advantage of using defined features instead of raw intensity values is that increased robustness w.r.t. brightness changes and image noise can be achieved. However, feature-based approaches quickly become difficult for complex high-dimensional medical images due to the task of defining hand-crafted features. To achieve high performance with feature-based approaches, external markers with clear, distinct features for motion analysis can be inserted [6, 55, 337]. However, this increases the effort for integration and can even require surgical implantation in medical applications such as radiotherapy [108]. Area-based approaches find direct correspondence based on image intensity values of a selected area or region without defining a distinct feature, i.e., the step of explicit feature selection is omitted. Hence, the task focuses on matching image intensity values rather than feature detection, and subsequent matching. To detect motion between medical images, correlation-based methods have widely been adopted [146, 284, 337, 383]. These methods search for correspondence between pixel intensities of image regions or even entire images using defined similarity measures. In particular, given a template image extracted from an image region, optimal similarity, i.e., correspondence is searched in a second image [222]. Based on the correspondence of pixel intensities of image regions, motion, i.e., a translation vector between images, can be estimated. A widely adopted approach to perform motion analysis based on a similarity measure such as NCC, that we introduced in Section 2.3, is block matching, where a

template image is extracted from an initial state and then searched in the current image of a sequence based on multiple comparisons [56, 411]. In general, such an approach can also be called template matching [518]. Notably, by using correspondence between a template and a search image, we typically estimate a single translation or motion vector for a target. To estimate pixel-wise displacements between images, optical flow methods have been presented [147, 204, 222, 296]. These methods estimate an entire displacement vector field, i.e., pixels motions of brightness between images [204]. However, optical flow methods generally rely on the assumption that the brightness, i.e., the pixel values of two corresponding pixels in images, remains constant, which can quickly become problematic for sequences of medical images [454].

Recently, deep learning approaches have shown numerous success stories for motion analysis and tracking considering images from the natural domain [303]. For example, it has been shown that deep learning can be used to estimate optical flow from 2D images in an end-to-end fashion [118, 214]. The authors of this approach refer to this network as FlowNet. Also, it has been demonstrated that end-to-end 2D object tracking, or region of interest (ROI) tracking, can be performed with a Fully-Convolutional Siamese Network called SiamFC that combines cross-correlation and learned features of a template and a search region by means of a CNN [47]. Overall, deep learning combined with sequences of 2D images has gained significant attention within the last years and demonstrated promising results to overcome limitations of conventional approaches [11, 303]. However, typical tissue structures and object movements are inherently three-dimensional and for many medical applications, volumetric imaging is advantageous, e.g., for prostate radiation therapy [75]. Although some modalities such as OCT and US provide not only volumetric images but also allow for a high temporal resolution, so far, there is very limited work that has considered spatio-temporal deep learning and end-to-end motion analysis using such data. From a machine learning perspective, this requires designing and performing a regression task between a sequence of volumes and relevant motion parameters, such as a three-dimensional translation vector of a target. Notably, this task needs to be performed in real-time. Considering our research questions of this work, this raises the question of whether end-to-end motion analysis can be performed effectively from sequences of volumetric data with spatio-temporal deep learning.
In this work, we address motion analysis and deep learning using sequences of volumetric OCT and US image data. We study object position estimation and markerless tissue motion analysis using sequences of volumetric OCT images. In addition to that, we study markerless tissue motion analysis using sequences of volumetric US images in the context of radiotherapy. Here, we try to perform motion estimation and forecasting directly from the image sequence. We summarize related studies in more depth in the following two sections and we further highlight open challenges we aim to address with our work.

5.1.2 US-based Motion Analysis

US-based motion analysis is relevant for clinical scenarios such as minimally invasive procedures [125] or motion compensation during radiotherapy [98,211,282, 409, 411]. In this work, we put focus on the latter. Parts of this section have been published in our study presented in [42]. In particular, during radiotherapy motion compensation is required to maintain a fixed geometric relationship between the target that is subject to motion and the beam of ionizing radiation [20, 108, 112]. Moreover, it needs to be performed in real-time and for long-term periods ranging over several minutes. To compensate for these motions and thereby reduce the motion margins during treatment planning, various approaches have been developed, such as moving the beams by a robot [89,128,445]. However, to address the motion compensation task two problems need to be solved: motion estimation and motion forecasting. The latter is required to address system latencies that introduce a lag of several milliseconds between motion estimation and compensation, which can lead to substantial tracking errors [360, 466]. These latencies are typically in the range from 150 ms-500 ms [98, 217, 360], depending on imaging parameters, image processing, and system adjustment latency. To address the motion forecasting task, e.g., methods from time series processing are used to predict the future target position based on the motion history [126]. X-ray-guided motion compensation is successfully used in practice, however, this approach is limited due to the correlation of external and internal motion, the ionizing radiation of the imaging system, and the placement of fiducial markers [108]. Another approach is MRI-guided radiation therapy [280]. However, these systems are rather complex and remain expensive [234]. Recently, US has been proposed to perform the motion estimation task [98, 211, 282, 409, 411]. As outlined in Section 2.2.1, advantages of US include its non-ionizing nature and the high spatial and temporal resolution.

There have been tremendous efforts to improve motion estimation from sequences of 2D US images to address motion compensation during radiotherapy. These methods include block matching with NCC [311, 411], or optical flow [455, 494]. To overcome the limitations of conventional approaches, a range of deep learning methods has been presented for sequences of 2D images. For example, Nouri et al. [335] perform motion estimation in 2D US sequences based on similarity learning of low-dimensional image patch embeddings learned by means of a CNN. Gomariz et al. [175] adopt SiamFC [47] for liver landmark tracking using 2D US sequences. This approach relies on a two-path CNN architecture and performs motion tracking based on correlation between learned features of a reference block (template) and a search region. To incorporate temporal consistency over time, the authors combine the estimation of the current frame with all estimations of landmark positions of the past frames. Liu et al. [282] extend the concept of SiamFC for US tracking and propose a cascaded SiamFC, where the target position is predicted at different resolutions using two Siamese CNNs. Wu et al. [499] also extended the idea of SiamFC and fuse results from multiple resolutions to perform target tracking from ultrasound sequences. Considering the concept of SiamFC, it stands out that mainly two images are used for motion

estimation, which limits temporal consistency and does not make direct use of the complete spatio-temporal information present in an acquired image sequence. The latter requires learning from multiple images. Rangamani et al. [367] propose an encoder-decoder CNN combined with an RNN for estimating landmark heat maps using sequences of 2D US images and demonstrate that using past motion information, i.e., additional past frames improves the performance compared to only using the current frame for prediction. Similar, Huang et al. [211] use ConvLSTMs [506] combined with a CNN to learn from sequences of 2D US image data. To perform the tracking task, this approach uses a CNN to extract features from the images, which are then processed by a ConvLSTMs.

Recently there have been efforts to combine volumetric US imaging with radiation therapy systems [160,218,392,395], and also different conventional approaches for 4D US tracking have been proposed recently. A first approach is to estimate the motion between volume pairs in an incremental fashion or non-incremental fashion [194]. In this context, Harris et al. [194] demonstrate that 4D speckle tracking can be performed using 3D cross-correlation. However, the authors also conclude that while only using a pair of volumes is highly efficient, it limits the performance due to the accumulation of errors or limited correlation between the volumes of a pair. Hence, using multiple volumes for motion analysis was proposed to overcome these limitations. This can be achieved by, e.g., using a multi-frame registration approach [341]. Also, conventional approaches have been developed that consider data from the entire available 4D US sequence [21, 377, 471]. Similar to studies using multiple 2D images, these works demonstrate the benefits of using multiple volumes in terms of performance. However, the substantial data size that results from using multiple volumes increases the run-times, notably up to several seconds [98, 341]. This quickly becomes problematic as motion analysis during radiotherapy typically needs to be performed within a few milliseconds to reduce additional delays between compensation and the actual motion [360].
Moreover, a 2D deep learning approach for landmark tracking from volumetric images has also been adapted, which considers 2D representations of a volumetric template and a volumetric search region by extracting orthogonal planes [212]. While this method shows the advantages of deep learning compared to conventional 4D approaches, it takes about 300 ms for motion estimation. The run-time results from searching a template image in a search region, making real-time motion estimation difficult. Also, a faster deep learning approach has been presented with a run-time of around 125 ms that combines NCC with learned multi-scale features of a template and a search region [195]. Still, this adds a notable lag between the estimation and the actual position of the target [360], and only two image volumes are used for motion analysis.
Faster approaches have been proposed that avoid volumetric image processing by performing in-plane and out-of-plane motion analysis based on lower dimensional surrogate images [307, 362, 371]. Moreover, tumor and surrounding tissue often appear similar in US images due to similar acoustic properties [78, 98], and noise, as well as imaging artifacts add to this problem [499]. Hence, typically distinct anatomical structures such as vessels are considered for motion estimation with

deep learning approaches [98]. We provide an overview of related studies using deep learning in Table 5.1.

In summary, 2D target tracking has been substantially studied, and various conventional and deep learning methods using sequences of 2D data have been demonstrated. Still, these 2D methods are typically limited to in-plane motion, and three-dimensional motion estimation remains an open problem. This requires motion analysis from 4D US data. However, current approaches are typically limited by run-times and the requirement for feature selection. In particular, markerless motion analysis without the presence of specific visual landmarks has hardly been studied [98]. Furthermore, current approaches for motion estimation typically perform tracking based on a template, a search region, and a similarity measure. This quickly becomes computationally demanding, limiting real-time processing of the data [212]. Another level of complexity results from the requirement to compensate for system latencies. To this end, various motion prediction methods that use prior motion estimates have been developed, representing a challenging topic itself [138, 226, 276, 451]. Hence, performing motion prediction directly from the image data with the same approach used for motion estimation would simplify the entire process. Also, by performing motion forecasting directly from the image data, abstract features can be used for the task compared to the lower-dimensional motion estimates used by predictive methods.

We address these challenges in our research questions of this work. Considering our first research question, we study whether end-to-end motion analysis can be performed from 4D US data and whether both motion estimation and forecasting could be combined into a single effective approach. Considering our second research question and given the challenging task of performing motion analysis over several minutes, i.e., processing several hundredths of volumetric US images in real-time, we compare pair-wise processing up to entire long-term sequences with spatio-temporal deep learning methods. We systematically develop and compare spatio-temporal 4D deep learning approaches in the context of markerless motion estimation from long-term US sequences. We report our results in Section 6.1.3.

Table 5.1: Summary of related works using deep learning approaches with US in the context of motion estimation and compensation.

Application	Approach	Data
2D liver landmark tracking [335]	2DCNN	3D US
2D liver landmark tracking [175, 282, 499]	2DSiamFC	3D US
2D liver landmark tracking [367]	2DCNN-LSTM	3D US
2D liver landmark tracking [211]	2DCNN-ConvLSTM	3D US
3D liver landmark tracking [212]	2DCNN	4D US
3D motion field estimation and prediction [371]	3DCNN	4D US
3D liver landmark tracking [195]	Siamese 3DCNN	4D US
3D deformation field estimation [307]	3DCNN	4D US

5.1.3 OCT-based Motion Analysis

OCT-based motion analysis is relevant in many scenarios, including intraoperative imaging and minimally invasive procedures [62, 134, 186, 359]. In particular, the high spatial and temporal resolution of OCT combined with the ability to perform volumetric imaging make this relatively new imaging modality interesting across a range of applications that are performed at sub-millimeter scale [62], including, e.g., laser cochleostomy [535], or ophthalmic microsurgery [62, 235]. In these scenarios, either motion compensation and target tracking or even both are typically required. Parts of this section have been published in our study presented in [38–40].

Motion compensation is highly relevant due to the limited field of view (FOV) of OCT, which is in the range few millimeters or centimeters [62, 247]. Thus, the relevant target area can be lost quickly during intraoperative imaging due to tissue or surgical tool movement. Hence, constant tracking of the ROI and adjustment of the FOV is beneficial or required, and performing this step manually is difficult and can disrupt the surgical workflow substantially [130, 188]. This leads to the task of motion estimation for automated motion compensation to fully leverage the advantages of OCT-based intraoperative imaging [62, 132, 134]. Another similar scenario for OCT-based motion analysis is to obtain information on surgical tool poses or tissue motion and location to guide surgical procedures such as laser cochleostomy [535], laser osteotomy [24], or ophthalmic microsurgery [62, 235]. In these scenarios, accurate tracking is critical to avoid damaging surrounding tissue [229].

One approach to address the motion analysis task is to use external tracking systems [133, 468]. However, this increases the complexity and requires efforts for integration. The high spatial and temporal resolution of OCT imaging motivates this modality for direct motion estimation [398]. Whereas there is still limited work so far, promising results have already been achieved using conventional tracking methods [235,255,331,394,396,397,535]. For example, it has been demonstrated that lateral displacement can be estimated between adjacent A-scans using cross-correlation [287, 489]. Considering volumetric data, Niemeijer et al. [331] demonstrate that registration between two volumetric OCT images can be performed by means of 3D SIFT [290]. Laves et al. [255] use and compare different conventional tracking approaches using 2D maximum intensity projections to perform volume of interest stabilization. Also, OCT-based markerless tracking using phase correlation [251] between volume pairs has been presented for homogeneous phantoms and tissue samples [398]. This approach has been extended, and a preliminary markerless tracking system has also been presented that solely relies on OCT images [393, 397, 398]. Moreover, pose estimation has been demonstrated with conventional tracking methods by estimating the motion at multiple locations [396].

Recently, deep learning methods have also been proposed for OCT-based target tracking, and motion compensation [168,171,254,336,432]. For example, Ntatsis et al. [336] present an approach to estimate motion between adjacent B-scans to compensate for motion artifacts during volumetric image acquisition. To this end, the authors present a custom 3DCNN approach that estimates translation parameters between adjacent B-scans. Considering volumetric data, Gessert et al. [171] demonstrate end-to-end marker pose estimation from single 3D OCT volumes using and comparing different 3DCNNs. This approach employed custom architectures that perform 3D convolutions across all spatial dimensions of a volumetric image to perform a regression task between the marker pose and the input image that visualizes the current marker position. Also, optical flow estimation using 2.5D OCT projections and deep learning has been presented by Laves et al. [254], adopting the concept of FlowNet [118, 214]. Considering the problem of motion compensation, Gessert et al. [168] propose a Siamese 3DCNN architecture to estimate the parameters of a motion compensation system using a distinct marker object. This approach performs a regression task using two volumetric OCT images and outputs parameters of a compensation system such that the FOV can be adjusted based on the translation between the volume pair. Adapting this Siamese 3DCNN approach, Sprenger et al. [432] demonstrate the feasibility marker-less motion estimation of perfused and non-perfused tissue of *in-vivo* xenograft tumors. This study also compared phase correlation and demonstrated that deep learning leads to superior tracking performance. We provide an overview of related studies using deep learning in Table 5.2.

Table 5.2: Summary of related works using deep learning approaches with OCT in the context of motion estimation and compensation.

Application	Approach	Data
Motion correction in retinal optical coherence tomography [336]	3DCNN	3D OCT
Tool tracking / marker position estimation [171]	3DCNN	4D OCT
Motion compensation for FOV adjustment [168]	Siamese 3DCNN	4D OCT
Marker-less motion estimation for minimally invasive procedures [432]	Siamese 3DCNN	4D OCT
Marker-less tracking for laser ablation [254]	FlowNet	4D OCT

In summary, OCT-based motion analysis needs to be performed accurately and in real-time to assist intraoperative imaging and minimally invasive procedures. To this end, both conventional and deep learning-based tracking has been demonstrated for the purpose of motion compensation and target tracking relying only on information from the OCT images. Deep learning has shown promising results to overcome limitations of conventional methods and allows, e.g., direct estimation of system parameters for compensation within a few milliseconds [168]. However, there is limited work that studies end-to-end deep learning and motion analysis using volumetric OCT images so far. It stands out that current deep learning-based methods only use very limited temporal information to address motion analysis and tracking tasks. In particular, pose estimation of a marker object was performed using a single volumetric image, and spatio-temporal relationships were not considered [171]. Deep learning-based motion estimation with volumetric OCT considered two images for the task, i.e., an initial template volume and an image volume at second time point [168, 254, 432], following the concept of registration-based motion estimation [398]. Notably, modern OCT systems could allow for acquiring entire sequences of OCT volumes in real-time as very high A-scan acquisition rates have been achieved, as outlined in Section 2.2.2. Thus, entire sequences are available and could be used for motion analysis. Ultimately, this could improve tracking performance and robustness. However, how and whether these spatio-temporal relationships can be learned efficiently and effectively with deep learning remains an interesting research direction.

We address these shortcomings in our work. Considering our first research question of this work, based on our studies presented in [38–40] and our methods presented in Section 4, we study marker position estimation and markerless motion compensation end-to-end using entire sequences of OCT data and spatio-temporal deep learning. Considering our second research question, we perform an analysis of different spatio-temporal deep learning methods that perform spatio-temporal feature learning in different fashions, e.g., using pairwise features up to the entire

spatio-temporal feature learning of the sequence. We also compare the performance of using a sequence compared to the previous approach that uses only a single volume or a volume pair. We report our results in Section 6.1.1 and Section 6.1.2 for marker position estimation and markerless motion compensation, respectively.

5.2 Dynamic Elastography for Tissue Characterization

In this section, we first introduce the relevant background and related work for dynamic elastography and highlight associated challenges. Afterwards, we focus on our two imaging modalities, US and OCT. While US is already widely established for elastography, we outline several challenges that remain unsolved and outline recent deep learning studies in this context. Second, we consider OCT and dynamic elastography, which is a rapidly growing field due to the advantages that result from the high spatial and temporal resolution of OCT. Here, we also highlight recent developments and challenges associated with estimating elastic parameters from sequences of OCT images that capture wave propagation.

5.2.1 Background

Diseases change the mechanical properties of tissue [510], and analyzing these properties can provide valuable information for diagnosis [182, 510], and can be used, e.g., for discriminating different liver fibrosis stages [148, 309], or for early tumor detection [306, 540]. For centuries, physicians have used manual compression to assess the stiffness of organs or tissue for diagnostic purposes [182, 267]. In addition to that, the mechanical properties of soft tissue are also of interest for surgical planning. For example, estimating the elastic characteristics of soft tissues improves tumor resection [67] and refines needle placement of biopsies [262,351]. Elastography techniques aim towards the quantification of mechanical properties of external as well as internal tissue structures using non-invasive imaging modalities [154, 340]. Several image modalities have been studied for elastography imaging, including MRI [302], US [156, 340], and OCT [330, 485]. These modalities provide information at different spatial scales allowing to detect a wide range of tissue pathologies with elastography techniques [213, 485].

Fundamentally, elastography techniques rely on the relationship between displacement and elasticity of a tissue [236]. These techniques provide images typically called elastograms, which give local information about tissue stiffness, e.g., the elastic modulus distribution of a target region [340]. Although biomechanical properties of tissue are typically more complex, mainly the Young's Modulus and the shear modulus are considered representative values for diagnostic purposes [436, 547]. In general, the Young's Modulus is defined by the one-dimensional Hooke's Law

$$E = \frac{\sigma_E}{\epsilon_E} = \frac{F/A}{\Delta L_E / L_E} \tag{5.1}$$

with σ_E for the stress, ϵ_E for the strain, F for applied force, A for cross-sectional area where the force is applied, L_E for original length and ΔL_E for the relative change in length. Elastic characteristics, i.e., Young's moduli of different example tissues, are reported in Table 5.3.

Table 5.3: Example Young's moduli of soft tissues. IDC refers to infiltrating ductal carcinoma.

Tissue	E [kPA]	Reference
Prostate (non-cancerous)	17.0 ± 9.0	[4]
Prostate (cancerous)	24.1 ± 14.5	[4]
Breast (normal fat)	3.25 ± 0.91	[382]
Breast (low-grade IDC)	10.40 ± 2.60	[382]
Breast (Intermediate-grade IDC)	19.99 ± 4.20	[382]
Breast (high-grade IDC)	42.52 ± 12.47	[382]

Elastography techniques can be divided into two groups, "static" / "quasi-static" elastography, and "dynamic" elastography [149,156,159,342]. The following paragraph is based on [156]. Both groups rely on measuring tissue displacement due to an applied force, while the type of excitation distinguishes the two groups. The concept of quasi-static elastography is to estimate the strain based on tissue displacement using image data before and after compression. Afterwards, elastic properties can be related based on the relationship of the one-dimensional Hooke's Law given in Equation 5.1, assuming an isotropic linear elastic solid [7]. However, estimating the stress value σ is typically not feasible during clinical practice [415]. Still, strain in softer tissue is usually larger compared to stiffer tissue, which allows qualitative differentiation of tissue stiffness [220]. Hence, strain values are typically visualized (strain elastogram) for qualitative comparison during clinical practice, and quantitative characterization with quasi-static elastography remains difficult [415]. Despite these limitations, quasi-static elastography is widely used and provides a valuable clinical tool, e.g., for classifying breast tumors [61]. The second technique is dynamic elastography, which allows for quantitative estimation of tissue elasticity [7, 389]. Here, waves of displacements are induced into a tissue, and the resulting wave propagation is then captured spatially and temporally with high-frequency imaging, i.e., sequences of images [415]. The term dynamic elastography can be related to the fact that dynamic stress is applied to tissue [415], or imaging of moving waves is performed. For the excitation of tissue, various techniques can be used, such as an acoustic radiation force impulse (ARFI) [92, 332, 415] or a mechanical vibrator [208, 525, 536]. A widely adopted approach for dynamic elastography is shear-wave elastography [7]. Here, a shear wave is induced into a tissue, and subsequently, the velocity of the propagating wave is estimated from sequences of image data. Note that a shear wave defines a wave where the particle motion is perpendicular to the direction of wave propagation, and the shear wave velocity defines the transverse wave propagation speed.

Typically, shear wave velocities of soft tissue are in the range $1\,\mathrm{m\,s^{-1}}$-$15\,\mathrm{m\,s^{-1}}$ [145] and are generated at frequencies in the range of 10 Hz-2000 Hz [159]. The fundamental concept of shear wave elastography is that shear wave velocity increases with increasing stiffness of tissue. Assuming an isotropic linear elastic model of tissue behavior, elastic properties such as the Young's modulus E can directly be related to shear wave velocity v using the relationship

$$E = \rho \cdot 2 \cdot (1 + \nu) \cdot v^2, \tag{5.2}$$

with ν for Poisson's ratio and ρ for the density [236, 340, 346]. Often a density of $\rho \sim 1000\ \mathrm{kg/m^3}$ is assumed for conversion of shear wave velocity v to the Young's Modulus E [236, 347]. For the Poisson's ratio, commonly a value close to $\nu = 0.5$ is assumed due to the high water content of tissue, which represents the case of a nearly incompressible medium [236]. Therefore, by estimating the shear wave velocity from sequences of image data, tissue stiffness, e.g., the Young's modulus, can be related, and by pixel-wise estimation of shear wave velocities, an entire elastogram can be constructed and visualized for an imaged tissue.

While shear-wave elastography could estimate specific elastic properties such as Young's Module in theory, performing such an approach in practice remains challenging [7]. First, estimating the shear wave velocity from image data is a challenging task itself. Second, deriving elastic properties from shear wave velocity requires strong theoretical assumptions, e.g., the assumption of a linear elastic material with isotropic mechanical properties [236, 340, 346]. Notably, these assumptions rarely hold in practice, and conversion of shear wave velocity to Young's modulus should be considered with caution [7, 236, 547]. Overall, shear-wave elastography involves four major challenges: shear wave generation, shear wave imaging/detection, elasticity estimation, and implementation [423]. Each of these challenges represents an entire research topic itself.
In this work, we focus on the data analysis aspects and aim toward end-to-end elasticity estimation using spatio-temporal deep learning and sequences of US and OCT data. From a machine learning perspective, dynamic elastography requires estimating a target value from a sequence of images that capture wave propagation over time. These target values could represent relevant properties such as the shear wave velocity, the Youngs modulus, or direct classification of different tissue types [267]. The relevant spatio-temporal features for this task are encoded in the spatio-temporal relationships of the wave propagation. However, analyzing these features in a hand-crafted fashion is known to be difficult [267]. Considering our first research question, this raises the question of whether multi-dimensional sequences of medical image data can be processed effectively with spatio-temporal deep learning to this end. Considering the aforementioned challenges, spatio-temporal deep learning could provide a promising approach that directly estimates target values from sequences of images in an end-to-end fashion by learning the direct relationship between local tissue elasticity and local wave propagation without requiring a specific material model and hand-crafted feature engineering. Similarly, by learning end-to-end relationships, the impact of systematic imaging artifacts could also be reduced. In the following two sections, we

explain US-based and OCT-based elastography in more depth, summarize related work, and highlight open challenges that we aim to address with our work.

5.2.2 US-based Elastography

Parts of this section have been published in our study presented in [184, 322]. Ultrasound shear wave elasticity imaging (US-SWEI) has already been demonstrated in 1998 [389], and since then, the field has developed rapidly. Today, it is widely used and various clinical applications have been demonstrated. Recent review papers that consider US-SWEI are given in [267, 268, 415, 449, 450], and examples of clinical applications include disease staging of breast lesions [515], thyroid nodules [402], or liver fibrosis [384]. For imaging of the wave propagation, commonly plane wave imaging is used [268, 449, 450], i.e., an entire image is acquired at once. Moreover, various techniques have been developed to improve shear wave excitation, and imaging quality, such as supersonic shear imaging or comb-push ultrasound shear elastography [45, 114, 425].

As outlined before, we focus on image sequence analysis to derive mechanical properties directly from the acquired image sequence in an end-to-end fashion. So far, most conventional approaches of US-SWEI first estimate displacement information, e.g., with the Loupas Algorithm [289] as a widely adopted method, and afterwards shear wave velocity estimation is performed based on the present spatio-temporal features. A range of methods have been proposed to this end, which can be divided into two groups time-of-flight (ToF) and transformation-based methods [225, 403].
ToF methods track the propagating shear wave over space and time. One approach considers a linear regression of the wave peaks in a 2D space-time image representation [343, 481]. Then, the slope of the regression corresponds to the shear wave velocity. Note that this relies on the same concepts of a spatio-temporal data representation that we introduced in Section 2.1. The 2D space-time image representation can be derived by, e.g., averaging across the axial image dimension when assuming that shear wave propagation is along the lateral image dimension. Another ToF approach tracks the entire waveform at two measurement points (pixels) with a known distance. Afterwards, a cross-correlation of the two signals is performed to estimate the time difference, and by dividing the known travel distance with the time difference of the signals, the shear wave velocity can be estimated [45, 424, 425, 452]. Often, shear wave propagation is assumed in a fixed direction, but the aforementioned approaches can also be used along multiple directions, i.e., axial and lateral, and thereby a resulting shear wave velocity independent of the propagating direction can be estimated using the individual components [424, 425].

The second approach for US-SWEI estimates the shear wave velocity using a transformed version of the data [240, 241, 301, 327, 333, 375, 462]. For example, using a 2D Fourier transformation of the 2D space-time map representation allows for detecting the dominant spatial and temporal frequencies of the imaged

wave propagation in the k-space, which then can directly be related to the shear wave velocity [333]. Recently, also a Fourier-based approach has been presented that performs shear wave velocity estimation in a sliding window fashion independent of the wave propagation direction [240, 241]. However, this approach requires intensive tuning of imaging and filter parameters [240]. Moreover, shear wave velocity estimation is influenced by the imaging depth [514], and estimations in stiffer materials are known to be difficult [480]. This becomes even more problematic in clinical application areas, such as laparoscopy, where small US transducer are required with lateral imaging widths in the millimeter range [215]. Hence, the shear wave is tracked over a smaller distance, making velocity estimation even more difficult and error-prone [241, 376]. Overall, various methods have been demonstrated to estimate shear wave velocity from a sequence of US data. However, shear wave velocity estimation remains a challenging task and requires notable problem-specific tuning, especially with noisy *in-vivo* data [104, 225, 304, 376, 403, 481]. In addition to that, subsequent conversion of the shear wave velocity to a relevant property adds another layer of complexity as outlined in the previous section.

Recently, deep learning methods have gained popularity in strain elastography [66,99,238,252] and SWE-imaging [3,225,464]. These methods promise to perform elastography without intensive data preprocessing and manual tuning. A recent review of deep learning elastography is given in [267]. In the following paragraph, we focus on our application scenario dynamic elastography, i.e., SWE-imaging. A recent review paper [267] highlighted that deep learning can be used successfully for classification tasks using shear wave elastography images and that this can lead to better performance than using B-mode US images or human-crafted features [152, 153, 158, 228, 352, 477, 517, 534]. These studies include among other, classification of beast tumors [152, 543], thyroid nodules [352], and chronic liver diseases [228]. To perform the classification, typically, a 2DCNN architecture, either pre-trained or custom designed, is applied to 2D shear wave elastography images. Also, approaches have been presented that utilize information from both 2D shear wave elastography images, and B-mode images [363,509,534,539]. While these studies demonstrate the value of shear wave information combined with deep learning, prior shear wave velocity estimation is required to obtain the shear wave elastography images used for classification. Shear wave estimation is difficult, as outlined previously, and hence it is likely affected by estimation errors which subsequently could impact the classification performance of machine learning algorithms that rely on this feature. This motivates end-to-end processing of the data without prior feature extraction, i.e., classification or estimation directly from the sequence of US image data that captures the wave propagation. Thereby, also the data processing pipeline could be simplified since fewer sequential data processing steps are involved [267].

It has been shown that shear wave velocity with uncertainty metrics can be estimated with deep learning using 2D space-time image representations derived from the sequences of 2D US image data [225]. These 2D space-time image represen-

tations are similar to those used for ToF approaches, and these studies perform a regression task using 2DCNNs applied to the 2D image representation. Moreover, Delaunay et al. [100] present a deep learning approach for displacement estimation from simulated RF-data for shear wave elastography. The authors adapt an encoder-decoder architecture that has first been proposed for strain elastography [99]. The architecture takes a pair of US image data and estimates the corresponding displacement field between the image pair. To improve the performance of the approach, the authors propose to use ConvLSTM modules [412] in the decoder part, such that also the displacement history is considered for estimation. However, the aforementioned studies only estimate shear wave velocity or the displacement field. Hence, conversion to material parameters or tissue classification remains an open problem. This motivates end-to-end processing of the data.

Considering end-to-end material parameter estimation, Vasconcelos et al. [464] show that elasticity and viscosity parameters can directly be estimated from simulated displacement data. This approach also uses 2D space-time image representations and presents a custom 2DCNN architecture to perform the regression task of the material parameter from the image data. Moreover, Ahmed et al. [3] demonstrate that entire elasticity maps and segmentation masks of lesions can be estimated with deep learning from simulated displacement data in an end-to-end fashion. This approach uses a sequence of 2D velocity maps as input, capturing the wave propagation, and outputs the corresponding elasticity maps and segmentation masks. For processing of the spatio-temporal data, Ahmed et al. [3] propose DSWE-Net an encoder-decoder architecture that uses a 3D spatio-temporal CNN in the encoder part and a two-path 2DCNN in the decoder part with one path for the 2D elasticity map and with one path for the 2D segmentation mask. The decoder and encoder are connected with a recurrent block using a ConvLSTM [412]. However, this approach requires training data which is extremely difficult to annotate. Hence, Ahmed et al. [3] rely on simulated training data, where data annotation can be automatically generated. While this approach shows encouraging results, the authors emphasize that the simulated training data does not sufficiently represent all aspects of real data and hence shows limited performance on such data. We provide an overview of related studies using deep learning and US-SWEI in Table 5.4.

In summary, deep learning methods have gained traction for US-SWEI, and there are two research streams. One considers various classification tasks using shear wave elastography images. So far, this stream represents the majority of previous studies. This could be explained by the fact that US-SWEI is already used in the clinical context, and processing of available shear wave elastography images with deep learning provides a promising decision support tool [267]. However, estimating the required shear wave elastography images remains challenging and error-prone. Thus, machine learning algorithms that build on this feature representation might be limited in performance and affected by assumptions that have been made to estimate the shear wave elastography images. The second

stream aims towards end-to-end data processing, and it has been shown, e.g., that material parameters can be directly estimated. However, the second stream typically relies on simulated data, and there have been limited evaluations on real data. This could be explained by the fact that the required data for training and evaluation is difficult to annotate and obtain in practice, which represents a major challenge for deep learning and elastography [267]. Moreover, so far, it has not been studied how the spatio-temporal relationships of the image sequence can be learned effectively. Overall, end-to-end processing with spatio-temporal deep learning and real sequences of US data has hardly been studied, and a systematic comparison of different spatio-temporal deep learning concepts is also missing.

In this work, we address these shortcomings in our research questions, and in contrast to previous work in the field, we do not rely on simulated data. Considering our first research question, we study whether end-to-end shear wave velocity estimation and localized elasticity estimation directly from sequences of US images can be performed effectively with deep learning. Considering our second research question, we study and compare our spatio-temporal deep learning methods presented in Section 4. We also evaluate how to process the image sequence and compare learning from the entire image sequence to using a lower-dimensional space-time map representation. Moreover, we present a training approach for localized elasticity estimation that only requires small spatio-temporal windows for training and thereby simplifies the data collection and annotation process. We report corresponding results for these experiments in Section 6.2.1 and Section 6.2.2.

Table 5.4: Summary of related works using deep learning approaches and US-SWEI. P denotes a pre-trained 2D architecture and C denotes a custom designed architecture. SWE refers to a 2D elastography map. SPT denotes 2D space-time image representations, and * denotes simulated data. (c.f. [267])

Application	Approach	Data
Thyroid nodule classification [352]	2DCNN (P)	SWE
Thyroid nodule classification [363]	2DCNN (P)	SWE + B-Mode
Breast tumor classification [543]	2DCNN (C)	SWE
Breast tumor classification [539]	2DCNN (C) / SVM	SWE + B-mode
Breast tumor classification [152]	2DCNN (P)	SWE
Breast tumor classification [534]	2DCNN (P)	SWE + B-mode
Liver fibrosis stage classification [57, 474, 477]	2DCNN (C)	SWE
Liver fibrosis stage classification [155, 158, 228, 541]	2DCNN (P)	SWE
Liver fibrosis stage classification [509]	2DCNN (P)	SWE + B-mode
Chronic liver disease assessment [228]	2DCNN (P)	SWE
Plantar fasciitis classification [153]	2DCNN (C)	SWE
Chronic kidney disease classification [530]	2DCNN (C)	SWE
Sarcopenia classificiation [517]	2DCNN (P)	SWE
Elasticity and viscosity parameter regression [464]	2DCNN (C)	SPT*
Elasticity parameter regression and lesion segmentation [3]	2D/3DCNN / ConvLSTM (C)	2D velocity maps*
Shear wave velocity regression with uncertainty estimation [225]	2DCNN (C)	SPT
Displacement estimation [100]	2DCNN / ConvLSTM (C)	RF data*

5.2.3 OCT-based Elastography

Parts of this section have been published in our study presented in [323, 324]. OCT for elastography is motivated by the high spatial resolution of the modality and the ability to perform 3D real-time imaging [236, 485]. In particular, using the phase information of the complex OCT signal is a promising approach allowing to measure displacements with nanometer sensitivity [547], as also outlined in Section 2.2.2. Thereby excitation forces can be reduced notably [274], and small deviations of elastic tissue properties can be detected [237]. This makes optical coherence elastography (OCE) suitable for application scenarios such as ophthalmology and cardiology where both of the aforementioned properties are essential [151,209,547]. However, while OCT brings appealing advantages, so far, OCE is still under research, and the clinical adaption is still in the early stage [547]. Recent review papers of the development of OCE and application scenarios are given in [236, 485, 547]. Typically ocular tissue is considered [357, 547–549], but also studies have been presented that analyze other tissue types such muscle [288,550], or brain [157, 546]. Recently, first *in-vivo* studies have been presented and, e.g., it has been demonstrated that *in-vivo* human corneal can be characterized with wave-based OCE [253, 365, 460, 547].

The following paragraph is based on [547]. Similar to our previous section that considers US-SWEI, we focus on the data analysis aspects in this section. In general, elasticity estimation based on dynamic OCE requires the measurement and subsequent analysis of a propagating mechanical wave induced in the tissue. Motion measurements with OCT data can be performed along the axial and lateral image axis based on speckle tracking techniques using the intensity/amplitude data of the OCT signal [201, 344, 399]. However, to utilize the aforementioned advantage of dynamic OCE, typically only small motions are induced into tissue, and tracking such small tissue displacements that might be only a fraction of a pixel is inaccurate with speckle tracking techniques [73,123,124,244]. Hence, typically the phase information of the OCT data is used, that allows to to measure displacements with nanometer sensitivity [243]. However, phase data is affected by various sources of noise, achieving phase stability remains a challenging problem, and estimating the actual phase difference requires phase-unwrapping, which can become problematic when the phase difference is greater than π [73]. Moreover, recall that 2D/3D OCT images are based on multiple A-scans acquired at different lateral locations, as outlined in Section 2.2.2. Moreover, the observed displacement characteristics of the propagating shear wave also depend on the acquisition protocol, while typically M-B acquisition or B-M acquisition are used. The former acquires a sequence of A-scans at a fixed lateral position and repeats this procedure for multiple lateral positions using a synchronized repeated motion excitation [485]. This results in 2D/3D images, which are similar to single 2D/3D spatial images where all A-scans were acquired at the same time with an apparent frame rate equal to the A-scan rate of the OCT system. Instead, A-scans are acquired laterally over time for B-M acquisition to obtain 2D/3D OCT images. This process is repeated to obtain a sequence of OCT images, i.e., repetitive B-mode or C-mode acquisition is performed [419, 420, 427]. This

approach requires only a single excitation, and the overall acquisition time is reduced. However, the wave field changes during the acquisition of a single OCT volume, and hence notable imaging artifacts can be present, which make displacement estimation and feature engineering for elasticity estimation even more challenging. First promising results have also been achieved to perform OCE with 4D OCT data [12, 283, 428, 549]. In particular, using 4D OCT data combined with reverberate wave fields where multiple shear waves travel in various directions has appealing advantages such a higher ratio of shear waves in tissue with elasticity estimation independent of the wave propagation direction [549]. However, so far, real-time data acquisition and analysis remains difficult. For example, a related study requires approximately 60 s per data acquisition [549]. After imaging and displacement estimation, wave propagation velocity can be estimated from the displacement data with conventional approaches similar to those used for US-SWEI that we outlined in the previous section, e.g., using ToF [426, 484], or Fourier-based estimators [189, 283]. While shear wave velocity estimation is a challenging task alone, subsequent estimation of material parameters becomes even more difficult with OCE. The authors of a recent review paper [547] empathize that due to the limited penetration depths of OCT imaging, measurements are often performed near the surface of the tissue, resulting in the presence of different wave types and without identification of the actual wave type false conclusion can be made about the tissue properties. This needs to be considered to correctly estimate mechanical properties from the estimated wave propagation velocity based on mechanical models.

In summary, measurement and subsequent analysis of propagating waves with OCT remain challenging and are an active research area. So far, previous studies typically follow the conventional approach, i.e., displacement estimation, wave propagation velocity estimation, and conversion to material parameters. Each of the involved steps is a challenging task, as outlined previously, and hence despite the advantages of OCT, tissue characterization remains difficult [547]. Similar to US-SWEI, these challenges related to data analysis could be addressed with effective deep learning methods by processing the data in an end-to-end fashion. In general, it has been shown that interpreting OCT data can be performed with deep learning considering classification, e.g., of retinal diseases [366, 511]. However, so far, there is very little work that considers deep learning and dynamic OCE. Still, a few related works consider deep learning and quasi-static OCE. It has been demonstrated that deep learning can be used to estimate strain field distributions from phase differences of the OCT signal in an end-to-end fashion [116]. The authors of this approach rely on simulated data for training and use a U-Net 2DCNN architecture which maps 2D phase difference images to 2D images of the corresponding strain field distributions. Considering deformations, it has also been shown that needle forces can directly be estimated from OCT data using deep learning [164, 166, 169, 170, 308, 325, 326]. Moreover, it has been demonstrated that end-to-end sample characterization of gelatin phantoms can be performed with deep learning performing compression-based OCE [308]. We provide an overview of related studies using deep learning in Table 5.5. These

studies demonstrate encouraging results that deep learning can be used to learn complex end-to-end relationships from OCT images with deep learning related to elastography without prior feature extraction.
We contribute to this research field, as we address end-to-end elasticity estimation with deep learning and OCE. Considering our first research question, we address whether spatio-temporal deep learning is able to learn the complex spatio-temporal relationships despite the aforementioned challenges and whether relevant properties such as elasticity can be estimated in an end-to-end fashion and in real-time. Considering our second research question, we evaluate different spatio-temporal deep learning networks for learning the spatio-temporal relationships and evaluate learning from lower-dimensional space-time map representations as well as entire 3D/4D image sequences. We report our corresponding results in Section 6.2.3 and Section 6.2.4 for 3D OCT data and 4D OCT data, respectively.

Table 5.5: Summary of related works using deep learning approaches and OCE.

Application	Approach	Data
Compression-based tissue elasticity characterization [308]	ConvGRU-CNN	2D OCT
Needle force estimation [169]	ConvGRU-2DCNN	2D OCT
Needle force estimation [170]	ConvGRU-2DCNN	2D OCT
Strain field distribution estimation [116]	2DCNN	3D OCT
Needle force estimation [166]	Siamese 3DCNN	4D OCT
Needle force estimation [326]	Siamese 3DCNN	4D OCT
Needle force estimation [164, 325]	3D/4DCNNs/ConvGRU	4D OCT

5.3 Summary

In this chapter, we presented the background and related work for target motion estimation and dynamic elastography, focusing on OCT and US. Considering motion analysis with US in the context of radiotherapy, processing of sequences of 2D images has been substantially studied, and deep learning methods have been demonstrated [98, 175, 211, 282, 307, 335, 367, 371, 499, 506]. However, there are only a few studies that consider volumetric image data, and so far, real-time processing and accurate tracking remain major challenges, especially if tracking is performed without the presence of distinct landmarks such as vessels [98]. Considering motion analysis with OCT in the context of target position estimation or motion compensation for FOV adjustment, deep learning methods have been

presented that perform motion analysis based on single volumetric images or image pairs [168, 171, 254, 432]. However, processing entire sequences of volumetric image data has hardly been addressed with deep learning. OCT and US allow for real-time volumetric imaging, and thus, entire sequences are available and could be used for motion analysis. This might not only be beneficial to improve the performance and robustness of motion estimation. In fact, motion forecasting to compensate for system latencies is another relevant problem typically addressed separately [138, 226, 276, 451]. This problem could be circumvented by using and learning the underlying spatio-temporal relationships of sequences of volumetric images that might even allow predicting a target motion into the future.

Considering dynamic elastography with a focus on shear wave elastography, it stands out that notable results have been achieved already, e.g., for classification tasks that use elastography maps as a feature [267]. However, accurate elasticity estimation remains a major problem [7]. Typically, the conventional data processing pipeline is considered, where, e.g., shear wave velocity is estimated and extracted as a feature. Subsequently, an elasticity parameter such as the Youngs Modulus is related [423]. However, often this involves assumptions that rarely hold in practice, and the task of shear wave velocity is difficult and error-prone itself [7, 236, 547]. Promising results have been achieved to estimate elasticity parameters from sequences of 2D US image data directly with deep learning [3, 464]. However, there have been limited evaluations on real data, and current approaches rely on simulated training data due to the difficulty of obtaining the required data annotation. Considering the recent developments of shear wave elastography based on OCT imaging highlights that this approach could bring many advantages, including the detection of small elasticity deviations [237]. However, measurement and subsequent analysis of propagating mechanical waves with OCT is an active research area with many unsolved questions [547]. In particular, volumetric OCT imaging for dynamic elastography could bring many advantages, such as estimations independent of the wave propagation direction [549]. However, so far, real-time volumetric elastography remains a major problem. Considering deep learning and OCT for shear wave elastography, it stands out that there is very limited work so far. Overall, considering the aforementioned challenges, spatio-temporal deep learning could provide a promising approach that directly estimates target values such as tissue elasticity from sequences of images in an end-to-end fashion by learning the direct relationship between tissue elasticity and wave propagation without requiring a specific material model and hand-crafted feature engineering.

Across our application scenarios, it stands out that there are many unsolved questions regarding spatio-temporal feature learning. From a conceptual level, both motion analysis and dynamic elastography require processing spatio-temporal features captured with multi-dimensional medical image sequences. This raises the question whether spatio-temporal deep learning is able to learn these complex spatio-temporal relationships effectively and whether training objectives, such as the current and future motions of a target or elasticity of tissue, can be estimated

in an end-to-end fashion. Thereby, many of the aforementioned challenges could be addressed, and data processing pipelines could be simplified substantially for motion analysis and dynamic elastography. In addition to that, it raises the general question of how data processing should be performed from an architectural and conceptual perspective, which is covered by our second research question of this work.

Chapter 6

Experiments and Evaluations

In this chapter, we study our proposed spatio-temporal deep learning approach in the context of motion analysis and dynamic elastography using sequences of OCT and US image data. We structure our experiment sections on the basis of the general definition of learning algorithms based on Mitchel (1997) [313], introduced in Chapter 3. We outline the learning task, the experience, the learning algorithms, and the performance measure. For each experiment section, we then report corresponding results, followed by a results-oriented discussion.
First, we present the results for our experiments regarding motion analysis with a focus on 4D spatio-temporal processing. We study marker object position estimation from OCT data and markerless motion analysis of tissue from sequences of volumetric OCT and US image data. In this context, we study our approaches ranging from single volume processing to short-term sequence processing up to entire long-term sequence processing of several hundreds of image volumes.
Second, we present the results for our experiments regarding dynamic elastography. We evaluate our spatio-temporal deep learning approach in the context of shear wave velocity estimation and elasticity estimation in an end-to-end fashion.
Considering our research questions, we focus on how sequences of images can be processed to improve the performance and whether the spatio-temporal relationships required to perform motion analysis and dynamic elastography can be learned in an end-to-end fashion without the requirement of prior feature selection. For each application scenario, we highlight the challenges, how the setting affects the learning task, and how these can be addressed by our methods.

6.1 Position Estimation and Motion Compensation

In this section, we study our spatio-temporal deep learning methods in the context of motion analysis from sequences of volumetric OCT and US image data. Considering our two main research questions, we analyze whether end-to-end motion analysis can be performed from such data and how this task can be addressed from an architecture and training perspective. Across the experiments in the following sections, we put a focus on the question of whether using sequences of images can improve the performance compared to the previous approach of using a single image or an image pair and how these complex spatio-temporal features can be learned from data in an end-to-end fashion. For the systematic evaluation of our methods, we consider the tasks of position estimation of a marker object from OCT data and markerless motion analysis of tissue from OCT and US image data. We use datasets that are acquired and annotated with experimental setups.

6.1.1 4D OCT-based Object Position Estimation

In this study, we address end-to-end position estimation of a marker object from sequences of OCT volumes, which is relevant for, e.g., surgery tool tracking during minimally invasive surgery [113] as highlighted in Section 5.1.3. We investigate the use of spatio-temporal deep learning for marker position estimation from sequences of volumetric OCT images in real-time. Previous work has used 3DCNNs combined with single volumetric images to locate small objects, but we aim to improve performance and consistency by using entire sequences of images. Parts of this section have been published in our study presented in [38]. The study [38] was a collaborative study, data acquisition was performed with equal contribution of N. Gessert and the author of this work. The aspects regarding spatio-temporal deep learning were developed and evaluated by the author of this thesis and are in the focus of this work.

Considering our first research question, we systematically evaluate our architecture concept using different spatio-temporal processing methods in comparison to 3DCNNs and evaluate whether a sequence of OCT volumes improves object position estimation performance. To this end, we perform position estimation from a single volume and from a sequence of volumes that also provide information on past positions of the marker.

Considering our second research question, we address the task of architecture design and contribute to the question of how such complex spatio-temporal relationships from sequences of OCT volumes can be learned and how previous 3D approaches can be extended to 4D spatio-temporal data processing. To directly address the substantially increased computational efforts by moving from 3D to 4D, we evaluate our transfer learning strategy that allows transferring previous 3D models to 4D spatio-temporal data processing.

Definition of the Learning Task

Our task is to estimate the three-dimensional position $y^{[n]} \in \mathbb{R}^3$ of a marker object at a time point $t = n$ from OCT volumes in an end-to-end fashion. This process is visualized in Figure 6.1. As a first approach, we estimate the marker position from single OCT volumes $x^{[n]} \in \mathbb{R}^{n_h \times n_w \times n_d \times n_c}$. Hence, we only use spatial information and try to learn a function $f_S \colon \mathbb{R}^{n_h \times n_w \times n_d \times n_c} \to \mathbb{R}^3$. However, using a single volume for position estimation does not utilize the spatio-temporal relationships of the underlying motion, which might be beneficial to improve performance and consistency. This motivates to incorporate additional temporal information. Hence, as a second approach, we also utilize past OCT volumes to estimate the current marker position. This approach then leads to position estimation from an entire sequence of volumes $x_t = [x^{[n-p_t]}, \ldots, x^{[n-1]}, x^{[n]}]$ to estimate the current position $y^{[n]}$ with p_t for the number of past image volumes. Hence, using our spatio-temporal deep learning approaches, we try to learn a function $f_{ST} \colon \mathbb{R}^{n_h \times n_w \times n_d \times n_t \times n_c} \to \mathbb{R}^3$ from data with $n_t = p_t + 1$.

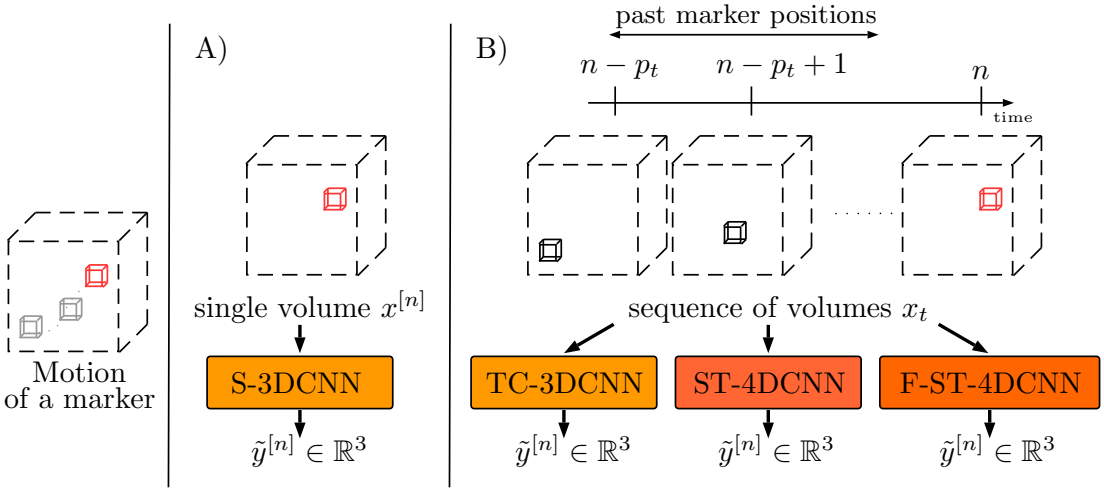

Figure 6.1: Movements of marker object (left). The marker is shown as a box for simplicity, the current position of the marker in the volumetric image is shown in red, and previous positions are indicated in gray. A) We process a single volume for position estimation with a 3DCNN approach, i.e., previous positions are not considered. B) We incorporate temporal information by using additional images of past marker positions. Our spatio-temporal deep learning approaches process an entire sequence of volumes to estimate the current marker position marked in red.

Experimental Setup and Data Sets

Our dataset consists of volumetric OCT images over time with corresponding ground truth annotation of a marker position. An experimental setup is used for data acquisition and annotation, see Figure 6.2. The data acquisition approach and the setup is based on the system presented in [393, 396]. It includes a swept-source OCT device (OMES, OptoRes), a second scanning stage with two mirror galvanometers, an achromatic lens, and a holder with a marker object that is

made of a polyoxymethylene block with a size of $1\,\mathrm{mm}^3$. The marker object is similar to previous studies that also consider OCT-based marker position estimation [168, 171]. We use volumetric OCT images with a size of $32 \times 32 \times 32$ voxels with a corresponding FOV of $3 \times 3 \times 3.5\,\mathrm{mm}^3$, resulting in an acquisition speed of 833 volumes per second. The second scanning stage includes two mirror galvanometers controlled by stepper motors that allow to shift the FOV in the lateral dimensions. The setup also includes a third stepper motor that shifts the FOV in the axial dimension by setting the reference arm pathlength of the OCT. Hence, the FOV of the OCT can be shifted in all spatial directions without moving the scan head. Note that only the relative movement between the FOV and the marker is relevant for position estimation. Hence, we move the FOV of the OCT using the stepper motors instead of the marker object, i.e., the marker object remains static during the experiments. To generate our dataset, the following steps are repeated to perform automatic data acquisition and annotation. First, a set of 60 to 90 target stepper motor positions are randomly generated that shift the marker object within the FOV. Then, the target stepper motor positions are connected using piecewise cubic spline interpolation. Afterwards, 500 motor points / target positions are sampled from the spline function, resulting in a smooth motion trajectory for the marker object. Second, one volumetric OCT image is acquired in a step-and-shoot fashion for each target motor position and ground truth annotation of the marker position is given by corresponding target motor position. We repeat this procedure to obtain the full dataset that consists of 7000 volumetric OCT images with corresponding ground truth annotation of the marker object position.

Figure 6.2: The experimental setup for data acquisition and annotation: Marker object (left); OCT setup (right). Figure adapted from [38].

Methods

To estimate the marker position from a sequence of volumes, we use and adapt our CNN architecture concept and implement four different CNN backbones combined with three different types of convolutions for 4D data processing that we introduced in Section 4.1. We explain specific details of our approaches in the following paragraphs. As backbone architectures, we consider and implement the concepts of ResNet [197], Inception [216, 442, 443], ResNeXt [504] and DenseNet [210].

S-3D. First, we estimate the marker position from a single volume. To this end, we use a single volume $x^{[n]} \in \mathbb{R}^{n_h \times n_w \times n_d \times 1}$ as input to estimate the corresponding marker position $y^{[n]} \in \mathbb{R}^3$ within the volume. To address this learning task, we combine our CNN architecture with the different backbone concepts with 3D

convolutions and pooling operations. As a result, we can perform end-to-end processing of the volumetric image to the current marker position. This approach is similar to previous work [171] and serves as our baseline.

TC-3D. Second, we incorporate temporal information by also using past image volumes. As a first approach to address this learning task, we use the same architectures as for single volume processing, except that we concatenate multiple consecutive volumes along the channel dimension of the input of the network. Hence, we consider the channel dimension of the input as a temporal dimension. The corresponding network input is $x_t \in \mathbb{R}^{n_h \times n_w \times n_d \times 1 \cdot n_t}$.

ST-4D. Third, we estimate the marker position from a sequence in an end-to-end fashion and perform joint spatio-temporal feature learning throughout the entire architecture. To this end, we replace each 3D convolution and 3D pooling of our CNN architecture with the corresponding 4D counterparts. This can also be interpreted as inflating the kernels of the 3D architectures to 4D. First, we can use the same architecture concept of the previous 3D approach without major adaptions, and second, it allows using a transfer learning strategy from 3D to 4D. Second, as explained in Section 4.1, we first train a 3DCNN, then we scale the fine-tuned weights with the temporal kernel size, and subsequently, we copy the weights along the temporal dimension and train the 4DCNN. Thereby, we initialize the kernel weights of a 4D architecture with fine-tuned weights of the corresponding 3D architecture. The input of this approach is $x_t \in \mathbb{R}^{n_h \times n_w \times n_d \times n_t \times 1}$.

F-ST-4D. Fourth, we still follow the concept of spatio-temporal feature learning but now disentangle spatial and temporal data processing to reduce the number of parameters. To this end, we use factorized spatio-temporal convolution and split a 4D spatio-temporal convolution into two subsequent convolutions, i.e., one spatial and one temporal convolution [364]. Thereby, we can achieve similar parameters compared to our 3D architectures. The input of this approach is also $x_t \in \mathbb{R}^{n_h \times n_w \times n_d \times n_t \times 1}$.

Architecture Details. Each architecture consists of an initial part with five convolutional layers, followed by two corresponding architecture modules. For the initial convolutional layers, we use 16 feature maps in the first layer and 22 in the following. The first and the second module of each architecture consists of four and five building blocks, respectively. We use a stride of two for the first block of each module and up-sample the feature dimension by a factor of two. For ResNeXt, we use four parallel convolutional paths. As our fourth architecture, we implement a DenseNet backbone. The architecture consists of one initial convolutional layer with 28 feature maps followed by three modules, while each module consists of five densely connected convolutional layers combined with bottleneck layers [197]. The modules are connected with transition layers [210]. We use a growth rate of 14 for our DenseNet architecture. For the 4D convolutions, we choose a temporal stride of one. Moreover, we use a kernel size of three for the architecture blocks. All methods are implemented in Tensorflow.

Training. We train our network with the MSE loss function between the estimated and actual marker position for 350 epochs with a batch size of $m_b = 18$ using Adam [242] for optimization. During training, we evaluate the performance of a network every ten epochs on the validation set and use the best network for our final evaluation on the test dataset. The learning rate is initially set to $\mathrm{lr} = 0.002$ and is reduced in a step-wise fashion. To improve convergence, we normalize the inputs of our networks. For our transfer learning approach, we use the same hyperparameters, except that we reduce the initial learning rate to $\mathrm{lr} = 0.0001$. For our approaches that use a sequence of volumes, we use $n_t = 5$ volumes, and the corresponding target $y^{[n]} \in \mathbb{R}^3$ of the sequence refers to the last position of the marker in one sequence, as also indicated in Figure 6.1.

Performance Measure - Evaluation and Metrics

We randomly split our dataset and use 5000 volumes for training and 1000 each for validating and testing our models. For evaluation, we consider the regression metrics MAE and rMAE. Using a simple calibration between motor steps and image coordinates, the MAE is given in micrometers. To compare our models, we test for significant differences in the median of the MAE of our methods using the Wilcoxon signed-rank test with a significance level of $\alpha = 5\%$.

Results

We report performance metrics for our different methods in Table 6.1. Our results show that using 4D spatio-temporal data improves performance for all architectures compared to position estimation from a single volume. Across the different spatio-temporal data processing methods, 4D spatio-temporal convolutions significantly ($\mathrm{p} < 0.05$) perform best, reducing the MAE by 30 % on average compared to single-volume processing. Considering the different types of backbone architectures, performance differences are small. Considering our transfer learning approach from 3D to 4D, convergence of the validation losses during training are shown in Figure 6.3. Our results highlight that substantially faster convergence is given for all architectures when fine-tuned 3D weights are used for initialization. The inference times are ∼6 ms and ∼20 ms for 3D and 4D architectures, respectively, using an nvidia GeForce GTX 1080 Ti for evaluation.

Table 6.1: Position estimation results. Comparison of our CNN architecture in combination with the different types of convolutions and backbone concepts. The corresponding number of parameters are given by #numParam. (Compare [38])

	Type	MAE [µm]	rMAE (10^{-3})	#numParam.
ResNet	S-3D	15.87 ± 14.40	13 ± 11	409 755
	TC-3D	13.39 ± 10.96	11 ± 9	411 483
	F-ST-4D	12.36 ± 10.15	10 ± 8	454 575
	ST-4D	$\mathbf{11.79 \pm 9.79}$	$\mathbf{9 \pm 8}$	1 137 459
Inception	S-3D	17.65 ± 15.48	14 ± 12	428 521
	TC-3D	14.83 ± 11.84	12 ± 9	430 249
	F-ST-4D	13.23 ± 11.36	10 ± 9	475 006
	ST-4D	$\mathbf{11.87 \pm 9.66}$	$\mathbf{9 \pm 8}$	1 161 568
ResNeXt	S-3D	16.96 ± 15.53	13 ± 12	392 367
	TC-3D	13.00 ± 12.16	10 ± 10	394 095
	F-ST-4D	12.32 ± 10.99	10 ± 9	432 903
	ST-4D	$\mathbf{11.87 \pm 10.93}$	$\mathbf{9 \pm 9}$	1 012 215
DenseNet	S-3D	16.03 ± 13.69	13 ± 11	406 723
	TC-3D	14.39 ± 11.57	11 ± 9	445 139
	F-ST-4D	12.51 ± 10.07	10 ± 8	420 205
	ST-4D	$\mathbf{11.54 \pm 9.51}$	$\mathbf{9 \pm 8}$	1 080 683

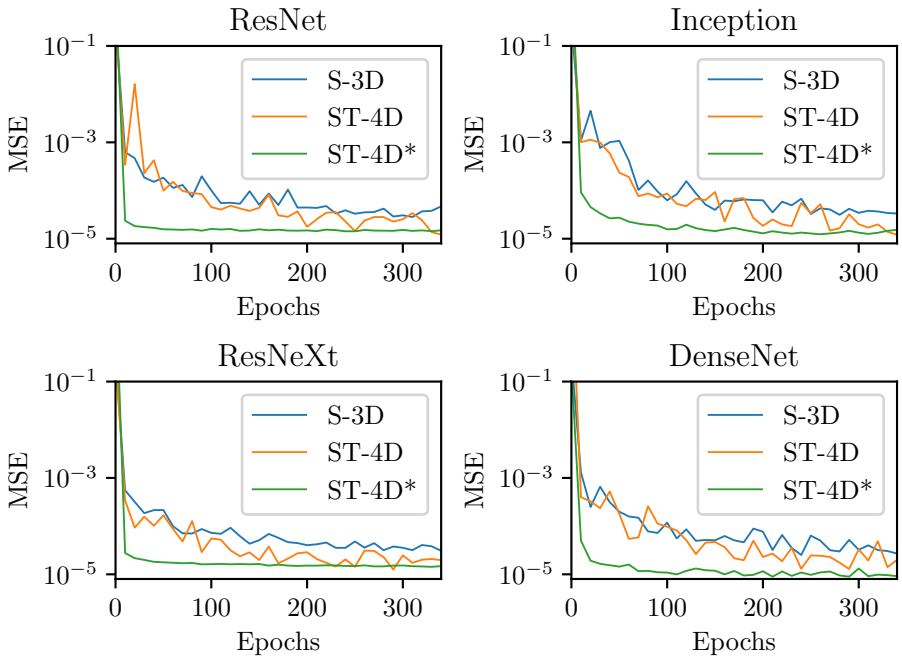

Figure 6.3: Comparison of the convergence of the validation loss of the different architectures using S-3D and ST-4D convolutions. ST-4D* refers to 4D convolutions initialized with fine-tuned 3D weights.

Discussion

We study OCT-based marker object position estimation using single volumetric images and sequences of volumetric images. This is relevant, e.g., for surgical tool tracking during minimally invasive procedures [62,134,186,359]. We compare the performance of our spatio-temporal approaches to 3DCNN architectures similar to those that have been proposed for pose estimation of a marker from single OCT volumes [171]. Our results regarding position estimation from a single volume are in a similar range compared to the previous study [171]. We present and compare different deep learning approaches for marker object position estimation and find that those using a sequence of volumetric images consistently outperform their 3D counterpart, see Table 6.1. These results are consistent across four different backbone architectures. This agrees with our expectation that subsequent object positions on a smooth trajectory captured with fast volumetric imaging result in rich 4D spatio-temporal data that can be utilized to improve position estimation performance. Our results highlight that these spatio-temporal relationships can be learned and utilized by our spatio-temporal deep learning approaches end-to-end from sequences of volumetric images. Also, our findings demonstrate that motion estimation can be performed within a few milliseconds with our 4DCNN approaches, i.e., position estimations can be provided up to 50 Hz. In summary, related to our first research question, these results demonstrate that 4D spatio-temporal relationships can be learned effectively from sequences of volumetric OCT data with our architecture concept and that such an approach can improve the performance in comparison to position estimation from a single volume.

Considering how spatio-temporal processing should be performed, even the simplest spatio-temporal processing approach, where we use the channel dimension of the input as a temporal dimension (TC-3D) clearly outperforms the previous approach (S-3DN) with a 15 % lower MAE on average. Yet, our results show that performance can further be improved with joint spatio-temporal feature learning. This indicates that using the channel dimension of the input as temporal dimension is not optimal for temporal processing. Across all architectures, 4D spatio-temporal convolutions clearly lead to the best performance with 30 % lower MAE on average compared to single-volume processing. This indicates that joint spatio-temporal feature learning is beneficial for the task at hand. We also find that factorized convolutions are effective and allow for less parameter intensive versions with competitive performance compared to full 4D convolutions. These results are similar to previous findings on 3D spatio-temporal data [364]. Considering the different backbones, while DenseNet shows the lowest overall error, the performance is not notably different compared to the other backbone architectures. This indicates that there are no clear architecture-dependent effects. Still, comparing the different concepts, ResNet and DenseNet can be considered advantageous due to their simpler architecture design and fewer hyperparameter choices and thus represent a good starting point for future studies.
A disadvantage of the 4DCNN approach is that it results in increased model complexity and increased training times compared to the 3D approach. To bring this into perspective, training our 3DCNNs takes around 3 hours, and training a

4DCNN takes around 15 hours for 350 epochs. Hence, training a 4DCNN could quickly range over several days or even weeks for future applications when large-scale datasets are used for training. To address this challenge, we presented a transfer learning strategy. To achieve a performance similar to training a 4D network from scratch for 350 epochs, only approximately 100 epochs are required, see Figure 6.3. Thus, training time can be reduced by 70 % with our transfer learning approach. These insights hold for all four backbone architectures. This is an important finding, as it enables to transfer 3D architectures to 4D with only a moderate increase in training effort. Similar, this is an interesting finding for future work that could use our transfer learning strategy when moving from previous 3D approaches to 4D data.

Overall, considering our first research question, our results demonstrate that OCT-based marker object position estimation can be improved by using 4D spatio-temporal deep learning and sequences of volumetric image data. This can be performed end-to-end and in real-time with our architecture concept. Considering our second research question, we find that learning joint spatio-temporal features by means of 4D spatio-temporal convolutions performs best across our different methods. Moreover, performing transfer learning from 3DCNNs to 4D spatio-temporal CNNs is an interesting direction to reduce the computational efforts during training.

6.1.2 4D OCT-based Tissue Motion Compensation

In the previous section, we demonstrated position estimation of a marker object using 4D OCT data and spatio-temporal deep learning. In this study, we address marker-less motion estimation of tissue for motion compensation using 4D OCT data. This is required for, e.g., FOV adjustment during surgery or motion compensation for treatment [219, 535] and requires estimating the motion between two-time points. Typically, two images are used to this end that represent a template image, and an image that represents the current state [168, 254, 398, 432]. However, such an approach uses little temporal information and might be limited in performance, e.g., if there is a small overlap between the original template and the current state, resulting from large motion. Moreover, the spatio-temporal nature of the motion is neglected and not utilized. In addition to that, only a few studies consider marker-less motion estimation, as outlined in Section 5.1. We address both of these aspects and evaluate to learn a three-dimensional translation vector of a tissue region from an entire sequence of OCT volumes compared to the approach of only using two images. Parts of this section have been published in our studies presented in [39, 40]. The studies [39, 40] were collaborative studies, and data acquisitions were performed with equal contribution of N. Gessert and the author of this work. The aspects regarding spatio-temporal deep learning were developed and evaluated by the author of this thesis and are in the focus of this work.

Considering our first research question, we study and evaluate whether markerless motion estimation in real-time using 4D OCT data can be performed with spatio-temporal deep learning in an end-to-end fashion and whether such an approach improves performance. To this end, we evaluate and compare motion estimation for compensation using only two volumes up to entire sequences. Also, we study the impact of image rotations and motion distortions on the performance.

Considering our second research question, we study how additional temporal information can be used to improve performance and how this can be addressed from an architecture and training perspective. We make use of additional temporal information not only at the input of the model using different methods but also at the model output by using a regularization strategy that forces the model to estimate motion information for previous time steps within the 4D sequence.

Definition of the Learning Task

We consider a fixed ROI showing a target tissue region in an OCT volume for motion analysis. Our goal is to estimate the three-dimensional translation vector $T_{0,n} \in \mathbb{R}^3$ of the target region between $t = 0$ and $t = n$ in an end-to-end fashion given a sequence of volumes $x_t = [x^{[0]}, x^{[1]}, \ldots, x^{[n]}]$ with $x^{[i]} \in \mathbb{R}^{n_h \times n_w \times n_d \times n_c}$. This process is visualized in Figure 6.4 and Figure 6.5. As a first approach, we estimate the relative translation $T_{0,n}$ by only using a volume pair, i.e., the template $x^{[0]}$ and the moving state $x^{[n]}$. Here, we try to learn a function $f_S \colon \mathbb{R}^{n_h \times n_w \times n_d \times 2 \times n_c} \rightarrow \mathbb{R}^3$. As a second approach, we incorporate additional temporal information and also utilize intermediate OCT volumes to estimate the translation vector. Hence, we use an entire sequence $x_t = [x^{[0]}, x^{[1]}, \ldots, x^{[n]}]$ and try to learn a function $f_{ST} \colon \mathbb{R}^{n_h \times n_w \times n_d \times n_t \times n_c} \rightarrow \mathbb{R}^3$ with $n_t \geq 2$.

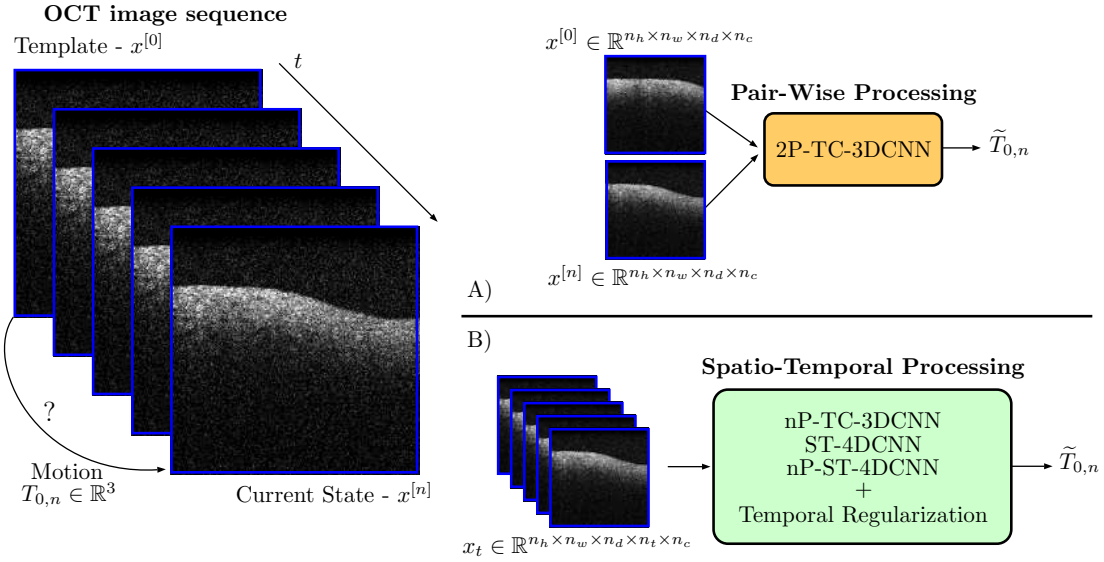

Figure 6.4: Our approach for motion estimation using sequences of OCT images. We perform all experiments with 3D volumetric OCT images and show 2D images only for simplicity of the visualization. (A) We follow the previous approach and only use an image pair to estimate the relative motion, i.e., translation $T_{0,n} \in \mathbb{R}^3$ between the images. (B) We incorporate additional temporal information at the model input and output and process an entire sequence of volumetric OCT images for motion estimation. Figure adapted from [40].

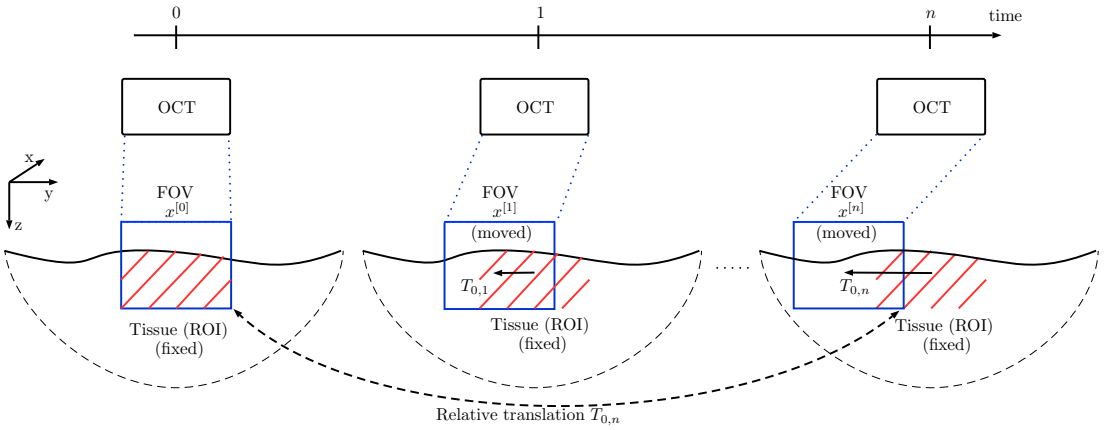

Figure 6.5: Our approach and data acquisition strategy. We use a fixed ROI and move the FOV in a step-wise fashion. In this way, we acquire a sequence of OCT volumes x_t with the corresponding ground truth annotation, i.e., relative translations $T_{0,i}$ between the initial volume $x^{[0]}$ and a volume $x^{[i]}$ of a sequence. Figure adapted from [40].

Experimental Setup and Data Sets

Our dataset consists of volumetric OCT images over time with corresponding ground truth annotation of the translation of a target region. The dataset is acquired with a setup similar to the previous Section 6.1.1. This setup is based on the system presented in [393, 396]. The setup is shown in Figure 6.6, and it consists of a robot (IRB 120, ABB) with an attached chicken breast sample, a swept-source OCT device (OMES, OptoRes) with a scan head, a second scanning stage with two mirror galvanometers, and lenses for beam focusing. We use volumes with a size of $32 \times 32 \times 32$ voxels with a corresponding FOV of approximately $5 \times 5 \times 3.5$ mm^3 and a single OCT volume can be acquired in approximately 1.2 ms. Similar to the setup used in our previous experiment, the FOV can be translated without moving the OCT scan head in all spatial directions using the second scanning stage and a stepper motor that changes the pathlength of the reference arm. Considering this and that only the relative translation is relevant for motion analysis, the FOV is moved instead of the chicken breast sample. The robot is used to automatically position the chicken breast sample for data acquisition of different ROIs. In this way, the setup can be exploited for automatic data acquisition and annotation by repeating the following steps.
First, a template volume $x^{[0]}$ is acquired in which the FOV completely overlaps with a target ROI. Second, the stepper motors translate the FOV in a step-wise fashion using smooth motion patterns, and for each position, an OCT volume is acquired in a step-and-shoot fashion, see Figure 6.5. This results in an image sequence that corresponds to a relative motion between a target ROI and the FOV. For our experiments, a single motion pattern consists of a sequence of five target translations $T_t = [T_{0,0}, T_{0,1}, T_{0,2}, T_{0,3}, T_{0,4}], T_{0,i} \in \mathbb{R}^3$. The translation $T_{0,i}$ defines the relative shift of the ROI between $t = 0$ and $t = i$, as also indicated in Figure 6.5. To generate a smooth motion pattern for data acquisition, spline interpolation is used between three positions, i.e., a starting point $T_{0,0} = [0, 0, 0]$, a randomly sampled endpoint $T_{0,4}$ and a connection point $T_{0,c}$ randomly sampled between the starting and endpoint that introduces curvature for the resulting motion pattern. Afterwards, intermediate target shifts $(T_{0,1}, T_{0,2}, T_{0,3})$ are sampled from the spline function and by using different distances between $T_{0,0}$ and $T_{0,4}$, various motion patterns with different magnitudes of motions are simulated. Figure 6.7 shows example trajectories. For a target ROI, data is acquired for 200 motion patterns. Lastly, the robot is used to position a new ROI into the OCT FOV, and the steps are repeated. To generate the entire dataset, 4D spatio-temporal OCT data is acquired for 40 ROIs. Using our motion patterns of five target translations, each acquired image sequence x_t consists of five OCT volumes with the corresponding ground truth annotation of the relative translation.
Our current setup only considers translation, and images are acquired in a step-and-shoot fashion, i.e., rotations and motion artifacts that might occur during the clinical practice are not considered. For our current setup, motion distortions are unlikely due to the high temporal resolution [523]. Still, motion artifacts might occur for slower OCT systems. For our experiments, both of the aforementioned aspects are addressed in a post-processing step. Rotations are simulated by rotating each volume $x^{[i]}$ of a sequence around the axial axis with a rotation matrix

R such that $\widetilde{x}^{[i]} = R(\gamma^{[i]})x^{[i]}$ with $\gamma^{[i]} = \frac{\gamma_{max}}{4} \cdot i$, $\forall i \in [0, 4]$ using a maximal rotation γ_{max}. Notably, we also adapt the ground truth translation vector accordingly. Moreover, we also simulate image artifacts that might result from fast and irregular motions. Following the findings of previous studies [247, 507, 523], we simulate motion distortions in a post-processing step by randomly shifting B-scans of an OCT volume along the lateral or axial axis with a probability p_{dist}. For our experiments, we evaluate shifting B-scans by one or two pixels and consider different values for p_{dist}.

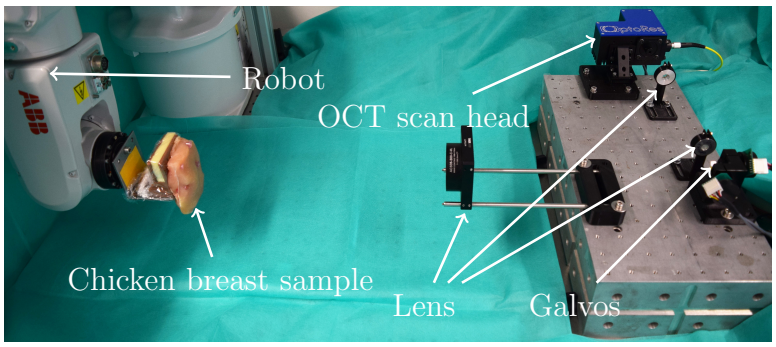

Figure 6.6: The experimental setup for data acquisition and annotation of the 4D OCT dataset. Figure adapted from [40].

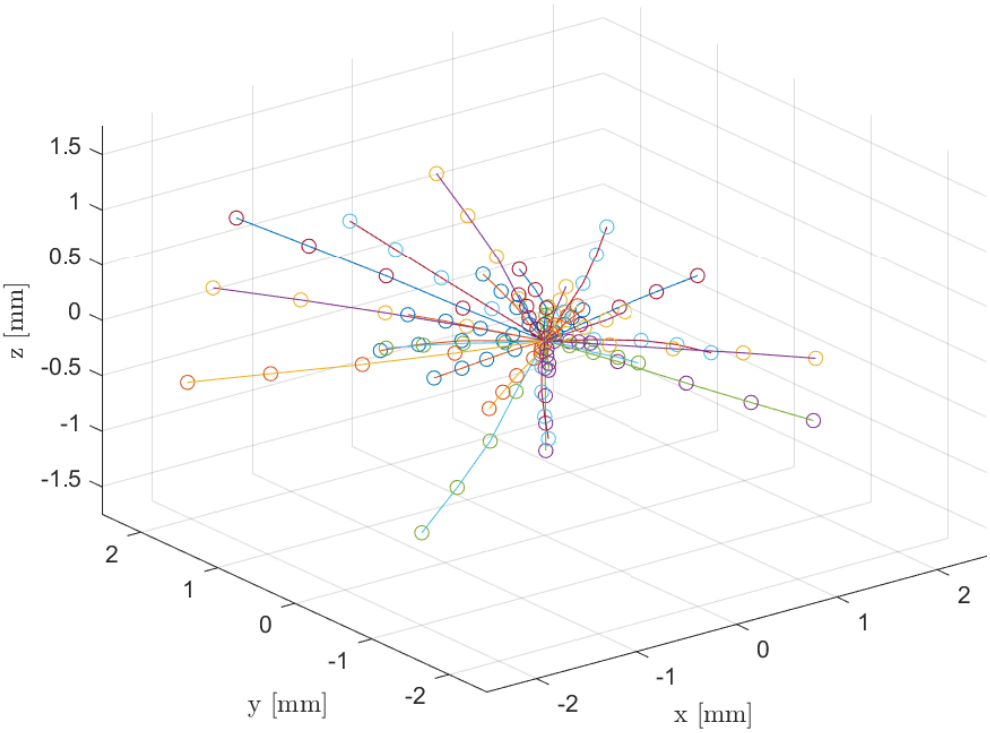

Figure 6.7: Example trajectories of the dataset. We show 30 example trajectories for the translations in the three spatial dimensions. A single trajectory consists of a sequence of five target shifts $T_{0,i}$ with $i \in 0, 1, 2, 3, 4$ shown as circles. Figure adapted from [40].

Methods

For motion compensation, we estimate the relative translation using two OCT volumes up to entire sequences by using and adapting our CNN framework with a DenseNet backbone. We choose this backbone concept due to its simplicity and based on our results from the previous experiment. Considering our different spatio-temporal deep learning methods introduced in Chapter 4, we systemically evaluate to extend pair-wise processing up to entire short-term sequences to address our learning task. To this end, we evaluate and compare five different spatio-temporal processing methods. We explain specific details of our approaches in the following paragraphs.

2P-TC-3DCNN. First, we consider the previous deep learning approach of using only two volumes as the input for motion estimation [168]. We adapt our CNN architecture concept to a two-path architecture to address this learning task. The inputs of the network are the initial volume $x^{[0]}$ and the last volume $x^{[4]}$ of a sequence to estimate the relative translation $T_{0,4}$ of a target region between the volume pair. The corresponding network input is $x_t \in \mathbb{R}^{n_h \times n_w \times n_d \times 1 \times 2}$. This approach serves as our baseline.

2P-TC-3DCNN-I. Second, we use a sequence of volumetric images for motion compensation but still use 2P-TC-3DCNN. This can be achieved by estimating the relative translation between the initial and last volume based on the sum of the relative translations between two subsequent volumes of a sequence. In this way, we can use the same network but use information from the entire sequence without increasing the model complexity or changing the architecture in general. Notably, the computational efforts are increased. That is, the network receives the input pairs $[x^{[0]}, x^{[1]}]$, $[x^{[1]}, x^{[2]}]$, $[x^{[2]}, x^{[3]}]$, $[x^{[3]}, x^{[4]}]$ and the estimations are added end-to-end to obtain the final network estimation $\widetilde{y} = \widetilde{T}_{0,4} = \sum_{i=0}^{3} \widetilde{T}_{i,i+1}$. As a result, the translation vector is obtained from incremental relative estimations between subsequent volume pairs. We train *2P-TC-3DCNN-I* end-to-end using the relative translation $T_{0,4}$ between the initial and the last volume as the training target. The corresponding network input is $x_t \in \mathbb{R}^{n_h \times n_w \times n_d \times n_t \times 1}$, which is then divided into the volume pairs.

nP-TC-3DCNN. Third, we extend the idea of 2P-TC-3DCNN to the processing of an entire sequence of volumes without estimating incremental translations. We adapt our architecture concept to a multi-path Siamese architecture concept, where the number of paths is equal to the sequence length n_t. At the fusion point of the multi-paths, the outputs are stacked in the channel dimension. Subsequently, we process the combined output with a 3DCNN by using 3D convolution and pooling operations for our DenseNet baseline architecture. This approach processes the spatio-temporal information by means of channel-wise interactions at the fusion point, and no joint spatio-temporal feature learning is performed throughout the architecture. The corresponding network input is $x_t \in \mathbb{R}^{n_h \times n_w \times n_d \times n_t \times 1}$.

ST-4DCNN. Fourth, we process the sequence of images directly end-to-end by adapting our architecture concept to a 4D spatio-temporal CNN. This can be achieved by using 4D convolutions and pooling operations for our architecture. In this way, we perform joint spatio-temporal feature learning from network input to network output, i.e., from the image sequence to the relative translation vector of a target region. The corresponding network input is $x_t \in \mathbb{R}^{n_h \times n_w \times n_d \times n_t \times 1}$.

nP-ST-4DCNN. Fifth, we consider our mixed 3D/4D spatio-temporal approach which combines a multi-path Siamese 3DCNN architecture and a 4D spatio-temporal CNN. Using this concept, each volume of a sequence is first processed with a multi-path Siamese 3DCNN, and afterwards, we concatenate the outputs along the temporal dimension. Then, we perform joint spatio-temporal processing by using 4D convolutions and pooling operations for our DenseNet baseline architecture. In this way, each volume of a sequence is first pre-processed, and afterwards, joint spatio-temporal feature learning is performed. Also, by sharing parameters for the Siamese part, we reduce the number of parameters compared to the full 4D spatio-temporal CNN approach (ST-4DCNN). The corresponding network input is $x_t \in \mathbb{R}^{n_h \times n_w \times n_d \times n_t \times 1}$.

Architecture Details. All our architectures consist of initial convolutional layers, followed by three DenseNet blocks which are connected with transition layers with a spatial stride of two for downsampling. For our multi-path Siamese architectures, we use three subsequent convolutional layers with 20 feature maps each to process the input volumes. After the siamese part, we use a convolutional layer with a spatial stride of two for downsampling. Afterwards, we use our DenseNet architecture, while each DenseNet block of our architecture consists of two convolutional layers with a growth rate of 10. All methods are implemented in Tensorflow.

Training. To train our different approaches for end-to-end motion compensation using 4D spatio-temporal OCT data, we minimize the MSE loss function between the defined ground-truth translation vector $y = T_{0,4} \in \mathbb{R}^3$ and our estimated translation vector $\widetilde{y} = \widetilde{T}_{0,4} \in \mathbb{R}^3$, i.e., the estimated relative motion between an initial volume $x^{[0]}$ and a final volume $x^{[4]}$. This leads to the training loss function

$$\mathcal{L} = \frac{1}{m_b} \sum_{j=1}^{m_b} \left\| T_{0,4}^{\{j\}} - \widetilde{T}_{0,4}^{\{j\}} \right\|^2 . \tag{6.1}$$

We also evaluate our temporal regularization strategy introduced in Section 4.4.1. To this end, we formulate an auxiliary task where we also estimate relative translations of past volumes w.r.t. the initial position using the image sequence $x_t = [x^{[0]}, x^{[1]}, x^{[2]}, x^{[3]}, x^{[4]}]$ as input. We perform this step to also integrate additional temporal information at the model output and to provide more information about the motion trajectory. We hypothesize that, thereby, more consistent and robust performance can be achieved. Hence, we also consider to estimate the relative translations for two previous volumes of a sequence, i.e., $x^{[3]}$ and $x^{[2]}$. This

requires to extent the network output to also estimate $\widetilde{T}_{0,3} \in \mathbb{R}^3$ and $\widetilde{T}_{0,2} \in \mathbb{R}^3$ such that the final output of network is $\widetilde{y} \in \mathbb{R}^9$. Note, the auxiliary task is not required to perform motion compensation during evaluation. Using our concept from Section 4.4.1 this leads to

$$\mathcal{L} = \frac{1}{m_b} \sum_{j=1}^{m_b} \sum_{i=0}^{2} \theta_i \left\| T_{0,4-i}^{\{j\}} - \widetilde{T}_{0,4-i}^{\{j\}} \right\|^2 \tag{6.2}$$

We set $\theta_{i=0} = 1$, and for our experiments, we evaluate and report different values for the weighting parameters $\theta_{i\neq 0} \in [0, 1]$. We tune the learning rate individually for each of our approaches and train all our models for 150 epochs, using Adam [242] for optimization with a batch size of $m_b = 50$. During training, we evaluate the performance of a network every ten epochs on the validation set and use the best network for our final evaluation on the test dataset. To improve convergence, we normalize the inputs of our networks.

Performance Measure - Evaluation and Metrics

To evaluate our models on previously unseen tissue regions, we strictly split our data based on the different ROIs and use data from 30 ROIs for training and from 5 ROIs each for validation and testing. For comparison of our methods, we report the MAE and the rMAE for our experiments. Using a simple calibration between motor steps and image coordinates, the MAE is given in micrometers. We test our results for significant differences in the median of the rMAE using Wilcoxon signed-rank test with $\alpha = 5\%$ significance level.

Results

First, we report the performance metrics for our different methods in Table 6.2 and show the number of parameters and inference times for the models in Table 6.3. Our results show that processing 4D spatio-temporal data with nP-ST-4DCNN significantly ($p < 0.05$) performs best with an inference time of around 9 milliseconds. The second best approach is 2P-TC-3DCNN-I, with an inference time of around 5 milliseconds.
Second, Figure 6.8 shows the MAE w.r.t. different motion magnitudes. For all our approaches, the MAE increases with an increasing magnitude of the motion. The MAE increases the most for 2P-TC-3DCNN and the least for nP-ST-4DCNN with an increasing magnitude of the motion.
Third, we systematically consider the impact of rotations and motion distortions on the performance of our approach nP-ST-4DCNN and report the results in Table 6.4 and Table 6.5, respectively. Our results show that rotations only have a notable impact on performance for larger rotations, angles $\gamma_{max} \geq 20°$. Similar, motion distortions only have a major impact for a large amount of motion distortions ($p_{dist} \geq 50\%$) and when not considered during training.
Fourth, we combine our temporal regularization strategy with nP-ST-4DCNN. Our results in Table 6.6 show that our regularization strategy significantly ($p < 0.05$) improves performance for a weighting of $\theta_1 = 0.75$ and $\theta_2 = 0.75$.

Table 6.2: Results for the different models for motion estimation. The MAE is given in mm. (Compare [40])

	MAE_x	MAE_y	MAE_z	rMAE
2P-TC-3DCNN	0.45 ± 0.52	0.42 ± 0.52	0.18 ± 0.15	0.34 ± 0.39
2P-TC-3DCNN-I	0.20 ± 0.21	0.15 ± 0.16	0.13 ± 0.12	0.16 ± 0.17
nP-TC-3DCNN	0.35 ± 0.45	0.18 ± 0.25	0.11 ± 0.09	0.21 ± 0.26
ST-4DCNN	0.22 ± 0.21	0.20 ± 0.24	0.13 ± 0.11	0.19 ± 0.19
nP-ST-4DCNN	$\mathbf{0.16 \pm 0.18}$	$\mathbf{0.13 \pm 0.15}$	$\mathbf{0.10 \pm 0.09}$	$\mathbf{0.13 \pm 0.14}$

Table 6.3: Inference times and number of parameters (#numParam.) for all models (Compare [40])

	inference time [ms]	#numParam.
2P-TC-3DCNN	$\mathbf{3.74 \pm 0.52}$	**143 913**
2P-TC-3DCNN-I	5.84 ± 0.32	**143 913**
nP-TC-3DCNN	5.23 ± 0.27	208 713
ST-4DCNN	9.78 ± 0.74	270 283
nP-ST-4DCNN	9.34 ± 0.67	258 323

Table 6.4: Evaluation of the motion estimation performance for different rotation angles during motion. The relative rotation around the axial axis between the initial template volume and the last volume of a sequence is given by γ_{max}. Results are shown for the architecture nP-ST-4DCNN. The MAE is given in mm. (Compare [40])

γ_{max}	MAE_x	MAE_y	MAE_z	rMAE
2°	0.16 ± 0.18	0.13 ± 0.15	0.09 ± 0.09	0.13 ± 0.14
5°	0.16 ± 0.18	0.14 ± 0.15	0.10 ± 0.09	0.14 ± 0.14
10°	0.17 ± 0.18	0.16 ± 0.15	0.10 ± 0.09	0.15 ± 0.14
20°	0.19 ± 0.20	0.23 ± 0.19	0.10 ± 0.09	0.18 ± 0.16

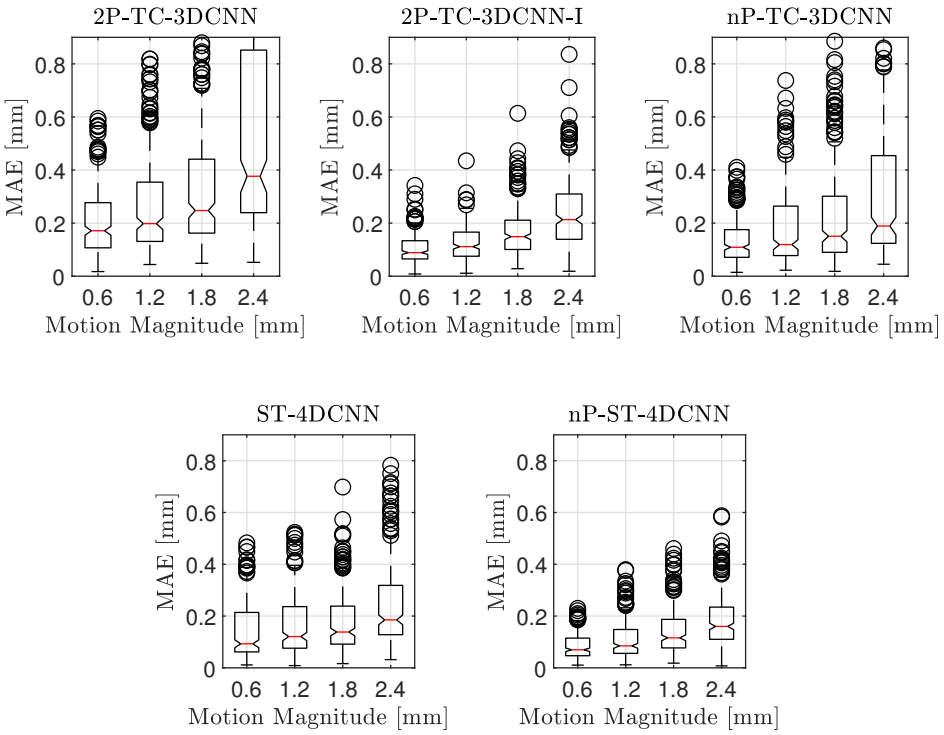

Figure 6.8: MAE for increasing motion magnitudes between $x^{[0]}$ and $x^{[n]}$. Figure adapted from [40].

Table 6.5: Evaluation of the motion estimation performance for different motion distortions. p_{dist} refers to the probability that a B-scan is shifted. E-1 and E-2 refer to shifting the B-scans one and two voxels during evaluation, respectively. T/E-2 refers to shifting the B-scans two voxels during training and evaluation. Results are shown for the architecture nP-ST-4DCNN. The MAE is given in mm. (Compare [40])

Type	p_{dist}	MAE_x	MAE_y	MAE_z	rMAE
E-1	50%	0.31 ± 0.33	0.29 ± 0.29	0.14 ± 0.11	0.25 ± 0.24
E-1	25%	0.20 ± 0.22	0.20 ± 0.20	0.11 ± 0.10	0.17 ± 0.17
E-1	10%	0.16 ± 0.18	0.16 ± 0.17	0.10 ± 0.09	0.14 ± 0.15
E-2	50%	0.33 ± 0.35	0.28 ± 0.28	0.14 ± 0.12	0.25 ± 0.24
E-2	25%	0.20 ± 0.21	0.20 ± 0.21	0.12 ± 0.10	0.17 ± 0.17
E-2	10%	0.17 ± 0.18	0.15 ± 0.16	0.10 ± 0.09	0.14 ± 0.14
T/E-2	50%	0.18 ± 0.21	0.15 ± 0.15	0.10 ± 0.08	0.14 ± 0.15

Table 6.6: Performance metrics for our architecture nP-ST-4DCNN combined with our temporal loss regularization approach using different weighing factors θ_1, θ_2. The MAE is given in mm. (Compare [40])

θ_1	θ_2	MAE_x	MAE_y	MAE_z	rMAE
0	0	0.16 ± 0.18	0.13 ± 0.15	0.10 ± 0.09	0.13 ± 0.14
1	0	0.15 ± 0.22	0.12 ± 0.13	0.11 ± 0.10	0.13 ± 0.15
0.75	0	0.14 ± 0.13	0.11 ± 0.10	0.13 ± 0.10	0.14 ± 0.11
0.5	0	0.10 ± 0.09	0.14 ± 0.11	0.10 ± 0.08	0.12 ± 0.10
0.25	0	0.11 ± 0.11	0.14 ± 0.13	0.11 ± 0.09	0.12 ± 0.11
1	1	0.11 ± 0.10	0.19 ± 0.17	0.10 ± 0.09	0.14 ± 0.12
0.75	**0.75**	$\mathbf{0.09 \pm 0.09}$	$\mathbf{0.11 \pm 0.10}$	$\mathbf{0.10 \pm 0.08}$	$\mathbf{0.10 \pm 0.09}$
0.75	0.5	0.12 ± 0.10	0.10 ± 0.11	0.10 ± 0.08	0.11 ± 0.10

Discussion

We study markerless OCT-based motion estimation, which is relevant for, e.g., FOV adjustment during surgery or motion compensation for treatment [219, 535]. Our results in Table 6.20 show that the previous approach of a two-path method (2P-TC-3DCNN) [168, 432] that uses the start (template) and the end volume (current state) performs worse than all our other spatio-temporal deep learning methods that use sequences of volumes. This highlights that 4D spatio-temporal information improves the performance and also highlights that our spatio-temporal deep learning approaches are well suited to learn the underlying spatio-temporal features of the motion. Our results show that for faster movements that lead to a smaller overlap between subsequent volumes of a sequence, using an entire sequence is beneficial, see Figure 6.32. This can likely be explained by the reduced overlap between the template volume and the current volume, which makes it more difficult to find correspondence between the volumes. This supports our assumption that entire sequences should be used for motion compensation. Our results confirm this assumption, and we observe a notably improved performance for larger motion magnitudes using our spatio-temporal deep learning approaches in comparison to the previous approach that only uses two volumes (2P-TC-3DCNN) [168, 432]. Moreover, we also evaluate how rotations between subsequent volumes and motion distortions affect performance for our best performing approach nP-ST-4DCNN. Although we did not consider rotations during training, our results in Table 6.4 show that performance remains robust. Future work could also integrate rotations in the training process, which likely will reduce the impact of rotations on the performance. Moreover, this approach could also allow to estimate not only the translation of a target region but also the relative rotation angles. Furthermore, for our current OCT setup where a single volume can be acquired in 1.2 ms, motion artifacts are unlikely considered the findings of Zawadzki et [523]. However, we simulate the problem of potential motion artifacts that result from fast and irregular motions in a post-processing step to ensure that our approach is also applicable to slower OCT devices. For a slower OCT system, motion artifacts would be present during training and evalu-

ation. Table 6.5 shows that in such a scenario, motion artifacts only have a minor impact on performance. Our results indicate that with our approach also robust performance can be achieved in the presence of rotations and motion artifacts. Thus, considering our research questions of this work, 4D spatio-temporal features can be learned from 4D OCT data to perform markerless motion estimation and using 4D spatio-temporal data can improve the performance and robustness for various magnitudes of motion.

Considering our second research question of this work, we evaluate and compare different spatio-temporal deep learning approaches using the same underlying CNN architecture concept. We find that the different methods can be effectively integrated into the architecture and that motion compensation can be performed with all approaches. However, there are substantial performance differences between the methods. Our results show that our approach, nP-TC-3DCNN, which is a direct extension of a two-path method to multiple volumes, improves motion estimation performance. However, our approaches that perform joint spatio-temporal feature learning by means of 4D operations perform even better. Comparing our different spatio-temporal deep learning methods, nP-ST-4DCNN performs best. This mixed 3D/4D spatio-temporal deep learning approach improves performance by a larger margin compared to the previous approach that only considers volume pairs [168, 432]. Our second best approach is 2P-TC-3DCNN-I, where motion estimation is performed using image-pairs of the sequence. Our results in Table 6.20 and Figure 6.8 clearly demonstrate that nP-ST-4DCNN leads to a lower MAE and fewer outliers compared to the second best approach. This indicates that joint-spatio-temporal feature learning from an entire sequence end-to-end performs better than combining pair-wise motion estimations or using channel-wise interactions. Moreover, comparing nP-ST-4DCNN and ST-4DCNN highlights that a mixed 3D/4D approach performs better than a single-stream 4D approach. These results indicate that multi-path processing is an effective pre-processing step of the 4D spatio-temporal data that improves overall performance before learning joint-spatio-temporal features. Similar results have also been found in the natural image domain [505, 545]. Our results in Table 6.3 show that all our approaches can perform motion estimation within a few milliseconds. Nevertheless, the improved performance of our best performing approach nP-ST-4DCNN comes at the cost of an increased inference time, making our second best approach 2P-TC-3DCNN-I with notably lower inference time a competitive approach. Thus, further exploring mixed 3D/4D approaches for motion analysis is an interesting direction, and, e.g., fusing the multi-path at a later stage in the architecture could be an effective approach to reduce the complexity of the 4D spatio-temporal deep learning approach.
Furthermore, we also evaluated to use additional temporal information as a regularization approach by formulating an auxiliary task. This approach only requires the adaption of the loss function and to extend the output of the model, i.e., no major adaptions of the approach are required. Our results in Table 6.6 demonstrate that thereby performance can be improved. This is an interesting finding, as our regularization approach can easily be performed for different methods and

only requires adjusting the output of the model without substantial architecture changes. Still, our results show that fine-tuning of the weighing parameters θ is required to improve the performance. Incorporating additional auxiliary tasks, such as estimating the velocity of a motion, could also be an interesting direction for future studies.

Overall, considering our first research question, our findings highlight that using entire sequences of OCT volumes and 4D spatio-temporal processing with our deep learning methods can be performed and improves motion estimation performance. Our results show that motion estimation can be performed markerless, end-to-end, and within a few milliseconds. Considering our second research question, learning joint-spatio-temporal features performs best, and using a mixed 3D/4D spatio-temporal approach turns out to be effective for motion estimation. Moreover, temporal information can also be used at the output of the network for regularization to improve performance even further.

6.1.3 4D US-based Tissue Motion Compensation

In this study, we evaluate our 4D spatio-temporal deep learning approaches in the context of motion analysis during radiotherapy using long-term sequences of US image volumes, i.e., sequences that range from several seconds up to several minutes. Real-time motion analysis is an important problem in radiation therapy, however, it remains challenging, especially using 4D spatio-temporal data [20, 98, 108, 112, 212]. Parts of this section have been published in our study presented in [42] (©2023 IEEE). The study [42] was a collaborative study, the development of the experimental setup, data acquisition, and evaluation with conventional image processing methods were performed by J. Sprenger. The aspects regarding spatio-temporal deep learning were developed and evaluated by the author of this thesis and are in the focus of this work.

Considering our first research question, we study and evaluate whether markerless motion analysis using long-term US image sequences can be performed with our spatio-temporal deep learning approach in an end-to-end fashion and in real-time. Also, we evaluate whether motion forecasting and estimation can be performed with a single approach directly from the image data. In this way, multiple challenges of motion compensation during radiotherapy could be addressed with a single approach directly from the image data, as outlined in Chapter 5.

Considering our second research question, we evaluate how this task can be performed from an architecture and training perspective. In particular, we study and evaluate how sequences of hundreds of volumes can be processed efficiently and effectively and whether using entire long-term 4D sequences improves performance compared to using volume pairs or short-term sequences for motion analysis. We present and develop all these approaches based on our spatio-temporal architecture concept of this work.

Definition of the Learning Task

We perform markerless motion analysis using a fixed ROI displaying a target region in the US volume, see Figure 6.9 and Figure 6.10. Motion estimation is performed between a template volume and US volumes that capture the motion of the target over time. Formally, a sequence of US volumes is given by $x_t = [x^{[0]}, x^{[1]}, \dots, x^{[n]}]$ with $x^{[i]} \in \mathbb{R}^{n_h \times n_w \times n_d \times n_c}$ with the template volume $x^{[0]}$ at time step $t = 0$ and the volume at the current time step $t = n$ is denoted by $x^{[n]}$. We try to estimate the three-dimensional translation vector $T_{0,n} \in \mathbb{R}^3$ of the target between $t = 0$ and $t = n$ from the sequence of volumetric US data x_t in real-time and in an end-to-end fashion. For motion forecasting, we also consider to predict the translation vector $T_{0,n+p_h} \in \mathbb{R}^3$ for $t = n + p_h$ using a prediction horizon $p_h \in \mathbb{N}$. First, we follow the common approach and estimate the translation by only using a volume pair, i.e., the template $x^{[0]}$ and the moving state $x^{[n]}$. Here, we try to learn a function $f_S \colon \mathbb{R}^{n_h \times n_w \times n_d \times 2 \times n_c} \to \mathbb{R}^3$. As a second approach, we incorporate additional temporal information to estimate the translation vector and future translation vectors. Hence, we use an entire sequence $x_t = [x^{[0]}, x^{[1]}, \dots, x^{[n]}]$ and try to learn a function $f_{ST} \colon \mathbb{R}^{n_h \times n_w \times n_d \times n_t \times n_c} \to \mathbb{R}^{3+p_h \cdot 3}$. Thereby, we try to perform both motion estimation and forecasting with a single machine learning model directly from a sequence of 3D US images.

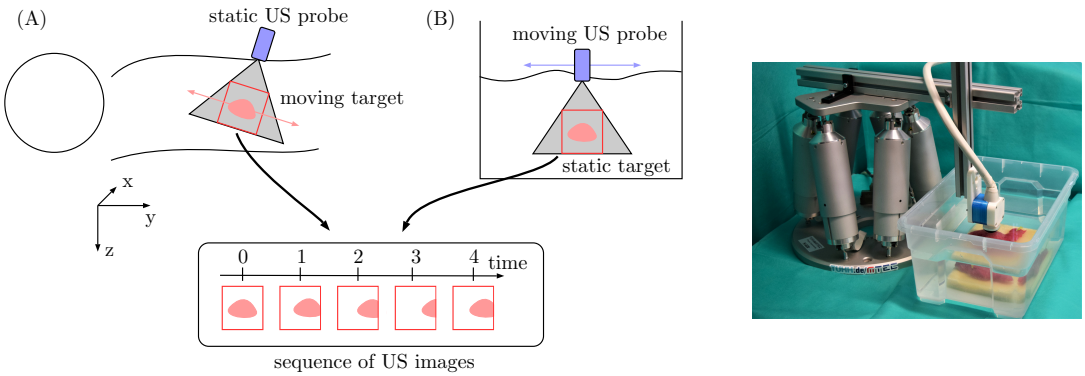

Figure 6.9: The general problem setting. (A) A target moving inside a patient is observed with a static US probe. (B) A moving ultrasound probe with respect to a static target. Note that the resulting image volume sequence of both scenarios is the same. We consider volumetric images and we only show 2D images for simplicity. Right, the experimental setup for data acquisition is shown. Figure adapted from [42] (©2023 IEEE).

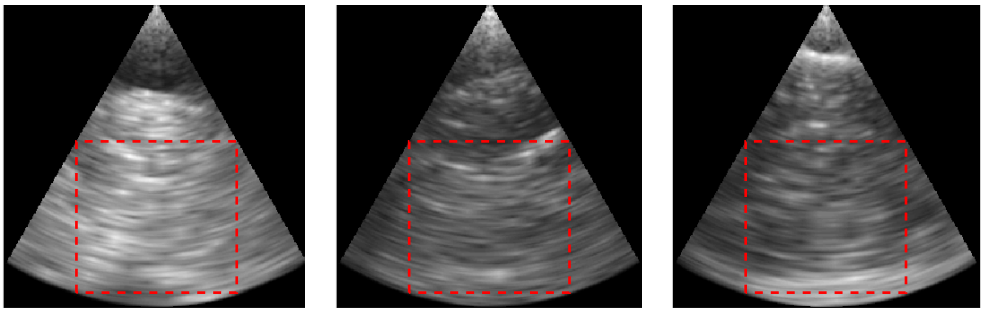

Figure 6.10: The ROI considered for tracking is visualized by the red box. Exemplary B-scans (slices) extracted from US volumes. From left to right, we show chicken tissue, bovine liver, and turkey tissue. Note that no distinct landmarks are visible. Figure adapted from [42] (©2023 IEEE).

Experimental Setup and Data Sets

For evaluation and training of our methods, we consider a dataset that consists of sequences of volumetric US images with ground truth annotations of translation vectors. The dataset is acquired with an experimental setup that allows for automated data acquisition with precise ground truth annotation. We also presented a similar setup in our study presented in [431]. The setup includes a matrix transducer (custom volume probe, Vermon) with a center frequency of 3 MHz, a US system (Griffin, Cephasonics Ultrasound), and a hexapod robot (H-820, Physik Instrumente). For imaging, volumes with a size of $268 \times 268 \times 268$ voxels are acquired with an acquisition rate of up to 11 volumes per second. The FOV covers approximately $40 \times 40 \times 40$ mm^3 assuming a homogenoues speed of sound of 1540 m s^{-1}.
The setup and the approach for data acquisition are shown in Figure 6.9. To acquire the dataset, the robot moves the US probe in three spatial dimensions according to a trajectory in a container filled with water. Recording volumetric images and robot positions then allows for automatic data acquisition and annotation. Note, only relative motion between the transducer and tissue is relevant for data acquisition. The formulate trajectories for data acquisition, a database of actual motion traces recorded during CyberKnife treatments at Georgetown University Hospital is used [138]. To acquire an image sequence according to a trajectory, segments of respiratory movements with a duration of around 25 seconds are extracted from the motion traces of the database, resulting in sequences with a length of around 280 US volumes each. No particular breathing phase is considered as starting point for a trajectory, and motion patterns with various motion behaviors, maximum amplitudes, and breathing cycles are used for data acquisition. To consider different target appearances and types, data acquisition is performed for chicken breast tissue, bovine liver, and turkey breast tissue using multiple tissue samples within each tissue type. Figure 6.10 shows exemplary B-scans for the different tissue types.
During data acquisition, only translational movements are performed without any deformations of the target tissue. However, deformation can be present during the actual clinical application [294]. Thus, we address this aspect in a post-

processing step, following the approach of [86, 372]. To synthetically deform the volumes of an image sequence, we use displacement vectors that are interpolated from a coarse $3 \times 3 \times 3 \times 5$ grid, where the last axis refers to the temporal dimension. Using our sequences of 280 volumes, this results in a temporal grid distance of approximately one breathing cycle. We draw the displacements of the grid points from a Gaussian distribution with a standard deviation σ_D. We sample displacements for all three spatial dimensions. Afterwards, voxel-wise displacement vectors are interpolated for the entire sequence, and the displacement field is used to deform an image sequence. We systematically evaluate different degrees of deformations by using different values for σ_D, and also we vary the number of grid points that are used for displacement.

Methods

To address the learning task and the challenge of learning from a sequence of hundreds of volumes, we adapt our spatio-temporal deep learning architecture in different fashions and study pair-wise image processing up to entire long-term sequences. We explain specific details of our approaches and adaptions in the following paragraphs. As a general architecture backbone, we use the concept of DenseNet [210] due to its parameter efficiency, strengthened feature propagation, and the results of our previous experiments. For motion analysis, we use a fixed ROI with a size of $128 \times 128 \times 128$ voxels, i.e., $\sim 19 \times 19 \times 19$ mm^3, see Figure 6.10.

2P-TC-3DCNN. First, we perform motion analysis based on volume pairs. To this end, we adapt our architecture concept to a two-path Siamese 3D-CNN approach. We train this approach in an end-to-end fashion to estimate the translation vector between a template volume and the current state using two different concepts. As a first concept, we directly estimate the translation between the volume pair $(x^{[0]}, x^{[n]})$, we refer to this approach as non-incremental (NI) estimation. The output of this approach is $\widetilde{y}^{[n]} = \tilde{T}_{0,n} \in \mathbb{R}^3$, i.e., the three-dimensional translation vector between the volume pair. As a second concept, we estimate the translation between consecutive volume pairs $(x^{[n-1]}, x^{[n]})$, and to obtain the translation vector between the template and the current volume, we then add the estimated incremental translations $\tilde{T}_{0,n} = \sum_{i=1}^{n} \tilde{T}_{i-1,i}$. The output of this approach for a volume pair $(x^{[i-1]}, x^{[i]})$ is $\widetilde{y}^{[i]} = \tilde{T}_{i-1,i} \in \mathbb{R}^3$. We call these approaches 2P-TC-3DCNN-NI and 2P-TC-3DCNN-I.

nP-ST-4DCNN. Second, we use additional temporal information for incremental motion estimation. To this end, we adapt our multi-path Siamese 4D-CNN approach that extends pair-wise processing to entire sequences. Using this approach, we use a local motion history for I-motion estimation and consider a history of three image volumes. The output of this approach for a sub-sequence of volumes $(x^{[i-4]}, x^{[i-3]}, x^{[i-2]}, x^{[i-1]}, x^{[i]})$ is $\widetilde{y}^{[i]} = \tilde{T}_{i-1,i} \in \mathbb{R}^3$. We hypothesize that by using additional temporal information, incremental motion estimation performance can be improved compared to the simple approach that only uses two volumes.

DenseConvGRU. Third, we estimate the translation between the volume pair $(x^{[0]}, x^{[n]})$ using the entire available temporal information between the volumes, i.e., the entire available sequence of volumetric images. This requires processing of sequences with variable lengths and also requires an efficient approach to handle sequences of several hundredths of volumes in real-time that are available after a few seconds of tracking. To this end, we study and evaluate our mixed approach of ConvGRU modules and our CNN architecture concept that fits these requirements. As outlined in Section 4.3, our approach can be used in a many-to-many fashion using variable sequence length, i.e., an input sequence of US volumes can be mapped to an output sequence of motion estimates. Motion estimates are given by the output of the network $\tilde{y} = \tilde{T}_{0,i} \in \mathbb{R}^3$ in one shot for each input volume $x^{[i]}$, and motion analysis can be performed for an ongoing input sequence. We consider this as particularly suited for the task and inherently efficient, as only the new input volume $x^{[n]}$ needs to be processed by the network. Recall from Section 4.3 that the motion history is encoded at different scales in the previous outputs of the ConvGRU modules, i.e., $z_{front}^{[n-1]}$, $z_{middle}^{[n-1]}$, $z_{end}^{[n-1]}$. We hypothesize that motion analysis benefits from spatio-temporal features at different scales. To analyze this aspect, we perform an in depths analysis w.r.t. the architecture variants, and we consider a module placed at the front, middle, end, or at all positions. We refer to these variants as DenseConvGRU-front/middle/end/all.

Architecture Details. For our 3D and 4DCNN approaches, we consider three convolutional layers for the initial multi-path part of our methods, with 14 feature maps for each layer. After the multi-path part of our networks, we use a convolutional layer with 28 feature maps, followed by our DenseNet architecture. For the DenseNet part of our CNN-based architectures, we use three DenseNet blocks connected with transition layers [210], while each block has three layers and a growth rate of 10. To account for the increased complexity of our DenseConvGRU approach, we choose two instead of three convolutional layers within the DenseNet blocks, and we replace the last DenseNet block with a single convolution.

Training. To train our approaches 2P-TC-3DCNN and nP-ST-4DCNN that consider image pairs and short-term sequences, respectively, we use a MAE loss function between the ground truth and estimated translations. Training a recurrent approach with 4D data and long-term sequences is a complex task, as outlined in Section 4.4.2. Hence, we use our training approach that combines truncated BPTT and curriculum learning. For $\widetilde{n}_t > r$, we sample $r \sim \mathcal{U}(b, \widetilde{n}_t)$ from a discrete uniform distribution in each training iteration. In this way, we increase the variance during training. Using a MAE loss function for our regression task to perform motion estimation, this leads to

$$\mathcal{L}(\tilde{T}, T) = \frac{1}{m_b} \frac{1}{b} \sum_{i=1}^{m_b} \sum_{l=r-b}^{r} \left| \tilde{T}_{0,l}^{\{i\}} - T_{0,l}^{\{i\}} \right| \tag{6.3}$$

with m_b for the number of samples in a training batch. Moreover, using approach DenseConvGRU that uses information from the entire image sequence, we also

address motion forecasting directly from the image sequence. To this end, we extend our network output to $\widetilde{y} \in \mathbb{R}^{3+p_h \cdot 3}$ and adapt our loss function to

$$\mathcal{L}(\tilde{T}, T) = \frac{1}{m_b} \frac{1}{b} \sum_{i=1}^{m_b} \sum_{l=r-b}^{r} \sum_{j=0}^{p_h} \left| \tilde{T}_{0,l+j}^{\{i\}} - T_{0,l+j}^{\{i\}} \right| \tag{6.4}$$

For optimization, we use the Adam optimizer [242]. For our CNN approaches, we use a batch size of $m_b = 20$ and a learning rate of $\mathrm{lr} = 0.001$. For our DenseConvGRU approaches, we use a batch size of $m_b = 5$ and use a learning rate of $\mathrm{lr} = 0.0005$. We normalize the voxel intensities the inputs to have a zero mean and standard deviation of one. We train our DenseConvGRU approach for 70 epochs, using $\widetilde{n}_t = r = b = 50$ for 20 epochs, and then we set $\widetilde{n}_t = 150$ for 20 epochs, and for the last 30 epochs, we set $\widetilde{n}_t = 280$. To reduce the computational complexity and memory requirements, we downsample the input volumes from a size of $128 \times 128 \times 128$ voxels to a size of $32 \times 32 \times 32$ voxels. Moreover, to augment our training data, we randomly swap the lateral volume axis, while also transforming the labels accordingly.

NCC. Moreover, we consider results from NCC as a conventional tracking method [193, 194, 431] for comparison. We evaluate results from this approach applied in an I-fashion (NCC-I) and NI-fashion (NCC-NI). Outlier estimations are filtered for the NI-motion estimation approach by comparing the position estimation at time step $t = n$ with the position estimation at the previous time step $t = n - 1$. Estimations with NCC are performed using the full resolution images.

Performance Measure - Evaluation and Metrics

The data sets, and our data splits are summarized in Table 6.7. Note, we do not use data from the same motion trace or tissue sample across our different sets. Moreover, we only train our approaches with data that comes from chicken breast tissue. In this way, we can evaluate our approaches for completely unseen tissue types and structures. Also, we use one long-term trajectory of seven minutes for evaluation. Following previous studies [97, 98, 282], we consider the TE as our main performance metric, which is directly relevant for margin planning during radiation therapy. The TE is estimated using the Euclidean distance between the estimated position $\tilde{T}_{0,n}$ and the ground truth position $T_{0,n}$. We report mean and standard deviation as well as the 95% percentile of the TE across all estimations. Also, we report the aCC as a relative metric. We test for significant differences in the median of the TE of our methods using Wilcoxon signed-rank test with a significance level of $\alpha = 5\%$.

Table 6.7: The different tissue datasets with the corresponding trajectories. We report the number of trajectories (#numT) as well as the mean and standard deviation of the amplitudes across the trajectories of a set. (Compare [42] (©2023 IEEE))

set	sequence	#numT	amplitude [mm]
training (chicken)	liver	20	11.22 ± 10.19
	pancreas	10	5.91 ± 2.19
validation (chicken)	liver	8	14.55 ± 4.97
	pancreas	2	6.81 ± 0.79
test (chicken)	liver	10	10.29 ± 3.58
	pancreas	7	7.97 ± 1.23
test (liver)	liver	9	10.58 ± 2.95
test (turkey)	liver	2	12.18 ± 0.53
	pancreas	3	10.26 ± 12.36

Results

Motion estimation. We report performance metrics for motion estimation for our different methods in Table 6.8. We compare all deep learning approaches w.r.t. inference time in Table 6.9. We also report the case where the initial position is considered for all subsequent positions and refer to this approach as "No Tracking". Our results show that all our deep learning approaches outperform the conventional tracking approach and lead to significantly ($p < 0.05$) improved performance compared to "No Tracking". Across the datasets, DenseConvGRU-all significantly ($p < 0.05$) outperforms all other methods. We report performance metrics for motion estimation for our variants of DenseConvGRU in Table 6.10. We report the performance of our methods w.r.t. translation distances in Figure 6.11 and Figure 6.12. We also report the TE w.r.t. the sequence length, i.e., tracking duration in Figure 6.13. Our results show that the TE increases substantially for increased translation distance using the NI-estimation approach, see Figure 6.11. Moreover, our results demonstrate that the TE increases substantially over time for the I-estimation approach, see Figure 6.12. Our approach DenseConvGRU-all with multi-scale spatio-temporal recurrence shows high performance across the entire range of translation distances, and the TE remains low even with an increased tracking duration. Figure 6.14 shows the estimations for example trajectories. We also report results for a trajectory of seven minutes using our best performing approach DenseConvGRU-all, see Figure 6.15. The results show that motion estimation can also be performed for such long periods of time with consistent performance over time, although we train the network with sequences with a length of smaller than 26 seconds. We also report performance metrics for different sequence lengths b considered for truncated BPTT during training in Figure 6.16. Our results show that for $b = 50$, tracking performance remains similar over time.
Furthermore, we retrain DenseConvGRU-all for different ROI size and report performance metrics for motion estimation in Table 6.11. Our results show that motion estimation can also be performed for smaller ROIs with a significantly

($p < 0.05$) lower TE than performing no tracking. However, the motion estimation performance decreases for smaller ROIs. Moreover, results for experiments with synthetically deformed volumes during evaluation are given in Figure 6.17. We study deformations of different degrees by varying the number of grid points that have a displacement different from zero and by using different standard deviations for our deformation sampling. Our results show that the TE increases only for substantial deformations.

Table 6.8: Motion estimation. Combined refers to the combination of all three datasets. (Compare [42] (©2023 IEEE))

		TE (mm)			
Set	Method	Mean	Std	95%	aCC
Test (chicken)	No Tracking	3.32	2.84	8.58	-
	NCC-NI	0.98	1.25	4.25	0.69
	2P-TC-3DCNN-NI	0.76	1.17	2.44	0.61
	NCC-I	2.63	2.03	6.63	0.47
	2P-TC-3DCNN-I	1.79	1.53	4.41	0.76
	nP-ST-4DCNN-I	1.15	1.24	3.99	0.59
	DenseConvGRU-all	**0.34**	**0.24**	**0.79**	**0.81**
Test (liver)	No Tracking	3.54	2.66	8.61	-
	NCC-NI	0.53	0.34	1.04	0.91
	2P-TC-3DCNN-NI	0.92	1.29	3.33	0.82
	NCC-I	2.27	1.48	5.24	0.63
	2P-TC-3DCNN-I	3.39	2.90	9.49	0.71
	nP-ST-4DCNN-I	1.68	1.49	4.54	0.71
	DenseConvGRU-all	**0.39**	**0.22**	**0.79**	**0.92**
Test (turkey)	No Tracking	4.38	3.61	11.1	-
	NCC-NI	2.22	2.92	9.21	0.64
	2P-TC-3DCNN-NI	0.92	0.82	2.50	0.76
	NCC-I	3.35	2.58	8.85	0.78
	2P-TC-3DCNN-I	1.79	1.28	4.40	0.80
	nP-ST-4DCNN-I	1.48	1.01	3.32	0.72
	DenseConvGRU-all	**0.36**	**0.24**	**0.83**	**0.94**
Combined	No Tracking	3.58	2.96	9.42	-
	NCC-NI	1.12	1.61	5.04	0.70
	2P-TC-3DCNN-NI	0.83	1.16	2.69	0.70
	NCC-I	2.82	2.13	7.20	0.56
	2P-TC-3DCNN-I	2.26	2.13	5.86	0.73
	nP-ST-4DCNN-I	1.36	1.31	4.05	0.64
	DenseConvGRU-all	**0.35**	**0.24**	**0.80**	**0.87**

Table 6.9: Inference time and number of parameters (#numParam.) of the different deep learning methods tested on an nvidia TITAN RTX-24GB. (Compare [42] (©2023 IEEE))

Method	inference time [ms]	#numParam.
nP-ST-4DCNN-I	37.06 ± 0.42	479 325
DenseConvGRU-all	4.99 ± 0.30	547 090
DenseConvGRU-front	3.47 ± 0.26	**207 022**
DenseConvGRU-middle	3.11 ± 0.15	293 407
DenseConvGRU-end	$\mathbf{2.36 \pm 0.07}$	397 117
2P-TC-3DCNN(NI/I)	3.25 ± 0.12	208 439

Table 6.10: Motion estimation. Results for DenseConvGRU with different ConvGRU module placements. Results are reported for the combination of all test datasets. (Compare [42] (©2023 IEEE))

	TE (mm)			
Method	Mean	Std	95%	aCC
No Tracking	3.58	2.96	9.42	-
DenseConvGRU-all	**0.35**	**0.24**	**0.80**	**0.87**
DenseConvGRU-front	0.50	0.41	1.34	0.84
DenseConvGRU-middle	0.46	0.38	1.17	0.79
DenseConvGRU-end	0.51	0.37	1.25	0.82

Table 6.11: Motion estimation. Results for DenseConvGRU using smaller ROIs. Results are reported for the combination of all test datasets. (Compare [42] (©2023 IEEE))

	TE [mm]			
ROI (mm^3)	Mean	Std	95%	aCC
$\mathbf{19 \times 19 \times 19}$	**0.35**	**0.24**	**0.80**	**0.87**
$10 \times 10 \times 10$	0.52	0.44	1.45	0.64
$5 \times 5 \times 5$	0.85	1.07	2.43	0.63
No Tracking	3.58	2.96	9.42	-

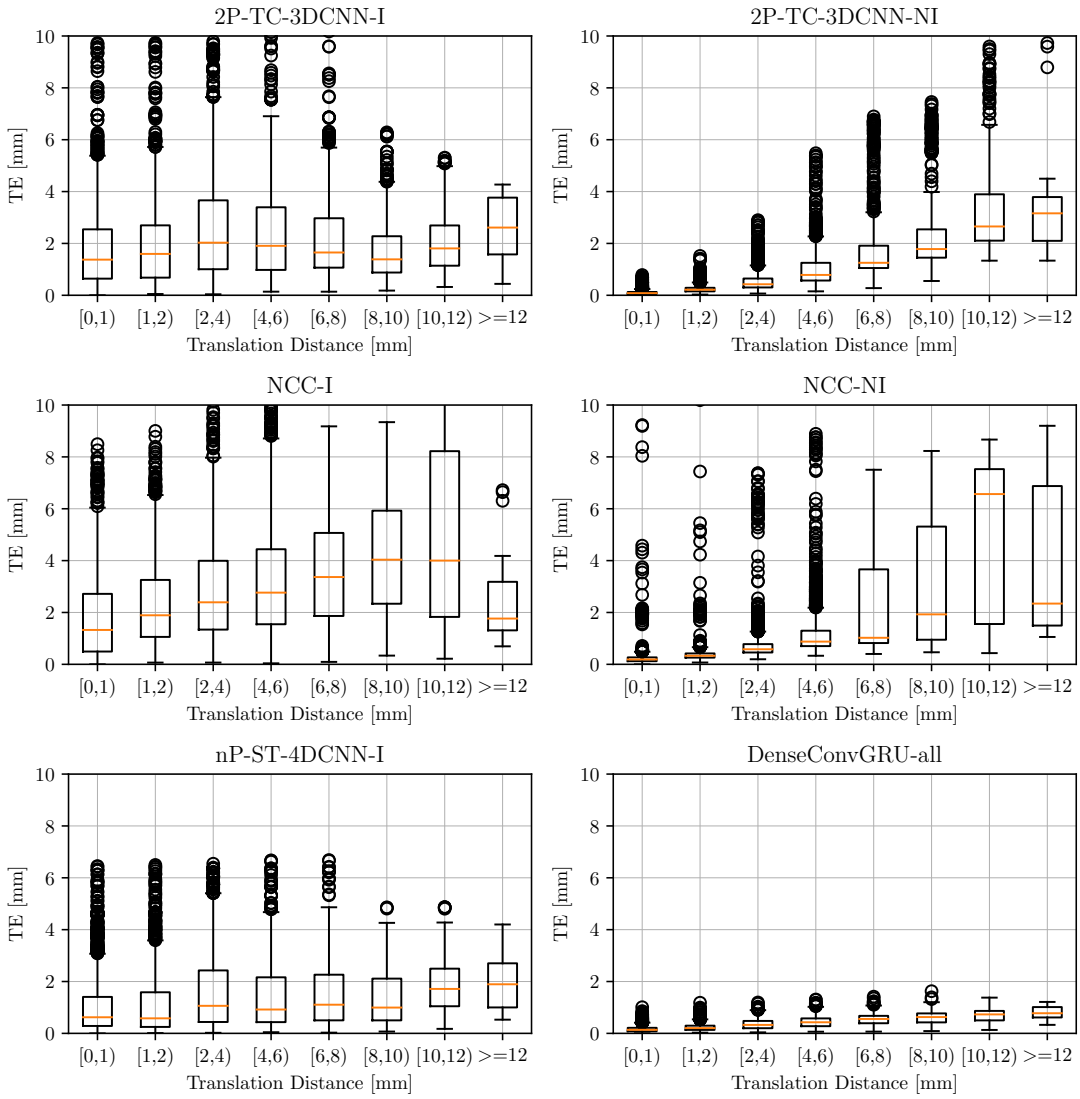

Figure 6.11: Motion estimation. Shown is the TE w.r.t. different translation distances. Results are reported for the combination of all test datasets. Figure adapted from [42] (©2023 IEEE).

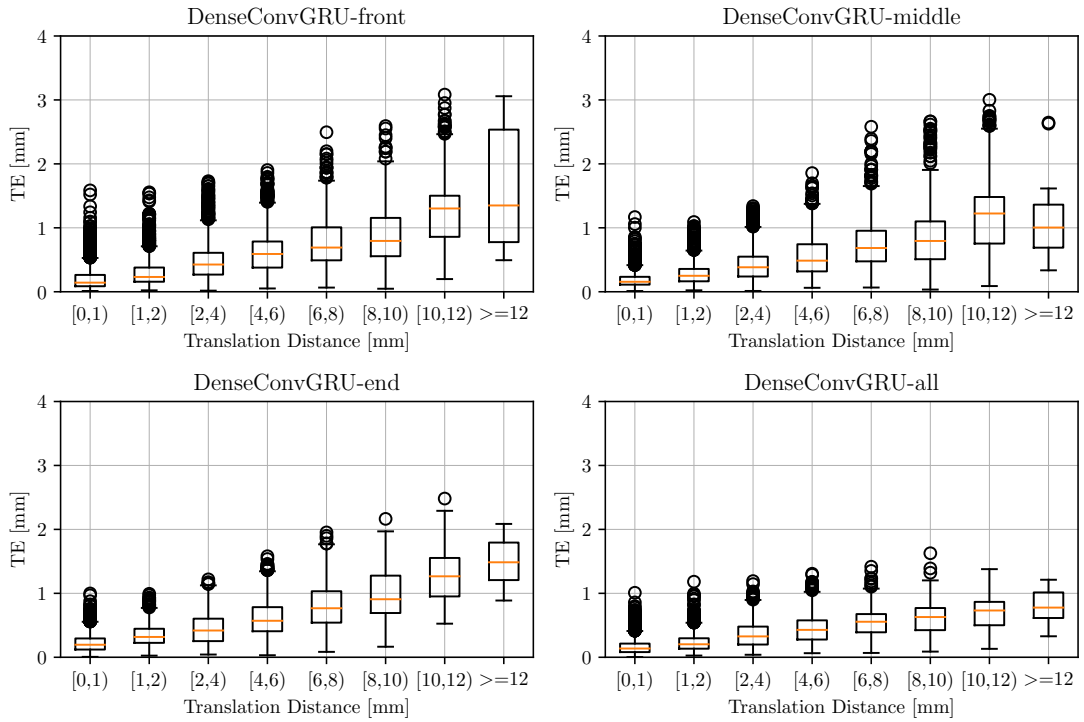

Figure 6.12: Motion estimation. Evaluation of different ConvGRU module placements for our architecture DenseConvGRU. Shown is TE w.r.t. different translation distances. Results are reported for the combination of all test datasets. Figure adapted from [42] (©2023 IEEE).

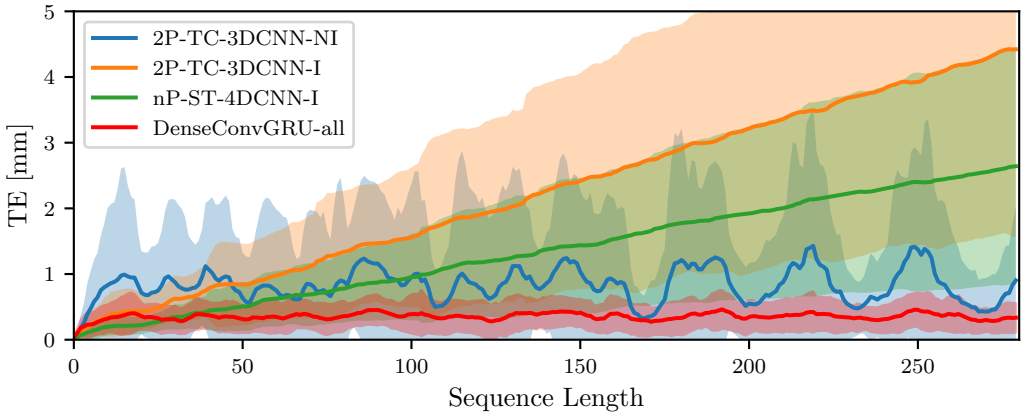

Figure 6.13: Motion estimation. Performance over sequence length, i.e., tracking time, shown for the combination of all test datasets. The solid lines refer to the mean value over all trajectories. The shaded backgrounds refer to the standard deviation. The sequence length is given in number of volumes. Figure adapted from [42] (©2023 IEEE).

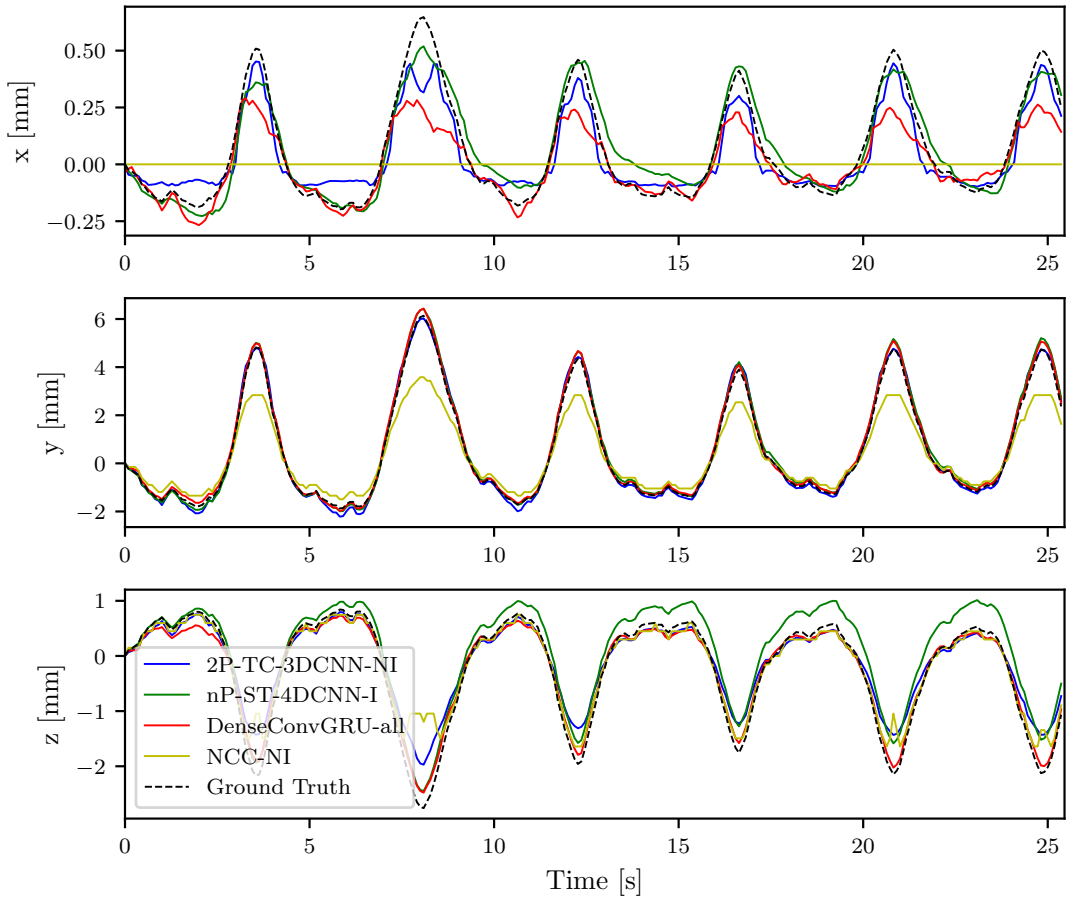

Figure 6.14: Motion estimation. Results for one example trajectory from the test set. We show the actual and estimated translation for the three spatial dimensions (x,y,z) over time. Note that the range of motion is substantially different for the different axes. Figure adapted from [42] (©2023 IEEE).

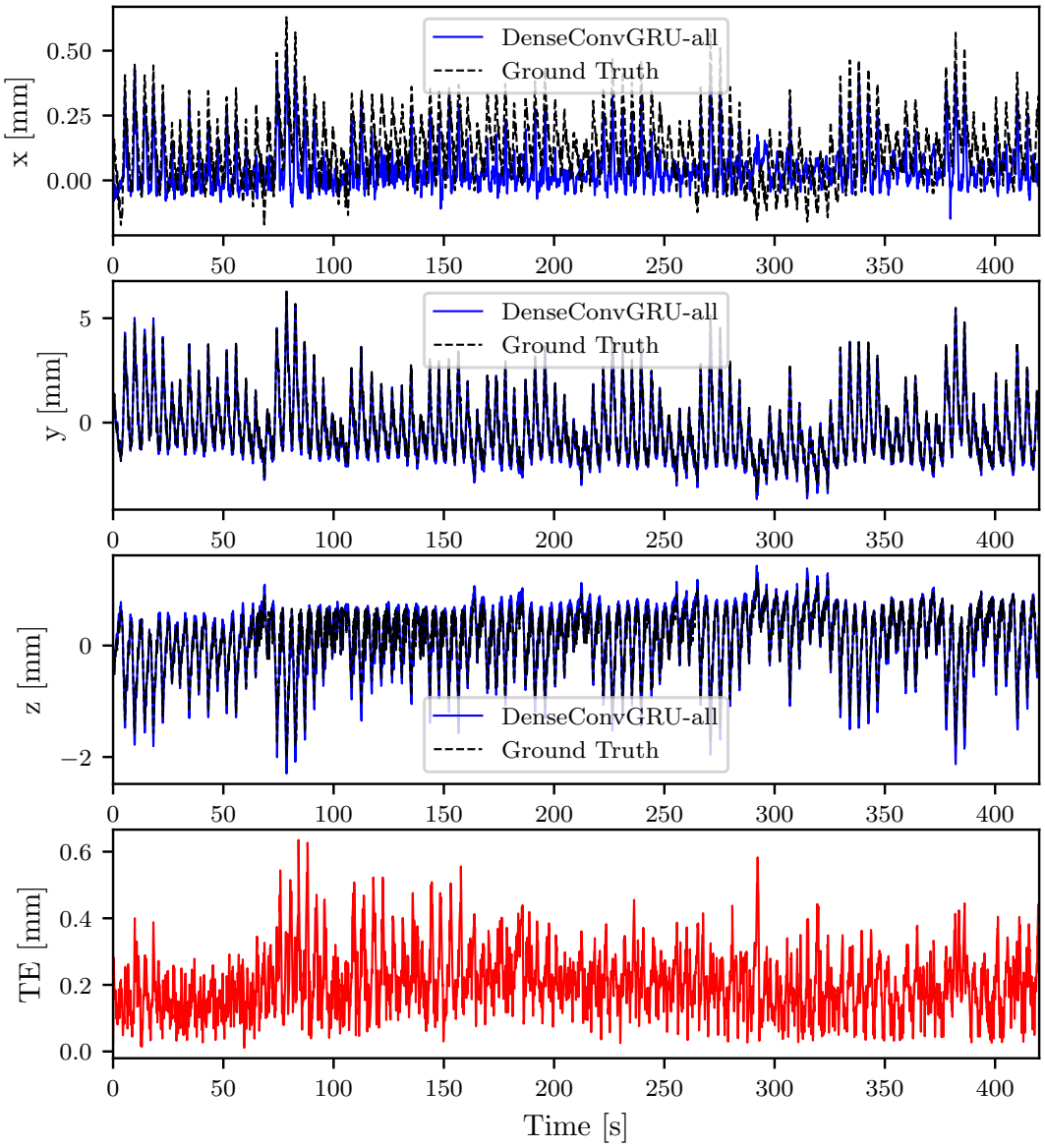

Figure 6.15: Motion estimation for a sequence with a duration of seven minutes using DenseConvGRU-all. Shown is the actual and estimated translation for the three spatial dimensions and the resulting TE over time (bottom). Note that the range of motion is substantially different for the different axes. Figure adapted from [42] (©2023 IEEE).

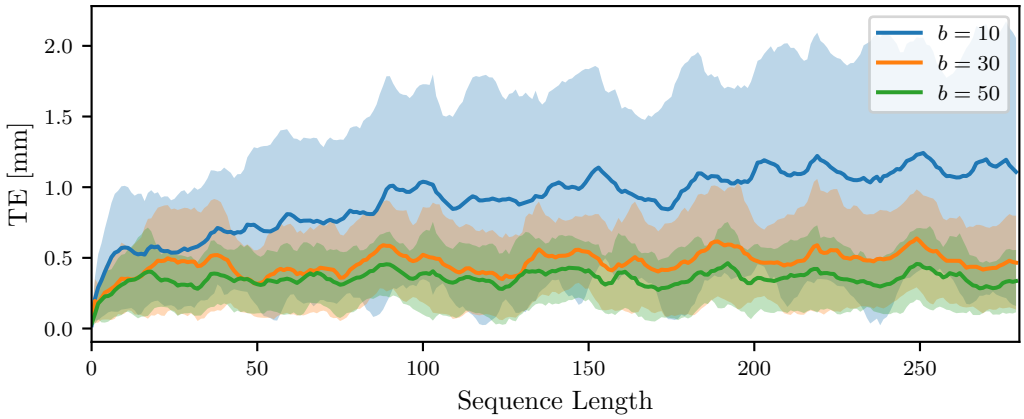

Figure 6.16: Motion estimation. Performance over sequence length using different lengths of b for backpropagation during training. Results are shown for our network DenseConvGRU-all and for the combination of all test datasets. The solid lines refer to the mean value over all trajectories. The shaded backgrounds refer to the standard deviation. The sequence length is given in number of volumes. Figure adapted from [42] (©2023 IEEE).

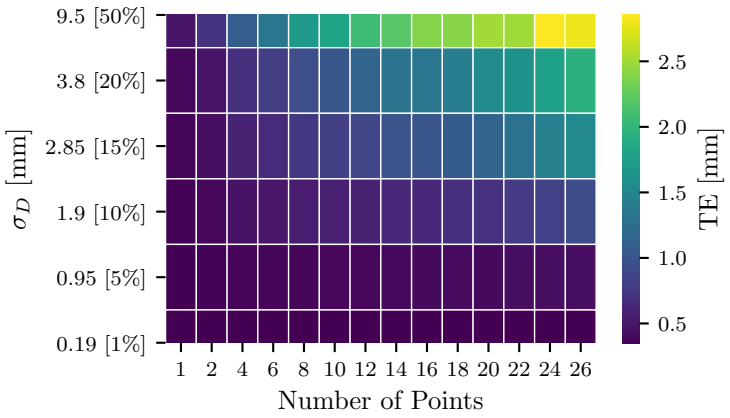

Figure 6.17: Motion estimation performance using synthetically deformed volumes during evaluation. We report the TE for a varying number of grid points used for displacement. The vectors for grid point displacements are sampled from a Gaussian distribution with a standard deviation σ_D. The numbers in brackets describe σ_D relative to the ROI size. Results are shown for our approach DenseConvGRU-all. Figure adapted from [42] (©2023 IEEE).

Motion forecasting. Moreover, we evaluate our approach DenseConvGRU-all to compensate for different latencies up to approximately 900 ms. This leads to a prediction horizon up to $p_h = 10$ and we report corresponding performance metrics in Table 6.12. For reference we also report the resulting $\mathrm{TE}(\tilde{T}_{0,n}, T_{0,n+p_h})$ when there is a latency between the estimation $\tilde{T}_{0,n}$ and the actual position $T_{0,n+p_h}$, i.e., when no motion forecasting is performed. Our results show that with motion forecasting, the TE is significantly ($p < 0.05$) lower for all latencies. However, the TE increases with an increased prediction horizon.

Table 6.12: Motion forecasting. Results are shown for the combination of all test datasets using our approach DenseConvGRU-all. p_h is given in number of time steps, while each time step corresponds to approximately 90 ms. (Compare [42] (©2023 IEEE))

p_h	$\mathrm{TE}(\tilde{T}_{0,n}, T_{0,n+p_h})$ [mm]			$\mathrm{TE}(\tilde{T}_{0,n+p_h}, T_{0,n+p_h})$ [mm]		
	mean	std	95%	mean	std	95%
0	0.35	0.24	0.80	-	-	-
1	0.62	0.43	1.47	0.45	0.32	1.06
2	1.06	0.76	2.53	0.52	0.41	1.30
3	1.51	1.11	3.63	0.78	0.66	2.08
4	1.95	1.43	4.66	1.00	0.87	2.83
5	2.37	1.74	5.67	1.22	1.01	3.37
10	4.03	2.89	9.44	2.31	2.26	7.30

Discussion

In this experiment, we address end-to-end motion analysis using 4D US data in the context of motion compensation during radiotherapy. Our findings demonstrate that this task can be performed with our 4D spatio-temporal deep learning approach. It stands out that all our deep learning approaches outperform the conventional approach that is used for comparison, see Table 6.8. This highlights that all our different concepts for spatio-temporal data processing can be integrated effectively into our architecture concept. Our results show that, in particular, our approach DenseConvGRU-all performs well. Our results show DenseConvGRU-all shows high performance for the different tissue types of our test datasets, although only chicken tissue is considered for our training data. These results highlight that robust spatio-temporal features can be learned end-to-end from data with our approach such that motion estimation can be performed for different target appearances and motion patterns, even in the absence of distinct landmarks. These promising results indicate that our approach could effectively be used for different targets during clinical practice. This is a valuable contribution considering that tracking during radiotherapy needs to be performed for different targets such as pancreas, liver, or prostate, and typically this requires manual adaption and landmark selection.
Moreover, tissue deformations that might occur during practice change the appearance of a target region over time, which makes motion analysis even more difficult. Our results in Figure 6.17 demonstrate that the TE increases only for

substantial deformations. These results indicate that deformations are not a limiting factor for our approach. So far, we only considered deformation during evaluation. Integrating deformations also during training is an interesting direction for future work to further increase the robustness and performance in this scenario. However, synthetically deforming volumes is time-consuming, with runtimes in the range of minutes, which makes online data augmentation with such an approach difficult. Still, future work could generate datasets with deformed volumes offline before training. In addition to that, our results show that motion estimation can also be performed for even smaller ROIs, see Table 6.11. Considering that our approach is particularly fast, motion could be estimated for multiple regions in parallel. Such an approach could be used to improve the consistency and also to estimate target deformations based on the relative motion between the regions. We also address system latencies and consider the task of motion forecasting directly from the image sequences. Typically, motion forecasting for latency compensation is performed with predictive methods that use the motion history [138, 226, 276, 451]. Our results show that motion forecasting can be performed directly from the image sequence. This further demonstrates that effective spatio-temporal features can be learned from the sequences of US data such that target motion can even be extrapolated into the future. Notably, this can be performed with the same approach used for motion estimation, and hence our results show that motion estimation and forecasting can be combined into a single end-to-end approach. These results are promising to simplify otherwise complex data processing pipelines into a single end-to-end approach. Overall, considering our research questions, our findings demonstrate that long-term motion analysis can be performed end-to-end and in real-time from long-term sequences of volumetric US with our approach DenseConvGRU-all. Also, our results show that this can be performed for different targets and motion patterns and that motion forecasting for system latency compensation can directly be performed from sequences of US image data.

Considering our second research question, our results show that DenseConvGRU-all performs best and that there are notable differences between our approaches. As a first approach, we adapt our spatio-temporal deep learning concept to perform motion estimation using image pairs or short-term sequences. Our results show that nP-ST-4DCNN-I outperforms 2P-TC-3DCNN-I. Similar to the results from the previous Section 6.1.2. These results further indicate that spatio-temporal data is beneficial for motion analysis and outperforms the simple approach of using only two images. However, for the I-estimation approach, the TE increases substantially over time due to the accumulation of estimation errors, see Figure 6.13. Thus, our results suggest that a NI-estimation approach is favorable for long-term tracking. However, this approach shows poor performance for larger translation distances, see Figure 6.11. Hence, both approaches show their limitations when it comes to motion analysis, similar to previous findings on this topic using conventional methods [194]. Our results show that we can overcome these limitations with our approach DenseConvGRU. We develop this approach based on our architecture concept by integrating ConvGRU modules at different scales.

We suppose that motion analysis in the presence of different motion patterns benefits from spatio-temporal features at different scales. Our results confirm this hypothesis and show that our approach of multi-scale spatio-temporal recurrence is beneficial, outperforming all other previous variants, i.e., considering the module at the front, middle, or end. Another advantage of our DenseConvGRU approach is that temporal relationships are considered via ConvGRU modules, which only requires to process the current input volume to perform motion analysis. Also, we perform motion analysis end-to-end directly from a fixed ROI without performing multiple template matching procedures. Combining these factors allows us to perform motion analysis with an inference time of 4.99 ± 0.30 ms (≈ 200 Hz) for our best performing approach DenseConvGRU-all. This is a valuable contribution considering that typically using multiple volumes increases the inference time substantially, and previous studies demonstrated inference time in the range of 300 ms up to 11 s [21, 22, 212, 341, 377, 471]. Faster run-times have also been achieved with 4D US data by carefully selecting parameters for the US data, and registration method [21, 195, 341], or by using lower-dimensional surrogate data instead of the actual 4D spatio-temporal data [307, 371].
Another challenge is training with long-term 4D data, which we address with our training approach, which combines curriculum learning [31] and truncated BPTT [446]. Although we perform backpropagation only for a short sub-sequence, we demonstrate motion estimation can be performed for much longer sequences up to seven minutes, see Figure 6.15. Thus, our findings demonstrate that also effective training can be performed to perform motion analysis with long-term sequences of volumetric US data.

Overall, considering our first research question, our findings highlight that end-to-end motion analysis can be performed with our spatio-temporal deep learning approach within a few milliseconds using entire long-term sequences of US volume. Our results show that motion analysis can be performed marker-less, for different target tissues and appearances as well as motion patterns. We also demonstrate that both motion estimation and forecasting can be performed with a single approach end-to-end using sequences of volumetric US images. Considering our second research, learning from entire image sequences and using our hybrid approach of a CNN and DenseConvGRU modules at different scales turns out to be effective to perform long-term motion analysis.

6.1.4 Summary

In the previous sections, we investigated our spatio-temporal deep learning methods in the context of position estimation and motion compensation using 4D spatio-temporal OCT and US data. We designed and evaluated several of our proposed 4D deep learning methods and compared them to previous approaches that perform estimations from single images or image pairs. We considered marker position estimation using 4D OCT data and marker-less tissue motion analysis using 4D OCT and 4D US data. For our experiments, we used experimental setups that allowed for automatic data acquisition and annotation.

Considering our first research question, we studied whether motion analysis can be performed end-to-end and in real-time directly from sequences of volumetric image data using our spatio-temporal deep learning approaches. This requires learning complex end-to-end relationships directly from sequences of multi-dimensional medical image data. In particular, marker-less motion analysis requires learning robust features from the data to generalize to different target appearances. Our findings demonstrated that our spatio-temporal deep learning approaches could generalize to unseen tissue types, appearances, and motion patterns. Across several experiments, our results showed that this could be performed with high performance and run-times in the range of milliseconds. We highlighted that also end-to-end multi-task learning could be performed effectively, which allows combining motion estimation and forecasting directly into a single model. This is a valuable contribution as our approach can directly address system latencies, which are typically present during motion compensation. Moreover, by combing this into a single approach, it reduces the efforts that result from addressing these tasks separately. Thereby, the data processing pipeline can be simplified. Regarding the flexibility of spatio-temporal deep learning, we found that our spatio-temporal CNN architecture concept can be used without major adaptions for OCT and US data and for marker position estimation as well as tissue motion analysis.

Considering our second research question, we demonstrated that using 4D spatio-temporal actually improves motion estimation performance compared to using only single or pair-wise volumetric images for analysis. Similar, we found that additional temporal information at the output of the model can also be used for regularization by enforcing the model to learn an extended motion pattern. However, the actual approach that is used for spatio-temporal data processing has a major impact on the performance, highlighting the importance of our comparison. We found that learning joint spatio-temporal features performs best and clearly outperforms, e.g., a naive approach that uses the channel dimension as a temporal dimension. This demonstrates the advantage of joint spatio-temporal feature learning in an end-to-end fashion. While inference times in the range of milliseconds can be achieved, allowing for real-time applications, training is substantially time consuming and can range over several days. Considering this, our results showed that transfer learning from spatial to spatio-temporal data is a promising approach to reduce training times. Also, we proposed and evaluated a training scheme that copes with the computational requirements of long-term 4D data processing. We demonstrated that with our architecture and training concept, precise motion analysis from entire long-term 4D sequences could also be performed for sequences of several hundreds of volumes ranging over several minutes. Overall, our results show that learning from multi-dimensional medical image sequences can be performed for motion analysis with our spatio-temporal deep learning approach. Our findings highlight that thereby performance can be improved and data processing can be simplified. Comparing our different approaches, the approach of joint spatio-temporal feature learning shows promising results.

6.2 Dynamic Elastography for Tissue Characterization

In this section, we present the results of our spatio-temporal deep learning methods considering our second application scenario, i.e., dynamic elastography for tissue characterization. Here, processing of multi-dimensional medical image sequences is required to infer mechanical properties of tissue from imaged wave propagation as outlined in Section 5.2. We study end-to-end shear wave elastography using both sequences of US and OCT data. In the context of our first research question, we analyze whether mechanical properties of tissue can directly be estimated from sequences of multi-dimensional medical image data in an end-to-end fashion with our spatio-temporal deep learning methods. Using sequences of US data, we consider the task of shear wave velocity estimation, followed by localized elasticity estimation in an end-to-end fashion. Afterwards, we present our results considering sequences of OCT data and study end-to-end estimation of mechanical properties from 3D and 4D OCT data. Considering our second research question, we evaluate different input strategies and architecture concepts for processing the spatio-temporal data, compare performance to conventional approaches, and contribute to the question of how spatio-temporal data processing should be performed. For our experiments, we consider data from tissue-mimicking gelatin phantoms with varying stiffness.

6.2.1 3D US-based Shear Wave Velocity Estimation

In this study, we address shear wave velocity estimation for tissue characterization from sequences of US images in an end-to-end fashion. Parts of this section have been published in our study presented in [184]. The study [184] was a collaborative study, the development of the experimental setup, data acquisition, and evaluation with conventional image processing methods were performed by S. Grube. The aspects regarding spatio-temporal deep learning were developed and evaluated by the author of this thesis and are in the focus of this work.

Considering our first research question, we study whether shear wave velocity can be determined directly with spatio-temporal CNNs from real US-SWEI data in an end-to-end fashion and whether this approach has the potential to overcome limitations of conventional approaches. In particular, we study the impact of lateral imaging widths on shear wave velocity estimation. This is relevant, e.g., during laparoscopy where probes with lateral imaging width in the range of a few millimeters are used [215].

Considering our second research question, we present and evaluate different spatio-temporal deep learning approaches to estimate shear wave velocity directly from an image sequence. We evaluate how the learning task can be addressed and how the spatio-temporal relationships can be learned. We evaluate the performance of a selected feature representation that is typically used [225, 343, 464, 481] in comparison to learning from the entire image sequence.

Definition of the Learning Task

Our task is to estimate shear wave velocity $v \in \mathbb{R}_+$ directly from sequences of 2D US images $x_t = [x^{[0]}, x^{[1]}, \ldots, x^{[n]}]$ with $x^{[i]} \in \mathbb{R}^{n_h \times n_w \times n_c}$ that capture displacement information over time resulting from shear wave propagation. As a first approach, we consider a lower-dimensional feature representation that is typically used by conventional approaches, and we try to learn a function $f_{v_1} \colon \mathbb{R}^{n_w \times n_t \times n_c} \to \mathbb{R}$. The feature representation is generated by averaging along the axial depth axis of the US images. This assumes the wave propagation direction along the lateral axis. While this reduces the dimensionality, it might also remove valuable information. As a second approach, we hence consider the entire available image sequence and try to learn a function $f_{v_2} \colon \mathbb{R}^{n_h \times n_w \times n_t \times n_c} \to \mathbb{R}$. Our approaches are visualized in Figure 6.18.

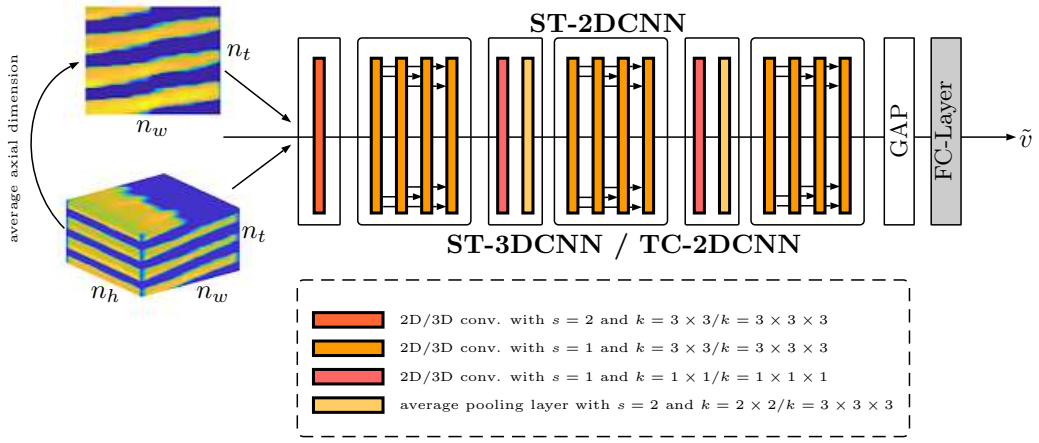

Figure 6.18: Our deep learning approaches for shear wave velocity estimation from sequences of 2D US images that capture displacement information. Our ST-2CNN approach estimates shear wave velocity using a lower dimensional space-time map representation. Our ST-3DCNN estimates shear wave velocity from the original image sequence and performs joint spatio-temporal feature learning throughout the architecture. Our TC-2DCNN approach also considers the original image sequence, but here all the temporal dimension is processed in the first layer of the network by using the channel dimension of the input as the temporal dimension. The last layer of our architecture concept is a regression layer that outputs shear wave velocity $\tilde{v}$. Note that our actual input images are gray-scale and colored inputs are only shown for improved visualization.

Experimental Setup and Data Sets

For evaluation of our methods, we use a dataset consisting of 2D US images over time that capture shear wave propagation in tissue-mimicking gelatin phantoms. We use data from phantoms with varying stiffness that results from different gelatin to water ratios. The full dataset consists of gelatin phantoms with gelatin concentrations $C \in [7.5, 10, 12.5, 15]$ given in percentage points (p.p.). For each concentration, we consider data from one phantom and use data from two different sides of each phantom. The dataset is acquired with an experimental setup

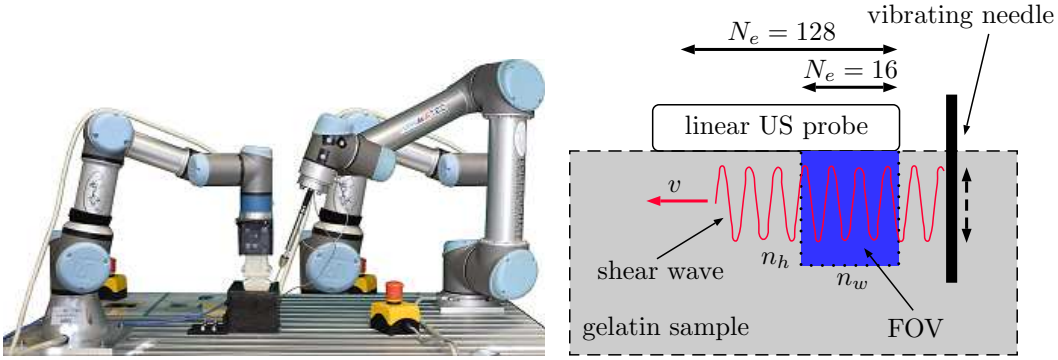

Figure 6.19: (Left) Experimental setup for data acquisition. (Right) Schematic drawing of the data acquisition approach. Shear wave are generated with a vibrating needle and imaging is performed with a linear US probe. By decreasing the number of piezoelectric elements N_e for image reconstruction, a varying lateral width n_w is simulated. Figure adapted from [184].

shown in Figure 6.19. The setup includes a 256 channel US system (Griffin, Cephasonics), a linear US probe (Ultrasonix L14-5\38), and a needle attached to a piezo-actuator to generate shear waves. A first robot (UR5, Universal Robots) is used to position the needle, and a second robot is used to position the US probe (UR3, Universal Robots). For each side of a phantom, the needle is inserted at 24 different positions, and at each position, ten image sequence acquisitions are performed. Measurements at different positions and sides of a phantom are performed to increase the variability in scatterer distribution. For data acquisition, the needle is inserted into a phantom, and a sinusoidal vibration of the needle with 300 Hz is performed and a sequence of $n_t = 70$ US images at a depth of 10 mm to 30 mm is acquired using plane wave imaging with an image acquisition rate of 6000 Hz. The full dataset consists of 480 US image sequences for each of the four concentrations. To obtain different datasets with image sequences that corresponds to different lateral imaging width n_w, US images are reconstructed with only a subset of the raw data from all $N_e = 128$ piezoelectric elements, see Figure 6.19. The different datasets for the different lateral imaging widths consist of image sequences that correspond to $N_e \in [8, 16, 32, 64, 128]$. Loupas algorithm [289] is used to visualize scatterer displacement. For ground truth annotation and for comparison, we consider shear wave velocity estimated with a conventional approach. We use the results for the for the largest lateral imaging width with $N_e = 128$ piezoelectric elements for ground truth annotation. Conventional shear wave velocity estimation is performed as follows. First, a space-time map is generated by averaging along the axial depth axis of US images [239, 333]. Afterwards, shear wave velocity is estimated based on the detected maximum of the spatial and temporal frequency using a fast Fourier transformation [239, 333]. Results that we consider as ground truth are summarized in Table 6.13. Summarized, our dataset consists of US image sequences x_t that capture shear wave propagation over time, and the corresponding label is given by the shear wave velocity v determined for the largest lateral imaging width with $N_e = 128$ piezoelectric elements.

Table 6.13: Ground truth shear wave velocities v for the different gelatin concentrations C determined using $N_e = 128$ piezoelectric elements. We consider the mean value of all estimated shear wave velocities across all measurements of one concentration as ground truth value. (Compare [184])

C [p.p.]	7.5	10	12.5	15
v [$\mathrm{m\,s^{-1}}$]	3.63	4.56	5.97	7.09

Methods

To estimate shear wave velocity from sequences of US images, we implement and compare three different spatio-temporal deep learning approaches using our CNN architecture concept of this work. We explain specific details of our approaches in the following paragraphs. As a backbone, we consider the concept of DenseNet [210]. Our different approaches are visualized in Figure 6.18.

ST-2DCNN. As a first approach, we consider the lower dimensional representation and try to learn from such data. To address this task, we use our CNN architecture in a 2D fashion, i.e., by using 2D convolutional and pooling operations. Using this input type, the 2D spatio-temporal CNN is trained to estimate shear wave velocity from a 1D+t image representation, i.e., a single 2D image. This approach receives the input $x'_t \in \mathbb{R}^{n_w \times n_t \times 1}$. While this reduces the dimensionality of the problem, it assumes and only considers wave propagation along the lateral axis. This motivates learning from the original image data.

TC-2DCNN. For our second and third approaches, we consider processing the entire image sequence and compare two different strategies for 3D spatio-temporal data processing. As a simple baseline approach, we consider the concept of a time-channel, where the input's channel dimension is considered as a temporal dimension. Note that with this approach, we can use the same architecture as for the processing of the lower dimensional input, i.e., we can still use a 2DCNN, but we can process the entire image sequence. This approach receives the input $x_t \in \mathbb{R}^{n_h \times n_w \times n_t}$. However, all the temporal information is processed in the first layer of the CNN, which might limit the performance.

ST-3DCNN. As a third approach, we consider joint spatio-temporal feature learning throughout the network and use 3D spatio-temporal convolutions and pooling operations for our CNN architecture concept. Thereby, we can learn abstract end-to-end relationships to estimate shear wave velocity from the original image sequence. This network receives the input $x_t \in \mathbb{R}^{n_h \times n_w \times n_t \times 1}$.

Architecture Details. For our CNNs, we use one initial convolutional layer with 18 feature maps and a stride of two for all input dimensions. For our DenseNet backbone, we use three DenseNet blocks which are connected with transition layers. For each DenseNet block, we use four convolutional layers with a growth rate of 12. All methods are implemented in PyTorch.

Training. We train our architecture with a MSE loss function for 100 epochs using the Adam optimizer [242] with a learning rate of lr = 0.001 and a batch size of $m_b = 10$. During training, we evaluate the performance of a network every epoch on the validation set and use the best network for our final evaluation on the test dataset. After 75 epochs, we reduce the learning rate by a factor of two every 10 epochs. To improve convergence, we normalize the pixel intensities of each input to have a zero mean and standard deviation of one. We resize the spatial dimensions of the input images to 32×32 pixels. In this way, we can use the same architecture for the different number of piezoelectric elements and the resulting varying image sizes. Moreover, the computational efforts are reduced compared to the full resolution approaches. We train each approach for each dataset corresponding to a different number of piezoelectric elements.

Performance Measure - Evaluation and Metrics

We perform a two-fold cross-validation approach to train and evaluate our methods. In each fold, we use one side of our phantoms for testing. For training and validation in each fold, we use 70% and 30% of the remaining data, respectively. We consider the MAE between the estimated and ground truth shear wave velocity as evaluation metric. We test for significant differences in the median of the MAE of our methods using the Wilcoxon signed-rank test with a significance level of $\alpha = 5\%$.

Results

We report our results in Table 6.14. Across all experiments, our deep learning approaches are significantly ($p < 0.05$) better than the ToF approach, and learning from the original image sequence performs best. Comparing our different deep learning approaches shows that ST-3DCNN performs significantly ($p < 0.05$) better than ST-2DCNN and TC-2DCNN. The approach TC-2DCNN performs worst. We report the estimated shear wave velocities of the ToF approach in Figure for comparison 6.20, and we compare the estimated shear wave velocities of our deep learning approaches in Figure 6.21. It stands out that for $N_e < 32$, the ToF approach almost fails completely, and shear wave velocity is notably overestimated for all gelatin concentrations. In contrast to that, the performance of our deep learning approaches only decreases slightly. Our results show that estimating faster shear wave velocities results in a higher MAE for all our methods, see Figure 6.22. Inference times and number of parameters are reported in Table 6.15.

Table 6.14: MAE of the estimated shear wave velocities in $\mathrm{m\,s^{-1}}$ for the different number of piezoelectric elements N_e. The corresponding lateral imaging width is given by n_w. For ToF* we remove outlier values and only consider values in the range between 0 $\mathrm{m\,s^{-1}}$ and 15 $\mathrm{m\,s^{-1}}$.

	N_e (n_w)			
	64 (18.9 [mm])	32 (9.3 [mm])	16 (4.5 [mm])	8 (2.1 [mm])
ToF	2.01 ± 8.63	4.63 ± 2.37	28.08 ± 24.98	78.13 ± 63.27
ToF*	1.02 ± 0.69	4.27 ± 1.71	8.92 ± 1.58	9.48 ± 1.05
ST-2DCNN	0.29 ± 0.26	0.47 ± 0.39	0.67 ± 0.52	0.80 ± 0.62
TC-2DCNN	0.29 ± 0.25	0.49 ± 0.42	0.83 ± 0.60	1.02 ± 0.72
ST-3DCNN	$\mathbf{0.22 \pm 0.26}$	$\mathbf{0.38 \pm 0.38}$	$\mathbf{0.58 \pm 0.47}$	$\mathbf{0.74 \pm 0.57}$

Table 6.15: Number of parameters (#numParam.) for our deep learning approaches and inference time evaluated on an nvidia TITAN RTX.

	inference time [ms]	#numParam.
ST-2DCNN	2.49 ± 0.05	**119 045**
TC-2DCNN	$\mathbf{2.45 \pm 0.05}$	129 413
ST-3DCNN	2.83 ± 0.09	243 785

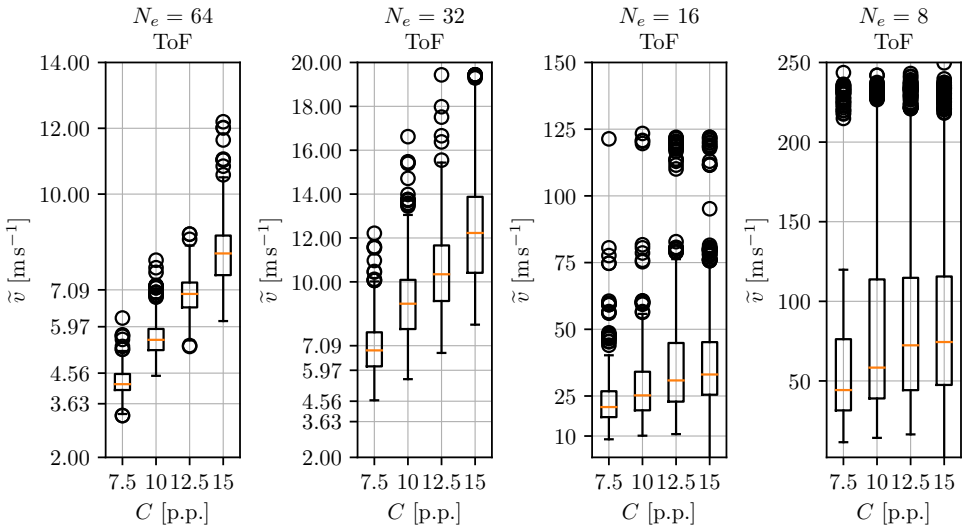

Figure 6.20: Boxplots of the estimated shear wave velocities based on the conventional approach. Note that we use different y-scales for the different number of piezoelectric elements N_e. From left to right, we show values for a decreasing number of piezoelectric elements, i.e., a reduced lateral imaging width. Figure adapted from [184].

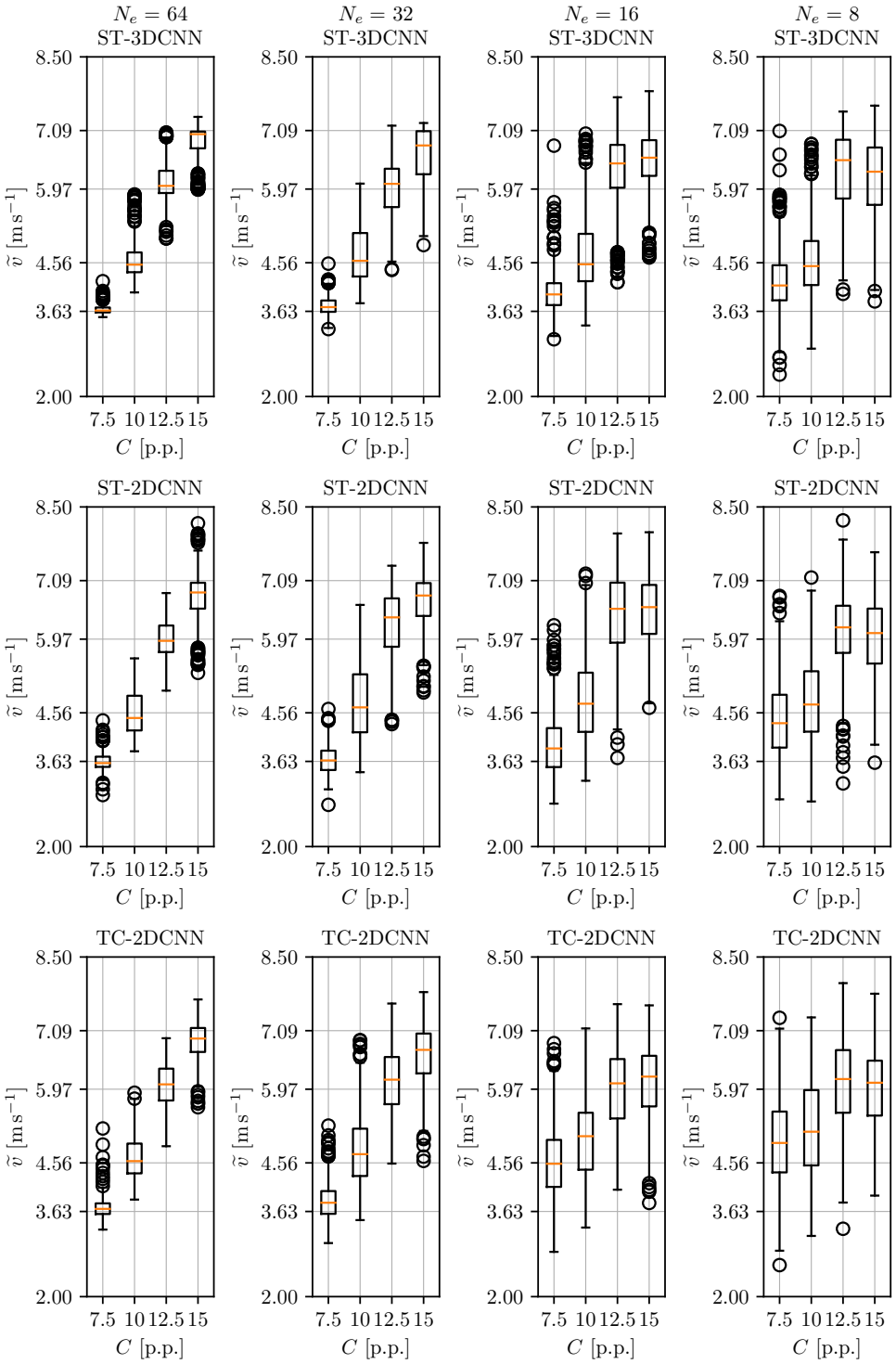

Figure 6.21: Boxplots of the estimated shear wave velocities using our deep learning approaches. From left to right, we show values for a decreasing number of piezoelectric elements that results in a reduced lateral imaging width.

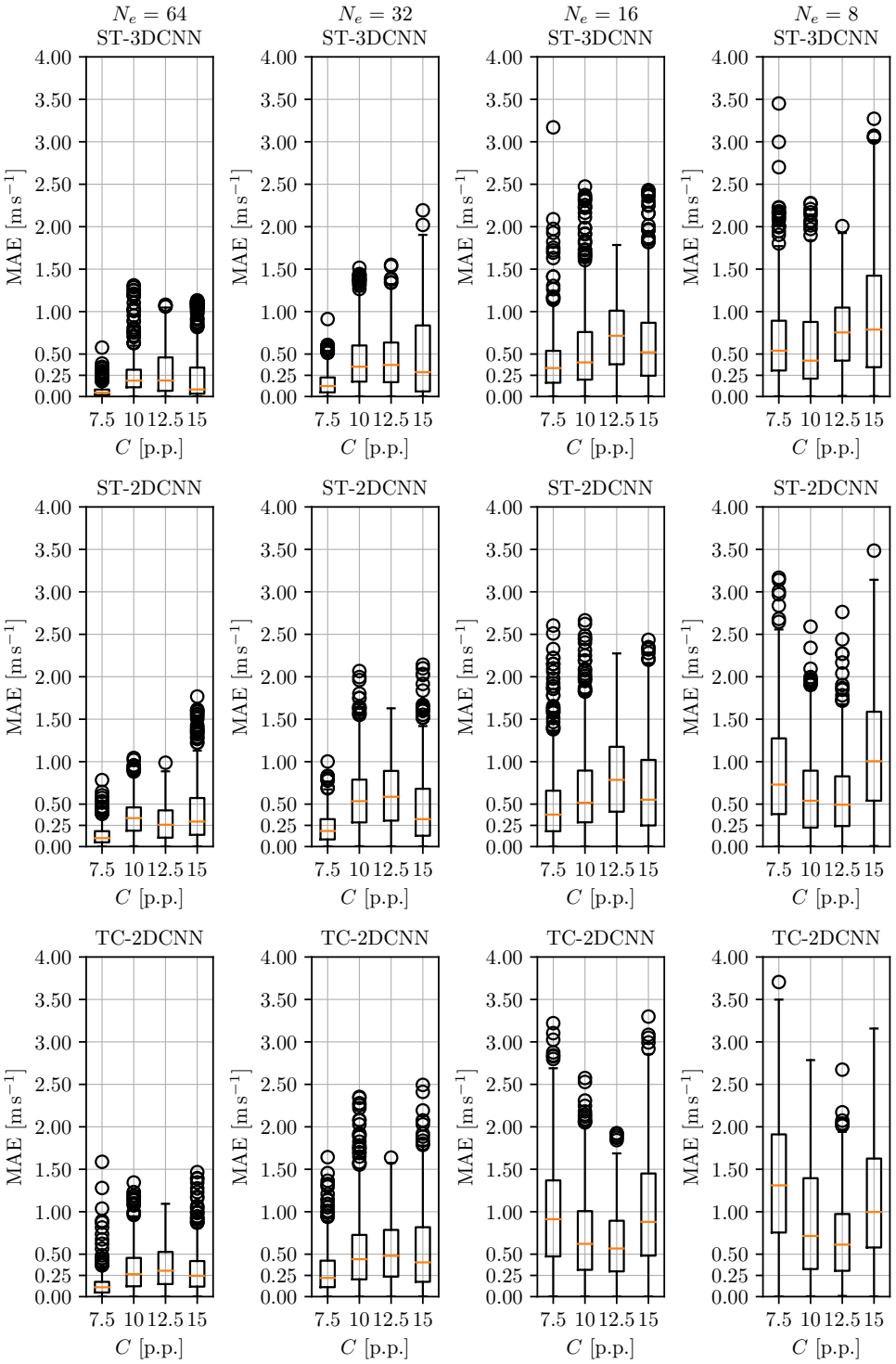

Figure 6.22: Boxplots of the MAE of the estimated shear wave velocities using our deep learning approaches. From left to right, we show values for a decreasing number of piezoelectric elements that results in a reduced lateral imaging width.

Discussion

In this experiment, we study shear wave velocity estimation for tissue characterization from sequences of US images in an end-to-end fashion and also evaluate the impact of reduced spatial information. A reduced imaging width is relevant in clinical application areas, such as laparoscopy, where small US probes in the range of a few millimeters are required [51, 215]. To address the learning task, we adapt our spatio-temporal architecture concept in a 2D and 3D fashion and evaluate learning from a lower dimensional space-time map representation and the entire image sequence.
Considering our first research question of this work, our findings demonstrate that shear wave velocity can be estimated directly from a sequence of image data in an end-to-end fashion with our spatio-temporal deep learning architecture concept within a few milliseconds. Moreover, we find that our architecture concept can be used and adapted across different lateral imaging widths and input types and that robust performance can be achieved across a range of phantom elasticities that result in different shear wave velocities. However, our findings show that using a reduced lateral imaging width is challenging for shear wave velocity estimation. It stands out that the conventional approach overestimates shear wave velocity, notably when the imaging width is reduced. These results confirm previous findings on this topic [241, 376]. We also observe that a smaller imaging width impacts all our deep learning approaches. However, our ST-3DCNN approach can still differentiate between the groups $C \in [7.5, 10]$ and $C \in [12.5, 15]$ even for the smallest lateral imaging width of 2.1 mm for $N_e = 8$, see Figure 6.21.

To perform the learning task, we present and compare different approaches. Our results demonstrate that shear wave velocity can be estimated from lower-dimensional space-time map representations with deep learning similar to previous findings on this topic [225]. However, we find that using the entire spatio-temporal information from the image sequence performs better. This indicates that valuable information for the task is lost when the space-time map representation is used and that processing the entire available spatio-temporal data is beneficial. A potential explanation is that the entire available spatio-temporal data provides a more consistent representation of the shear wave propagation and thereby, e.g., reduces the effect of noise and artifacts. An additional explanation is that using the entire available spatio-temporal data allows learning features from both spatial directions while using a space-time map representation assumes shear wave propagation only along the lateral image axis. Moreover, our results show that the approach for processing the 3D spatio-temporal data has a notable impact on the performance. Processing the entire image sequence with TC-2DCNN, i.e., using the channel dimension of the input as temporal dimension, even performs worse than using the lower dimensional image representation. These results indicate that using the channel dimension of the input for temporal processing is not suitable to capture the complex spatio-temporal dynamics of the wave propagation and that joint spatio-temporal feature learning should be performed. We find that ST-3DCNN clearly outperforms all other approaches for estimating faster shear wave velocities, see Figure 6.22.

Overall, considering our first research question, our results highlight that end-to-end shear wave velocity estimation can be performed with our spatio-temporal deep learning approach, this includes estimations for small lateral imaging widths. Considering our second research question, our results highlight that using the entire spatio-temporal data is beneficial. Comparing different approaches highlights that our ST-3DCNN approach shows promising results, outperforming the approach where the channel dimension is considered as temporal dimension.

6.2.2 3D US-based Elasticity Imaging

In the previous section, we demonstrated that shear wave velocity estimation could be performed from image sequence with our spatio-temporal deep learning approach in an end-to-end fashion. However, as outlined in Section 5.2, given shear wave velocity, it is typically still difficult to estimate actual elastic properties such as the Young's modulus. This motivates the question of whether end-to-end estimation of elastic tissue properties from sequences of US image data can also be performed with our approach. Parts of this section have been published in our study presented in [322] (©2022 IEEE). The study [322] was a collaborative study, the development of the experimental setup, data acquisition, and evaluation with conventional image processing methods were performed by M. Neidhardt. The aspects regarding spatio-temporal deep learning were developed and evaluated by the author of this thesis and are in the focus of this work.

Considering our first research question, we study whether entire elasticity maps can be estimated from real ultrasound SWEI data with spatio-temporal deep learning. This requires the complex task of end-to-end elasticity estimation from sequences of image data and thereby circumvents intermediate steps such as shear wave velocity estimation followed by conversion to elastic properties using mechanical models. We systematically study whether our spatio-temporal deep learning approach generalizes to elasticities not present during training. Also, we put a focus on the flexibility of our approach and evaluate performance w.r.t. different US push locations for wave excitation.

Considering our second research question, we address how this task can be performed and how the learning problem can be formulated to obtain entire elasticity maps with localized estimations. To this end, we present and evaluate a localized training approach. In particular, we train our spatio-temporal deep learning approach end-to-end to learn the relationship between localized shear wave propagation and local elasticity using localized spatio-temporal windows. Our approach is visualized in Figure 6.23.

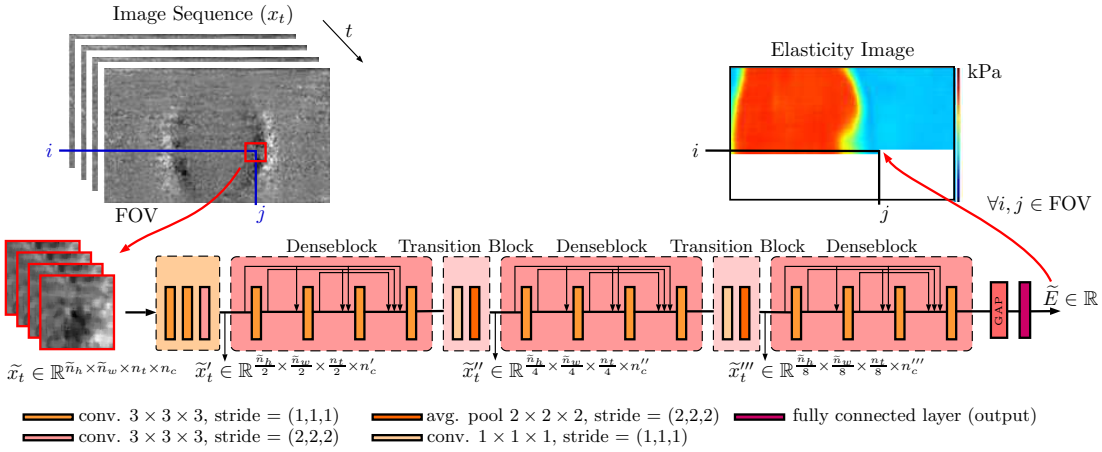

Figure 6.23: Our approach and architecture for end-to-end localized elasticity imaging. We estimate a global elasticity map by estimating local elastic properties $\widetilde{E}$ pixel-wise at $\forall i, j \in$ FOV with our architecture using localized 3D spatio-temporal windows $\widetilde{x}_t$ as input. Figure adapted from [322] (©2022 IEEE).

Definition of the Learning Task

We address the task of estimating local elastic properties of tissue in an end-to-end fashion using sequences of US images that capture wave propagation over time. Given a sequence of 2D images, $x_t = [x^{[0]}, x^{[2]}, \ldots, x^{[n]}]$ with $x^{[i]} \in \mathbb{R}^{n_h \times n_w \times n_c}$ which capture shear wave propagation over time, we try to estimate the elasticity locally using a spatio-temporal window $\widetilde{x}_t \in \mathbb{R}^{\widetilde{n}_h \times \widetilde{n}_w \times n_t \times n_c}, \widetilde{x}_t \subset x_t$ centered at pixel location $o = [i, j]$. Hence, we develop and evaluate a spatio-temporal deep learning approach for learning $f_E \colon \mathbb{R}^{\widetilde{n}_h \times \widetilde{n}_w \times n_t \times n_c} \to \mathbb{R}$. An entire elasticity map can then be estimated by applying our approach pixel-wise, see Figure 6.23.

Experimental Setup and Data Sets

For evaluation of our methods, we use a dataset consisting of 2D US images over time that capture shear wave propagation in tissue-mimicking gelatin phantoms. We use data from phantoms with varying stiffness that results from different gelatin to water ratios. The full dataset consists of gelatin phantoms with gelatin concentrations $C \in [5.0, 7.5, 10.0, 12.5, 15.0, 17.5]$ given in p.p.. To obtain a ground truth elasticity value, i.e., the Young's modulus for each concentration, unconfined compression tests are performed using cylindrical phantoms of each concentration. Details can be found in our corresponding publication [322]. Results for the estimated Young's moduli are shown in Table 6.16. Note that the considered elasticity range is similar to values of soft tissue in the literature [4, 382]. For training and testing of our methods, we consider data from homogeneous phantoms with a block shape ($\sim 100 \times 100 \times 100\,\mathrm{mm}^3$) of each concentration. Moreover, to test our methods on homogeneous phantoms with stiffer inclusions, we consider data from phantoms with embedded circular inclusions with a radius of approximately 5 mm and 10 mm. In addition to that, we also use data from one gelatin phantom with an embedded chicken heart tissue.

The dataset is acquired with an experimental setup shown in Figure 6.24. It consists of a 128-channel ultrasound system (Cicada, Cephasonics Inc), a linear array probe (128 elements, 0.29 mm pitch, center frequency 7.5 MHz), a serial robot (UR3, Universal Robots) and a force sensor (Nano43, ATI). Using the robot, data acquisition is performed at 80 randomly chosen positions for each gelatin block. At each position, data is acquired for seven different push locations shown in Figure 6.24. Shear wave excitation is performed with an unfocused push sequence (120 V, 2000 push cycles, 10 mm depth) using a continuous segment of 11 piezoelectric elements. The shear wave propagation is imaged with plane wave imaging with an imaging frequency of 7000 Hz. For data processing and enhancement of the scatterer displacement, the Loupas algorithm is applied [289]. Each image sequence x_t consists of $n_t = 35$ subsequent images with a FOV of 20 $\times$ 33 mm and a resolution of 250 $\times$ 600 pixels along the depth and lateral axis, respectively. The corresponding sequence-level ground truth y is given by the Young's modulus E_{gt} measured for the gelatin concentration of the phantom.

Table 6.16: Estimated ground truth Young's modulus E for the different gelatin concentrations C determined with unconfined compression tests. (Compare [322])

C [p.p.]	5.0	7.5	10	12.5	15	17.5
E [kPa]	17.42	37.55	56.04	72.64	97.22	126.05

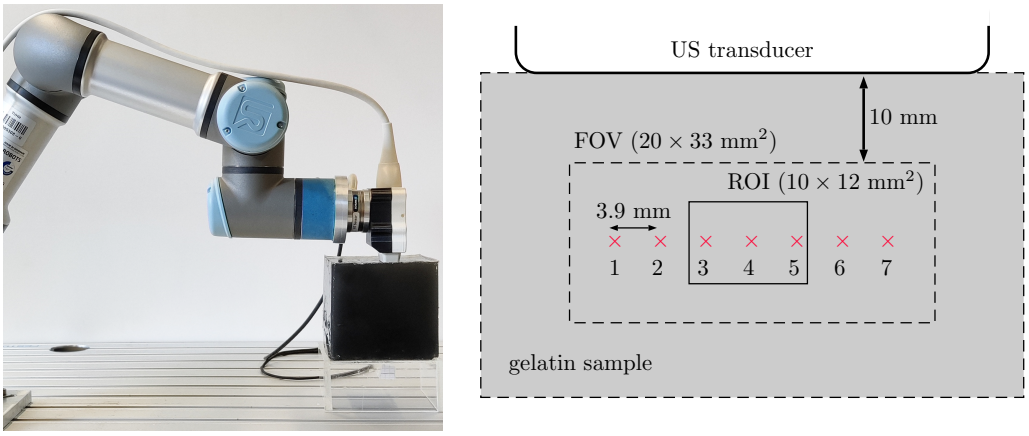

Figure 6.24: The setup for ultrasound shear wave data acquisition. (Left) The experimental setup for data acquisition. (Right) A schematic drawing of the seven different push locations relative to the ROI that are used for data acquisition. A push location is indicated by a red cross. Figure adapted from [322] (©2022 IEEE).

Methods

To develop our approach for local elasticity estimation from 3D spatio-temporal windows, we present both an architecture and a specialized training approach. Our architecture is based on our CNN concept of this work and is shown in Figure 6.23. We explain specific details of our approach in the following paragraphs.

ST-3DCNN. As the backbone architecture of our CNN architecture concept, we use the concept of DenseNet [210]. We combine our approach with 3D spatio-temporal convolutions and pooling operations to jointly learn features from space and time of the spatio-temporal windows $\widetilde{x}_t$. This results in a 3D spatio-temporal CNN, and we consider a spatio-temporal window size of $65 \times 65 \times n_t$ as the baseline for our experiments. We also evaluate and compare smaller spatio-temporal windows $\widetilde{x}_t$ with a size $33 \times 33 \times n_t$, $17 \times 17 \times n_t$, $9 \times 9 \times n_t$ and $5 \times 5 \times n_t$ with $n_t = 35$ frames. We use the same architecture for the different spatio-temporal window sizes, except that we adapt the spatial kernel stride of our architecture in Figure 6.23, i.e., we use a spatial stride of one for the spatio-temporal window sizes of $9 \times 9 \times 35$ and $5 \times 5 \times 35$.

Architecture Details. For our ST-3DCNN architecture we use three initial convolutional layers with 24 feature maps each, followed by three DenseNet blocks with four convolutional layers each with a growth rate of 10. Further details are given in Figure 6.23. All methods are implemented in PyTorch.

Patch-Wise Training. We train our network using homogeneous phantoms, and hence we assign the corresponding ground truth elasticity E_{gt} of a corresponding phantom to a spatio-temporal window $\widetilde{x}_t$. For training, we minimize the MSE loss function between the defined target ground truth elasticity E_{gt} and our estimated elasticity E_p based on a spatio-temporal window. In this way, we train our network to learn the relationship between elasticity and shear-wave propagation for a small local region in an end-to-end fashion. Each network is trained for 250 epochs with a batch size of $m_b = 250$ using Adam [242] for optimization with a learning rate of $\mathrm{lr} = 0.0001$. We divide the learning rate after 150 epochs by a factor of two every 50 epochs. For training of our approaches, we consider spatio-temporal windows $\widetilde{x}_t$ that are located within a defined ROI with a size of 121×181 pixels (10×12 mm^2), see Figure 6.24. In each training epoch, we consider one spatio-temporal window $\widetilde{x}_t$ with a random location within the ROI from every image sequence x_t in our training data set. We normalize the pixel intensities of each input $\widetilde{x}_t$ to have a zero mean and standard deviation of one. We apply horizontal and vertical flipping, multiple 90° rotations, Gaussian blur, and randomized input erasing of the input data for data augmentation.

ToF. For comparison, we consider the results of a ToF approach performed similar to Song et al. [425]. Shear wave velocity estimation is performed by dividing a known travel distance with the time difference of two signals [45, 424, 425, 452]. Estimations are performed based on a cross-correlation of the time-varying signals measured at an equivalent distance of 65 pixels along the lateral axis. Estimates

which are not in the physically plausible range of $0.1\,\mathrm{m\,s^{-1}}$-$10\,\mathrm{m\,s^{-1}}$ are considered failed estimates. After shear wave velocity estimation, the Young's modulus is related using

$$E_{ToF} = \theta_E \cdot \rho \cdot 2(1+\nu) \cdot v^2 \tag{6.5}$$

with the density $\rho = 1000\,\mathrm{kg\,m^{-3}}$ and the Possion's ratio $\nu = 0.5$ [390]. A scaling factor $\theta_E \in \mathbb{R}_+$ is used to correct for constant errors between the estimates E_{ToF} and the ground truth Young's modulus E_{gt} measured with indentation experiments. Minimizing the offset between E_{ToF} and E_{gt} yields $\theta_E = 0.75$ for the dataset.

Performance Measure - Evaluation and Metrics

We evaluate our approach with two training scenarios, with and without all elasticities present during training. For the first scenario, we split our data based on the 80 different positions used for data acquisition of each phantom. We perform four-fold cross-validation, and in each fold, we randomly split the data and use data from 60 positions of each phantom for training and data from 10 positions each for validation and testing.
Second, we systematically leave out elasticities during training to test our approach on unseen elasticities. To this end, we perform cross-validation and leave out the entire data of one gelatin concentration, i.e., elasticity. Thereby, we evaluate our approach to interpolate between different elasticities. We do not leave out the boundary elasticities, i.e., $C = 5$ p.p. and $C = 17.5$ p.p., to avoid out of distribution estimations for the regression task. This results in four-fold cross-validation, and in each fold, we split the data and use 50% of the data for testing and 50% for validation.
For all our training scenarios, we remove push at locations one and seven completely from our training data. We perform this step to evaluate our approach on unseen push locations further away from the ROI. For evaluation of our approach using inclusion phantoms, we train the network for additional 10 epochs with inhomogeneous phantom data to account for wave reflections at boundaries that are not present in homogeneous phantoms.
We consider the MAE between the estimated and ground truth elasticity as evaluation metric. In addition to that, we also report the pCC, inference time, and throughput for our deep learning methods. For our inclusion phantoms, we also report the Dice coefficient to evaluate the segmentation performance of the stiffer inclusions.

Results

We report the pixel-wise MAE of ST-3DCNN for our ROI w.r.t. the different push locations and elasticities in Figure 6.25. For comparison, we also report the results of the ToF approach. Our results show that considering all elasticities during training leads to a lower MAE compared to the scenario where we systematically leave out elasticities. For both scenarios, our results show that the MAE is independent of the push location and that the MAE increases with increasing

elasticity. Figure 6.26 visualizes the estimations of our ST-3DCNN in comparison to ToF. For this and the following evaluations, if not indicated otherwise, we consider the training scenario, where we leave out the evaluated elasticities. Figure 6.26 further highlights that ST-3DCNN leads to consistent estimations, independent of the push location, and also allows for estimations inside the push location, where the ToF approach fails completely. We also report results for the entire FOV using push one, four, and seven in Figure 6.27. Note that push one and seven are not considered during training and that we train our approach only based on data within the ROI. Our results show that robust estimations with a low standard deviation can also be performed outside the ROI and for the push locations that have not been considered during training. However, our results show that for lower phantom elasticity (37.55 kPa) estimations far away from the push location become difficult for ST-3DCNN and ToF, i.e., our results show a high standard deviation.

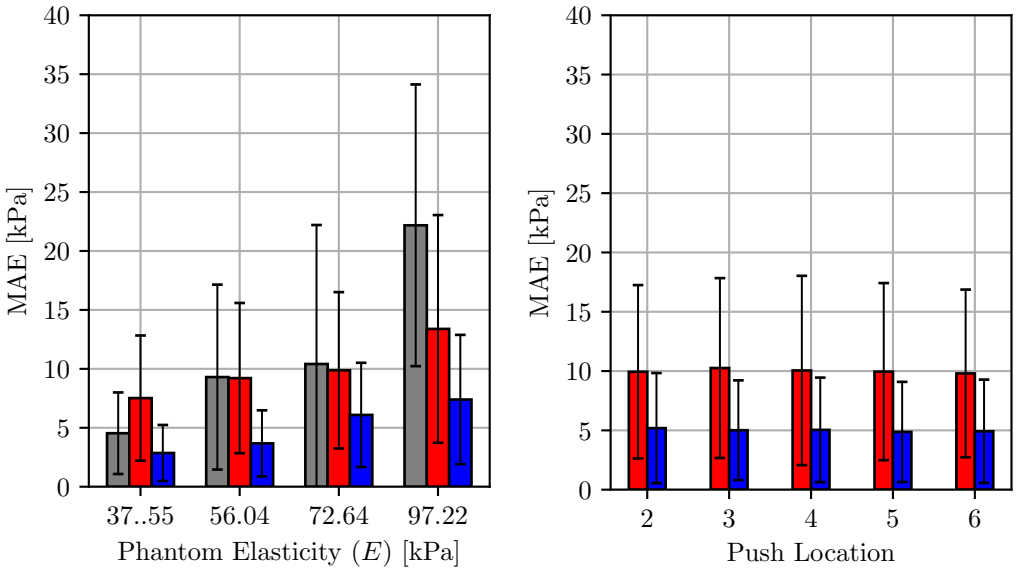

Figure 6.25: Localized elasticity estimation performance. (Left) MAE w.r.t. different phantom elasticity, considering the results of all push locations. (Right) MAE w.r.t. the different push locations. (Red) Spatio-temporal CNN, where evaluated elasticity is left out during training; (Blue) Spatio-temporal CNN, where evaluated elasticity is also present during training; (Grey) ToF, note failed estimations are excluded. Figure adapted from [322] (©2022 IEEE).

Moreover, we report the performance for different spatio-temporal window sizes in Table 6.17. Our results show that the largest window size performs best and that performs gradually reduces for a reduced window size. The largest spatial input size of 65×65 pixels ($\sim 4 \times 5\,\text{mm}^2$) improves performance by 30% compared to the smallest spatial size of 5×5 pixels ($\sim 0.32 \times 0.4\,\text{mm}^2$). However, this comes at the cost of a notably reduced throughout.

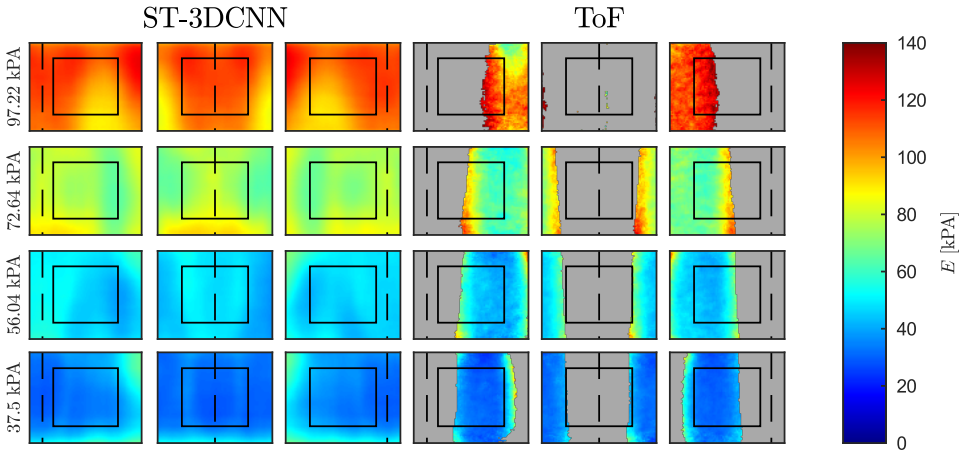

Figure 6.26: Localized elasticity estimation performance. Estimations of the Young's modulus with our ST-3DCNN approach (left) and with the conventional ToF method (right) for different push locations and ground truth elasticities (different rows). The push location is indicated by the black dashed line for push 2 (left line), 4 (middle line), and 6 (right line). For each pixel, we show the mean Young's modulus across our different measurements of the test dataset. Failed estimates are indicated in gray; the black square indicates the ROI with a size of 10×12 mm^2. We use a spatio-temporal window size of $65 \times 65 \times 35$. Figure adapted from [322] (©2022 IEEE).

Table 6.17: MAE and pCC for different window sizes using our ST-3DCNN approach. Throughput denotes the number of positions, i.e., pixel locations, for which elasticity can be estimated within one second using an nvidia Tesla V100-SXM2-32GB with a batch size of $m_b = 500$. (Compare [322] (©2022 IEEE))

$\widetilde{n}_h \times \widetilde{n}_w$	MAE [kPa]	pCC [%]	Throughput [pixels/s]
65×65	**9.99 ± 7.49**	**93.39**	508
33×33	10.64 ± 7.48	88.55	1 688
17×17	11.83 ± 8.25	87.54	2 348
9×9	12.63 ± 9.01	86.53	5 651
5×5	14.40 ± 10.82	72.81	**15 745**

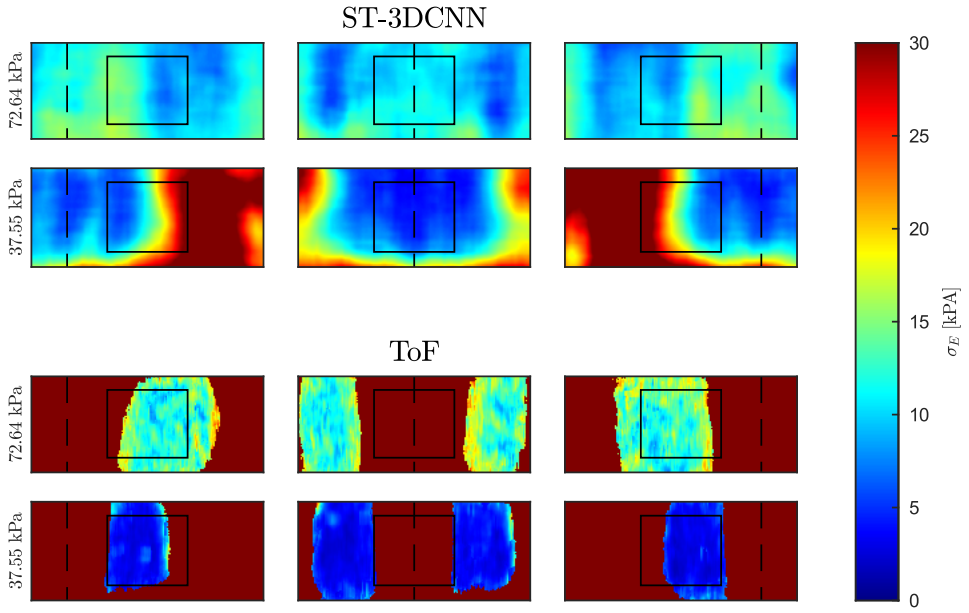

Figure 6.27: Localized elasticity estimation for the entire FOV for three different push locations (columns) and different elasticities (rows). We show the standard deviations (σ_E) of the pixel-wise estimated Young's moduli across our different measurements of the test dataset. The black rectangle indicates the ROI used for training. The black dashed line indicates the push locations. From left to right, we report results for push location one, four, and seven. We use a spatio-temporal window size of $65 \times 65 \times 35$ pixels. Figure adapted from [322] (©2022 IEEE).

Furthermore, we show results for the phantoms with embedded cylindrical inclusions in Figure 6.28 and report corresponding performance metrics in Table 6.18 and Table 6.19. For each phantom, we consider the mean of the estimations across nine push and imaging sequences, and we consider the training scenario where all elasticities are present during training. For the estimation of the Dice coefficient, we use the mean Young's modulus of inclusion and background as the binarization threshold. We also provide estimations for a varying binarization threshold in Figure 6.29. Our results show that also high performance can be achieved for inclusion phantoms with ST-3DCNN, outperforming the ToF approach. Lastly, we report the results for the phantom with the embedded chicken heart tissue in Figure 6.30. Note that we do not have ground truth values for the elasticity of chicken heart, and hence we do not provide any performance metrics. Our results show that also for such a phantom, estimations are feasible with our approach.

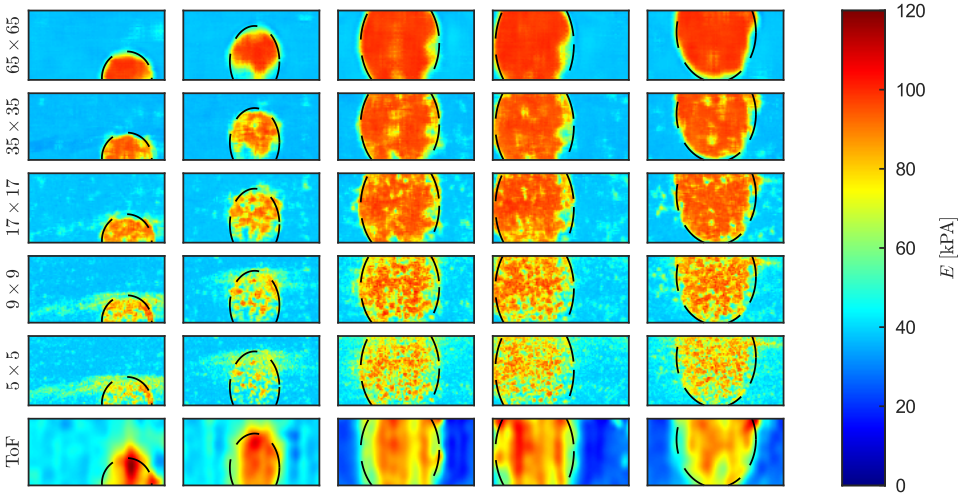

Figure 6.28: Localized elasticity estimation for the different phantoms with stiff inclusions, shown in the different columns. The results show the estimations for the entire FOV. The first five columns show the result for our ST-3DCNN approach with different spatio-temporal window sizes. The last column shows the ToF results. We indicate the shape of the inclusion with a dashed line. The ground truth elasticity of the backgrounds of the phantoms is 37.55 kPa and the inclusions have a ground truth elasticity of 97.22 kPa. Figure adapted from [322] (©2022 IEEE).

Table 6.18: Performance metrics for inclusion (in) and background (bg) with the corresponding mean μ for all five inclusion shapes. Results are shown for spatio-temporal windows with a size of 65×65 pixels. The inclusion shapes are displayed in Figure 6.28 with, e.g., column one referring to phantom #1. (Compare [322] (©2022 IEEE))

Method	#	MAE_{in} [kPa]	MAE_{bg} [kPa]	Dice
ST-3DCNN	1	10.16 ± 13.13	0.97 ± 2.30	0.93
	2	19.52 ± 19.06	1.06 ± 3.25	0.81
	3	4.70 ± 8.32	2.42 ± 5.01	0.98
	4	5.50 ± 10.98	2.13 ± 5.25	0.96
	5	6.11 ± 11.51	2.27 ± 5.84	0.95
	μ	$\mathbf{7.50 \pm 12.87}$	$\mathbf{1.64 \pm 4.32}$	**0.93**
ToF	1	15.39 ± 10.14	9.19 ± 10.09	0.79
	2	13.28 ± 10.38	8.37 ± 0.89	0.85
	3	17.14 ± 9.66	14.49 ± 9.24	0.89
	4	13.87 ± 9.05	13.60 ± 9.77	0.93
	5	19.91 ± 10.15	11.99 ± 10.89	0.86
	μ	$\mathbf{16.28 \pm 10.05}$	$\mathbf{11.11 \pm 10.08}$	**0.86**

Table 6.19: Performance metrics for all inclusion shapes reported for the inclusion (in) and background (bg) with all studied spatio-temporal window sizes. (Compare [322] (©2022 IEEE))

$\widetilde{n}_h \times \widetilde{n}_w$	MAE_{in} [kPa]	MAE_{bg} [kPa]	Dice
65 × 65	**7.50 ± 12.87**	**1.64 ± 4.32**	**0.93**
33 × 33	11.45 ± 13.55	1.91±5.05	0.90
17 × 17	14.29 ± 13.59	4.21±7.81	0.86
9 × 9	22.74 ± 14.43	5.30 ± 7.54	0.75
5 × 5	29.68 ± 13.84	6.37 ± 7.47	0.60

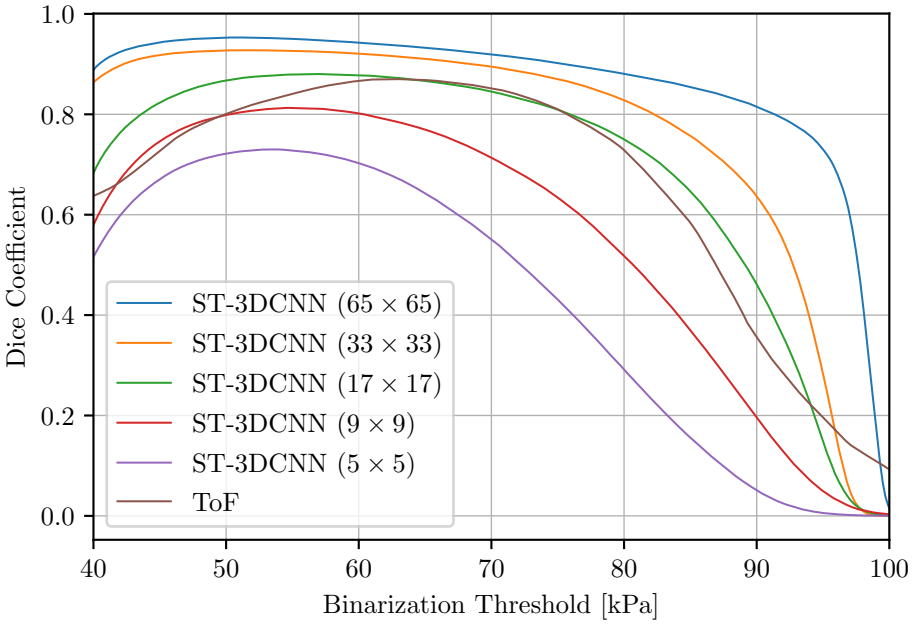

Figure 6.29: Segmentation performance vs. binarization threshold. Results are shown for ToF and our ST-3DCNN approach using different spatio-temporal window sizes. We show the mean value for all inclusion phantoms. Figure adapted from [322] (©2022 IEEE).

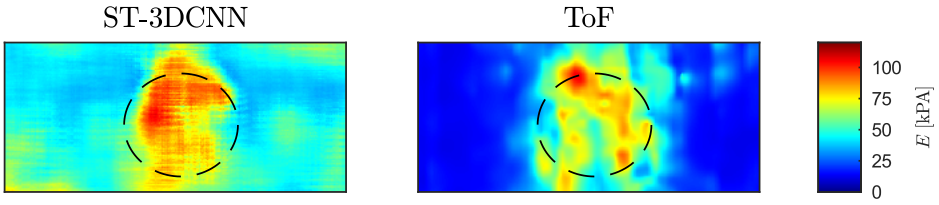

Figure 6.30: Localized elasticity estimation for a gelatin phantom with an embedded chicken heart tissue with ST-3DCNN and ToF. We show the estimations for the entire FOV and indicate the position of the chicken heart tissue with a dashed circle. The background of the phantom has a ground truth elasticity of 37.55 kPa. Figure adapted from [322] (©2022 IEEE).

Discussion

In this experiment, we study localized elasticity estimation from sequences of US images with our spatio-temporal deep learning approach. To this end, we present an architecture and training concept and evaluate the performance on gelatin phantoms considering various push locations and elasticities. We train our approach to estimate the Young's modulus as a surrogate for tissue elasticity in an end-to-end fashion. Thereby, we simplify the traditional data processing pipeline, which typically includes shear wave velocity extraction and subsequent estimation of the Young's modulus [7, 236, 547].
Our results demonstrate that localized elasticity estimation can be performed with our ST-3DCNN and our training approach across a wide range of elasticities. Our findings also highlight that the estimation of elasticity in stiffer phantoms with ST-3DCNN is consistent, which is known to be challenging [104, 225, 304, 376, 403, 481]. These findings demonstrate that the complex relationships of elasticity and spatio-temporal features of the wave propagation can be learned with our approach in an end-to-end fashion. Moreover, our results demonstrate that estimations can be performed independently of the relative push location and also for push locations that have not been considered during training. These results further indicate that the underlying spatio-temporal relationships of the wave propagation can be learned well from the sequences of image data allowing for robust and high performance. Furthermore, we compare the training scenario where we consider all elasticities during training compared to the case where we systemically leave out elasticities that are considered for evaluation. As expected, our results show that estimating known elasticities improves the performance of our ST-3DCNN approach, and in this training scenario, our approach outperforms ToF substantially across the entire range of elasticities. Note that for future applications additional elasticities can be integrated in the training dataset. However, our results show that even elasticities that have not been considered during training can be estimated. This is an important property of our approach to provide consistent results across a continuous range of elasticities. Also, note that the ToF approach leads to failed estimates, while our ST-3DCNN approach provides consistent results across all our experiments. Figure 6.26 highlights that with

ST-3DCNN, we can perform estimations even within the push location and our results show that also these complex spatio-temporal relationships can be learned in an end-to-end fashion. These results are an interesting finding and valuable contribution considering that such aspects have not been studied in any previous work that uses deep learning methods in combination with simulated data [3,464]. Although our results demonstrate that estimations can be performed with ST-3DCNN for a much larger FOV than the ROI that is considered for training, estimations show an increased standard deviation with an increased distance to the push location, see Figure 6.27. These results are similar to the ToF approach. These findings indicate that our approach fails when no or limited displacement information is present in the data, highlighting that the actual spatio-temporal relationships of the wave propagation are learned and used by ST-3DCNN. Similar, these results show that our approach does not rely on other phantom specific characteristics that would be independent of the distance to the push location.

Considering our second research question, we present and study a localized training approach. We evaluate whether the spatio-temporal relationships required for localized elasticity estimation can be learned from small localized spatio-temporal windows. In this context, we systematically compare different spatio-temporal window sizes. Our results show that increasing the spatial window improves the performance, see Table 6.20. These results are also similar to our findings from the previous Section 6.2.1, where the results demonstrated that performance decreases for smaller lateral imaging width. However, our results show that real-time estimations could become difficult for pixel-wise estimations for a large FOV using the largest spatio-temporal window size that results in a throughput of 508 pixels/s for our current hardware setup, see Table 6.20. Hence, using a smaller window size such as 33×33 pixels might be a good trade-off between performance and inference time. Another approach could be to downsample the 65×65 pixels spatio-temporal window to 33×33 pixels. Also, computations could be shared across overlapping regions during testing to improve the throughput [405]. Moreover, only sparse estimations could be performed for a FOV, e.g., only estimations for every nth pixel could be performed, and thereby similar inference times can be achieved for different FOVs sizes. Also, more powerful hardware could be used, and estimations could be performed in parallel using multiple GPUs. Thus, our approach can be scaled effectively, and estimating an entire elasticity map within a few milliseconds could be achieved. Evaluating such aspects that improve the efficiency of our approach is an interesting direction for future work.
We also evaluate our spatio-temporal deep learning approach on gelatin phantoms with stiff inclusions. Our results show that estimations for inclusion phantoms can be performed, outperforming the ToF approach. Also, our results demonstrate that elasticity estimation for a phantom with an embedded chicken heart is also feasible, although training is performed only on gelatin phantoms. This further indicates that our spatio-temporal deep learning approach generalizes well and that the underlying spatio-temporal relationships of wave propagation and elasticity can be learned with our approach. Considering that we use different spatio-temporal window sizes, a smaller spatial area could lead to more distinct

boundaries. However, our results show that an increased window size leads to more consistent estimations and overall the boundary is more distinctly visible, see Figure 6.28. These results indicate that the reduced performance of smaller spatio-temporal windows outweighs the benefit that could result from more localized estimations. An interesting direction for future work could be a multi-scale approach where different resolutions are processed end-to-end at the same time.

Overall, considering our first research question and our results, our results highlight that end-to-end estimation of localized elasticities can be performed from sequences of US images with spatio-temporal deep learning, independent of the wave propagation direction in an end-to-end fashion. Considering our second research question, our findings demonstrate that with our localized training approach and our ST-3DCNN architecture the relationship between localized shear wave propagation and local elasticity can learned with high performance. Our results highlight that more consistent and robust results can be achieved in comparison to the conventional approach.

6.2.3 3D OCT-based Elasticity Imaging

In this section, we evaluate our spatio-temporal deep learning approach combined with sequences of 2D OCT images for the task of elasticity imaging. Parts of this section have been published in our study presented in [323] (©2020 IEEE). The study [323] was a collaborative study, the development of the experimental setup, data acquisition, and evaluation with conventional image processing methods were performed by M. Neidhardt. The aspects regarding machine learning and deep learning were developed and evaluated by the author of this thesis and are in the focus of this work.

Considering our first research questions, we address whether material properties can also be estimated directly from sequences of 2D OCT data in an end-to-end fashion using our spatio-temporal deep learning approaches. Considering the fast acquisition rate of OCT, this requires processing and learning from several hundreds of images. We also contribute to the question of whether similar concepts can be shared across different imaging modalities and use similar architecture concepts as for US data processing. In addition to that, we evaluate whether such an approach can generalize to unknown elasticities.

Considering our second research question, we present and compare several spatio-temporal deep learning models. Similar to our experiments using 2D US data, we try to learn from sequences of entire images, i.e., sequences of 2D OCT data, and compare the performance to lower dimensional space-time map representations that are widely adopted by conventional approaches [329,547]. For comparison to our end-to-end approach, we also study material property estimation using shear wave velocity as an explicit feature combined with machine learning methods.

Definition of the Learning Task

Our task is to estimate material properties directly from sequences of 2D OCT images $x_t = [x^{[0]}, x^{[1]}, \ldots, x^{[n]}]$ with $x^{[i]} \in \mathbb{R}^{n_h \times n_w \times n_c}$, which capture displacement information over time resulting from wave propagation. We consider gelatin concentration $C \in \mathbb{R}_+$ of phantoms as surrogate value for material properties. As a first approach, we consider shear wave velocity $v \in \mathbb{R}_+$ as an explicit feature estimated with conventional methods and try to learn the mapping from shear wave velocity to gelatin concentration with machine learning regression methods. Hence, we try to learn the mapping $f_{C_1} \colon \mathbb{R} \to \mathbb{R}$. As a second approach, we use a lower-dimensional representation that is typically used for shear wave velocity estimation [329, 547]. Here, we try to learn a function $f_{C_2} \colon \mathbb{R}^{n_w \times n_t \times n_c} \to \mathbb{R}$. As a third approach, we define an end-to-end regression task, and we try to estimate gelatin concentration directly from the entire image sequence. Hence, we try to learn a function $f_{C_3} \colon \mathbb{R}^{n_h \times n_w \times n_t \times n_c} \to \mathbb{R}$. Thereby, we map an input image sequence directly to gelatin concentration C without feature selection or assumptions about the wave propagation.

Experimental Setup and Data Sets

For training and evaluation of our methods, we use a dataset consisting of 2D OCT images over time that capture wave propagation in tissue-mimicking gelatin phantoms. We use data from phantoms with varying stiffness that results from different gelatin to water ratios. The full dataset consists of gelatin phantoms with gelatin concentrations $C \in [4.2, 4.8, 5.6, 6.7, 8.3, 11.1]$ given in p.p. and we use data from two phantoms per gelatin concentration. The dataset is acquired with an experimental setup shown in Figure 6.31. It consists of a high-speed swept-source OCT device (OMES, Optores), a clinical needle attached to a piezoelectric actuator, and a robot (IRB 120, ABB) for automatic placement of the needle. For imaging, 2D OCT images are acquired at a scan rate of 30 kHz with an effective size of 32×250 pixels along lateral and depths axis, respectively, and a FOV of approximately 3×2 mm^2 in air.
For data acquisition, the robot inserts the needle within a gelatin phantom, and shear wave excitation is performed by vibrating the needle with a frequency of 100 Hz using a single burst function. A random time delay between OCT data acquisition and shear wave excitation is used to increase the variance of the data. We consider data from each phantom using two different orientations w.r.t. the FOV of the OCT, and for each orientation, we use data for four different needle positions 5 mm, 10 mm, 15 mm and 20 mm relative to the FOV, see Figure 6.31. For each combination of needle position and sample orientation, four measurements are used. Overall, this results in the full dataset with 64 2D OCT image sequences x_t for each concentration. All our methods are trained and evaluated with the unwrapped phase information of the OCT data.
Moreover, conventional shear wave velocity estimation is performed for the dataset similar to [329]. The displacement resulting from the wave propagation is visualized by calculating the phase difference of subsequent B-scans. Afterwards, values are averaged along the the axial z-direction, which results in a 1D+t rep-

resentation and we refer to this input as $x'_t \in \mathbb{R}^{n_w \times n_t \times 1}$. Afterwards, shear wave velocity is estimated by performing a linear regression of the wave peaks in the 2D space-time image representation. The slope of the regression corresponds to the shear wave velocity. Results of the conventional shear wave velocity estimation are shown in Figure 6.32.
In summary, our dataset consists of 2D OCT image sequences with $n_t = 400$ B-scans, each corresponding to 13.3 ms. In addition to that, shear wave velocity v information is available for each sequence x_t. The sequence-level label y is given by the gelatin concentration C of the phantom.

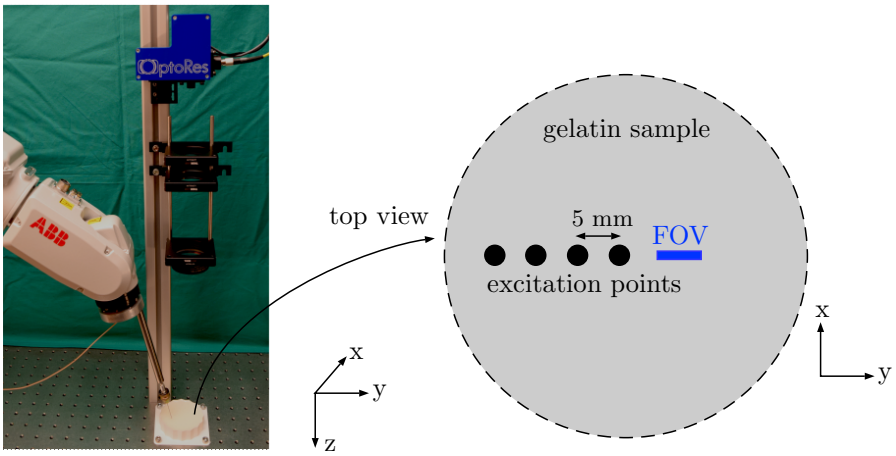

Figure 6.31: Experimental setup for data acquisition with a robot, piezoelectric actuator and needle, OCT scan head, and gelatin sample. For data acquisition shear wave excitation is performed at four different positions relative to the FOV. Figure adapted from [323] (©2020 IEEE).

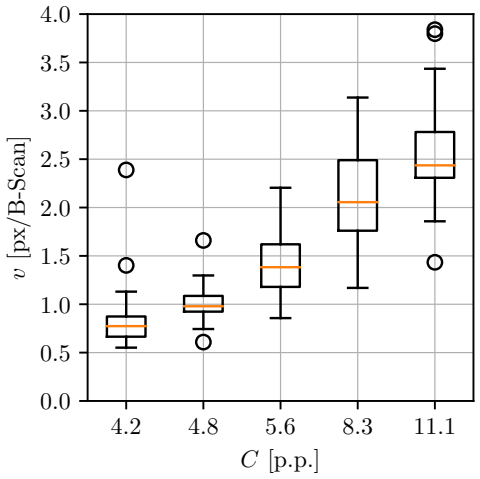

Figure 6.32: Results of the feature extraction, i.e., estimated shear wave velocities v for the different gelatin concentrations C. Figure adapted from [323] (©2020 IEEE).

Methods

To estimate gelatin concentration from a sequence of 2D OCT images, we consider three different strategies shown in Figure 6.33. We develop all these methods based on our spatio-temporal CNN concept of this work. As our backbone architecture, we use the idea of DenseNet [210]. To address our aforementioned learning task, we adapt and evaluate our approach in a 2D and 3D fashion. Specific details of our approaches are given in the following paragraphs.

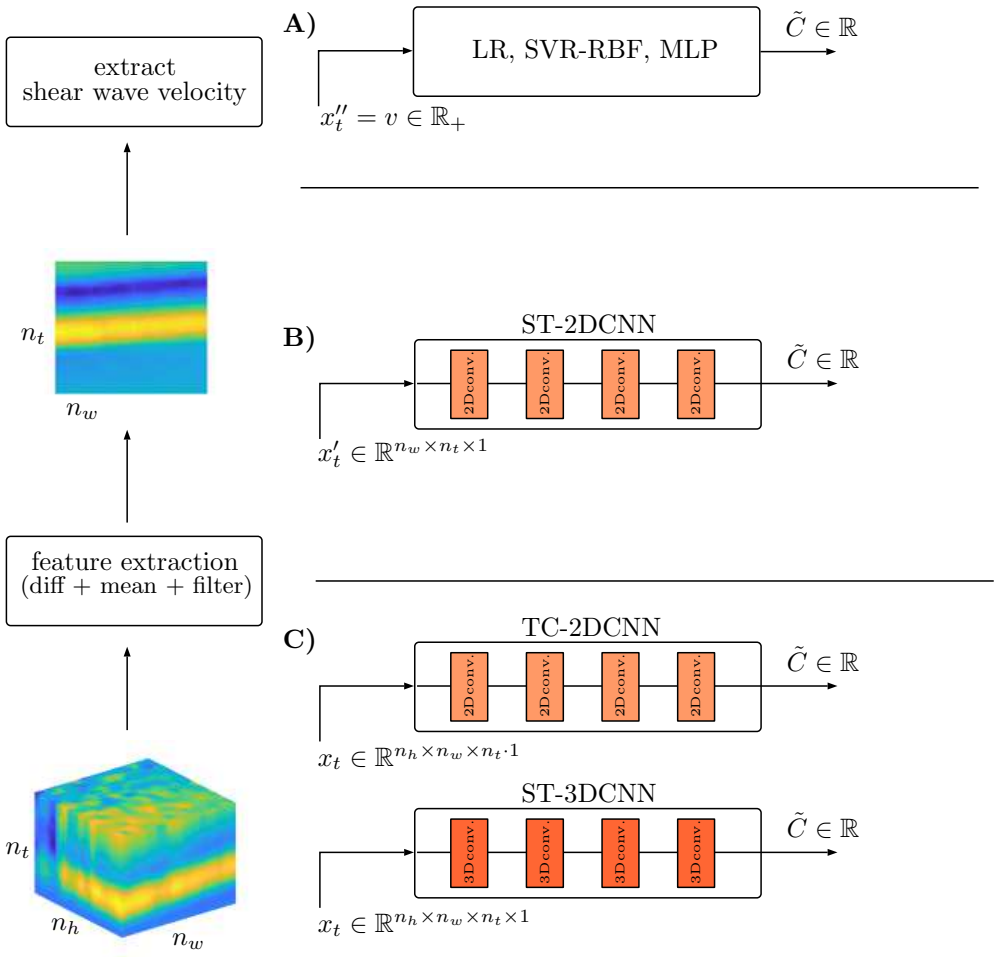

Figure 6.33: Our approaches for material property estimation from sequences of 2D OCT images that capture displacement information over time. We consider gelatin concentration C as a surrogate for a material property. (A) We follow the conventional pipeline and extract shear wave velocity and apply various machine learning methods for conversion of the velocity v to gelatin concentration. (B) We use a lower dimensional space-time map extracted from the image sequence and use a ST-2DCNN approach. (C) Our approaches, ST-3DCNN and TC-2DCNN, estimate gelatin concentration end-to-end from the original image sequence. The output of our methods is indicated by $\tilde{C}$. Note that our actual images are gray-scale and colored inputs are only shown for improved visualization.

LR, SVR-RBF, MLP. First, we consider shear wave velocity as an explicit feature for our task. For conversion of the estimated shear wave velocities to our surrogate material parameter, i.e., gelatin concentration, we consider three machine learning approaches. To perform a simple linear regression from shear wave velocity to gelatin concentration, we use linear regression (LR). Next, we consider non-linear regression approaches and use support vector regression (SVR) with a Gaussian kernel (RBF) as well as multi-layer fully connected neural networks (MLP). We evaluate two MLP configurations using two hidden layers and consider 250 and 500 hidden units in each layer.

ST-2DCNN. Second, we circumvent the feature extraction step, i.e., shear wave velocity estimation, and try to estimate the gelatin concentration directly from the lower-dimensional 2D space-time map representation of the image sequence. To address this learning task, we use our CNN architecture concept with 2D convolution and pooling operations. In this way, we can process the 2D image representation with our network in an end-to-end fashion.

TC-2DCNN. Third, we also circumvent the reduction of the dimensionality and try to learn from the original image sequence in an end-to-end fashion. As a first baseline approach, we use the concept where the input's channel dimension is considered a temporal dimension. In this way, we can still use 2D operations for our CNN architecture while learning from the entire sequence of 2D images.

ST-3DCNN. Fourth, we evaluate full spatio-temporal feature learning from the entire OCT image sequence and combine our CNN architecture concept with 3D spatio-temporal convolutions and pooling operations. Hence, this approach uses the entire sequence as input and thereby uses the entire spatial and temporal information of the wave propagation. Note that average pooling layers with a temporal stride and kernel larger than one are an effective way to reduce the computational efforts that result from the large number of OCT images. However, it also downsamples the temporal dimension and might reduce the performance. Hence, we also evaluate the case where we set the temporal stride to one. We refer to this scenario as *w/o*.

Architecture Details. Our CNN architectures consist of one initial convolutional layer with eight feature maps. We use a kernel stride of four and two for the temporal and spatial dimensions, respectively, in the first layer for downsampling. The initial convolutional layer is followed by three DenseNet blocks with a growth rate of six and three layers each. We connect the DenseNet blocks with transition layers and use a kernel and stride of four and two for the temporal and spatial dimensions, respectively, for the average pooling layer. We use a kernel size $k = k_w \times k_h \times k_t$ for our spatio-temporal convolutions with $k_t = 5$ and $k_w = k_h = 3$ for the temporal and spatial dimensions, respectively. Note for our ST-2DCNN approach, we use a similar architecture but use 2D spatio-temporal kernels, i.e., $k = k_w \times k_t = 3 \times 5$. For our TC-2DCNN approach, we use 2D spatial kernels $k = k_w \times k_h = 3 \times 3$. To reduce the computational efforts when

learning from the entire image sequence, we downsample the input images to a spatial size of 32×32. All methods are implemented in PyTorch.

Training. We train our methods for 300 epochs with a learning rate of $\mathrm{lr} = 0.001$ and use the MAE loss function between the ground truth C and estimated gelatin concentration $\widetilde{C}$. During training, we evaluate the performance of a network every ten epochs on the validation set and use the best network for our final evaluation on the test dataset. We normalize our inputs to have a zero mean and standard deviation of one. We use the Adam optimizer [242] with a batch size of $m_b = 15$.

Performance Measure - Evaluation and Metrics

We apply a two-fold cross-validation approach to test our different methods on previously unseen gelatin phantoms. For each fold, we leave out the data of one gelatin phantom of each concentration for testing and use 75% of the remaining data for training and 25% of the data for validation. Also, we evaluate our best performing approach in the case where the evaluated concentration is left out during training. Here, we perform a five-fold cross-validation approach and leave out the data of an entire concentration for testing, and use 75% of the remaining data for training and 25% of the data for validation. Note, we do not perform cross-validation on the border concentrations, i.e., $C = 4.2$ p.p. and $C = 11.1$ p.p., to avoid out of distribution predictions, i.e., extrapolation. For evaluation, we consider the regression metrics MAE, rMAE, and pCC. We test for significant differences in the median of the MAE of our methods using the Wilcoxon signed-rank test with a significance level of $\alpha = 5\%$.

Results

Performance metrics for our approaches are reported in Table 6.20 for the scenario where all concentrations are present during training. Considering the entire range of concentrations, our results show that our deep learning methods significantly ($\mathrm{p} < 0.05$) outperform the conventional machine learning methods that use shear wave velocity as an explicit feature. Using shear wave velocity as an explicit feature leads to similar performance for all methods, while linear regression performs worst and SVR-RBF performs best. Consistently across all metrics, our ST-3DCNN approach performs best, followed by our ST-2DCNN approach, TC-2DCNN performs the worst. Our results also show that the performance of our networks without temporal pooling (w/o) is reduced compared to our networks with temporal pooling.
Estimated gelatin concentrations and MAE w.r.t. gelatin concentration for the different methods are given in Figure 6.34 and Figure 6.35, respectively. Our approaches that use shear wave velocity as an explicit feature show a low MAE for $C < 5.6$. p.p.. However, the MAE increases substantially with an increasing gelatin concentration. Similar, the MAE increases notably for the higher concentrations using our ST-2DCNN approach that uses the lower dimensional image representation. Considering the results for the higher concentrations $C > 5.6$. p.p., our approach ST-3DCNN still shows good performance and performs best.

For the lower concentrations, there is no substantial difference between our deep learning methods. Furthermore, we evaluate our best performing approach ST-3DCNN, where the evaluated gelatin concentration is left-out during training, see Figure 6.36. The results demonstrate unknown gelatin concentrations can be estimated with a MAE$<$ 1 p.p. for $C < 8.3$ p.p.. However, the performance decreases compared to the scenario where all concentrations are present during training, especially for the stiffer phantoms. Lastly, inference times and the number of parameters for our deep learning approaches are given in Table 6.21.

Table 6.20: Results for gelatin estimation for the different methods. w/o refers to a network without temporal pooling.

	Input	MAE [p.p.]	rMAE	pCC [%]
LR	$v \in \mathbb{R}_+$	1.16 ± 1.22	0.45 ± 0.47	76.15
SVR-RBF	$v \in \mathbb{R}_+$	0.92 ± 1.02	0.35 ± 0.39	84.95
MLP-250	$v \in \mathbb{R}_+$	0.96 ± 1.01	0.37 ± 0.39	84.44
MLP-500	$v \in \mathbb{R}_+$	0.93 ± 1.01	0.36 ± 0.40	85.44
ST-2DCNN	(2D) 1D+t	0.58 ± 0.78	0.22 ± 0.30	92.78
ST-2DCNN w/o	(2D) 1D+t	0.75 ± 0.92	0.29 ± 0.35	89.11
TC-2DCNN	(3D) 2D+t	0.65 ± 1.03	0.25 ± 0.40	88.15
ST-3DCNN	**(3D) 2D+t**	$\mathbf{0.44 \pm 0.50}$	$\mathbf{0.17 \pm 0.19}$	**96.76**
ST-3DCNN w/o	(3D) 2D+t	0.93 ± 1.03	0.36 ± 0.40	84.96

Table 6.21: Number of parameters (#numParam.) for our deep learning approaches and inference time evaluated on an nvidia TITAN RTX.

	inference time [ms]	#numParam.
ST-2DCNN	$\mathbf{1.92 \pm 0.02}$	**18 857**
ST-2DCNN w/o	1.95 ± 0.07	**18 857**
TC-2DCNN	1.93 ± 0.03	47 537
ST-3DCNN	2.19 ± 0.06	65 753
ST-3DCNN w/o	5.43 ± 0.03	65 753

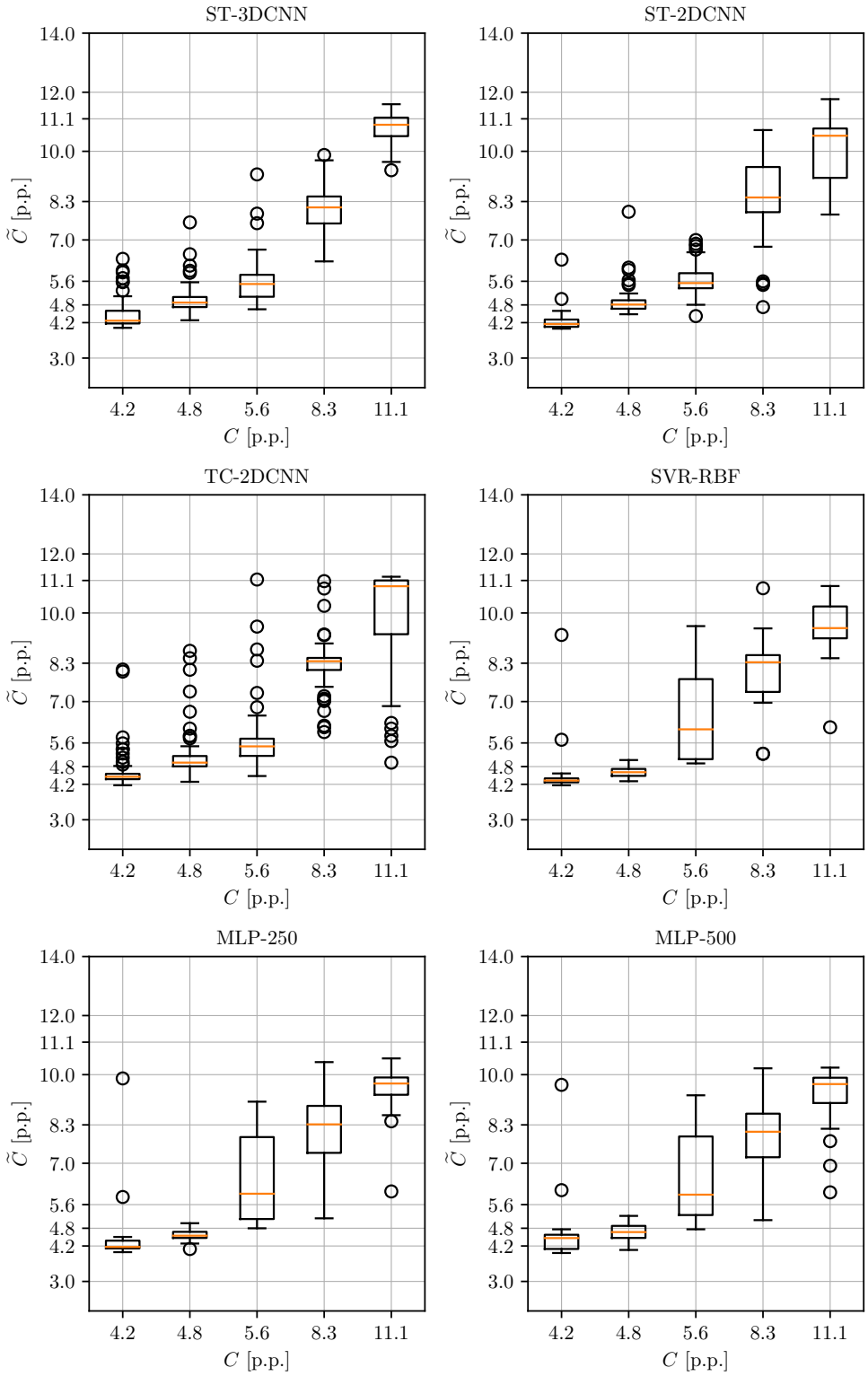

Figure 6.34: Boxplots of the estimated gelatin concentrations for our different methods.

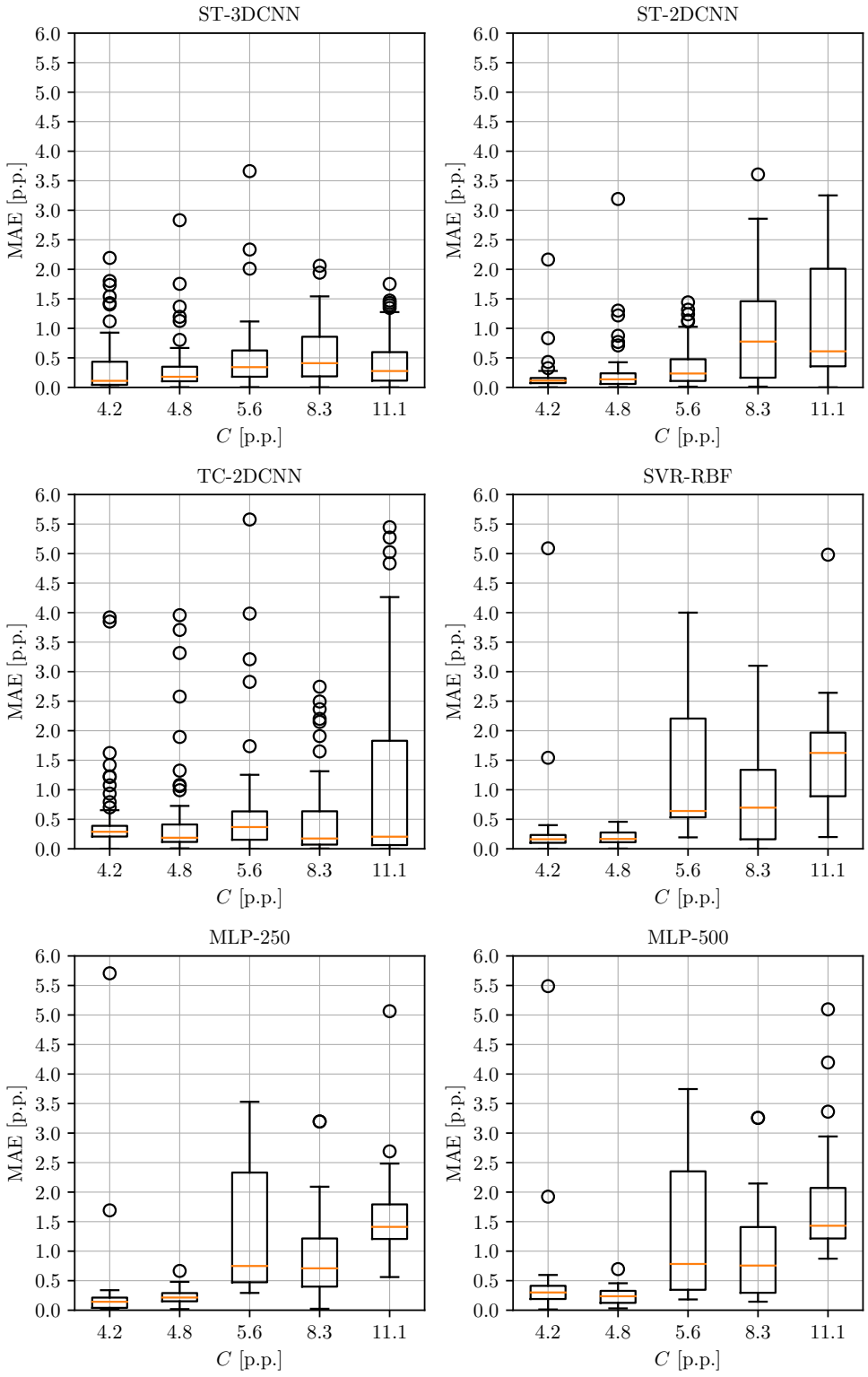

Figure 6.35: Boxplots of the MAE of the estimated gelatin concentrations using our different methods.

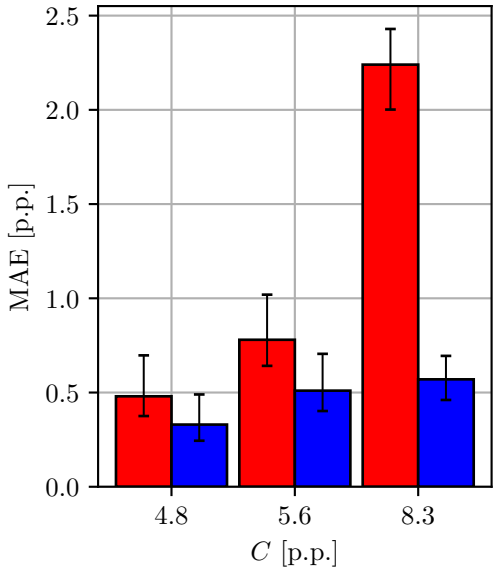

Figure 6.36: MAE with 95 % confidence intervals over gelatin concentration. (Red) ST-3DCNN where evaluated elasticity is left out during training; (Blue) ST-3DCNN where evaluated elasticity is also present during training.

Discussion

In this experiment, we study our spatio-temporal deep learning approach to estimate material properties end-to-end from sequences of 2D OCT images that capture shear wave propagation over time. For comparison, we consider conventional regression methods combined with feature extraction. As a feature for this task, we consider shear wave velocity estimated from the image data with a conventional ToF approach. Our results show that the estimated shear wave velocities are affected by noise, and differentiating the gelatin concentration with $C > 8.3$ p.p. seems to be difficult, see Figure 6.32. This highlights once again that shear wave velocity estimation itself is already a challenging task influenced by image artifacts and measurement noise. Naturally, noisy shear wave velocity estimations also affect the performance of methods that use shear wave velocity as an explicit feature. Our results show that conventional regression methods can lead to good results for the softer phantoms, i.e., $C = 4.2$ p.p. or $C = 4.8$ p.p.. However, estimating the gelatin concentration for the stiffer phantoms becomes difficult. In fact, none of our conventional regression approaches shows good performance for $C = 11.1$ p.p., see Figure 6.34. This can be explained by the shear wave velocity estimation results, where the shear wave velocity estimations for $C > 4.8$ p.p. notably overlap. These results confirm our assumption that end-to-end processing of the data should be performed to overcome limitations and errors of the feature extraction process.
Our ST-2DCNN approach, which uses the same data representation that was used for shear wave velocity extraction, already demonstrates improved performance. This highlights that more robust and effective features can be learned end-to-end from data. The performance is increased even further with our ST-

3DCNN approach that uses the entire sequences of image data and where joint spatio-temporal feature learning is performed. While our deep learning methods perform similar for the lower concentrations, ST-3DCNN outperforms all other methods for the higher concentrations. Across our different deep learning methods, we find that TC-2DCNN performs worst similar to our experiments in Section 6.2.1. These results further confirm our findings that spatio-temporal processing with such an approach is not ideal for learning the complex and rich features of shear wave propagation. We further study temporal processing with our ST-2DCNN and our ST-3DCNN approach and evaluate the effect of temporal pooling. This becomes relevant due to the large number of OCT images of a sequence ($n_t = 400$). We find that temporal pooling is essential to achieve good performance. ST-3DCNN without temporal pooling with a MAE of 0.93 ± 1.03 p.p. performs worse than our conventional regression methods, see Table 6.20. One possible explanation is that temporal pooling might act as regularization and hence improves performance. Also, temporal pooling increases the effective receptive field more quickly, which could lead to the performance improvements considering that the sequences consist of several hundreds of images. In summary, we find that our spatio-temporal deep learning approaches outperform the conventional pipeline, and ST-3DCNN performs best, clearly outperforming all other methods for stiffer phantoms. Hence, considering our second research question of this work, these results highlight the benefit of joint spatio-temporal feature learning in an end-to-end fashion using our ST-3DCNN approach. However, our results also show that using ST-2DCNN with a lower-dimensional space time map representation is also a viable approach. Further evaluating and comparing both methods using a large-scale dataset that also contains additional elasticities is an interesting direction for future work. We also evaluate ST-3DCNN to interpolate between gelatin concentrations, see Figure 6.36. However, our results show that this is a more challenging scenario, and performance decreases compared to the case where the evaluated gelatin concentration is also present during training. Notably, the performance decreases substantially with increasing gelatin concentration, i.e., faster shear wave velocity, similar to the conventional shear wave velocity estimation. This highlights the difficulty of the task and indicates that there there are complex spatio-temporal relationships that are difficult to generalize. These spatio-temporal relationships are likely influenced by artifacts that result from the acquisition process of OCT images, where an image is generated line-by-line. Still, when present during training, these relationships can be well captured with our ST-3DCNN approach. Further studying aspects w.r.t. data acquisition is an interesting direction for future work.

Overall, considering our first research question, our results show that using shear wave velocity as an explicit feature for tissue characterization is challenging, and that spatio-temporal deep learning presents a promising and simpler alternative to the conventional signal processing pipeline. Considering our second research question, performing this task in an end-to-end fashion directly from the sequence of image data, and learning spatio-temporal features jointly turns out to be most effective, especially for stiffer elasticities and resulting faster shear waves.

6.2.4 4D OCT-based Elasticity Imaging

In this experiment, we address elasticity estimation from a sequence of volumetric OCT images using reverberate or diffuse wave fields. This is motivated by the advantages that result from such an approach, such as a higher ratio of shear waves in tissue and elasticity estimation independent of the wave propagation direction [549]. Parts of this section have been published in our study presented in [324]. The study [324] was a collaborative study, development of the experimental setup and data acquisition were performed by M. Neidhardt. The aspects regarding spatio-temporal deep learning were developed and evaluated by the author of this thesis and are in the focus of this work.

Considering our first research question of this work, we study whether end-to-end elasticity estimation can be performed with our spatio-temporal deep learning approaches using diffuse wave fields and 4D spatio-temporal OCT data. Using such data that captures wave propagation entirely over space and time, we also contribute to the questions of whether estimations can be performed independent of the wave propagation direction and whether elasticities can be estimated that have not been used during training despite the challenging scenario.

Considering our second research question, we present and compare several different approaches that are based on our spatio-temporal deep learning architectures. We analyze how spatio-temporal feature learning can be performed, and evaluate to estimate elasticity from entire 4D OCT sequences using different concepts for processing the data.

Definition of the Learning Task

For our study, we consider tissue-miking gelatin phantoms with varying stiffness and consider gelatin concentration $C \in \mathbb{R}_+$ as surrogate value for material properties. We try to estimate material properties end-to-end from a sequence of 3D OCT images $x_t = [x^{[0]}, x^{[1]}, \ldots, x^{[n]}]$ with $x^{[i]} \in \mathbb{R}^{n_h \times n_w \times n_d \times n_c}$ which capture displacement over time resulting from a reverberate wave field. Formally, we try to learn a function $f_{C_1} \colon \mathbb{R}^{n_h \times n_w \times n_d \times n_t \times n_c} \to \mathbb{R}$.

Experimental Setup and Data Sets

For evaluation of our methods, we use a dataset consisting of volumetric OCT images over time that capture wave propagation in tissue-mimicking gelatin phantoms. We use data from homogeneous phantoms with varying stiffness that results from different gelatin to water ratios. The full dataset consists of gelatin phantoms with gelatin concentrations $C \in [5, 7.5, \ldots, 20.0]$ given in p.p.. For each concentration, we use data from six phantoms. The dataset is acquired with an experimental setup shown in Figure 6.37. It consists of a high-speed swept-source OCT system (OMES, OptoRes), a clinical needle attached to a piezoelectric actuator, and a robot (H-820.D1, Physik Instrumente). For data acquisition, C-scans are acquired with a temporal rate of 831 Hz and volume size of $3 \times 3 \times 2$ mm^3

in air ($32 \times 32 \times 470$ voxels) along the x, y, and z-axis, respectively. For each phantom, a sequence of volumetric images is acquired at the 52 positions shown in Figure 6.37. At each FOV position, a sequence of $n_t = 90$ OCT volumes is recorded, resulting in 4D spatio-temporal data of the wave propagation. Shear wave excitation is performed by vibrating the needle with 100 Hz and using the robot, data is acquired at the different positions indicated in Figure 6.37. The position of the needle remains fixed, i.e., it is not changed w.r.t. a phantom. All our methods are trained and evaluated with the unwrapped phase information of the OCT data and each volume is resized to $32 \times 32 \times 32$ voxels. In summary, the dataset consists of 4D OCT data x_t acquired at different positions w.r.t. the excitation point and the corresponding sequence-level y is given by the gelatin concentration of the phantom.

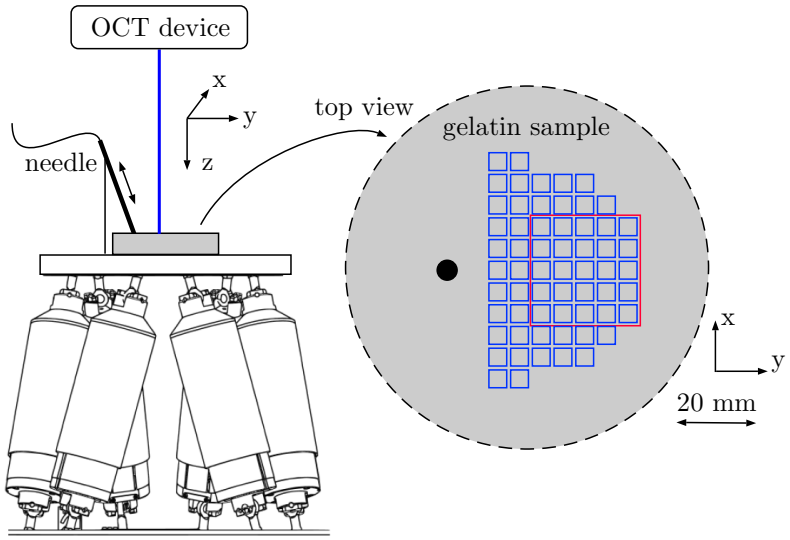

Figure 6.37: Experimental setup for data acquisition of the 4D OCE dataset. Shear waves are generated with a vibrating needle interested into a phantom. Data acquisition is performed at multiple phantom positions indicated by a blue square. The position of the needle is fixed. For the development and comparison of our approaches, we consider data within the red rectangle. Figure adapted from [324].

Methods

So far, there is very little work that considers deep learning and 4D OCE as outlined in Section 5.2.3. The complex and diffuse wave field and artifacts that might result from OCT data acquisition combined with the 4D data structure raises the question of how spatio-temporal feature learning can be performed. To this end, we evaluate a range of our spatio-temporal deep learning concepts visualized in Figure 6.38. We rely on our CNN architecture concept, and we use the idea of DenseNet [210] as our backbone architecture. Specific details of our approaches are given in the following paragraphs.

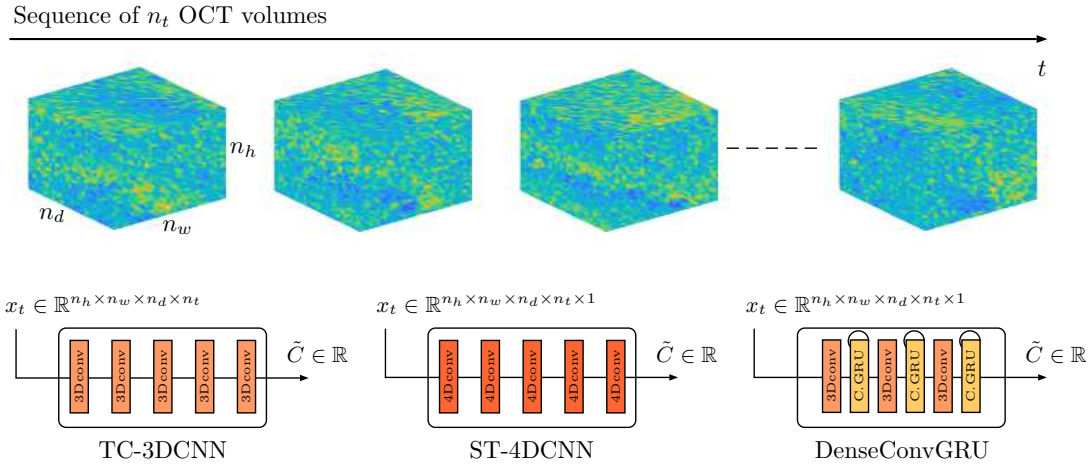

Figure 6.38: Our deep learning approaches estimate material properties in an end-to-end fashion using 4D OCT data that captures displacement over time resulting from a wave field. For the processing of an entire 4D sequence, we present and compare three different spatio-temporal deep learning approaches. TC-3DCNN uses the channel dimension of the input as a temporal dimension. ST-4DCNN performs joint spatio-temporal feature learning by means of convolutions, and DenseConvGRU performs spatio-temporal feature learning by means of spatio-temporal recurrence. The output of our methods is indicated by $\tilde{C}$. Note that our actual images are gray-scale and colored inputs are only shown for improved visualization.

TC-3DCNN. First, we try to learn from the 4D spatio-temporal data with our simple baseline approach, where we use the channel dimension of the input as the temporal dimension. To this end, we use our CNN architecture concept with 3D operations, and the input of this approach is $x_t \in \mathbb{R}^{n_h \times n_w \times n_d \times n_t}$.

ST-4DCNN. Second, we use our approach of a 4D spatio-temporal CNN that jointly learns features from the temporal and spatial dimensions by means of 4D spatio-temporal convolutions and pooling operations. To this end, we use our CNN architecture in a 4D fashion. The input of this approach is $x_t \in \mathbb{R}^{n_h \times n_w \times n_d \times n_t \times 1}$.

DenseConvGRU. Third, we consider our hybrid approach of a CNN and ConvGRU modulus at different scales. We systematically study and compare a ConvGRU module placed at different feature levels and place a module at the front, middle, or end of the architecture. Also, we evaluate a ConvGRU placed at all positions. We use this approach in a many-to-one fashion, i.e., we process the entire sequence and output a single value for the gelatin concentration. The input of this approach is also $x_t \in \mathbb{R}^{n_h \times n_w \times n_d \times n_t \times 1}$.

Architecture Details. Our architectures consists of one initial convolutional layer with 16 feature maps, followed by our DenseNet architecture, which consists of three DenseNet blocks with a feature growth rate of 12. For each DenseNet blocks, we use four convolutional layers. The DenseNet blocks are connected

with transition layer that downsample the spatial dimensions by a factor of two. To account for the increased number of parameters and complexity based on the ConvGRU modules, we use three instead of four convolutional layers for our DenseNet blocks. All methods are implemented in PyTorch.

Training. The dataset consists of sequences with an overall length of 90 volumes. To augment our training data, we randomly crop sub-sequences from the entire OCT sequences of the dataset. As a baseline, we use a sub-sequence length of $n_t = 10$ and also evaluate longer sequence length for our best performing approach. We normalize our inputs to have a zero mean and standard deviation of one. We train our network for 300 epochs with a batch size of $m_b = 10$ and a learning rate of $\mathrm{lr} = 0.001$ using Adam [242] for optimization combined with a MSE loss function between our estimations $\widetilde{C}$ and the target labels C. After 150 epochs, we reduce the learning rate by a factor of two every ten epochs.

Performance Measure - Evaluation and Metrics

We apply a three-fold cross-validation approach to test our different methods on previously unseen gelatin phantoms. Given data from six phantoms for each concentration, in each fold, we leave out the data of one phantom for testing and one for validation. Moreover, we also test our best performing approach in the case where the evaluated concentration is left-out during training. Here, we perform five-fold cross-validation, and in each fold, we leave out the data of an entire concentration. Note, we do not perform cross-validation on the border concentrations, i.e., $C = 5$ p.p. and $C = 20$ p.p., to avoid out of distribution estimations. Hence, this leads to five folds for our cross-validation, and in each fold, we use three phantoms for testing and three phantoms for validation. Note that we strictly split our validation and test phantoms, i.e., we do not use the same phantom for validation and testing across different folds or training strategies. For the comparison of our methods, we consider data acquired in a fixed ROI highlighted in Figure 6.37. We also evaluate and train our best performing approach on data from all positions. For evaluation, we consider the regression metrics MAE, rMAE, and pCC. We test for significant differences in the median of the MAE of our methods using the Wilcoxon signed-rank test with a significance level of $\alpha = 5\%$.

Results

We report performance metrics for our methods in Table 6.22. Comparing the different methods, our approach DenseConvGRU-End significantly ($\mathrm{p} \leq 0.05$) outperforms all other approaches with a MAE of 0.63 ± 0.87 p.p.. We report the estimated gelatin concentration and the MAE w.r.t. the different gelatin concentrations in Figure 6.39 and 6.40, respectively. Our results show that the MAE increases substantially for $C \geq 10$ p.p. for all our methods. Still, DenseConvGRU-end shows good performance and outperforms all other methods across the entire range of concentrations. Inference times and the number of parameters of the different methods are given in Table 6.23. The fastest approach is TC-3DCNN,

with an inference time of around 3 ms. The slowest approach is ST-4DCNN, with an inference time of around 45 ms.
Furthermore, we evaluate the impact of additional temporal information using our best performing approach DenseConvGRU-End, see Table 6.24. We consider averaging the estimations of multiple sub-sequences with a length of $n_t = 10$ volumes during evaluation or by training our approach end-to-end with longer sequences. Our results demonstrate that using an increased sequence length improves the performance, except for training with $n_t = 60$. Our results show that averaging sub-sequences is more effective than directly training the network with longer sequences. Note, we do not consider training the network end-to-end with $n_t = 90$ due to GPU memory limitations.
Moreover, we evaluate our approach DenseConvGRU-End in the scenario where the evaluated concentration is left-out during training. A performance comparison to the scenario where the concentration is also present during training is given in Figure 6.41. Our results also show that predicting concentrations that are not considered during training is difficult with a substantially increased MAE.
Furthermore, we evaluate our DenseConvGRU-End on phantoms without wave excitation to exclude that our network uses other features than the wave propagation and report the performance in Table 6.25. Our results show that gelatin concentration estimation becomes unfeasible without wave excitation, and the network estimates approximately the same concentration for all tested phantoms with different concentrations.
Lastly, we train our DenseConvGRU-End with data from the entire phantom and report the spatial distribution of the MAE in Figure 6.42. Our results show that estimations can be performed at all spatial locations with a tendency for a higher MAE close to the border of the phantom. However, overall, our results show an increased MAE at inconsistent positions for the different concentrations.

Table 6.22: Performance comparison of the different spatio-temporal deep learning methods for shear wave elastography from sequences of volumetric OCT data. Experiments are performed with a sequence length of $n_t = 10$.

Method	MAE [p.p.]	rMAE [p.p.]	pCC [%]
TC-3DCNN	1.19 ± 1.18	0.23 ± 0.24	94.45
ST-4DCNN	0.88 ± 1.05	0.18 ± 0.21	96.31
DenseConvGRU-Front	0.97 ± 1.02	0.19 ± 0.21	96.28
DenseConvGRU-Middle	0.79 ± 0.87	0.16 ± 0.17	97.30
DenseConvGRU-End	$\mathbf{0.63 \pm 0.80}$	$\mathbf{0.13 \pm 0.16}$	**97.96**
DenseConvGRU-All	0.85 ± 1.09	0.17 ± 0.22	96.12

Table 6.23: Inference time and number of parameters (#numParam.) for our different spatio-temporal deep learning methods. Experiments are performed with a sequence length of $n_t = 10$. For our evaluations, we use an nvidia TITAN RTX.

Method	inference time [ms]	#numParam.
TC-3DCNN	$\mathbf{2.71 \pm 0.08}$	**246 099**
ST-4DCNN	44.02 ± 0.93	6163 23
DenseConvGRU-Front	8.05 ± 0.08	459 249
DenseConvGRU-Middle	21.82 ± 0.24	695 205
DenseConvGRU-End	23.11 ± 0.17	1 079 052
DenseConvGRU-All	32.76 ± 0.14	1 361 736

Table 6.24: Evaluation of the impact of the temporal dimension on performance. Experiments are performed for our approach DenseConvGRU-End. Avg. refers to dividing a sequence into sub-sequence lengths with a $n_t = 10$ and averaging the individual results afterwards. Full. refers to training the network with the entire sequence length. Note, we do not train our network with the full length of $n_t = 90$ volumes due to memory limitations.

	#vols	MAE [p.p.]	rMAE	pCC [%]	inference time [ms]
	10	0.63 ± 0.87	0.13 ± 0.16	97.96	$\mathbf{23.11 \pm 0.17}$
Avg.	30	0.60 ± 0.77	0.12 ± 0.15	98.10	51.30 ± 0.17
	60	0.57 ± 0.74	0.11 ± 0.15	98.27	97.33 ± 0.32
	90	$\mathbf{0.57 \pm 0.73}$	$\mathbf{0.11 \pm 0.15}$	**98.31**	147.96 ± 0.99
Full.	30	0.58 ± 0.80	0.12 ± 0.16	98.05	56.0 ± 0.36
	60	0.77 ± 1.03	0.15 ± 0.21	96.70	109.34 ± 0.61

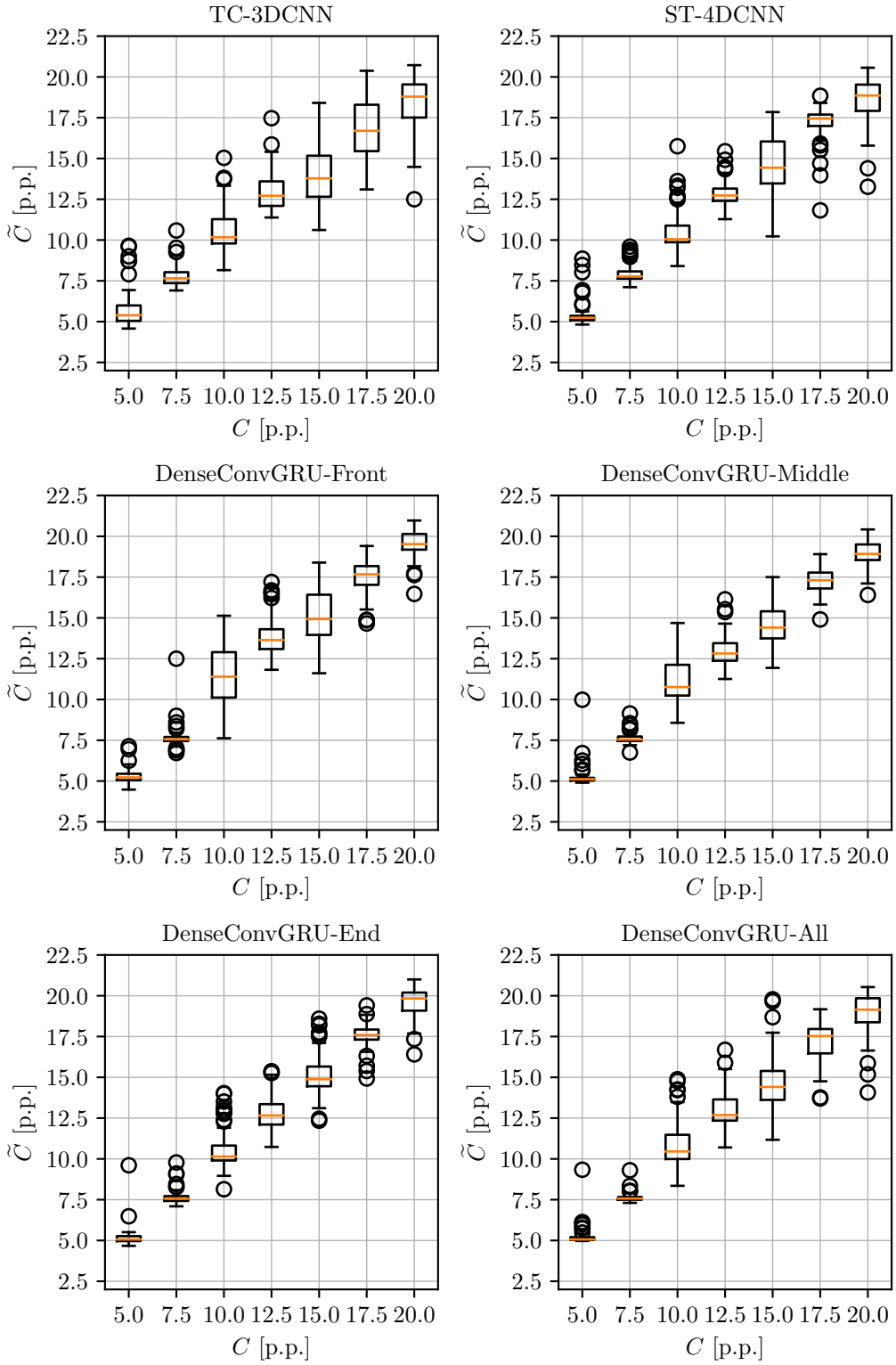

Figure 6.39: Boxplots of the estimated gelatin concentrations for our different methods.

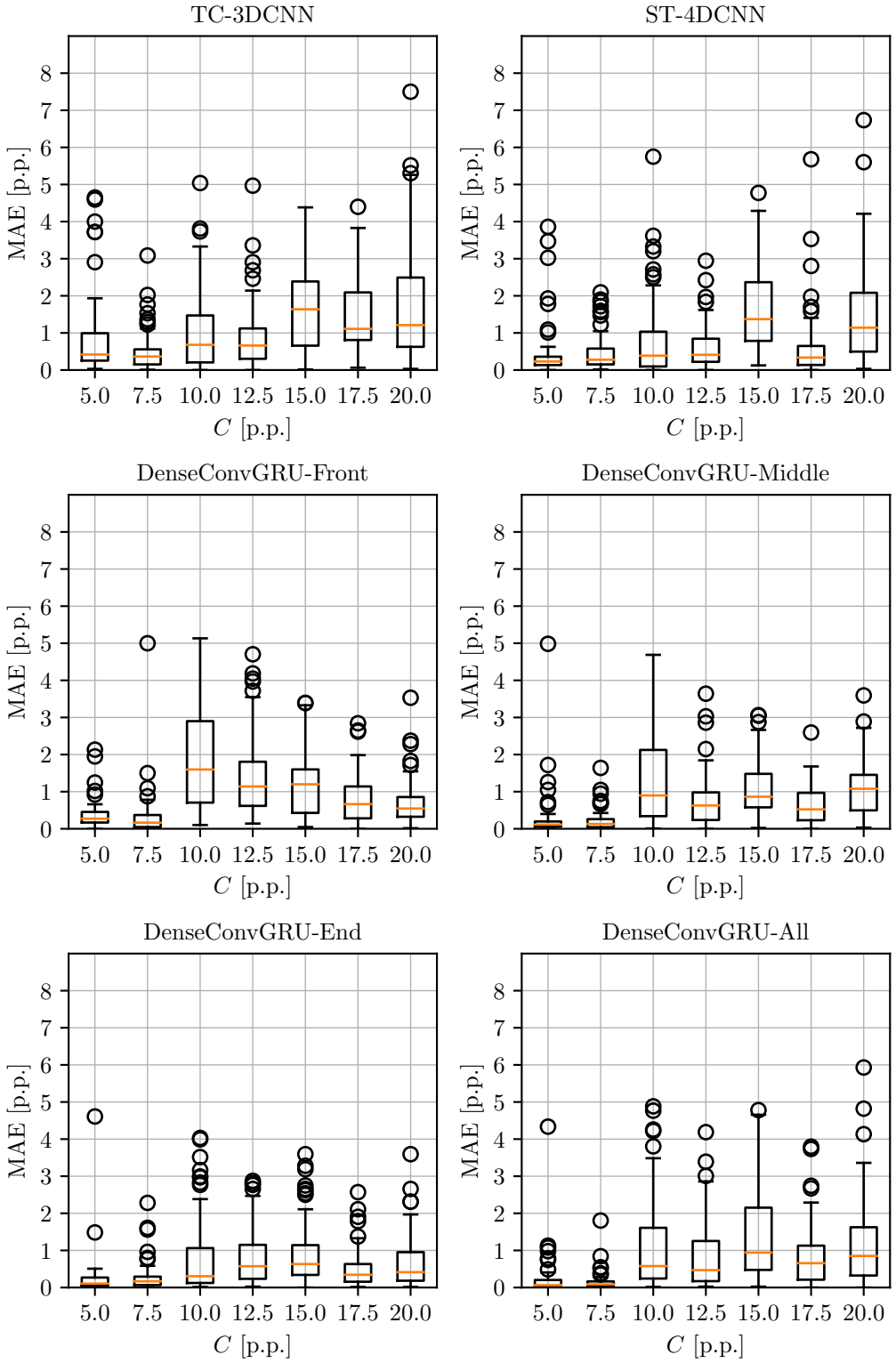

Figure 6.40: Boxplots of the MAE of the estimated gelatin concentrations using our different methods.

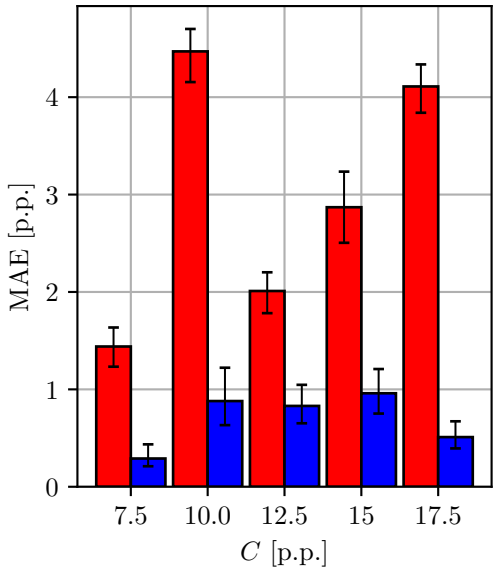

Figure 6.41: MAE with 95 % confidence intervals over gelatin concentration. (Red) 3DCNN where evaluated elasticity is left out during training; (Blue) 3DCNN where evaluated elasticity is also present during training.

Table 6.25: Evaluation of our approach DenseConvGRU-End on phantoms without wave excitation.

Concentration [p.p.]	7.5	12.5	17.5
Mean Estimation [p.p.]	15.13 ± 0.9	15.34 ± 0.56	15.36 ± 0.98
Mean Estimation Error [p.p.]	7.63 ± 0.9	2.84 ± 0.56	2.15 ± 0.97

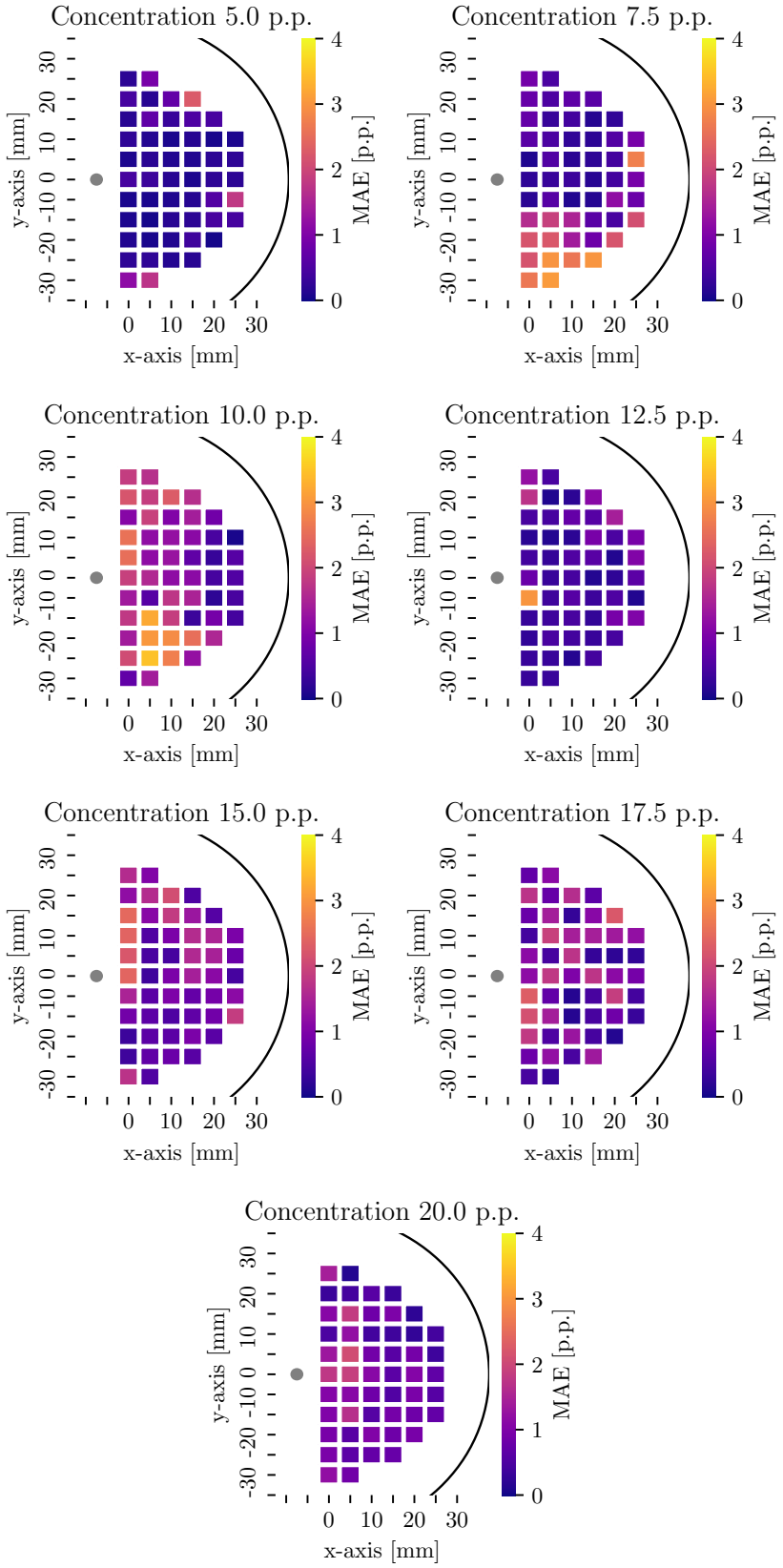

Figure 6.42: MAE for the different positions and gelatin concentrations using our approach DenseConvGRU-End. The gray circle indicates the excitation position. Figure adapted from [324].

Discussion

In this experiment, we study end-to-end elasticity estimation using sequences of volumetric OCT data that capture the 4D spatio-temporal relationships of wave propagation over time. So far, there is very little work that considers deep learning and 4D OCE as outlined in Section 5.2.3. To address the learning task, we study and compare six different spatio-temporal deep learning methods using our compact CNN architecture concepts and estimate gelatin concentration as a surrogate for elasticity.
We assume that the volumetric OCT images and the spatio-temporal relationships between the volumes of a sequence are affected by artifacts. These artifacts likely result from the changing wave field during data acquisition of a volumetric OCT image. To avoid such artifacts, previous approaches used multiple excitations and synchronized data acquisition such that a volumetric image appears as if all A-scans were acquired at the same time with an apparent frame rate equal to the A-scan rate of the OCT system [547,549]. However, this results in substantial data acquisition times, e.g., 60 s per data acquisition [549]. Instead, our data is acquired without synchronized data acquisition or repeated excitations and hence can be acquired in real-time.

Comparing our approaches highlights that DenseConvGRU-End performs best and that TC-3DCNN performs worst, see Table 6.22. The former approach shows good performance across the entire range of concentrations, see Figure 6.40. While this approach performs best, it indicates that first, learning a compact and robust spatial representation, followed by joint spatio-temporal feature learning, is beneficial in this challenging scenario and outperforms the inverted approach as well as joint spatio-temporal feature learning throughout the entire architecture. Our results in Figure 6.41 highlight that DenseConvGRU-end generalizes well between different phantoms and resulting speckle distributions. This is an important property and represents a relevant scenario for the clinical application, as relevant tissues and elasticities can be included in the training data. However, our results in Figure 6.40 show that estimating higher concentrations $C \geq 10$ p.p. leads to notably increased MAE for all our methods. This becomes even more severe when the evaluated concentration is left-out during training, see Figure 6.41. This suggests complex spatio-temporal characteristics influenced or even defined by artifacts that are difficult to generalize, especially for faster shear wave propagation. We assume that the characteristics are dependent on the acquisition rate and scan pattern of the OCT imaging as well as the type of excitation. However, our findings indicate that such spatio-temporal relationships can be learned and used for end-to-end estimation of elasticity with our approach. Evaluating and studying these aspects w.r.t. different data acquisition strategies is an interesting direction for future work. To ensure that our networks only rely on displacement information resulting from wave propagation, we also evaluate DenseConvGRU-end on phantoms without wave excitation. Here, our network fails to estimate the concentration, which demonstrates the dependency on displacement information, see Table 6.25.

We also study the spatial distribution of the MAE w.r.t. the wave excitation, see Figure 6.42. Our results show no clear dependency between MAE and the relative position. This indicates that estimations can be performed independently of the location relative to the excitation point and that single estimations are affected by localized phantom inclusions or inhomogeneities. Overall, our findings suggest that even complex and abstract spatio-temporal relationships influenced by artifacts can be learned and utilized by our spatio-temporal deep learning approaches for end-to-end elasticity estimation using 4D OCT data.
We further evaluate the impact of the temporal dimension, i.e., number of volumes, on the performance using our best performing approach DenseConvGRU-End, see Table 6.24. Here, we find that using an increased sequence length can improve the performance and consistency, but this comes at the cost of substantially increased inference times. Our results also highlight that performance even decreases for $n_t = 60$ when trained in an end-to-end fashion. We assume that this behavior results from reduced variance during training that results from the longer sequences that are used for random cropping during training. Visa versa, these results indicate that our random cropping approach during training with a sequence length of $n_t = 10$ is an effective method to increase the variance during training. Taking these findings into account, our results suggest that using a few volumes such as $n_t = 10$ and performing our random cropping approach during training, combined with our architecture DenseConvGRU-End is a good starting point for future studies.

Overall, considering our first research question, our results show that elasticity estimation can be performed end-to-end with our spatio-temporal deep learning approaches using 4D OCT data, despite the challenging scenario that results from the diffuse wave fields and the data acquisition process of the OCT images. Our findings show that elasticity estimations can be performed in real-time and independent of the measuring position relative to the excitation point. However, interpolating between elasticities is difficult, and including relevant elasticities during training is recommended to achieve high performance. Future work could evaluate this aspect in more detail by including additional phantoms and gelatin concentrations in the training data. Considering our second research question and comparing different spatio-temporal deep learning methods, we find that first learning compact spatial representations and performing spatio-temporal processing afterwards, using DenseConvGRU-end, turns out to perform well in this challenging scenario.

6.2.5 Summary

In the previous experiments, we investigated whether biomechanical properties can be estimated end-to-end with spatio-temporal deep learning from sequences of US and OCT images that capture wave propagation. In this context, we focused on evaluating different spatio-temporal deep learning approaches that can be used for both 3D and 4D data processing.

Regarding our first research question of this work, whether multi-dimensional sequences of medical image data can be processed effectively with deep learning, our results demonstrated that data processing with spatio-temporal deep learning could be performed effectively for dynamic elastography, i.e., end-to-end and in real-time. Our results also highlighted that our spatio-temporal CNN architecture concept could be used for OCT and US data and for 2D and 3D image data. That is, a similar concept can be shared for different modalities and dimensionalities. Thereby, the data processing of medical image sequences can be simplified compared to the previous approach that requires fine-tuned feature extraction and subsequent model building. In fact, our approach is designed to identify wave patterns in an end-to-end fashion without handcrafted feature extraction or physical model. Our results showed that estimations with such an approach could be accomplished. We showed that estimations could be performed independent of the position relative to the excitation point and that our approaches generalized well to different phantoms. However, we observed a decrease in performance when estimating completely unknown gelatin concentrations that were not present during training, particularly for higher concentrations and faster wave velocities. Across several experiments, our findings showed that our spatio-temporal deep learning approach outperformed the conventional approach in terms of performance, flexibility, and consistency. Overall, these findings indicate that the complex end-to-end relationships to estimate elastic properties from multi-dimensional medical image sequences can be learned with spatio-temporal deep learning.

Considering our second research question, how spatio-temporal feature learning can be performed with deep learning, we find that using the entire spatio-temporal data is beneficial for dynamic elastography and performs better than, e.g., using a lower-dimensional space-time map representation which is commonly used by conventional approaches. Similar, our results highlighted that learning spatio-temporal features throughout the architecture by means of spatio-temporal convolutions is superior compared to a simple baseline approach where the channel dimension is considered as temporal dimension. This highlights the value of joint spatio-temporal feature learning and processing for the task. In case of diffuse wave fields and abstract spatio-temporal patterns, our results showed that learning a compact spatial representation first, followed by joint spatio-temporal processing, can improve the performance and consistency. We also demonstrated that localized estimation of elastic properties could be performed with our patch-wise training strategy. Overall, our findings show that learning end-to-end from entire image sequences is beneficial for dynamic elastography and that joint spatio-temporal feature learning is recommended. Furthermore, our results highlight that a spatio-temporal CNN architecture is well suited to perform end-to-end regression of material parameters from sequences of multi-dimensional medical image data.

Chapter 7

Discussion

In this chapter, we discuss our results and the findings of this work from a more general perspective in the context of our research questions. We highlight the scientific contributions of the study and bring together the findings of our experiments. First, we consider processing sequences of medical image data with spatio-temporal deep learning from a general perspective and address whether multi-dimensional sequences of medical image data can be processed effectively with such an approach. In this part, we focus on the general findings regarding spatio-temporal deep learning. Second, we focus on the different spatio-temporal deep learning concepts and discuss how multi-dimensional spatio-temporal feature learning can be performed. Third, we discuss aspects related to training data and highlight interesting directions for future work.

7.1 Processing Sequences of Medical Image Data

Processing sequences of medical image data is essential in many clinical scenarios ranging from image-guided interventions to computer-assisted diagnosis. In this work, we focus on end-to-end motion analysis and elasticity estimation using spatio-temporal deep learning and sequences of US and OCT data. From a machine learning perspective, motion analysis and dynamic elastography require estimating a target value from a sequence of images that capture spatial and temporal relationships. Considering motion analysis, the target value represents, e.g., the current position of a target or even the future position. The latter is beneficial or even required to address system latencies, resulting in a lag between the estimation and the target's actual position. Considering dynamic elastography, the target values represent elastic properties of tissue such as the shear wave velocity, the Young's modulus, or direct classification of different tissue types [267]. Fast volumetric imaging can be performed with imaging modalities such as US or OCT. Thus, spatio-temporal features of a dynamic process can be captured in three-dimensional space and time.
Considering our first research question of this work, we study whether sequences of medical image data can be processed effectively with an end-to-end spatio-temporal deep learning approach. That is, whether competitive performance can be achieved, whether real-time constraints can be met, and whether a similar deep learning concept can be shared across different tasks and imaging modalities. The latter is beneficial to reduce network design efforts for individual tasks

and imaging modalities, which can shift the problem of handcrafting features to handcrafting specific architectures. For example, comparing deep learning approaches of motion analysis such as SiamFC [47], and dynamic elastography such as DSWE-Net [3], it stands out that two very distinctly different architectures are used. Thus, an effective deep learning approach for analyzing multi-dimensional medical image sequences that is compact and unified would be beneficial. In this way, only minor architecture adaptations are required to address different clinical problems and imaging modalities as well as dimensionalities. As a result of Chapter 4, we presented such a unified spatio-temporal CNN architecture concept for end-to-end regression and evaluated the effectiveness across different application scenarios in Chapter 6.

To evaluate whether complex end-to-end relationships can be learned effectively from spatio-temporal medical image data, we studied motion analysis using 4D OCT data and 4D US data with our spatio-temporal deep learning concept. For this application scenario, volumetric imaging is beneficial or even required due to the inherent three-dimensional nature of targets and motion. A challenge of motion analysis is that it needs to be performed in real-time for different tissue types, appearances, and motion patterns. This brings the challenging task of learning complex end-to-end relationships directly from 4D spatio-temporal data. So far, this has hardly been demonstrated considering motion analysis with OCT [168,171,254,432], or US [98,175,211,282,307,335,367,371,499,506]. We address motion analysis as an end-to-end regression task using our spatio-temporal deep learning approach with sequences of volumetric images as inputs. By performing end-to-end regression, we estimate target motion within a few milliseconds in a single forward pass without performing multiple comparisons. Other approaches typically rely on matching a template image and search images such as SiamFC [47] that is widely adopted for US-based 2D tracking [175, 282, 499]. Notably, such an approach has also been demonstrated using 3D US images with run-times in the range of 300 ms [212].
Our results demonstrated that motion analysis using sequences of volumetric data could be performed efficiently and effectively in an end-to-end fashion and that short-term as well as long-term motion analysis can be performed (Section 6.1). In particular, our findings regarding markerless tracking are promising. For both of our imaging modalities, we demonstrated that after training, tracking could be performed without distinct landmarks, such as vessels, and for new unseen tissue areas or even completely new tissue types. So far, such results have hardly been presented in previous studies that usually rely on landmarks and the same tissue type considering US-based tracking [98]. Thus, our results are an important contribution demonstrating that spatio-temporal deep learning can learn effective features for tracking end-to-end from 4D spatio-temporal data. We demonstrated that even motion forecasting could be performed directly by using and learning the underlying 4D spatio-temporal relationships of sequences of volumetric images (Section 6.1.3). We combine motion estimation and forecasting into a single effective approach. Thus, system latencies can be directly addressed. This can be performed considering the future motion as an additional target during training.

Thus, only the loss function and the output of the network need to be adapted. This contribution allows simplifying otherwise complex data processing pipelines, e.g., during motion compensation in radiotherapy, which involves motion forecasting as an additional problem after the task of motion estimation [138,226,276,451].

As our second application scenario to study whether complex end-to-end relationships can be learned effectively, we considered dynamic elastography (Section 6.2). During dynamic elastography waves of displacements are induced into a tissue, and the resulting wave propagation is then captured spatially and temporally with sequences of images [415]. Afterwards, spatio-temporal relationships of the wave propagation are related to elastic properties of tissue [156]. However, analyzing these features in a hand-crafted fashion is known to be difficult [267]. The conventional data processing pipeline includes many unsolved challenges, including shear wave velocity estimation, and estimation of elasticity parameters such as the Youngs Modulus [7, 236, 547]. Deep learning could be a promising approach to overcome these challenges. However, related work considering US-SWEI heavily relies on simulated training data [3, 464], which introduces a limiting factor for the performance on real data. Considering OCE, so far, very little work considers deep learning and end-to-end dynamic elastography. Using the same spatio-temporal architecture concept as for motion analysis, we address end-to-end shear wave velocity and tissue elasticity estimation by performing a regression task directly from sequences of images. An advantage resulting from our end-to-end regression approach is that we learn the direct relationship between tissue elasticity and wave propagation. Thus, we do not require a specific material model and handcrafted feature engineering. Our results showed that end-to-end elasticity estimation could be performed with our spatio-temporal deep learning approach. Our findings demonstrate that this approach generalizes to unseen elasticities, phantoms, and wave propagation directions. These results indicate that the underlying spatio-temporal relationships of dynamic elastography can be learned effectively with spatio-temporal deep learning.
Furthermore, we present elastography with deep learning and volumetric OCT data (Section 6.2.4). In the future, such an approach could be valuable to perform estimations independent of the wave propagation in all spatial directions. Another advantage of our approach is that we achieve substantially faster runtimes compared to related work considering 4D OCE [549]. We performed 4D OCE without synchronized data acquisition and thus performed data acquisition in one-shot, which likely resulted in diffuse spatio-temporal features due to the sequential data acquisition of OCT. Still, we achieved robust and high performance. These results indicate that robust spatio-temporal feature learning can be performed, despite artifacts that might result from the data acquisition procedures. This is an interesting finding for imaging modalities such as OCT, where images are acquired line-by-line. Further exploring the impact of different excitations and acquisition parameters is an interesting and important direction for future work.
Moreover, we developed and studied the concept of localized elasticity estimation by training with small localized spatio-temporal windows (Section 6.2.2). Our

findings demonstrate that we can perform localized elasticity estimations with such an approach. This training strategy constitutes an important contribution and could be an important step in overcoming data collection and annotation challenges. An advantage of this training approach is that we do not require entire elasticity maps as ground truth, which is in strict contrast compared to encoder-decoder approaches that estimate the entire elasticity map at once [3]. Thus, only sparse annotations could be used with our training approach, which saves annotations efforts, and also allows to focus training on areas where the annotation has a high reliability and quality. Overall, these results strongly indicate that with spatio-temporal deep learning, the data processing of dynamic elastography could be simplified compared to the previous approach that requires fine-tuned feature extraction and subsequent model building.

In summary, we demonstrated marker position estimation and markerless motion analysis using OCT and US data as well as for short-term and long-term sequences ranging over several minutes. Moreover, we demonstrated end-to-end elasticity estimation using OCT and US data. These results indicate that our spatio-temporal architecture concept can be used without major adaptions across different tasks and imaging modalities, which reduces architecture design efforts and thus provides a good starting point for future studies. Moreover, across our application scenarios, we achieved high performance with inference times in the range of a few milliseconds. Our findings demonstrate that spatio-temporal deep learning can effectively address end-to-end processing of sequences of medical image data.

Overall, the results across our application scenarios are promising and indicate that spatio-temporal deep learning could be an important milestone in achieving high and robust performance with sequences of medical image data. In the future spatio-temporal deep learning could be a valuable contribution to improve automatic FOV adjustment during OCT-based intraoperative imaging [62, 132, 134], to obtain information on surgical tool poses or tissue motion and location to guide surgical procedures such as laser cochleostomy [535], laser osteotomy [24]. Considering sequences of volumetric US images, in the future spatio-temporal deep learning could be a valuable contribution to improve motion compensation during radiotherapy [98, 211, 282, 409, 411], in particular, by combing both motion estimation and forecasting into a single approach. Considering dynamic elastography, spatio-temporal deep learning could be a valuable contribution to providing quantitative estimations of tissue elasticities. These estimations could improve clinical applications such as tumor resection [67], needle placement of biopsies [262, 351], or diagnostic decisions regarding discriminating different liver fibrosis stages [148, 309], or early tumor detection [306, 540].

7.2 Spatio-Temporal Deep Learning Approaches

Spatio-temporal feature learning with deep learning is known to be a difficult task with several architecture choices to learn effectively and efficiently from video

data [503]. Similar, it has been pointed out that the spatial and temporal nature of image sequences requires tailored deep learning approaches to process such multi-dimensional and rich data [369]. We address sequences of medical image data with a focus on regression problems and we address how spatio-temporal feature learning can be performed with deep learning and contribute to the question of how high and robust performance can be achieved. In Chapter 4, we presented different operations and architecture building blocks that can be adapted to different imaging modalities, dimensions, and tasks. As a result, we developed and presented a unified architecture concept for both 3D and 4D spatio-temporal data. Our approach can be adapted to pair-wise processing up to entire long-term sequences and can be used for image-level and sequence-level estimations of regression problems. Moreover, we also presented different training strategies to perform multi-task learning and to address training with long-term sequences of volumetric images. Considering our first research question, our results demonstrate that data analysis can be performed efficiently and effectively with our unified architecture concept, as discussed in the previous section. In this section, we discuss our different spatio-temporal deep learning approaches for multi-dimensional medical image sequence analysis, including spatio-temporal CNNs, multi-path Siamese CNNs, and mixed approaches of a CNN and a ConvGRU at multi-scales.

As a first simple baseline, we combined our CNN architecture with the concept of using the channel dimension of the input as the temporal dimension. This approach brings the advantage that an architecture with similar computational costs and inference time compared to single image processing can be used. Note that compared to a CNN for single image processing, the only adaption is the first layer of the network. It stands out that even such a simple approach allows improving performance compared, e.g., using a 3DCNN applied to a single OCT volume for marker position estimation (Section 6.1.1) or compared to conventional shear wave velocity estimation (Section 6.2.1 and Section 6.2.3). This is an interesting finding that indicates that the approach of a time-channel at the input-level is an effective baseline approach for the analysis of medical image sequences that is easy to adapt when first initial results are of interest. However, across all our experiments, we find that stacking multiple images of a sequence in the channel dimension of the input is not optimal for achieving high performance. The approach of joint spatio-temporal feature learning using spatio-temporal convolutions or by means of ConvGRU modules clearly outperforms this simple baseline approach across all our experiments. These results indicate that the approach of a time-channel is limited in terms of spatio-temporal feature learning, as also demonstrated in the natural image domain [457]. We also observed similar results for our variants of two-path (2P-TC-3DCNN) and multi-path (nP-TC-3DCNN) architectures, where the joint processing is performed based on channel-wise interactions after the Siamese part (Section 6.1.2). Recall that images are first processed individually and then concatenated along the channel dimension. Afterwards, processing is performed similar to the approach of TC-3D/4DCNN. Hence, our experiments indicate that joint spatio-temporal feature learning should

be performed to achieve high performance and to leverage the spatio-temporal data structure of medical image sequences.

Considering spatio-temporal feature learning, our results demonstrated that spatio-temporal convolutions are an effective approach to learning from entire sequences of images in an end-to-end fashion. Spatio-temporal CNNs have extensively been studied in the natural image domain and demonstrated promising results, e.g., for tasks such as human action recognition [438]. However, limited studies consider sequences of medical image data, particularly sequences of volumetric images. We present 4D spatio-temporal CNNs, and our results demonstrated that end-to-end learning could be performed from sequences of volumetric data. Across the different application scenarios and imaging modalities, we found that robust and high performance can be achieved with such an approach. An advantage of spatio-temporal convolutions is that they are conceptually simple and that they represent a direct extension of CNNs for image analysis to sequences of medical image analysis. Spatio-temporal convolutions increase the number of parameters compared to spatial convolutions or the time-channel approach. However, across our experiments, we did not observe notable overfitting that may result from the increased number of parameters. We also evaluated 4D factorized convolutions that allow for less parameter-intensive versions than full spatio-temporal convolutions. Here, a spatio-temporal convolutional layer is replaced by two successive convolutional layers, i.e, one that performs a spatial convolution and one that performs a temporal convolution. However, we did not observe any performance advantages (Section 6.1.1). Overall, across our experiments, our results indicate that the increased number of parameters of our spatio-temporal CNNs are not a limiting factor for learning from sequences of medical image data. In fact, as also outlined in Chapter 3, the effective capacity of a deep learning model is much more complex and cannot be attributed solely to the number of parameters [233]. Nevertheless, the computational efforts of 4D spatio-temporal CNNs in the range of several days even for our small datasets cannot be neglected. We demonstrated transfer learning from 3D to 4D spatio-temporal data to reduce computational requirements and training times. We first trained a 3DCNN to learn relevant spatial features, then inflated the architecture to a 4D spatio-temporal CNN, and then trained with 4D spatio-temporal data. Our findings highlighted that thereby training times could be reduced (Section 6.1.1). These results indicate that such a transfer learning strategy could counter the increased computational efforts of 4D spatio-temporal CNNs combined with medical image sequences.

Learning from sequences of image data also raises conceptual questions regarding the data. Considering motion analysis, we evaluated dividing a sequence into image-pairs or sub-sequences or to use the entire sequence at once. Our results highlight that pair-wise approaches such as 2P-TC-3DCNN can lead to good results. However, it results in poor performance when there is a small overlap between the template volume and the current volume (Section 6.1.2 and Section 6.1.3). Thus, we presented nP-ST-4DCNN a Siamese approach that processes multiple volumes and performs joint spatio-temporal processing. This approach

demonstrated promising results considering short-term motion analysis. This approach first processes each image of a CNN individually with Siamese convolutional layers. Afterwards, joint spatio-temporal feature learning is performed by means of spatio-temporal convolutions. Note that the Siamese part of the architecture leads to reduced parameters compared to the approach, where the entire image sequence is processed with a spatio-temporal CNN. Considering 4D data, our approach is a mixed approach between a 3D spatial and 4D spatio-temporal CNN. Notably, mixing 2D and 3D modeling has also been found to be effective for video processing [69].
Considering dynamic elastography analysis, we also evaluated using a lower-dimensional representation of the sequence typically used by conventional approaches [343, 481] and has also recently been used by deep learning approaches [225, 464]. Our results showed consistently that using the entire image sequences leads to higher performance compared to using the lower-dimensional space time map representation (Section 6.2.1 and Section 6.2.3). Overall, across both of our application scenarios, our results indicate that entire image sequences, i.e, the entire spatio-temporal data should be used and processed with spatio-temporal deep learning. This is an interesting finding as it demonstrates that end-to-end learning does not only reduce the requirement of feature selection, it also improves the performance. These results are similar to a study that considers deep learning and different data representations of medical image data [162], which also concluded that using higher-dimensional data typically outperforms using lower-dimensional data representations.

However, processing an entire sequence quickly leads to substantial computational efforts and run-times that limit real-time processing. We presented DenseConvGRU, a hybrid approach of ConvGRU units and a CNN built based on state-of-the-art architecture modules. A key advantage of this approach is that it is highly efficient during evaluation when an output is required for each image of a sequence. This results from recurrent processing of the data, where only the current image needs to be processed to generate the output, and the information from the history of the sequence is encoded in the hidden states of the memory units [413, 506]. Our experiments regarding long-term motion analysis using 4D US data demonstrated that this approach performs well, outperforming all other approaches (Section 6.1.3). A key concept of our approach is that we introduce spatio-temporal recurrence at different feature scales that allow learning localized and global spatio-temporal relationships. Previously, spatio-temporal recurrence has usually been considered at the front, or the end of architecture [115, 164, 211, 293, 319, 461, 486]. We expected that features at different scales are beneficial to represent small-scale and large-scale motions. Our results confirmed our expectation and we found multi-scale spatio-temporal recurrence particularly beneficial for motion analysis. Moreover, our results demonstrated that DenseConvGRU is also effective for aggregating spatio-temporal information of an entire sequence. Our results regarding 4D OCE highlighted that in the case of diffuse wave fields and abstract spatio-temporal patterns, learning a compact spatial representation first, followed by joint spatio-temporal processing,

can improve the performance and consistency compared to the approach using a spatio-temporal CNN (Section 6.2.4).
However, training with long-term sequences of volumetric images using recurrent approach results in substantial computational requirements. To address this problem, we presented a training approach that combines truncated BPTT [136, 221, 446, 493] and curriculum learning [31]. Our findings highlight that with this approach, training can be performed with shorter sequences while still achieving high performance for much longer sequences during testing (Section 6.1.3). This is an important finding, which indicates that DenseConvGRU is a viable approach to perform long-term tracking and motion analysis during, e.g., radiotherapy.

Another aspect of our architecture is the CNN backbone concept. We mostly relied on the concept of DenseNet [210], due to the advantages of a reduced number of parameters and strengthened feature propagation. We also performed a comparison of different backbones, which demonstrated no notable difference between different architecture concepts such as ResNet [197], Inception [216, 442, 443], DenseNet [210] and ResNeXt [504] (Section 6.2). These results indicate that the exact backbone CNN architecture concept has no substantial impact on the performance. Hence, we consider the backbone concept a hyperparameter that can be tuned and adapted. Considering our results across our different experiments, we achieved high performance using DenseNet as a backbone. This indicates that the concept of DenseNet for the backbone is a good default choice for spatio-temporal data processing and hence is recommended for future studies.

In summary, our results regarding architectures and concepts for learning from medical image sequences indicate that promising results can be achieved by using entire image sequences and learning joint spatio-temporal features. Our results highlighted that spatio-temporal convolutions and ConvGRU modules at different feature scales are well-suited to this end.

7.3 Training Data and Future Research

For a long time, spatio-temporal CNNs demonstrated limited performance regarding video analysis tasks [192]. Training of spatio-temporal deep learning models typically requires large-scale datasets with high-accuracy annotation. Collecting such datasets remains time-consuming and expensive and can also be affected by labeling errors. For example, annotation errors have been identified as a limiting factor for previous studies considering motion analysis from ultrasound image sequences [98]. Similar, related studies that consider deep learning and elastography rely on simulated data for training [3, 464], which has been identified as a limiting factor for the performance on real data. In general, data collection and annotation are fundamental problems of deep learning and medical image analysis [26, 49, 281].
To evaluate and compare our methods, we considered data from experimental setups that allowed for automatic data acquisition and even automatic annotation

for our experiments regarding motion analysis. The advantages of the automated setups are that we obtain annotation with high accuracy and that we have a controlled setup for our data. However, the experimental setups for data acquisition also bring limitations and do not include all aspects present during the clinical setting, e.g., deformation or rotations of a tracking target. We addressed these aspects with post-processing steps and performed experiments using synthetic deformations, rotations, or motion artifacts. Our results demonstrated that our spatio-temporal deep learning approach could still achieve high performance in these challenging scenarios, which indicates that such aspects could also be addressed in the clinical setting. For our experiments regarding elastography, we relied on gelatin phantoms. We assumed a homogeneous and distinct elasticity for particular phantoms. However, small deviations of elasticity and localized inhomogeneities are likely. Still, our spatio-temporal deep learning approach performed well, and these results indicate that noisy annotated training data can also be handled. This is an important finding, as noisy annotations can also be expected when data from the clinical practice are collected and annotated.

Moreover, our motion analysis and elastography results demonstrate that our spatio-temporal deep learning approach generalizes well. However, we still obverse reduced performance, e.g., for our experiments regarding dynamic elastography, where we estimate new unknown elasticities that were not considered during training. This effect can likely be reduced by using more training data with additional variance, but overall, it highlights the importance of the training data. This should be considered for future studies. Similar, we expect a lower performance using data from the actual clinical setting without additional training on such data. This problem could be reduced by acquiring and using more data for training that represents the clinical setting even more closely. Ideally, data from the actual clinical setting is acquired and annotated. However, this is known to be time-consuming and error-prone. Thus, automated setups for data acquisition and annotation could play an important role in collecting the required data to train robust deep learning models. Hence, an interesting direction for future work could be not only an evaluation in the clinical setting but also the development of automated setups that provide additional and improved training data. Large-scale datasets acquired with data acquisition setups combined with data from the actual clinical setting could be an effective approach.
In general, collecting more data is an effective approach to improve the performance. However, this also brings up an additional challenge, that is the processing of large-scale 3D/4D datasets. Notably, this requires building effective data pipelines and processing strategies to handle training with such data, including effective data loading, pre-processing, and storage. This challenge is also relevant and known in the natural image domain that considers video data processing, where large-scale datasets of thousands of videos ranging over several minutes are used for training [69, 369, 408]. While we focused on architecture aspects and strategies to perform spatio-temporal feature learning, addressing aspects regarding data pipelines is also an interesting direction for future work.

Another interesting direction for future work is raw data processing. Our results demonstrate that motion analysis can be performed, e.g., without distinct visual landmarks. This highlights the ability of spatio-temporal deep learning to learn complex and rich features end-to-end. Thus, raw data processing of the imaging modalities without data pre-processing could be an interesting step. For example, performing US-SWEI, spatio-temporal deep learning could directly process the raw RF-data of the piezoelectric elements. Thereby, data processing pipelines could be simplified even more. Notably, this scenario is challenging, and we expect large training datasets to be important.

Another relevant direction for future work is studying our approaches in other application scenarios, where sequences of medical image data are relevant, such as fMRI data [261], or hyperspectral image analysis [292]. Notably, our results demonstrated the flexibility of our approach. Thus, our approach could provide promising results in these scenarios where sequences of images are present, also, where the sequence dimension is not necessarily a temporal dimension but another sequential dimension of the data, such as during hyperspectral imaging. We already achieved first promising results in these scenarios using similar methods to those we presented in this work [25, 36, 37, 43, 129].

Moreover, future work could explore concepts to further reduce the computational requirements and training times of spatio-temporal deep learning, especially when high-resolution and large-scale images are used. Considering our results, further exploring of mixed 3D/4D spatio-temporal architectures or spatio-temporal feature learning using channel-wise interactions could be interesting directions. Our results demonstrated that multi-task learning could effectively be addressed. In the future, also cross-modality learning could be evaluated where a single architecture is trained, e.g., for motion analysis using multiple modalities such as OCT and US. Moreover, Transformer approaches [117,465] are gaining traction for medical image analysis, and studying such approaches for medical image sequences is an interesting direction for future work. However, so far, there are many open challenges considering Transformer methods and medical image analysis, including the requirement for large-scale datasets and pre-training and also substantial computational complexity [269]. However, as notable efforts are made regarding Transformer, we expect the associated challenges to reduce in the coming years making such an approach an interesting direction for future studies, e.g., to address motion analysis or dynamic elastography.

Chapter 8

Conclusion

In this thesis, we studied and presented spatio-temporal deep learning methods for analyzing medical image sequences. Sequences of medical image data have numerous clinical applications, yet analyzing the high-dimensional spatio-temporal data remains challenging, especially with real-time constraints [46, 98]. Deep learning has recently shown promising results for efficient end-to-end processing of medical image data in various clinical applications, including pioneering results for processing multi-dimensional medical image data [14, 162, 281]. However, spatio-temporal feature learning with deep learning is a difficult task and requires tailored deep learning approaches to process such multi-dimensional and rich data [369]. This results in several architecture choices to learn effectively and efficiently from such data [503]. Moreover, medical imaging modalities such as OCT and US even allow for real-time volumetric imaging, which allows to capture dynamic processes in their entirety, spatially and temporally. However, effectively analyzing such 3D/4D spatio-temporal data has many unsolved challenges and usually requires highly specialized data processing pipelines, feature extraction approaches, and model building [7, 14, 21, 55, 98, 267, 281, 377, 471].
This results in the two fundamental research questions for our work. First, whether multi-dimensional sequences of medical image data can be processed effectively with deep learning. Second, how multi-dimensional spatio-temporal feature learning can be performed with deep learning.

In Chapter 4, we presented a range of methods, including 3D/4D spatio-temporal CNNs (ST-3D/4DCNN), multi-path Siamese CNNs (nP-ST-4DCNN), and hybrid approaches of a CNN and ConvGRU modules (DenseConvGRU). To reduce the challenge of hand-crafting architectures for specific problems, we incorporated all these concepts into a compact and unified CNN architecture concept for end-to-end regression from sequences of medical image data. We also addressed learning spatio-temporal relationships end-to-end from sequences of volumetric image data with deep learning. Using OCT and US image data, we addressed the application scenarios, motion analysis, and dynamic elastography. We conceptualized the entire learning tasks to evaluate our methods and considered data acquired with experimental setups.
Considering motion analysis using OCT and US data, we studied object position estimation and markerless motion analysis of tissue (Section 6.1). Our results demonstrated that our spatio-temporal deep learning approach for end-to-end regression generalized to unseen tissue types, appearances, and motion patterns.

Across several experiments, our results showed that this could be performed with run-times in the range of milliseconds and with high performance even without distinct visual landmarks. A comparison of different methods highlighted that our approach of a multi-path Siamese network (nP-ST-4DCNN), and our hybrid approach of a CNN and ConvGRU modules at different scales (DenseConvGRU) performed best. In particular, the latter was beneficial for long-term motion analysis. Our results demonstrated that with our architecture and training concept, precise marker-less motion analysis from entire long-term 4D sequences could also be performed for sequences of several hundreds of volumes ranging over several minutes. We also combined the task of motion estimation and forecasting into a single approach, allowing us to directly address system latencies of motion compensation systems, e.g., during radiotherapy.
Considering dynamic elastography, our findings demonstrated that with our end-to-end regression approach, the direct relationship between tissue elasticity and wave propagation could be learned from sequences of OCT and US images with spatio-temporal deep learning (Section 6.2). Our results showed that a spatio-temporal CNN (ST-3D/4DCNN) is well suited to this end and that such an approach generalized to unseen elasticities, phantoms, and wave propagation directions and that localized elasticity estimation can be performed. We also presented a training concept using localized spatio-temporal windows allowing for localized elasticity estimation and generation of entire elasticity maps.

We addressed both of our application scenarios and imaging modalities with the same architecture concept, processed data in an end-to-end fashion, and achieved high performance with inference times in the range of a few milliseconds. Thus, considering our first research question, these results demonstrate that sequences of medical image data can be processed effectively with spatio-temporal deep learning. Considering our second research question, we approached the task with different concepts for learning spatio-temporal relationships from data and also considered different data representations. Across our experiments, we found that joint spatio-temporal feature learning performed well and that 3D/4D spatio-temporal CNNs and ConvGRU modules at different feature scales are effective to this end. Moreover, we found that entire image sequences, i.e., the entire 3D/4D spatio-temporal data, can and should be processed end-to-end with spatio-temporal deep learning. Our results demonstrated that, thereby, higher and more robust performance could be achieved compared to previous approaches that only process a single image representation or pair-wise images of a sequence.

In summary, in this thesis, we presented and evaluated a range of spatio-temporal deep learning methods for end-to-end regression using sequences of medical images. An interesting direction for future work could be an evaluation in the clinical setting but also the development of automated setups that provide additional and improved training data. Training spatio-temporal deep learning with large-scale datasets acquired with data acquisition setups combined with data from the actual clinical setting could be an effective approach to achieve high and robust performance during clinical practice. Furthermore, future work could also

evaluate our approach in other scenarios, such as fMRI and hyperspectral image analysis, where sequences of medical image data are also relevant.
Overall, our findings highlight that spatio-temporal deep learning is a viable approach for motion analysis and dynamic elastography and that processing entire sequences end-to-end utilizing joint spatio-temporal feature learning leads to promising results. Ultimately, our findings demonstrate that spatio-temporal deep learning can effectively address end-to-end processing of sequences of medical image data, including sequences of volumetric images. In the future, we expect spatio-temporal deep learning to provide further advancement of medical image analysis.

Bibliography

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)

[2] Adelson, E.H., Bergen, J.R.: Spatiotemporal energy models for the perception of motion. Journal of the Optical Society of America 2(2), 284–299 (1985)

[3] Ahmed, S., Kamal, U., Hasan, M.K.: DSWE-Net: A deep learning approach for shear wave elastography and lesion segmentation using single push acoustic radiation force. Ultrasonics 110, 106283 (2021)

[4] Ahn, B.M., Kim, J., Ian, L., Rha, K.H., Kim, H.J.: Mechanical property characterization of prostate cancer using a minimally motorized indenter in an ex vivo indentation experiment. Urology 76(4), 1007–1011 (2010)

[5] Al-Dhabyani, W., Gomaa, M., Khaled, H., Fahmy, A.: Dataset of breast ultrasound images. Data Brief 28, 104863 (2020) (2019)

[6] Alam, F., Rahman, S.U., Ullah, S., Gulati, K.: Medical image registration in image guided surgery: Issues, challenges and research opportunities. Biocybernetics and Biomedical Engineering 38(1), 71–89 (2018)

[7] Alam, S.K., Garra, B.S.: Tissue elasticity imaging: Volume 1: Theory and methods. Elsevier (2019)

[8] Alhersh, T., Stuckenschmidt, H., Rehman, A.U., Belhaouari, S.B.: Learning human activity from visual data using deep learning. IEEE Access 9, 106245–106253 (2021)

[9] Alom, M.Z., Taha, T.M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M.S., Van Esesn, B.C., Awwal, A.A.S., Asari, V.K.: The history began from alexnet: A comprehensive survey on deep learning approaches. arXiv preprint arXiv:1803.01164 (2018)

[10] Aly, I., Rizvi, A., Roberts, W., Khalid, S., Kassem, M.W., Salandy, S., du Plessis, M., Tubbs, R.S., Loukas, M.: Cardiac ultrasound: An anatomical and clinical review. Translational Research in Anatomy 22, 100083 (2021)

[11] Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M., Farhan, L.: Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data 8(1), 1–74 (2021)

[12] Ambroziński, Ł., Song, S., Yoon, S.J., Pelivanov, I., Li, D., Gao, L., Shen, T.T., Wang, R.K., OâDonnell, M.: Acoustic micro-tapping for non-contact 4D imaging of tissue elasticity. Scientific Reports 6(1), 38967 (2016)

[13] Antico, M., Sasazawa, F., Wu, L., Jaiprakash, A., Roberts, J., Crawford, R., Pandey, A.K., Fontanarosa, D.: Ultrasound guidance in minimally invasive robotic procedures. Medical Image Analysis 54, 149–167 (2019)

[14] Anwar, S.M., Majid, M., Qayyum, A., Awais, M., Alnowami, M., Khan, M.K.: Medical image analysis using convolutional neural networks: A review. Journal of Medical Systems 42(11), 1–13 (2018)

[15] Azad, R., Asadi-Aghbolaghi, M., Fathy, M., Escalera, S.: Bi-directional ConvLSTM U-Net with densley connected convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 0–0 (2019)

[16] Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)

[17] Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., Baskurt, A.: Sequential deep learning for human action recognition. In: International Workshop on Human Behavior Understanding. pp. 29–39. Springer (2011)

[18] Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018)

[19] Ballas, N., Yao, L., Pal, C., Courville, A.: Delving deeper into convolutional networks for learning video representations. arXiv preprint arXiv:1511.06432 (2015)

[20] Ballhausen, H., Li, M., Hegemann, N., Ganswindt, U., Belka, C.: Intra-fraction motion of the prostate is a random walk. Physics in Medicine & Biology 60(2), 549 (2014)

[21] Banerjee, J., Klink, C., Niessen, W.J., Moelker, A., van Walsum, T.: 4D Ultrasound tracking of liver and its verification for tips guidance. IEEE Transactions on Medical Imaging 35(1), 52–62 (2015)

[22] Banerjee, J., Klink, C., Vast, E., Niessen, W.J., Moelker, A., van Walsum, T.: A combined tracking and registration approach for tracking anatomical landmarks in 4D ultrasound of the liver. In: MICCAI Workshop: Challenge on Liver Ultrasound Tracking. pp. 36–43 (2015)

[23] Baur, C., Denner, S., Wiestler, B., Navab, N., Albarqouni, S.: Autoencoders for unsupervised anomaly segmentation in brain MR images: A comparative study. Medical Image Analysis 69, 101952 (2021)

[24] Bayhaqi, Y.A., Hamidi, A., Canbaz, F., Navarini, A.A., Cattin, P.C., Zam, A.: Deep-Learning-Based Fast Optical Coherence Tomography (OCT) Image Denoising for Smart Laser Osteotomy. IEEE Transactions on Medical Imaging 41(10), 2615–2628 (2022)

[25] Behrendt, F., Bengs, M., Bhattacharya, D., Krüger, J., Opfer, R., Schlaefer, A.: Capturing Inter-Slice Dependencies of 3D Brain MRI-Scans for Unsupervised Anomaly Detection. In: Medical Imaging with Deep Learning (2022)

[26] Behrendt, F., Bengs, M., Rogge, F., Krüger, J., Opfer, R., Schlaefer, A.: Unsupervised Anomaly Detection in 3D Brain MRI using Deep Learning with impured training data. In: 2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI). pp. 1–4. IEEE (2022)

[27] Bello, G.A., Dawes, T.J., Duan, J., Biffi, C., De Marvao, A., Howard, L.S., Gibbs, J.S.R., Wilkins, M.R., Cook, S.A., Rueckert, D., et al.: Deep-learning cardiac motion analysis for human survival prediction. Nature Machine Intelligence 1(2), 95–104 (2019)

[28] Bengio, Y.: Practical recommendations for gradient-based training of deep architectures. In: Neural networks: Tricks of the Trade, pp. 437–478. Springer (2012)

[29] Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(8), 1798–1828 (2013)

[30] Bengio, Y., Frasconi, P., Simard, P.: The problem of learning long-term dependencies in recurrent networks. In: IEEE International Conference on Neural Networks. pp. 1183–1188. IEEE (1993)

[31] Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 41–48 (2009)

[32] Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks 5(2), 157–166 (1994)

[33] Bengs, M., Behrendt, F., Krüger, J., Opfer, R., Schlaefer, A.: Three-dimensional deep learning with spatial erasing for unsupervised anomaly segmentation in brain MRI. International Journal of Computer Assisted Radiology and Surgery 16, 1413–1423 (2021)

[34] Bengs, M., Behrendt, F., Laves, M.H., Krüger, J., Opfer, R., Schlaefer, A.: Unsupervised anomaly detection in 3D brain MRI using deep learning with multi-task brain age prediction. In: Medical Imaging 2022: Computer-Aided Diagnosis. vol. 12033, pp. 291–295. SPIE (2022)

[35] Bengs, M., Bockmayr, M., Schüller, U., Schlaefer, A.: Medulloblastoma tumor classification using deep transfer learning with multi-scale EfficientNets. In: Medical Imaging 2021: Digital Pathology. vol. 11603, pp. 70–75. SPIE (2021)

[36] Bengs, M., Gessert, N., Laffers, W., Eggert, D., Westermann, S., Mueller, N.A., Gerstner, A.O., Betz, C., Schlaefer, A.: Spectral-spatial recurrent-convolutional networks for in-vivo hyperspectral tumor type classification. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 690–699. Springer (2020)

[37] Bengs, M., Gessert, N., Schlaefer, A.: 4D spatio-temporal deep learning with 4D fMRI data for autism spectrum disorder classification. In: International Conference on Medical Imaging with Deep Learning (2019)

[38] Bengs, M., Gessert, N., Schlaefer, A.: 4D spatio-temporal convolutional networks for object position estimation in OCT volumes. Current Directions in Biomedical Engineering 6(1) (2020)

[39] Bengs, M., Gessert, N., Schlaefer, A.: A deep learning approach for motion forecasting using 4D OCT data. In: International Conference on Medical Imaging with Deep Learning (2020)

[40] Bengs, M., Gessert, N., Schlüter, M., Schlaefer, A.: Spatio-temporal deep learning methods for motion estimation using 4D OCT image data. International Journal of Computer Assisted Radiology and Surgery 15, 943–952 (2020)

[41] Bengs, M., Pant, S., Bockmayr, M., Schüller, U., Schlaefer, A.: Multi-scale input strategies for medulloblastoma tumor classification using deep transfer learning. Current Directions in Biomedical Engineering 7(1), 63–66 (2021)

[42] Bengs, M., Sprenger, Johanna, G.S., Neidhardt, M., Schlaefer, A.: Real-time motion analysis with 4D deep learning for ultrasound-guided radiotherapy. IEEE Transactions on Biomedical Engineering 70(9), 2690–2699 (2023)

[43] Bengs, M., Westermann, S., Gessert, N., Eggert, D., Gerstner, A.O., Mueller, N.A., Betz, C., Laffers, W., Schlaefer, A.: Spatio-spectral deep learning methods for in-vivo hyperspectral laryngeal cancer detection. In: Medical Imaging 2020: Computer-Aided Diagnosis. vol. 11314, pp. 369–374. SPIE (2020)

[44] Bercoff, J.: Ultrafast ultrasound imaging. Ultrasound Imaging-Medical applications pp. 3–24 (2011)

[45] Bercoff, J., Tanter, M., Fink, M.: Supersonic shear imaging: A new technique for soft tissue elasticity mapping. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 51(4), 396–409 (2004)

[46] Bertholet, J., Knopf, A., Eiben, B., McClelland, J., Grimwood, A., Harris, E., Menten, M., Poulsen, P., Nguyen, D.T., Keall, P., et al.: Real-time intrafraction motion monitoring in external beam radiotherapy. Physics in Medicine & Biology 64(15), 15TR01 (2019)

[47] Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: European Conference on Computer Vision. pp. 850–865. Springer (2016)

[48] Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., Modi, K., Ghayvat, H.: CNN variants for computer vision: History, architecture, application, challenges and future scope. Electronics 10(20), 2470 (2021)

[49] Bhattacharya, D., Becker, B.T., Behrendt, F., Bengs, M., Beyersdorff, D., Eggert, D., Petersen, E., Jansen, F., Petersen, M., Cheng, B., et al.: Supervised Contrastive Learning to Classify Paranasal Anomalies in the Maxillary Sinus. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part III. pp. 429–438. Springer (2022)

[50] Bianco, S., Cadene, R., Celona, L., Napoletano, P.: Benchmark analysis of representative deep neural network architectures. IEEE Access 6, 64270–64277 (2018)

[51] Billings, S., Deshmukh, N., Kang, H.J., Taylor, R., Boctor, E.M.: System for robot-assisted real-time laparoscopic ultrasound elastography. In: Medical Imaging 2012: Image-Guided Procedures, Robotic Interventions, and Modeling. vol. 8316, pp. 589–596. SPIE (2012)

[52] Bishop, C.M.: Pattern Recognition and Machine Learning, vol. 128. Springer (2006)

[53] Blumer, A., Ehrenfeucht, A., Haussler, D., Warmuth, M.K.: Learnability and the Vapnik-Chervonenkis dimension. Journal of the ACM (JACM) 36(4), 929–965 (1989)

[54] Borchani, H., Varando, G., Bielza, C., Larranaga, P.: A survey on multi-output regression. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 5(5), 216–233 (2015)

[55] Bouget, D., Allan, M., Stoyanov, D., Jannin, P.: Vision-based and marker-less surgical tool detection and tracking: A review of the literature. Medical Image Analysis 35, 633–654 (2017)

[56] Boukerroui, D., Noble, J.A., Brady, M.: Velocity estimation in ultrasound images: A block matching approach. In: International Conference on Image Processing and Machine Intelligence. vol. 2732, pp. 586–598. Springer (2003)

[57] Brattain, L.J., Ozturk, A., Telfer, B.A., Dhyani, M., Grajo, J.R., Samir, A.E.: Image processing pipeline for liver fibrosis classification using ultrasound shear wave elastography. Ultrasound in Medicine & Biology 46(10), 2667–2676 (2020)

[58] Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a "siamese" time delay neural network. Advances in Neural Information Processing Systems 6 (1993)

[59] Burckhardt, C.B.: Speckle in ultrasound B-mode scans. IEEE Transactions on Sonics and Ultrasonics 25(1), 1–6 (1978)

[60] Cai, L., Gao, J., Zhao, D.: A review of the application of deep learning in medical image classification and segmentation. Annals of Translational Medicine 8(11) (2020)

[61] Carlsen, J.F., Ewertsen, C., Lönn, L., Nielsen, M.B.: Strain elastography ultrasound: An overview with emphasis on breast cancer diagnosis. Diagnostics 3(1), 117–125 (2013)

[62] Carrasco-Zevallos, O.M., Viehland, C., Keller, B., Draelos, M., Kuo, A.N., Toth, C.A., Izatt, J.A.: Review of intraoperative optical coherence tomography: Technology and applications. Biomedical Optics Express 8(3), 1607–1637 (2017)

[63] Carreira, J., Zisserman, A.: Quo vadis, action recognition? A new model and the kinetics dataset. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. pp. 4724–4733. IEEE (2017)

[64] Caruana, R.: Multitask learning: A knowledge-based source of inductive bias. In: Proceedings of the Tenth International Conference on Machine Learning. pp. 41–48. Citeseer (1993)

[65] Cense, B., Nassif, N.A., Chen, T.C., Pierce, M.C., Yun, S.H., Park, B.H., Bouma, B.E., Tearney, G.J., De Boer, J.F.: Ultrahigh-resolution high-speed retinal imaging using spectral-domain optical coherence tomography. Optics Express 12(11), 2435–2447 (2004)

[66] Chan, D.Y., Morris, D.C., Polascik, T.J., Palmeri, M.L., Nightingale, K.R.: Deep convolutional neural networks for displacement estimation in ARFI imaging. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 68(7), 2472–2481 (2021)

[67] Chan, H.W., Uff, C., Chakraborty, A., Dorward, N., Bamber, J.C.: Clinical application of shear wave elastography for assisting brain tumor resection. Frontiers in Oncology 11, 112 (2021)

[68] Chan, V., Perlas, A.: Basics of ultrasound imaging. In: Atlas of Ultrasound-Guided Procedures in Interventional Pain Management, pp. 13–19. Springer (2011)

[69] Chen, C.F.R., Panda, R., Ramakrishnan, K., Feris, R., Cohn, J., Oliva, A., Fan, Q.: Deep analysis of CNN-based spatio-temporal representations for action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6165–6175 (2021)

[70] Chen, K., Tao, W.: Once for all: A two-flow convolutional neural network for visual tracking. IEEE Transactions on Circuits and Systems for Video Technology 28(12), 3377–3386 (2017)

[71] Chen, Z., Zeng, Z., Shen, H., Zheng, X., Dai, P., Ouyang, P.: DN-GAN: Denoising generative adversarial networks for speckle noise reduction in optical coherence tomography images. Biomedical Signal Processing and Control 55, 101632 (2020)

[72] Chicco, D.: Siamese neural networks: An overview. Artificial Neural Networks pp. 73–94 (2021)

[73] Chin, L., Curatolo, A., Kennedy, B.F., Doyle, B.J., Munro, P.R., McLaughlin, R.A., Sampson, D.D.: Analysis of image formation in optical coherence elastography using a multiphysics approach. Biomedical Optics Express 5(9), 2913–2930 (2014)

[74] Chinn, S., Swanson, E., Fujimoto, J.: Optical coherence tomography using a frequency-tunable optical source. Optics Letters 22(5), 340–342 (1997)

[75] Chinnaiyan, P., Tomé, W., Patel, R., Chappell, R., Ritter, M.: 3D-ultrasound guided radiation therapy in the post-prostatectomy setting. Technology in Cancer Research & Treatment 2(5), 455–458 (2003)

[76] Chmarra, M.K., Grimbergen, C., Dankelman, J.: Systems for tracking minimally invasive surgical instruments. Minimally Invasive Therapy & Allied Technologies 16(6), 328–340 (2007)

[77] Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259 (2014)

[78] Cho, S.H., Lee, J.Y., Han, J.K., Choi, B.I.: Acoustic radiation force impulse elastography for the evaluation of focal solid hepatic lesions: Preliminary findings. Ultrasound in Medicine & Biology 36(2), 202–208 (2010)

[79] Choma, M.A., Sarunic, M.V., Yang, C., Izatt, J.A.: Sensitivity advantage of swept source and Fourier domain optical coherence tomography. Optics Express 11(18), 2183–2189 (2003)

[80] Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1, pp. 539–546. IEEE (2005)

[81] Choy, C., Gwak, J., Savarese, S.: 4D spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3075–3084 (2019)

[82] Chrupała, G., Kádár, A., Alishahi, A.: Learning language through pictures. arXiv preprint arXiv:1506.03694 (2015)

[83] Chung, D., Tahboub, K., Delp, E.J.: A two stream siamese convolutional neural network for person re-identification. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1983–1991 (2017)

[84] Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)

[85] Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Gated feedback recurrent neural networks. In: International Conference on Machine Learning. pp. 2067–2075. PMLR (2015)

[86] Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 424–432. Springer (2016)

[87] Clark, D., Badea, C.: Convolutional regularization methods for 4D, x-ray CT reconstruction. In: Medical Imaging 2019: Physics of Medical Imaging. vol. 10948, pp. 574–585. SPIE (2019)

[88] Cobbold, R.S.: Foundations of biomedical ultrasound. Oxford University Press (2006)

[89] Colvill, E., Booth, J., Nill, S., Fast, M., Bedford, J., Oelfke, U., Nakamura, M., Poulsen, P., Worm, E., Hansen, R., Ravkilde, T., Rydhög, J., Pommer, T., Rosenschold, P., Lang, S., Guckenberger, M., Groh, C., Herrmann, C., Verellen, D., Poels, K., Wang, L., Hadsell, M., Sothmann, T., Blanck, O., Keall, P.: A dosimetric comparison of real-time adaptive and non-adaptive radiotherapy: A multi-institutional study encompassing robotic, gimbaled, multileaf collimator and couch tracking. Radiother Oncol 119(1), 159–165 (2016)

[90] Colyer, S.L., Evans, M., Cosker, D.P., Salo, A.I.: A review of the evolution of vision-based motion analysis and the integration of advanced computer vision methods towards developing a markerless system. Sports Medicine-Open 4(1), 1–15 (2018)

[91] Cossmann, M., Welzel, J.: Evaluation of the atrophogenic potential of different glucocorticoids using optical coherence tomography, 20-MHz ultrasound and profilometry; A double-blind, placebo-controlled trial. British Journal of Dermatology 155(4), 700–706 (2006)

[92] Dahl, J.J., Pinton, G.F., Palmeri, M.L., Agrawal, V., Nightingale, K.R., Trahey, G.E.: A parallel tracking method for acoustic radiation force impulse imaging. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 54(2), 301–312 (2007)

[93] Dai, Z., Heckel, R.: Channel normalization in convolutional neural network avoids vanishing gradients. arXiv preprint arXiv:1907.09539 (2019)

[94] Damerjian, V., Tankyevych, O., Souag, N., Petit, E.: Speckle characterization methods in ultrasound images–A review. IRBM 35(4), 202–213 (2014)

[95] Dantas, R.G., Costa, E.T., Leeman, S.: Ultrasound speckle and equivalent scatterers. Ultrasonics 43(6), 405–420 (2005)

[96] Darken, C., Chang, J., Moody, J., et al.: Learning rate schedules for faster stochastic gradient search. In: Neural Networks for Signal Processing. vol. 2. Citeseer (1992)

[97] De Luca, V., Benz, T., Kondo, S., König, L., Lübke, D., Rothlübbers, S., Somphone, O., Allaire, S., Bell, M.L., Chung, D., et al.: The 2014 liver ultrasound tracking benchmark. Physics in Medicine & Biology 60(14), 5571 (2015)

[98] De Luca, V., Banerjee, J., Hallack, A., Kondo, S., Makhinya, M., Nouri, D., Royer, L., Cifor, A., Dardenne, G., Goksel, O., et al.: Evaluation of 2D and 3D ultrasound tracking algorithms and impact on ultrasound-guided liver radiotherapy margins. Medical Physics 45(11), 4986–5003 (2018)

[99] Delaunay, R., Hu, Y., Vercauteren, T.: An unsupervised learning approach to ultrasound strain elastography with spatio-temporal consistency. Physics in Medicine & Biology 66(17), 175031 (2021)

[100] Delaunay, R., Hu, Y., Vercauteren, T.: An unsupervised learning-based shear wave tracking method for ultrasound elastography. In: Medical Imaging 2022: Ultrasonic Imaging and Tomography. vol. 12038, pp. 149–155. SPIE (2022)

[101] Demi, L.: Practical guide to ultrasound beam forming: Beam pattern and image reconstruction analysis. Applied Sciences 8(9), 1544 (2018)

[102] Demi, L., Verweij, M.D., Van Dongen, K.W.: Parallel transmit beamforming using orthogonal frequency division multiplexing applied to harmonic imaging-A feasibility study. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 59(11), 2439–2447 (2012)

[103] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255. Ieee (2009)

[104] Deng, Y., Rouze, N.C., Palmeri, M.L., Nightingale, K.: On system-dependent sources of uncertainty and bias in ultrasonic quantitative shear-wave imaging. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 63(3), 381–93 (2016)

[105] Dennis, E.L., Thompson, P.M.: Functional brain connectivity using fMRI in aging and Alzheimerâs disease. Neuropsychology Review 24, 49–62 (2014)

[106] Devalla, S.K., Subramanian, G., Pham, T.H., Wang, X., Perera, S., Tun, T.A., Aung, T., Schmetterer, L., Thiery, A.H., Girard, M.J.: A deep learning approach to denoise optical coherence tomography images of the optic nerve head. Scientific Reports 9(1), 1–13 (2019)

[107] DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)

[108] Dhont, J., Harden, S., Chee, L., Aitken, K., Hanna, G., Bertholet, J.: Image-guided radiotherapy to manage respiratory motion: Lung and liver. Clinical Oncology 32(12), 792–804 (2020)

[109] Diba, A., Fayyaz, M., Sharma, V., Arzani, M.M., Yousefzadeh, R., Gall, J., Van Gool, L.: Spatio-temporal channel correlation networks for action classification. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 284–299 (2018)

[110] Diba, A., Fayyaz, M., Sharma, V., Karami, A.H., Arzani, M.M., Yousefzadeh, R., Van Gool, L.: Temporal 3D convnets: New architecture and transfer learning for video classification. arXiv preprint arXiv:1711.08200 (2017)

[111] Dice, L.R.: Measures of the amount of ecologic association between species. Ecology 26(3), 297–302 (1945)

[112] Dieterich, S., Green, O., Booth, J.: SBRT targets that move with respiration. Physica Medica 56, 19–24 (2018)

[113] Dogangil, G., Davies, B., Rodriguez y Baena, F.: A review of medical robotics for minimally invasive soft tissue surgery. Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine 224(5), 653–679 (2010)

[114] Doherty, J.R., Trahey, G.E., Nightingale, K.R., Palmeri, M.L.: Acoustic radiation force elasticity imaging in diagnostic ultrasound. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 60(4), 685–701 (2013)

[115] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2625–2634 (2015)

[116] Dong, B., Huang, N., Bai, Y., Xie, S.: Deep-learning-based approach for strain estimation in phase-sensitive optical coherence elastography. Optics Letters 46(23), 5914–5917 (2021)

[117] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)

[118] Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2758–2766 (2015)

[119] Drexler, W., Fujimoto, J.G.: Optical coherence tomography: Technology and applications. Springer Science & Business Media (2008)

[120] Dubey, S.R., Singh, S.K., Chaudhuri, B.B.: Activation functions in deep learning: A comprehensive survey and benchmark. Neurocomputing (2022)

[121] Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research 12(Jul), 2121–2159 (2011)

[122] Dumoulin, V., Visin, F.: A guide to convolution arithmetic for deep learning. arXiv preprint arXiv:1603.07285 (2016)

[123] Duncan, D.D., Kirkpatrick, S.J.: Performance analysis of a maximum-likelihood speckle motion estimator. Optics Express 10(18), 927–941 (2002)

[124] Duncan, D.D., Kirkpatrick, S.J.: Processing algorithms for tracking speckle shifts in optical elastography of biological tissues. Journal of Biomedical Optics 6(4), 418–426 (2001)

[125] Dunnhofer, M., Antico, M., Sasazawa, F., Takeda, Y., Camps, S., Martinel, N., Micheloni, C., Carneiro, G., Fontanarosa, D.: Siam-U-Net: Encoder-decoder siamese network for knee cartilage tracking in ultrasound images. Medical Image Analysis 60, 101631 (2020)

[126] Dürichen, R., Wissel, T., Ernst, F., Schlaefer, A., Schweikard, A.: Multivariate respiratory motion prediction. Physics in Medicine & Biology 59(20), 6043 (2014)

[127] Dvornek, N.C., Ventola, P., Pelphrey, K.A., Duncan, J.S.: Identifying autism from resting-state fMRI using long short-term memory networks. In: Machine Learning in Medical Imaging: 8th International Workshop, MLMI 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 10, 2017, Proceedings 8. pp. 362–370. Springer (2017)

[128] Echner, G., Kilby, W., Lee, M., Earnst, E., Sayeh, S., Schlaefer, A., Rhein, B., Dooley, J., Lang, C., Blanck, O., Lessard, E., Maurer, C.J., Schlegel, W.: The design, physical properties and clinical utility of an iris collimator for robotic radiosurgery. Physics in Medicine & Biology 54(18), 5359 (2009)

[129] Eggert, D., Bengs, M., Westermann, S., Gessert, N., Gerstner, A.O., Mueller, N.A., Bewarder, J., Schlaefer, A., Betz, C., Laffers, W.: In vivo detection of head and neck tumors by hyperspectral imaging combined with deep learning methods. Journal of Biophotonics 15(3), e202100167 (2022)

[130] Ehlers, J.P., Dupps, W.J., Kaiser, P.K., Goshe, J., Singh, R.P., Petkovsek, D., Srivastava, S.K.: The prospective intraoperative and perioperative ophthalmic imaging with optical coherence tomography (PIONEER) study: 2-year results. American Journal of Ophthalmology 158(5), 999–1007 (2014)

[131] El-Gazzar, A., Quaak, M., Cerliani, L., Bloem, P., Wingen, G.v., Mani Thomas, R.: A hybrid 3DCNN and 3DC-LSTM based model for 4D spatio-temporal fMRI data: An ABIDE autism classification study. In: OR 2.0 Context-Aware Operating Theaters and Machine Learning in Clinical Neuroimaging, pp. 95–102. Springer (2019)

[132] El-Haddad, M.T., Malone, J.D., Hoang, N.T., Tao, Y.K.: Deep-learning based automated instrument tracking and adaptive-sampling of intraoperative OCT for video-rate volumetric imaging of ophthalmic surgical maneuvers. In: Optical Coherence Tomography and Coherence Domain Optical Methods in Biomedicine XXIII. vol. 10867, pp. 30–35. SPIE (2019)

[133] El-Haddad, M.T., Tao, Y.K.: Automated stereo vision instrument tracking for intraoperative OCT guided anterior segment ophthalmic surgical maneuvers. Biomedical Optics Express 6(8), 3014–3031 (2015)

[134] El-Haddad, M.T., Tao, Y.K.: Advances in intraoperative optical coherence tomography for surgical guidance. Current Opinion in Biomedical Engineering 3, 37–48 (2017)

[135] Elangovan, A., Jeyaseelan, T.: Medical imaging modalities: A survey. In: 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS). pp. 1–4. IEEE (2016)

[136] Elman, J.L.: Finding structure in time. Cognitive Science 14(2), 179–211 (1990)

[137] Elsken, T., Metzen, J.H., Hutter, F.: Neural architecture search: A survey. The Journal of Machine Learning Research 20(1), 1997–2017 (2019)

[138] Ernst, F., Dürichen, R., Schlaefer, A., Schweikard, A.: Evaluating and comparing algorithms for respiratory motion prediction. Physics in Medicine & Biology 58(11), 3911 (2013)

[139] Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M., Thrun, S.: Dermatologist-level classification of skin cancer with deep neural networks. Nature 542(7639), 115–118 (2017)

[140] Farzad, A., Mashayekhi, H., Hassanpour, H.: A comparative performance analysis of different activation functions in LSTM networks for classification. Neural Computing and Applications 31, 2507–2521 (2019)

[141] Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6202–6211 (2019)

[142] Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1933–1941 (2016)

[143] Fercher, A.F., Hitzenberger, C.K., Kamp, G., El-Zaiat, S.Y.: Measurement of intraocular distances by backscattering spectral interferometry. Optics Communications 117(1-2), 43–48 (1995)

[144] Fiaz, M., Mahmood, A., Jung, S.K.: Deep siamese networks toward robust visual tracking. In: Visual Object Tracking with Deep Neural Networks. IntechOpen London, UK (2019)

[145] Fink, M., Tanter, M.: Multiwave imaging and super resolution. Physics Today 63(2), 28–33 (2010)

[146] Fonseca, L.M., Manjunath, B.: Registration techniques for multisensor remotely sensed imagery. PE & RS- Photogrammetric Engineering & Remote Sensing 62(9), 1049–1056 (1996)

[147] Fortun, D., Bouthemy, P., Kervrann, C.: Optical flow modeling and computation: A survey. Computer Vision and Image Understanding 134, 1–21 (2015)

[148] Franchi-Abella, S., Corno, L., Gonzales, E., Antoni, G., Fabre, M., Ducot, B., Pariente, D., Gennisson, J., Tanter, M., Corréas, J.: Feasibility and diagnostic accuracy of supersonic shear-wave elastography for the assessment of liver stiffness and liver fibrosis in children: A pilot study of 96 patients. Radiology 278(2), 554–562 (2015)

[149] Franchi-Abella, S., Elie, C., Correas, J.M.: Ultrasound elastography: Advantages, limitations and artefacts of the different techniques from a study on a phantom. Diagnostic and Interventional Imaging 94(5), 497–501 (2013)

[150] Freund, J., Buijs, M., Savci-Heijink, C., de Bruin, D., de la Rosette, J., van Leeuwen, T., Laguna, M.: Optical coherence tomography in urologic oncology: A comprehensive review. SN Comprehensive Clinical Medicine 1(2), 67–84 (2019)

[151] Fujimoto, J.G., Pitris, C., Boppart, S.A., Brezinski, M.E.: Optical coherence tomography: An emerging technology for biomedical imaging and optical biopsy. Neoplasia 2(1-2), 9–25 (2000)

[152] Fujioka, T., Katsuta, L., Kubota, K., Mori, M., Kikuchi, Y., Kato, A., Oda, G., Nakagawa, T., Kitazume, Y., Tateishi, U.: Classification of breast masses on ultrasound shear wave elastography using convolutional neural networks. Ultrasonic Imaging 42(4-5), 213–220 (2020)

[153] Gao, J., Xu, L., Bouakaz, A., Wan, M.: A deep Siamese-Based plantar fasciitis classification method using shear wave elastography. IEEE Access 7, 130999–131007 (2019)

[154] Gao, L., Parker, K., Lerner, R., Levinson, S.: Imaging of the elastic properties of tissue - A review. Ultrasound in Medicine & Biology 22(8), 959–977 (1996)

[155] Gao, L., Zhou, R., Dong, C., Feng, C., Li, Z., Wan, X., Liu, L.: Multi-modal active learning for automatic liver fibrosis diagnosis based on ultrasound shear wave elastography. In: 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI). pp. 410–414. IEEE (2021)

[156] Garra, B.S.: Elastography: History, principles, and technique comparison. Abdominal Imaging 40(4), 680–697 (2015)

[157] Gary, R.G., Zvietcovich, F., Rolland, J.P., Mestre, H., Giannetto, M., Nedergaard, M., Parker, K.J.: A preliminary study on using reverberant shear wave fields in optical coherence elastography to examine mice brain ex vivo. In: Optical Elastography and Tissue Biomechanics VI. vol. 10880, pp. 55–61. SPIE (2019)

[158] Gatos, I., Tsantis, S., Spiliopoulos, S., Karnabatidis, D., Theotokas, I., Zoumpoulis, P., Loupas, T., Hazle, J.D., Kagadis, G.C.: Temporal stability assessment in shear wave elasticity images validated by deep learning neural network for chronic liver disease fibrosis stage assessment. Medical Physics 46(5), 2298–2309 (2019)

[159] Gennisson, J.L., Deffieux, T., Fink, M., Tanter, M.: Ultrasound elastography: Principles and techniques. Diagnostic and Interventional Imaging 94(5), 487–495 (2013)

[160] Gerlach, S., Kuhlemann, I., Jauer, P., Bruder, R., Ernst, F., Fürweger, C., Schlaefer, A.: Robotic ultrasound-guided SBRT of the prostate: Feasibility with respect to plan quality. International Journal of Computer Assisted Radiology and Surgery 12(1), 149–159 (2017)

[161] Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: Continual prediction with LSTM. Neural Computation 12(10), 2451–2471 (2000)

[162] Gessert, N.: Deep learning with multi-dimensional medical image data. Ph.D. thesis, Technische Universität Hamburg (2020)

[163] Gessert, N., Bengs, M., Krüger, J., Opfer, R., Ostwaldt, A.C., Manogaran, P., Schippling, S., Schlaefer, A.: 4D deep learning for multiple sclerosis lesion activity segmentation. arXiv preprint arXiv:2004.09216 (2020)

[164] Gessert, N., Bengs, M., Schlüter, M., Schlaefer, A.: Deep learning with 4D spatio-temporal data representations for OCT-based force estimation. Medical Image Analysis 64, 101730 (2020)

[165] Gessert, N., Bengs, M., Wittig, L., Drömann, D., Keck, T., Schlaefer, A., Ellebrecht, D.B.: Deep transfer learning methods for colon cancer classification in confocal laser microscopy images. International Journal of Computer Assisted Radiology and Surgery 14, 1837–1845 (2019)

[166] Gessert, N., Beringhoff, J., Otte, C., Schlaefer, A.: Force estimation from OCT volumes using 3D CNNs. International Journal of Computer Assisted Radiology and Surgery 13(7), 1073–1082 (2018)

[167] Gessert, N., Gromniak, M., Schlüter, M., Schlaefer, A.: Two-path 3D CNNs for calibration of system parameters for OCT-based motion compensation. In: SPIE Medical Imaging 2019: Image-Guided Procedures, Robotic Interventions, and Modeling (2018)

[168] Gessert, N., Gromniak, M., Schlüter, M., Schlaefer, A.: Two-path 3D CNNs for calibration of system parameters for OCT-based motion compensation. In: Medical Imaging 2019: Image-Guided Procedures, Robotic Interventions, and Modeling. vol. 10951, p. 1095108. International Society for Optics and Photonics (2019)

[169] Gessert, N., Priegnitz, T., Saathoff, T., Antoni, S.T., Meyer, D., Hamann, M.F., Jünemann, K.P., Otte, C., Schlaefer, A.: Needle tip force estimation using an oct fiber and a fused Convgru-CNN architecture. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part IV 11. pp. 222–229. Springer (2018)

[170] Gessert, N., Priegnitz, T., Saathoff, T., Antoni, S.T., Meyer, D., Hamann, M.F., Jünemann, K.P., Otte, C., Schlaefer, A.: Spatio-temporal deep learning models for tip force estimation during needle insertion. International Journal of Computer Assisted Radiology and Surgery 14(9), 1485–1493 (2019)

[171] Gessert, N., Schlüter, M., Schlaefer, A.: A deep learning approach for pose estimation from volumetric OCT data. Medical Image Analysis 46, 162–179 (2018)

[172] Gholami, P., Lakshminarayanan, V.: Optical coherence tomography image retinal database. Ann Arbor, MI: Inter-University Consortium for Political and Social Research pp. 02–20 (2019)

[173] Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. pp. 249–256. JMLR Workshop and Conference Proceedings (2010)

[174] Glorot, X., Bordes, A., Bengio, Y.: Deep Sparse Rectifier Neural Networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. pp. 315–323 (2011)

[175] Gomariz, A., Li, W., Ozkan, E., Tanner, C., Goksel, O.: Siamese networks with location prior for landmark tracking in liver ultrasound sequences. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019). pp. 1757–1760. IEEE (2019)

[176] Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)

[177] Goodfellow, I.J., Vinyals, O., Saxe, A.M.: Qualitatively characterizing neural network optimization problems. arXiv preprint arXiv:1412.6544 (2014)

[178] Gossage, K.W., Smith, C.M., Kanter, E.M., Hariri, L.P., Stone, A.L., Rodriguez, J.J., Williams, S.K., Barton, J.K.: Texture analysis of speckle in optical coherence tomography images of tissue phantoms. Physics in Medicine & Biology 51(6), 1563 (2006)

[179] Gour, N., Khanna, P.: Speckle denoising in optical coherence tomography images using residual deep convolutional neural network. Multimedia Tools and Applications 79(21), 15679–15695 (2020)

[180] Graves, A.: Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850 (2013)

[181] Graves, A., Mohamed, A.r., Hinton, G.: Speech recognition with deep recurrent neural networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-2013). pp. 6645–6649. IEEE (2013)

[182] Greenleaf, J.F., Fatemi, M., Insana, M.: Selected methods for imaging elastic properties of biological tissues. Annual Review of Biomedical Engineering 5(1), 57–78 (2003)

[183] Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J.: LSTM: A search space odyssey. IEEE Transactions on Neural Networks and Learning Systems 28(10), 2222–2232 (2016)

[184] Grube, S., Bengs, M., Neidhardt, M., Latus, S., Schlaefer, A.: Ultrasound shear wave velocity estimation in a small field of view via spatio-temporal deep learning. In: Medical Imaging 2023: Image Processing. vol. 12464, pp. 491–495. SPIE (2023)

[185] Guayacán, L.C., Rangel, E., Martínez, F.: Towards understanding spatio-temporal parkinsonian patterns from salient regions of a 3D convolutional network. In: 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). pp. 3688–3691. IEEE (2020)

[186] Gunalan, A., Mattos, L.S.: Towards OCT-guided endoscopic laser surgery - A review. Diagnostics 13(4), 677 (2023)

[187] Gupta, A., Anpalagan, A., Guan, L., Khwaja, A.S.: Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. Array 10, 100057 (2021)

[188] Hahn, P., Carrasco-Zevallos, O., Cunefare, D., Migacz, J., Farsiu, S., Izatt, J.A., Toth, C.A.: Intrasurgical human retinal imaging with manual instrument tracking using a microscope-integrated spectral-domain optical coherence tomography device. Translational Vision Science & Technology 4(4), 1–1 (2015)

[189] Han, Z., Singh, M., Aglyamov, S.R., Liu, C.H., Nair, A., Raghunathan, R., Wu, C., Li, J., Larin, K.V.: Quantifying tissue viscoelasticity using optical coherence elastography and the Rayleigh wave model. Journal of Biomedical Optics 21(9), 090504 (2016)

[190] Hangiandreou, N.J.: AAPM/RSNA physics tutorial for residents: Topics in US: B-mode US: Basic concepts and new technology. Radiographics 23(4), 1019–1033 (2003)

[191] Hanin, B., Rolnick, D.: How to start training: The effect of initialization and architecture. Advances in Neural Information Processing Systems 31 (2018)

[192] Hara, K., Kataoka, H., Satoh, Y.: Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and imagenet? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6546–6555 (2018)

[193] Harris, E.J., Miller, N.R., Bamber, J.C., Evans, P.M., Symonds-Tayler, J.R.N.: Performance of ultrasound based measurement of 3D displacement using a curvilinear probe for organ motion tracking. Physics in Medicine & Biology 52(18), 5683 (2007)

[194] Harris, E.J., Miller, N.R., Bamber, J.C., Symonds-Tayler, J.R.N., Evans, P.M.: Speckle tracking in a phantom and feature-based tracking in liver in the presence of respiratory motion using 4D ultrasound. Physics in Medicine & Biology 55(12), 3363 (2010)

[195] He, J., Shen, C., Chen, Y., Huang, Y., Wu, J.: FPSN-FNCC: An accurate and fast motion tracking algorithm in 3D ultrasound for image-guided interventions. Physics in Medicine & Biology 66(15), 155012 (2021)

[196] He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1026–1034 (2015)

[197] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)

[198] Hepburn, M.S., Foo, K.Y., Wijesinghe, P., Munro, P.R., Chin, L., Kennedy, B.F.: Speckle-dependent accuracy in phase-sensitive optical coherence tomography. Optics Express 29(11), 16950–16968 (2021)

[199] Heywang, W., Lubitz, K., Wersing, W.: Piezoelectricity: Evolution and future of a technology, vol. 114. Springer Science & Business Media (2008)

[200] Hihi, S., Bengio, Y.: Hierarchical recurrent neural networks for long-term dependencies. Advances in Neural Information Processing Systems 8 (1995)

[201] Hild, F., Roux, S.: Digital image correlation: From displacement measurement to identification of elastic properties–A review. Strain 42(2), 69–80 (2006)

[202] Hochreiter, S.: Untersuchungen zu dynamischen neuronalen Netzen. Diploma, Technische Universität München 91(1) (1991)

[203] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation 9(8), 1735–1780 (1997)

[204] Horn, B.K., Schunck, B.G.: Determining optical flow. Artificial Intelligence 17(1-3), 185–203 (1981)

[205] Hornik, K.: Approximation capabilities of multilayer feedforward networks. Neural Networks 4(2), 251–257 (1991)

[206] Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks 2(5), 359–366 (1989)

[207] Hounsfield, G.N.: Computerized transverse axial scanning (tomography): Part 1. Description of system. The British Journal of Radiology 46(552), 1016–1022 (1973)

[208] Huang, C., Song, P., Mellema, D.C., Gong, P., Lok, U.W., Tang, S., Ling, W., Meixner, D.D., Urban, M.W., Manduca, A., et al.: Three-dimensional shear wave elastography on conventional ultrasound scanners with external vibration. Physics in Medicine & Biology 65(21), 215009 (2020)

[209] Huang, D., Swanson, E.A., Lin, C.P., Schuman, J.S., Stinson, W.G., Chang, W., Hee, M.R., Flotte, T., Gregory, K., Puliafito, C.A.: Optical coherence tomography. Science 254(5035), 1178–1181 (1991)

[210] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4700–4708 (2017)

[211] Huang, P., Yu, G., Lu, H., Liu, D., Xing, L., Yin, Y., Kovalchuk, N., Xing, L., Li, D.: Attention-aware fully convolutional neural network with convolutional long short-term memory network for ultrasound-based motion tracking. Medical Physics 46(5), 2275–2285 (2019)

[212] Huang, Y., He, J., Wu, X., Zhao, X., Wu, J.: Tracking 3D ultrasound anatomical landmarks via three orthogonal plane-based scale discriminative correlation filter network. Medical Physics 48(5), 2127–2135 (2021)

[213] Humphrey, J.D.: Continuum biomechanics of soft biological tissues. Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 459(2029), 3–46 (2003)

[214] Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2462–2470 (2017)

[215] Inoue, Y., Kokudo, N.: Elastography for hepato-biliary-pancreatic surgery. Surgery Today 44(10), 1793–1800 (2014)

[216] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning (ICML-15) (2015)

[217] Ipsen, S., Bruder, R., OâBrien, R., Keall, P.J., Schweikard, A., Poulsen, P.R.: Online 4D ultrasound guidance for real-time motion compensation by MLC tracking. Medical Physics 43(10), 5695–5704 (2016)

[218] Ipsen, S., Wulff, D., Kuhlemann, I., Schweikard, A., Ernst, F.: Towards automated ultrasound imaging-robotic image acquisition in liver and prostate for long-term motion monitoring. Physics in Medicine & Biology 66(9), 094002 (2021)

[219] Irsch, K., Lee, S., Bose, S.N., Kang, J.U.: Motion-compensated optical coherence tomography using envelope-based surface detection and Kalman-based prediction. In: Advanced Biomedical and Clinical Diagnostic and Surgical Guidance Systems XVI. vol. 10484, p. 104840Q. International Society for Optics and Photonics (2018)

[220] Itoh, A., Ueno, E., Tohno, E., Kamma, H., Takahashi, H., Shiina, T., Yamakawa, M., Matsumura, T.: Breast disease: Clinical application of US elastography for diagnosis. Radiology 239(2), 341–350 (2006)

[221] Jaeger, H.: Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach (2002)

[222] Jähne, B.: Digital Image Processing. Springer, 6. rev. and extended ed. edn. (2005)

[223] Jakubović, A., Velagić, J.: Image feature matching and object detection using brute-force matchers. In: 2018 International Symposium ELMAR. pp. 83–86. IEEE (2018)

[224] Ji, S., Xu, W., Yang, M., Yu, K.: 3D convolutional neural networks for human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(1), 221–231 (2012)

[225] Jin, F.Q., Carlson, L.C., Feltovich, H., Hall, T.J., Palmeri, M.L.: SweiNet: Deep Learning Based Uncertainty Quantification for Ultrasound Shear Wave Elasticity Imaging. arXiv preprint arXiv:2203.10678 (2022)

[226] Jöhl, A., Ehrbar, S., Guckenberger, M., Klöck, S., Meboldt, M., Zeilinger, M., Tanadini-Lang, S., Schmid Daners, M.: Performance comparison of prediction filters for respiratory motion tracking in radiotherapy. Medical Physics 47(2), 643–650 (2020)

[227] Jozefowicz, R., Zaremba, W., Sutskever, I.: An empirical exploration of recurrent network architectures. In: International Conference on Machine Learning. pp. 2342–2350. PMLR (2015)

[228] Kagadis, G.C., Drazinos, P., Gatos, I., Tsantis, S., Papadimitroulas, P., Spiliopoulos, S., Karnabatidis, D., Theotokas, I., Zoumpoulis, P., Hazle, J.D.: Deep learning networks on chronic liver disease assessment with fine-tuning of shear wave elastography image sequences. Physics in Medicine & Biology 65(21), 215027 (2020)

[229] Kahrs, L.A., Raczkowsky, J., Werner, M., Knapp, F.B., Mehrwald, M., Hering, P., Schipper, J., Klenzner, T., Wörn, H.: Visual servoing of a laser ablation based cochleostomy. In: Medical Imaging 2008: Visualization, Image-Guided Procedures, and Modeling. vol. 6918, pp. 789–799. SPIE (2008)

[230] Karaoğlu, O., Bilge, H.Ş., Uluer, İ.: Removal of speckle noises from ultrasound images using five different deep learning networks. Engineering Science and Technology, an International Journal 29, 101030 (2022)

[231] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1725–1732 (2014)

[232] Kasaragod, D.K., Lu, Z., Smith, L.E., Matcher, S.J.: Speckle texture analysis of optical coherence tomography images. In: Speckle 2010: Optical Metrology. vol. 7387, pp. 553–559. SPIE (2010)

[233] Kawaguchi, K., Kaelbling, L.P., Bengio, Y.: Generalization in deep learning. arXiv preprint arXiv:1710.05468 (2017)

[234] Keall, P., Nguyen, D., OâBrien, R., Zhang, P., Happersett, L., Bertholet, J., Poulsen, P.: A review of real-time 3D IGRT on standard-equipped cancer radiotherapy systems: Are we at the tipping point for the era of real-time radiotherapy? International Journal of Radiation Oncology, Biology, Physics 102(4), 922 (2018)

[235] Keller, B., Draelos, M., Tang, G., Farsiu, S., Kuo, A.N., Hauser, K., Izatt, J.A.: Real-time corneal segmentation and 3D needle tracking in intrasurgical OCT. Biomedical Optics Express 9(6), 2716–2732 (2018)

[236] Kennedy, B.F., Kennedy, K.M., Sampson, D.D.: A review of optical coherence elastography: Fundamentals, techniques and prospects. IEEE Journal of Selected Topics in Quantum Electronics 20(2), 272–288 (2013)

[237] Kennedy, K.M., Ford, C., Kennedy, B.F., Bush, M.B., Sampson, D.D.: Analysis of mechanical contrast in optical coherence elastography. Journal of Biomedical Optics 18(12), 121508 (2013)

[238] Kibria, M.G., Rivaz, H.: Gluenet: Ultrasound elastography using convolutional neural network. In: Simulation, Image Processing, and Ultrasound Systems for Assisted Diagnosis and Navigation: International Workshops, POCUS 2018, BIVPCS 2018, CuRIOUS 2018, and CPM 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16–20, 2018, Proceedings, pp. 21–28 (2018)

[239] Kijanka, P., Ambrozinski, L., Urban, M.W.: Two point method for robust shear wave phase velocity dispersion estimation of viscoelastic materials. Ultrasound in Medicine & Biology 45(9), 2540–2553 (2019)

[240] Kijanka, P., Urban, M.: Fast local phase velocity-based imaging: Shear wave particle velocity and displacement motion study. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 67(3), 526–537 (2019)

[241] Kijanka, P., Urban, M.W.: Local phase velocity based imaging: A new technique used for ultrasound shear wave elastography. IEEE Transactions on Medical Imaging 38(4), 894–908 (2018)

[242] Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)

[243] Kirby, M.A., Pelivanov, I., Song, S., Ambrozinski, L., Yoon, S.J., Gao, L., Li, D., Shen, T.T., Wang, R.K., O'Donnell, M.: Optical coherence elastography in ophthalmology. Journal of Biomedical Optics 22(12), 121720 (2017)

[244] Kirkpatrick, S.J., Wang, R.K., Duncan, D.D.: OCT-based elastography for large and small deformations. Optics Express 14(24), 11585–11597 (2006)

[245] Kirtane, T.S., Wagh, M.S.: Endoscopic optical coherence tomography (OCT): Advances in gastrointestinal imaging. Gastroenterology Research and Practice 2014 (2014)

[246] Klein, T., Wieser, W., Reznicek, L., Neubauer, A., Kampik, A., Huber, R.: Multi-mhz retinal oct. Biomedical Optics Express 4(10), 1890–1908 (2013)

[247] Kraus, M.F., Potsaid, B., Mayer, M.A., Bock, R., Baumann, B., Liu, J.J., Hornegger, J., Fujimoto, J.G.: Motion correction in optical coherence tomography volumes on a per A-scan basis using orthogonal scan patterns. Biomedical Optics Express 3(6), 1182–1199 (2012)

[248] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Communications of the ACM 60(6), 84–90 (2017)

[249] Van der Kruk, E., Reijne, M.M.: Accuracy of human motion capture systems for sport applications; state-of-the-art review. European Journal of Sport Science 18(6), 806–819 (2018)

[250] Krupinski, E.A., Jiang, Y.: Anniversary paper: Evaluation of medical imaging systems. Medical Physics 35(2), 645–659 (2008)

[251] Kuglin, C.D.: The phase correlation image alignment method. In: Proc. International Conference on Cybernetics and Society, 1975. pp. 163–165 (1975)

[252] KZ Tehrani, A., Mirzaei, M., Rivaz, H.: Semi-supervised training of optical flow convolutional neural networks in ultrasound elastography. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part III 23. pp. 504–513. Springer (2020)

[253] Lan, G., Aglyamov, S.R., Larin, K.V., Twa, M.D.: In vivo human corneal shear-wave optical coherence elastography. Optometry and Vision Science 98(1), 58 (2021)

[254] Laves, M.H., Ihler, S., Kahrs, L.A., Ortmaier, T.: Deep-learning-based 2.5 D flow field estimation for maximum intensity projections of 4D optical coherence tomography. In: Medical Imaging 2019: Image-Guided Procedures, Robotic Interventions, and Modeling. vol. 10951, p. 109510R. International Society for Optics and Photonics (2019)

[255] Laves, M.H., Schoob, A., Kahrs, L.A., Pfeiffer, T., Huber, R., Ortmaier, T.: Feature tracking for automated volume of interest stabilization on 4D-OCT images. In: Medical Imaging 2017: Image-Guided Procedures, Robotic Interventions, and Modeling. vol. 10135, p. 101350W. International Society for Optics and Photonics (2017)

[256] Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3361–3368. IEEE (2011)

[257] Leal-Taixé, L., Canton-Ferrer, C., Schindler, K.: Learning by tracking: Siamese CNN for robust target association. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 33–40 (2016)

[258] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521(7553), 436–444 (2015)

[259] LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Computation 1(4), 541–551 (1989)

[260] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)

[261] Lee, M.H., Smyser, C.D., Shimony, J.S.: Resting-state fMRI: A review of methods and clinical applications. American Journal of Neuroradiology 34(10), 1866–1872 (2013)

[262] Lee, S., Jung, Y., Bae, Y.: Clinical application of a color map pattern on shear-wave elastography for invasive breast cancer. Surgical Oncology 25(1), 44–48 (2016)

[263] van de Leemput, S.C., Prokop, M., van Ginneken, B., Manniesing, R.: Stacked bidirectional convolutional LSTMs for deriving 3D non-contrast ct from spatiotemporal 4D CT. IEEE Transactions on Medical Imaging 39(4), 985–996 (2019)

[264] Lei, Y., Fu, Y., Wang, T., Liu, Y., Patel, P., Curran, W.J., Liu, T., Yang, X.: 4D-CT deformable image registration using multiscale unsupervised deep learning. Physics in Medicine & Biology 65(8), 085003 (2020)

[265] Leitgeb, R., Hitzenberger, C., Fercher, A.F.: Performance of fourier domain vs. time domain optical coherence tomography. Optics Express 11(8), 889–894 (2003)

[266] Lewis, J.P.: Fast template matching. In: Vision Interface. vol. 95, pp. 15–19 (1995)

[267] Li, H., Bhatt, M., Qu, Z., Zhang, S., Hartel, M.C., Khademhosseini, A., Cloutier, G.: Deep learning in ultrasound elastography imaging: A review. Medical Physics 49(9), 5993–6018 (2022)

[268] Li, H., Flé, G., Bhatt, M., Qu, Z., Ghazavi, S., Yazdani, L., Bosio, G., Rafati, I., Cloutier, G.: Viscoelasticity imaging of biological tissues and single cells using shear wave propagation. Frontiers in Physics 9, 666192 (2021)

[269] Li, J., Chen, J., Tang, Y., Wang, C., Landman, B.A., Zhou, S.K.: Transforming medical imaging with Transformers? A comparative review of key properties, current progresses, and future perspectives. Medical Image Analysis p. 102762 (2023)

[270] Li, K., Li, X., Wang, Y., Wang, J., Qiao, Y.: CT-net: Channel tensorization network for video classification. arXiv preprint arXiv:2106.01603 (2021)

[271] Li, M.D., Arun, N.T., Gidwani, M., Chang, K., Deng, F., Little, B.P., Mendoza, D.P., Lang, M., Lee, S.I., OâShea, A., et al.: Automated assessment and tracking of COVID-19 pulmonary disease severity on chest radiographs using convolutional siamese neural networks. Radiology: Artificial Intelligence 2(4), e200079 (2020)

[272] Li, M.D., Chang, K., Bearce, B., Chang, C.Y., Huang, A.J., Campbell, J.P., Brown, J.M., Singh, P., Hoebel, K.V., Erdoğmuş, D., et al.: Siamese neural networks for continuous disease severity evaluation and change detection in medical imaging. NPJ Digital Medicine 3(1), 1–9 (2020)

[273] Li, W., Lin, X., Chen, X.: Detecting Alzheimer's disease Based on 4D fMRI: An exploration under deep learning framework. Neurocomputing 388, 280–287 (2020)

[274] Li, Y., Moon, S., Chen, J.J., Zhu, Z., Chen, Z.: Ultrahigh-sensitive optical coherence elastography. Light: Science & Applications 9(1), 1–10 (2020)

[275] Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J.: A survey of convolutional neural networks: Analysis, applications, and prospects. IEEE Transactions on Neural Networks and Learning Systems (2021)

[276] Lin, H., Shi, C., Wang, B., Chan, M.F., Tang, X., Ji, W.: Towards real-time respiratory motion prediction based on long short-term memory neural networks. Physics in Medicine & Biology 64(8), 085010 (2019)

[277] Lin, J., Gan, C., Han, S.: Tsm: Temporal shift module for efficient video understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7083–7093 (2019)

[278] Lin, M., Chen, Q., Yan, S.: Network in network. In: Conference on Learning Representations (ICLR) (2013)

[279] Lindenmaier, A.A., Conroy, L., Farhat, G., DaCosta, R.S., Flueraru, C., Vitkin, I.A.: Texture analysis of optical coherence tomography speckle for characterizing biological tissues in vivo. Optics Letters 38(8), 1280–1282 (2013)

[280] Liney, G.P., Whelan, B., Oborn, B., Barton, M., Keall, P.: MRI-linear accelerator radiotherapy systems. Clinical Oncology 30(11), 686–691 (2018)

[281] Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A., van Ginneken, B., Sánchez, C.I.: A Survey on Deep Learning in Medical Image Analysis. arXiv preprint arXiv:1702.05747 (2017)

[282] Liu, F., Liu, D., Tian, J., Xie, X., Yang, X., Wang, K.: Cascaded one-shot deformable convolutional neural networks: Developing a deep learning model for respiratory motion estimation in ultrasound sequences. Medical Image Analysis 65, 101793 (2020)

[283] Liu, H.C., Kijanka, P., Urban, M.W.: Four-dimensional (4D) phase velocity optical coherence elastography in heterogeneous materials and biological tissue. Biomedical Optics Express 11(7), 3795–3817 (2020)

[284] Liu, S., Yang, B., Wang, Y., Tian, J., Yin, L., Zheng, W.: 2D/3D multimode medical image registration based on normalized cross-correlation. Applied Sciences 12(6), 2828 (2022)

[285] Liu, X., Song, L., Liu, S., Zhang, Y.: A review of deep-learning-based medical image segmentation methods. Sustainability 13(3), 1224 (2021)

[286] Liu, X., Faes, L., Kale, A.U., Wagner, S.K., Fu, D.J., Bruynseels, A., Mahendiran, T., Moraes, G., Shamdas, M., Kern, C., et al.: A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: A systematic review and meta-analysis. The Lancet Digital Health 1(6), e271–e297 (2019)

[287] Liu, X., Huang, Y., Kang, J.U.: Distortion-free freehand-scanning OCT implemented with real-time scanning speed variance correction. Optics Express 20(15), 16567–16583 (2012)

[288] Loehr, J.A., Wang, S., Cully, T.R., Pal, R., Larina, I.V., Larin, K.V., Rodney, G.G.: NADPH oxidase mediates microtubule alterations and diaphragm dysfunction in dystrophic mice. eLife 7, e31732 (2018)

[289] Loupas, T., Powers, J., Gill, R.W.: An axial velocity estimator for ultrasound blood flow imaging, based on a full evaluation of the Doppler equation by means of a two-dimensional autocorrelation approach. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 42(4), 672–688 (1995)

[290] Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision. vol. 2, pp. 1150–1157. Ieee (1999)

[291] Lu, C., Hirsch, M., Scholkopf, B.: Flexible spatio-temporal networks for video prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6523–6531 (2017)

[292] Lu, G., Fei, B.: Medical hyperspectral imaging: A review. Journal of Biomedical Optics 19(1), 010901–010901 (2014)

[293] Lu, N., Wu, Y., Feng, L., Song, J.: Deep learning for fall detection: Three-dimensional CNN combined with LSTM on video kinematic data. IEEE Journal of Biomedical and Health Informatics 23(1), 314–323 (2018)

[294] Lu, X.Q., Shanmugham, L.N., Mahadevan, A., Nedea, E., Stevenson, M.A., Kaplan, I., Wong, E.T., La Rosa, S., Wang, F., Berman, S.M.: Organ deformation and dose coverage in robotic respiratory-tracking radiotherapy. International Journal of Radiation Oncology - Biology - Physics 71(1), 281–289 (2008)

[295] Lubana, E.S., Dick, R., Tanaka, H.: Beyond BatchNorm: Towards a unified understanding of normalization in deep learning. Advances in Neural Information Processing Systems 34, 4778–4791 (2021)

[296] Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI'81: 7th International Joint Conference on Artificial intelligence. vol. 2, pp. 674–679 (1981)

[297] Luo, C., Yuille, A.L.: Grouped spatial-temporal aggregation for efficient action recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5512–5521 (2019)

[298] Luo, P., Wang, X., Shao, W., Peng, Z.: Towards understanding regularization in batch normalization. arXiv preprint arXiv:1809.00846 (2018)

[299] Ma, Y., Chen, X., Zhu, W., Cheng, X., Xiang, D., Shi, F.: Speckle noise reduction in optical coherence tomography images based on edge-sensitive cGAN. Biomedical Optics Express 9(11), 5129–5146 (2018)

[300] Maier, A., Steidl, S., Christlein, V., Hornegger, J.: Medical imaging systems: An introductory guide. Springer (2018)

[301] Maksuti, E., Widman, E., Larsson, D., Urban, M.W., Larsson, M., Bjällmark, A.: Arterial stiffness estimation by shear wave elastography: Validation in phantoms with mechanical testing. Ultrasound in Medicine & Biology 42(1), 308–321 (2016)

[302] Manduca, A., Oliphant, T.E., Dresner, M.A., Mahowald, J., Kruse, S.A., Amromin, E., Felmlee, J.P., Greenleaf, J.F., Ehman, R.L.: Magnetic resonance elastography: Non-invasive mapping of tissue elasticity. Medical Image Analysis 5(4), 237–254 (2001)

[303] Marvasti-Zadeh, S.M., Cheng, L., Ghanei-Yakhdan, H., Kasaei, S.: Deep learning for visual tracking: A comprehensive survey. IEEE Transactions on Intelligent Transportation Systems (2021)

[304] McAleavey, S.A., Osapoetra, L.O., Langdon, J.: Shear wave arrival time estimates correlate with local speckle pattern. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 62(12), 2054–2067 (2015)

[305] McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics 5(4), 115–133 (1943)

[306] McKnight, A.L., Kugel, J.L., Rossman, P.J., Manduca, A., Hartmann, L.C., Ehman, R.L.: MR elastography of breast cancer: Preliminary results. American Journal of Roentgenology 178(6), 1411–1417 (2002)

[307] Mezheritsky, T., Romaguera, L.V., Kadoury, S.: 3D ultrasound generation from partial 2D observations using fully convolutional and spatial transformation networks. In: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI). pp. 1808–1811. IEEE (2020)

[308] Mieling, R., Sprenger, J., Latus, S., Bargsten, L., Schlaefer, A.: A novel optical needle probe for deep learning-based tissue elasticity characterization. Current Directions in Biomedical Engineering 7(1), 21–25 (2021)

[309] Mikołajczyk-Korniak, N., Tronina, O., Ślubowska, K., Perkowska-Ptasińska, A., Pacholczyk, M., Bączkowska, T., Durlik, M.: Dynamic elastography in diagnostics of liver fibrosis in patients after liver transplantation due to cirrhosis in the course of hepatitis C. In: Transplantation Proceedings. vol. 48, pp. 1725–1729. Elsevier (2016)

[310] Minakaran, N., de Carvalho, E.R., Petzold, A., Wong, S.H.: Optical coherence tomography (OCT) in neuro-ophthalmology. Eye 35(1), 17–32 (2021)

[311] Mirzaei, M., Asif, A., Fortin, M., Rivaza, H.: Spatio-temporal normalized cross-correlation for estimation of the displacement field in ultrasound elastography. Ultrasonics 102 (04 2018)

[312] Mishra, S., Tripathy, H.K., Acharya, B.: A precise analysis of deep learning for medical image processing. Bio-Inspired Neurocomputing pp. 25–41 (2021)

[313] Mitchell, T.M.: Machine Learning. McGraw-Hill (1997)

[314] Mohri, M., Rostamizadeh, A., Talwalkar, A.: Foundations of Machine Learning. MIT Press (2012)

[315] Montaldo, G., Tanter, M., Bercoff, J., Benech, N., Fink, M.: Coherent plane-wave compounding for very high frame rate ultrasonography and transient elastography. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 56(3), 489–506 (2009)

[316] Moradi, R., Berangi, R., Minaei, B.: A survey of regularization strategies for deep models. Artificial Intelligence Review 53(6), 3947–3986 (2020)

[317] Mozer, M.C.: Induction of multiscale temporal structure. Advances in Neural Information Processing Systems 4 (1991)

[318] Murphy, K.P.: Machine learning: A probabilistic perspective. MIT press (2012)

[319] Mutegeki, R., Han, D.S.: A CNN-LSTM approach to human activity recognition. In: 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC). pp. 362–366. IEEE (2020)

[320] Myronenko, A., Yang, D., Buch, V., Xu, D., Ihsani, A., Doyle, S., Michalski, M., Tenenholtz, N., Roth, H.: 4D CNN for semantic segmentation of cardiac volumetric sequences. In: International Workshop on Statistical Atlases and Computational Models of the Heart. pp. 72–80. Springer (2019)

[321] Nandy, A., Haldar, S., Banerjee, S., Mitra, S.: A survey on applications of siamese neural networks in computer vision. In: 2020 International Conference for Emerging Technology (INCET). pp. 1–5. IEEE (2020)

[322] Neidhardt, M., Bengs, M., Latus, S., Gerlach, S., Cyron, C.J., Sprenger, J., Schlaefer, A.: Ultrasound shear wave elasticity imaging with spatio-temporal deep learning. IEEE Transactions on Biomedical Engineering 69(11), 3356–3364 (2022)

[323] Neidhardt, M., Bengs, M., Latus, S., Schlüter, M., Saathoff, T., Schlaefer, A.: Deep learning for high speed optical coherence elastography. In: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI). pp. 1583–1586. IEEE (2020)

[324] Neidhardt, M., Bengs, M., Latus, S., Schlüter, M., Saathoff, T., Schlaefer, A.: 4D deep learning for real-time volumetric optical coherence elastography. International Journal of Computer Assisted Radiology and Surgery 16, 23–27 (2021)

[325] Neidhardt, M., Gessert, N., Gosau, T., Kemmling, J., Feldhaus, S., Schumacher, U., Schlaefer, A.: Force estimation from 4D OCT data in a human tumor xenograft mouse model. Current Directions in Biomedical Engineering 6(1) (2020)

[326] Neidhardt, M., Mieling, R., Bengs, M., Schlaefer, A.: Optical force estimation for interactions between tool and soft tissues. Scientific Reports 13(1), 506 (2023)

[327] Nenadic, I., Urban, M.W., Qiang, B., Chen, S., Greenleaf, J.: Model-free quantification of shear wave velocity and attenuation in tissues and its in vivo application. The Journal of the Acoustical Society of America 134(5), 4011–4011 (2013)

[328] Ng, M., Brown, E., Williams, A., Chao, M., Lawrentschuk, N., Chee, R.: Fiducial markers and spacers in prostate radiotherapy: Current applications. BJU International 113, 13–20 (2014)

[329] Nguyen, T., Arnal, B., Song, S., Huang, Z., Wang, R., OâDonnell, M.: Shear wave elastography using amplitude-modulated acoustic radiation force and phase-sensitive optical coherence tomography. Journal of Biomedical Optics 20(1), 016001 (2015)

[330] Nguyen, T.M., Song, S., Arnal, B., Huang, Z., OâDonnell, M., Wang, R.K.: Visualizing ultrasonically induced shear wave propagation using phase-sensitive optical coherence tomography for dynamic elastography. Optics Letters 39(4), 838–841 (2014)

[331] Niemeijer, M., Garvin, M.K., Lee, K., van Ginneken, B., Abràmoff, M.D., Sonka, M.: Registration of 3D spectral OCT volumes using 3D SIFT feature point matching. In: Medical Imaging 2009: Image Processing. vol. 7259, pp. 520–527. SPIE (2009)

[332] Nightingale, K., McAleavey, S., Trahey, G.: Shear-wave generation using acoustic radiation force: In vivo and ex vivo results. Ultrasound in Medicine & Biology 29(12), 1715–1723 (2003)

[333] Nightingale, K.R., Rouze, N.C., Rosenzweig, S.J., Wang, M.H., Abdelmalek, M.F., Guy, C.D., Palmeri, M.L.: Derivation and analysis of viscoelastic properties in human liver: Impact of frequency on fibrosis and steatosis staging. IEEE transactions on ultrasonics, ferroelectrics, and frequency control 62(1), 165–175 (2015)

[334] Nolan, C.P., Forde, E.J.: A review of the use of fiducial markers for image-guided bladder radiotherapy. Acta Oncologica 55(5), 533–538 (2016)

[335] Nouri, D., Rothberg, A.: Liver ultrasound tracking using a learned distance metric. In: Proc. MICCAI Workshop: Challenge on Liver Ultrasound Tracking. pp. 5–12 (2015)

[336] Ntatsis, K., Brea, L.S., De Jesus, D.A., Barbosa-Breda, J., van Walsum, T., Bennink, E., Klein, S.: Motion correction in retinal optical coherence tomography imaging using deep learning registration. In: Medical Imaging 2022: Image Processing. vol. 12032, pp. 334–343. SPIE (2022)

[337] Oliveira, F.P., Tavares, J.M.R.: Medical image registration: A review. Computer Methods in Biomechanics and Biomedical Engineering 17(2), 73–93 (2014)

[338] Ondrašovič, M., Tarábek, P.: Siamese visual object tracking: A survey. IEEE Access 9, 110149–110172 (2021)

[339] O'Neill, A.G., Jain, S., Hounsell, A.R., O'Sullivan, J.M.: Fiducial marker guided prostate radiotherapy: A review. The British Journal of Radiology 89(1068), 20160296 (2016)

[340] Ophir, J., Cespedes, I., Ponnekanti, H., Yazdi, Y., Li, X.: Elastography: A quantitative method for imaging the elasticity of biological tissues. Ultrasonic Imaging 13(2), 111–134 (1991)

[341] Øye, O.K., Wein, W., Ulvang, D.M., Matre, K., Viola, I.: Real time image-based tracking of 4D ultrasound data. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 447–454. Springer (2012)

[342] Ozturk, A., Grajo, J.R., Dhyani, M., Anthony, B.W., Samir, A.E.: Principles of ultrasound elastography. Abdominal Radiology 43(4), 773–785 (2018)

[343] Palmeri, M.L., Wang, M.H., Dahl, J.J., Frinkley, K.D., Nightingale, K.R.: Quantifying hepatic shear modulus in vivo using acoustic radiation force. Ultrasound in Medicine & Biology 34(4), 546–558 (2008)

[344] Pan, B., Qian, K., Xie, H., Asundi, A.: Two-dimensional digital image correlation for in-plane displacement and strain measurement: A review. Measurement Science and Technology 20(6), 062001 (2009)

[345] Park, J., Kang, J.B., Chang, J.H., Yoo, Y.: Speckle reduction techniques in medical ultrasound imaging. Biomedical Engineering Letters 4, 32–40 (2014)

[346] Parker, K.J., Doyley, M.M., Rubens, D.J.: Imaging the elastic properties of tissue: The 20 year perspective. Physics in Medicine & Biology 56(1), R1 (2010)

[347] Parker, K.J., Taylor, L.S., Gracewski, S., Rubens, D.J.: A unified view of imaging the elastic properties of tissue. The Journal of the Acoustical Society of America 117(5), 2705–2712 (2005)

[348] Parmar, H., Nutter, B., Long, R., Antani, S., Mitra, S.: Spatiotemporal feature extraction and classification of Alzheimerâs disease using deep learning 3D-CNN for fMRI data. Journal of Medical Imaging 7(5), 056001–056001 (2020)

[349] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems 32 (2019)

[350] Patra, A., Huang, W., Noble, J.A.: Learning spatio-temporal aggregation for fetal heart analysis in ultrasound video. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3. pp. 276–284. Springer (2017)

[351] Peker, A., Balci, P., Basara Akin, I., Özgül, H.A., Aksoy, S.Ö., Gürel, D.: Shear-wave elastography-guided core needle biopsy for the determination of breast cancer molecular subtype. Journal of Ultrasound in Medicine 40(6), 1183–1192 (2021)

[352] Pereira, C., Dighe, M., Alessio, A.: Comparison of machine learned approaches for thyroid nodule characterization from shear wave elastography images. In: Medical Imaging 2018: Computer-Aided Diagnosis. vol. 10575, p. 105751X. International Society for Optics and Photonics (2018)

[353] Perrot, V., Polichetti, M., Varray, F., Garcia, D.: So you think you can DAS? A viewpoint on delay-and-sum beamforming. Ultrasonics 111, 106309 (2021)

[354] Pfister, T., Simonyan, K., Charles, J., Zisserman, A.: Deep convolutional neural networks for efficient pose estimation in gesture videos. In: Asian Conference on Computer Vision. pp. 538–552. Springer (2014)

[355] Pflugfelder, R.: An in-depth analysis of visual tracking with siamese neural networks. arXiv preprint arXiv:1707.00569 (2017)

[356] Philip, R.C., Dauvermann, M.R., Whalley, H.C., Baynham, K., Lawrie, S.M., Stanfield, A.C.: A systematic review and meta-analysis of the fMRI investigation of autism spectrum disorders. Neuroscience & Biobehavioral Reviews 36(2), 901–942 (2012)

[357] Pitre, J.J., Kirby, M.A., Li, D.S., Shen, T.T., Wang, R.K., OâDonnell, M., Pelivanov, I.: Nearly-incompressible transverse isotropy (NITI) of cornea elasticity: Model and experiments with acoustic micro-tapping OCE. Scientific Reports 10(1), 1–14 (2020)

[358] Polyak, B.T.: Some methods of speeding up the convergence of iteration methods. USSR Computational Mathematics and Mathematical Physics 4(5), 1–17 (1964)

[359] Posarelli, C., Sartini, F., Casini, G., Passani, A., Toro, M.D., Vella, G., Figus, M.: What is the impact of intraoperative microscope-integrated OCT in ophthalmic surgery? Relevant applications and outcomes. A systematic review. Journal of Clinical Medicine 9(6), 1682 (2020)

[360] Poulsen, P.R., Cho, B., Sawant, A., Ruan, D., Keall, P.J.: Detailed analysis of latencies in image-based dynamic MLC tracking a. Medical Physics 37(9), 4998–5005 (2010)

[361] Powers, J., Kremkau, F.: Medical ultrasound systems. Interface Focus 1(4), 477–489 (2011)

[362] Preiswerk, F., De Luca, V., Arnold, P., Celicanin, Z., Petrusca, L., Tanner, C., Bieri, O., Salomir, R., Cattin, P.C.: Model-guided respiratory organ motion prediction of the liver from 2D ultrasound. Medical Image Analysis 18(5), 740–751 (2014)

[363] Qin, P., Wu, K., Hu, Y., Zeng, J., Chai, X.: Diagnosis of benign and malignant thyroid nodules using combined conventional ultrasound and ultrasound elasticity imaging. IEEE Journal of Biomedical and Health Informatics 24(4), 1028–1036 (2019)

[364] Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with pseudo-3D residual networks. In: 2017 IEEE International Conference on Computer Vision. pp. 5534–5542. IEEE (2017)

[365] Ramier, A., Eltony, A.M., Chen, Y., Clouser, F., Birkenfeld, J.S., Watts, A., Yun, S.H.: In vivo measurement of shear modulus of the human cornea using optical coherence elastography. Scientific Reports 10(1), 1–10 (2020)

[366] Ran, A.R., Tham, C.C., Chan, P.P., Cheng, C.Y., Tham, Y.C., Rim, T.H., Cheung, C.Y.: Deep learning in glaucoma with optical coherence tomography: A review. Eye 35(1), 188–201 (2021)

[367] Rangamani, A., Xiong, T., Nair, A., Tran, T.D., Chin, S.P.: Landmark detection and tracking in ultrasound using a CNN-RNN framework. In: NIPS 2016 3D Deep Learning Workshop (2016)

[368] Rawat, W., Wang, Z.: Deep convolutional neural networks for image classification: A comprehensive review. Neural Computation 29(9), 2352–2449 (2017)

[369] Rehman, A., Belhaouari, S.B.: Deep Learning for video classification: A review (2021)

[370] Reiber, J.H., Tu, S., Tuinenburg, J.C., Koning, G., Janssen, J.P., Dijkstra, J.: QCA, IVUS and OCT in interventional cardiology in 2011. Cardiovascular Diagnosis and Therapy 1(1), 57 (2011)

[371] Romaguera, L.V., Mezheritsky, T., Mansour, R., Carrier, J.F., Kadoury, S.: Probabilistic 4D predictive model from in-room surrogates using conditional generative networks for image-guided radiotherapy. Medical Image Analysis 74, 102250 (2021)

[372] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 234–241. Springer (2015)

[373] Rosa, R., Monteiro, F.C.: Performance analysis of speckle ultrasound image filtering. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization 4(3-4), 193–201 (2016)

[374] Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review 65(6), 386 (1958)

[375] Rouze, N.C., Wang, M.H., Palmeri, M.L., Nightingale, K.: Robust estimation of time-of-flight shear wave speed using a radon sum transformation. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 57(12), 2662–2670 (2010)

[376] Rouze, N.C., Wang, M.H., Palmeri, M.L., Nightingale, K.R.: Parameters affecting the resolution and accuracy of 2-D quantitative shear wave images. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 59(8), 1729–1740 (2012)

[377] Royer, L., Krupa, A., Dardenne, G., Le Bras, A., Marchand, E., Marchal, M.: Real-time target tracking of soft tissues in 3D ultrasound images based on robust visual information and mechanical simulation. Medical Image Analysis 35, 582–598 (2017)

[378] Ruder, S.: An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098 (2017)

[379] Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature 323(6088), 533–536 (1986)

[380] Sakata, L.M., DeLeon-Ortega, J., Sakata, V., Girkin, C.A.: Optical coherence tomography of the retina and optic nerve–a review. Clinical & Experimental Ophthalmology 37(1), 90–99 (2009)

[381] Salehinejad, H., Sankar, S., Barfett, J., Colak, E., Valaee, S.: Recent advances in recurrent neural networks. arXiv preprint arXiv:1801.01078 (2017)

[382] Samani, A., Zubovits, J., Plewes, D.: Elastic moduli of normal and pathological human breast tissues: An inversion-technique-based investigation of 169 samples. Physics in Medicine & Biology 52(6), 1565 (2007)

[383] San José Estépar, R., Westin, C.F., Vosburgh, K.G.: Towards real time 2D to 3D registration for ultrasound-guided endoscopic and laparoscopic procedures. International Journal of Computer Assisted Radiology and Surgery 4(6), 549–560 (2009)

[384] Sande, J.A., Verjee, S., Vinayak, S., Amersi, F., Ghesani, M.: Ultrasound shear wave elastography and liver fibrosis: A Prospective Multicenter Study. World Journal of Hepatology 9(1), 38 (2017)

[385] Sandrin, L., Catheline, S., Tanter, M., Fink, M.: 2D transient elastography. In: Acoustical Imaging, pp. 485–492. Springer (2002)

[386] Sandrin, L., Catheline, S., Tanter, M., Hennequin, X., Fink, M.: Time-resolved pulsed elastography with ultrafast ultrasonic imaging. Ultrasonic Imaging 21(4), 259–272 (1999)

[387] Sandrin, L., Tanter, M., Catheline, S., Fink, M.: Shear modulus imaging with 2-D transient elastography. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 49(4), 426–435 (2002)

[388] Santurkar, S., Tsipras, D., Ilyas, A., Madry, A.: How does batch normalization help optimization? Advances in Neural Information Processing Systems 31 (2018)

[389] Sarvazyan, A.P., Rudenko, O.V., Swanson, S.D., Fowlkes, J.B., Emelianov, S.Y.: Shear wave elasticity imaging: A new ultrasonic technology of medical diagnostics. Ultrasound in Medicine & Biology 24(9), 1419–1435 (1998)

[390] Sarvazyan, A.P., Urban, M.W., Greenleaf, J.F.: Acoustic waves in medical imaging and diagnostics. Ultrasound in Medicine & Biology 39(7), 1133–1146 (2013)

[391] Schäberle, W.: Ultraschall in der Gefäßdiagnostik: Therapieorientiertes Lehrbuch und Atlas. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg (2010)

[392] Schlosser, J., Gong, R.H., Bruder, R., Schweikard, A., Jang, S., Henrie, J., Kamaya, A., Koong, A., Chang, D.T., Hristov, D.: Robotic intrafractional US guidance for liver SABR: System design, beam avoidance, and clinical imaging. Medical Physics 43(11), 5951–5963 (2016)

[393] Schlüter, M.: Analysis of ultrasound and optical coherence tomography for markerless volumetric image guidance in robotic radiosurgery. Ph.D. thesis, Technische Universität Hamburg (2021)

[394] Schlüter, M., Fuh, M.M., Maier, S., Otte, C., Kiani, P., Hansen, N.O., Miller, R.D., Schlüter, H., Schlaefer, A.: Towards OCT-navigated tissue ablation with a picosecond infrared laser (PIRL) and Mass-spectrometric analysis. In: 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). pp. 158–161. IEEE (2019)

[395] Schlüter, M., Gerlach, S., Fürweger, C., Schlaefer, A.: Analysis and optimization of the robot setup for robotic-ultrasound-guided radiation therapy. International Journal of Computer Assisted Radiology and Surgery 14(8), 1379–1387 (2019)

[396] Schlüter, M., Glandorf, L., Gromniak, M., Saathoff, T., Schlaefer, A.: Concept for markerless 6D tracking employing volumetric optical coherence tomography. Sensors 20(9), 2678 (2020)

[397] Schlüter, M., Glandorf, L., Sprenger, J., Gromniak, M., Neidhardt, M., Saathoff, T., Schlaefer, A.: High-speed markerless tissue motion tracking using volumetric optical coherence tomography images. In: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI). pp. 1979–1982. IEEE (2020)

[398] Schlüter, M., Otte, C., Saathoff, T., Gessert, N., Schlaefer, A.: Feasibility of a markerless tracking system based on optical coherence tomography. In: Medical Imaging 2019: Image-Guided Procedures, Robotic Interventions, and Modeling. vol. 10951, pp. 32–37. SPIE (2019)

[399] Schmitt, J.M.: OCT elastography: Imaging microscopic deformation and strain of tissue. Optics Express 3(6), 199–211 (1998)

[400] Schmitt, J.M., Xiang, S., Yung, K.M.: Speckle in optical coherence tomography. Journal of Biomedical Optics 4(1), 95–105 (1999)

[401] Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 45(11), 2673–2681 (1997)

[402] Sebag, F., Vaillant-Lombard, J., Berbis, J., Griset, V., Henry, J., Petit, P.a., Oliver, C.: Shear wave elastography: A new ultrasound imaging mode for the differential diagnosis of benign and malignant thyroid nodules. The Journal of Clinical Endocrinology & Metabolism 95(12), 5281–5288 (2010)

[403] Seliverstova, E., Caenen, A., Bézy, S., Nooijens, S., Voigt, J.U., D'hooge, J.: Comparing myocardial shear wave propagation velocity estimation methods based on tissue displacement, velocity and acceleration data. Ultrasound in Medicine & Biology 48(11), 2207–2216 (2022)

[404] Seong, D., Jeon, D., Wijesinghe, R.E., Park, K., Kim, H., Lee, E., Jeon, M., Kim, J.: Ultrahigh-speed spectral-domain optical coherence tomography up to 1-MHz A-scan rate using space–time-division multiplexing. IEEE Transactions on Instrumentation and Measurement 70, 1–8 (2021)

[405] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013)

[406] Shannon, C.: Communication in the Presence of Noise. Proceedings of the IRE 37(1), 10–21 (1949)

[407] Sharma, H., Droste, R., Chatelain, P., Drukker, L., Papageorghiou, A.T., Noble, J.A.: Spatio-temporal partitioning and description of full-length routine fetal anomaly ultrasound scans. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019). pp. 987–990. IEEE (2019)

[408] Sharma, V., Gupta, M., Pandey, A.K., Mishra, D., Kumar, A.: A review of deep learning-based human activity recognition on benchmark video datasets. Applied Artificial Intelligence 36(1), 2093705 (2022)

[409] Shen, C., Shi, H., Sun, T., Huang, Y., Wu, J.: An online learning approach for robust motion tracking in liver ultrasound sequence. In: Chinese Conference on Pattern Recognition and Computer Vision (PRCV). pp. 440–451. Springer (2018)

[410] Shen, D., Wu, G., Suk, H.I.: Deep learning in medical image analysis. Annual Review of Biomedical Engineering 19, 221–248 (2017)

[411] Shepard, A.J., Wang, B., Foo, T.K., Bednarz, B.P.: A block matching based approach with multiple simultaneous templates for the real-time 2D ultrasound tracking of liver vessels. Medical Physics 44(11), 5889–5900 (2017)

[412] Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Advances in Neural Information Processing Systems 28 (2015)

[413] Shi, X., Gao, Z., Lausen, L., Wang, H., Yeung, D.Y., Wong, W.k., Woo, W.c.: Deep learning for precipitation nowcasting: A benchmark and a new model. Advances in Neural Information Processing Systems 30 (2017)

[414] Shinde, P.P., Shah, S.: A review of machine learning and deep learning applications. In: 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). pp. 1–6. IEEE (2018)

[415] Sigrist, R.M., Liau, J., El Kaffas, A., Chammas, M.C., Willmann, J.K.: Ultrasound elastography: Review of techniques and clinical applications. Theranostics 7(5), 1303 (2017)

[416] Silva, V.B., De Jesus, D.A., Klein, S., van Walsum, T., Cardoso, J., Brea, L.S., Vaz, P.G.: Signal-carrying speckle in optical coherence tomography: A methodological review on biomedical applications. Journal of Biomedical Optics 27(3), 030901 (2022)

[417] Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems. pp. 568–576 (2014)

[418] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR) (2015)

[419] Singh, M., Wu, C., Liu, C., Li, J., Schill, A., Nair, A., Larin, K.: Phase-sensitive optical coherence elastography at 1.5 million A-Lines per second. Optics Letters 40(11), 2588–2591 (2015)

[420] Singh, M., Han, Z., Nair, A., Schill, A., Twa, M.D., Larin, K.V.: Applanation optical coherence elastography: Noncontact measurement of intraocular pressure, corneal biomechanical properties, and corneal geometry with a single instrument. Journal of Biomedical Optics 22(2), 020502 (2017)

[421] Smith, L.N., Topin, N.: Super-convergence: Very fast training of neural networks using large learning rates. In: Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications. vol. 11006, pp. 369–386. SPIE (2019)

[422] So, H., Chen, J., Yiu, B., Yu, A.: Medical ultrasound imaging: To GPU or not to GPU? IEEE Micro 31(5), 54–65 (2011)

[423] Song, P.: Innovations in ultrasound shear wave elastography. Ph.D. thesis, College of Medicine-Mayo Clinic (2014)

[424] Song, P., Manduca, A., Zhao, H., Urban, M.W., Greenleaf, J.F., Chen, S.: Fast shear compounding using robust 2-D shear wave speed calculation and multi-directional filtering. Ultrasound in Medicine & Biology 40(6), 1343–1355 (2014)

[425] Song, P., Zhao, H., Manduca, A., Urban, M.W., Greenleaf, J.F., Chen, S.: Comb-push ultrasound shear elastography (CUSE): A novel method for two-dimensional shear elasticity imaging of soft tissues. IEEE Transactions on Medical Imaging 31(9), 1821–1832 (2012)

[426] Song, S., Huang, Z., Nguyen, T.M., Wong, E.Y., Arnal, B., O'Donnell, M., Wang, R.K.: Shear modulus imaging by direct visualization of propagating shear waves with phase-sensitive optical coherence tomography. Journal of Biomedical Optics 18(12), 121509 (2013)

[427] Song, S., Wei, W., Hsieh, B.Y., Pelivanov, I., Shen, T.T., O'Donnell, M., Wang, R.K.: Strategies to improve phase-stability of ultrafast swept source optical coherence tomography for single shot imaging of transient mechanical waves at 16 kHz frame rate. Applied Physics Letters 108(19), 191104 (2016)

[428] Song, S., Yoon, S.J., Ambroziński, Ł., Pelivanov, I., Li, D., Gao, L., Shen, T.T., O'Donnell, M., Wang, R.K.: Non-contact rapid optical coherence elastography by high-speed 4D imaging of elastic waves. In: Optical Coherence Tomography and Coherence Domain Optical Methods in Biomedicine XXI. vol. 10053, p. 100531Y. International Society for Optics and Photonics (2017)

[429] Sorensen, T.A.: A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. Biol. Skar. 5, 1–34 (1948)

[430] Spiesberger, W., Tasto, M.: Processing of medical image sequences. In: Image Sequence Analysis, pp. 381–428. Springer (1981)

[431] Sprenger, J., Bengs, M., Gerlach, S., Neidhardt, M., Schlaefer, A.: Systematic analysis of volumetric ultrasound parameters for markerless 4D motion tracking. International Journal of Computer Assisted Radiology and Surgery 17(11), 2131–2139 (2022)

[432] Sprenger, J., Neidhardt, M., Schlüter, M., Latus, S., Gosau, T., Kemmling, J., Feldhaus, S., Schumacher, U., Schlaefer, A.: In-vivo markerless motion detection from volumetric optical coherence tomography data using CNNs. In: Medical Imaging 2021: Image-Guided Procedures, Robotic Interventions, and Modeling. vol. 11598, pp. 400–405. SPIE (2021)

[433] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15(1), 1929–1958 (2014)

[434] Suetens, P.: Fundamentals of Medical Imaging. Cambridge University Press, 2 edn. (2009)

[435] Suganyadevi, S., Seethalakshmi, V., Balasamy, K.: A review on deep learning in medical image analysis. International Journal of Multimedia Information Retrieval 11(1), 19–38 (2022)

[436] Sun, C., Standish, B.A., Yang, V.X.: Optical coherence elastography: Current status and future applications. Journal of Biomedical Optics 16(4), 043001 (2011)

[437] Sun, L., Jia, K., Yeung, D.Y., Shi, B.E.: Human action recognition using factorized spatio-temporal convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4597–4605 (2015)

[438] Sun, Z., Ke, Q., Rahmani, H., Bennamoun, M., Wang, G., Liu, J.: Human action recognition from various data modalities: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)

[439] Sutskever, I., Martens, J., Hinton, G.E.: Generating text with recurrent neural networks. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11). pp. 1017–1024 (2011)

[440] Szabo, T.L., Lewin, P.A.: Ultrasound transducer selection in clinical imaging practice. Journal of Ultrasound in Medicine 32(4), 573–582 (2013)

[441] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI Conference on Artificial Intelligence. vol. 4, p. 12 (2017)

[442] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–9 (2015)

[443] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2818–2826 (2016)

[444] Szeliski, R.: Computer vision: Algorithms and applications. Springer Nature (2022)

[445] Takayama, K., Mizowaki, T., Kokubo, M., Kawada, N., Nakayama, H., Narita, Y., Nagano, K., Kamino, Y., Hiraoka, M.: Initial validations for pursuing irradiation using a gimbals tracking system. Radiotherapy and Oncology 93(1), 45–49 (2009)

[446] Tallec, C., Ollivier, Y.: Unbiasing truncated backpropagation through time. arXiv preprint arXiv:1705.08209 (2017)

[447] Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., Liu, C.: A survey on deep transfer learning. In: International Conference on Artificial Neural Networks. pp. 270–279. Springer (2018)

[448] Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. pp. 6105–6114. PMLR (2019)

[449] Tang, A., Cloutier, G., Szeverenyi, N.M., Sirlin, C.B.: Ultrasound elastography and MR elastography for assessing liver fibrosis: Part 1, principles and techniques. AJR. American Journal of Roentgenology 205(1), 22 (2015)

[450] Tang, A., Cloutier, G., Szeverenyi, N.M., Sirlin, C.B.: Ultrasound elastography and MR elastography for assessing liver fibrosis: Part 2, diagnostic performance, confounders, and future directions. AJR. American Journal of Roentgenology 205(1), 33 (2015)

[451] Tanner, C., Eppenhof, K., Gelderblom, J., Székely, G.: Decision fusion for temporal prediction of respiratory liver motion. In: 2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI). pp. 698–701. IEEE (2014)

[452] Tanter, M., Bercoff, J., Athanasiou, A., Deffieux, T., Gennisson, J.L., Montaldo, G., Muller, M., Tardivon, A., Fink, M.: Quantitative assessment of breast lesion viscoelasticity: Initial clinical results using supersonic shear imaging. Ultrasound in Medicine & Biology 34(9), 1373–1386 (2008)

[453] Taylor, G.W., Fergus, R., LeCun, Y., Bregler, C.: Convolutional learning of spatio-temporal features. In: European Conference on Computer Vision. pp. 140–153. Springer (2010)

[454] Tenbrinck, D., Schmid, S., Jiang, X., Schäfers, K., Stypmann, J.: Histogram-based optical flow for motion estimation in ultrasound imaging. Journal of Mathematical Imaging and Vision 47, 138–150 (2013)

[455] Teo, P.T., Guo, K., Fontaine, G., Ahmed, B., Alayoubi, N., Kehler, K., Sasaki, D., Pistorius, S.: Reducing the tracking drift of an uncontoured tumor for a portal-image-based dynamically adapted conformal radiotherapy treatment. Medical & Biological Engineering & Computing 57, 1657–1672 (2019)

[456] Tohno, E., Ueno, E., Watanabe, H.: Ultrasound screening of breast cancer. Breast Cancer 16, 18–22 (2009)

[457] Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4489–4497 (2015)

[458] Tran, D., Wang, H., Torresani, L., Feiszli, M.: Video classification with channel-separated convolutional networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5552–5561 (2019)

[459] Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6450–6459 (2018)

[460] Twa, M.D., Lan, G., Singh, M., Larin, K.: In-vivo human corneal elasticity imaging: A phase sensitive optical coherence elastography method. Investigative Ophthalmology & Visual Science 58(8), 4324–4324 (2017)

[461] Ullah, A., Ahmad, J., Muhammad, K., Sajjad, M., Baik, S.W.: Action recognition in video sequences using deep bi-directional LSTM with CNN features. IEEE Access 6, 1155–1166 (2017)

[462] Urban, M.W., Greenleaf, J.F.: Use of the radon transform for estimation of shear wave speed. The Journal of the Acoustical Society of America 132(3), 1982–1982 (2012)

[463] Varol, G., Laptev, I., Schmid, C.: Long-term temporal convolutions for action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 40(6), 1510–1517 (2017)

[464] Vasconcelos, L., Kijanka, P., Urban, M.W.: Viscoelastic parameter estimation using simulated shear wave motion and convolutional neural networks. Computers in Biology and Medicine 133, 104382 (2021)

[465] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in Neural Information Processing Systems 30 (2017)

[466] Vedam, S., Keall, P., Docef, A., Todor, D., Kini, V., Mohan, R.: Predicting respiratory motion for four-dimensional radiotherapy. Medical Physics 31(8), 2274–2283 (2004)

[467] Vesal, S., Gu, M., Maier, A., Ravikumar, N.: Spatio-temporal multi-task learning for cardiac MRI left ventricle quantification. IEEE Journal of Biomedical and Health Informatics 25(7), 2698–2709 (2020)

[468] Vienola, K.V., Braaf, B., Sheehy, C.K., Yang, Q., Tiruveedhula, P., Arathorn, D.W., de Boer, J.F., Roorda, A.: Real-time eye motion compensation for OCT imaging with tracking SLO. Biomedical Optics Express 3(11), 2950–2963 (2012)

[469] Viergever, M.A., Maintz, J.A., Klein, S., Murphy, K., Staring, M., Pluim, J.P.: A survey of medical image registration–under review. Medical Image Analysis 33, 140–144 (2016)

[470] Vignali, L., Solinas, E., Emanuele, E.: Research and clinical applications of optical coherence tomography in invasive cardiology: A review. Current Cardiology Reviews 10(4), 369–376 (2014)

[471] Vijayan, S., Klein, S., Hofstad, E.F., Lindseth, F., Ystgaard, B., Langø, T.: Motion tracking in the liver: Validation of a method based on 4D ultrasound using a nonrigid registration technique. Medical Physics 41(8Part1), 082903 (2014)

[472] Villegas, R., Yang, J., Hong, S., Lin, X., Lee, H.: Decomposing motion and content for natural video sequence prediction. arXiv preprint arXiv:1706.08033 (2017)

[473] Voulodimos, A., Doulamis, N., Doulamis, A., Protopapadakis, E., et al.: Deep learning for computer vision: A brief review. Computational Intelligence and Neuroscience 2018 (2018)

[474] Wang, C., Zheng, L., Li, Y., Xia, S., Lv, J., Hu, X., Zhan, W., Yan, F., Li, R., Ren, X.: Noninvasive Assessment of Liver Fibrosis and Inflammation in Chronic Hepatitis B: A Dual-task Convolutional Neural Network (DtCNN) Model Based on Ultrasound Shear Wave Elastography. Journal of Clinical and Translational Hepatology (000), 0–0 (2022)

[475] Wang, J., Zhu, H., Wang, S.H., Zhang, Y.D.: A review of deep learning on medical image analysis. Mobile Networks and Applications 26, 351–380 (2021)

[476] Wang, J., Xu, Y., Boppart, S.A.: Review of optical coherence tomography in oncology. Journal of Biomedical Optics 22(12), 121711 (2017)

[477] Wang, K., Lu, X., Zhou, H., Gao, Y., Zheng, J., Tong, M., Wu, C., Liu, C., Huang, L., Jiang, T., et al.: Deep learning Radiomics of shear wave elastography significantly improved diagnostic performance for assessing liver fibrosis in chronic hepatitis B: A prospective multicentre study. Gut 68(4), 729–741 (2019)

[478] Wang, L., Tong, Z., Ji, B., Wu, G.: Tdn: Temporal difference networks for efficient action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1895–1904 (2021)

[479] Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: European Conference on Computer Vision. pp. 20–36. Springer (2016)

[480] Wang, M., Byram, B., Palmeri, M., Rouze, N., Nightingale, K.: On the precision of time-of-flight shear wave speed estimation in homogeneous soft solids: Initial results using a matrix array transducer. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency control 60(4), 758–770 (2013)

[481] Wang, M.H., Palmeri, M.L., Rotemberg, V.M., Rouze, N.C., Nightingale, K.R.: Improving the robustness of time-of-flight based shear wave speed reconstruction methods using RANSAC in human liver in vivo. Ultrasound in Medicine & Biology 36(5), 802–813 (2010)

[482] Wang, R.K., Kirkpatrick, S., Hinds, M.: Phase-sensitive optical coherence elastography for mapping tissue microstrains in real time. Applied Physics Letters 90(16), 164105 (2007)

[483] Wang, R.K., Ma, Z., Kirkpatrick, S.J.: Tissue Doppler optical coherence elastography for real time strain rate and strain mapping of soft tissue. Applied Physics Letters 89(14), 144103 (2006)

[484] Wang, S., Larin, K.V.: Noncontact depth-resolved micro-scale optical coherence elastography of the cornea. Biomedical Optics Express 5(11), 3807–3821 (2014)

[485] Wang, S., Larin, K.V.: Optical coherence elastography for tissue characterization: A review. Journal of Biophotonics 8(4), 279–302 (2015)

[486] Wang, T., Li, J., Zhang, M., Zhu, A., Snoussi, H., Choi, C.: An enhanced 3DCNN-ConvLSTM for spatiotemporal multimedia data analysis. Concurrency and Computation: Practice and Experience 33(2), e5302 (2021)

[487] Wang, X., Gao, L., Song, J., Shen, H.: Beyond frame-level CNN: Saliency-aware 3-D CNN with LSTM for video action recognition. IEEE Signal Processing Letters 24(4), 510–514 (2016)

[488] Wang, X., Gao, L., Wang, P., Sun, X., Liu, X.: Two-stream 3-D convnet fusion for action recognition in videos with arbitrary size and length. IEEE Transactions on Multimedia 20(3), 634–644 (2017)

[489] Wang, Y., Wang, Y., Akansu, A., Belfield, K.D., Hubbi, B., Liu, X.: Robust motion tracking based on adaptive speckle decorrelation analysis of OCT signal. Biomedical Optics Express 6(11), 4302–4316 (2015)

[490] Wang, Z., She, Q., Smolic, A.: Action-net: Multipath excitation for action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13214–13223 (2021)

[491] Werbos, P.J.: Backpropagation through time: What it does and how to do it. Proceedings of the IEEE 78(10), 1550–1560 (1990)

[492] Wieser, W., Biedermann, B.R., Klein, T., Eigenwillig, C.M., Huber, R.: Multi-megahertz OCT: High quality 3D imaging at 20 million A-scans and 4.5 GVoxels per second. Optics Express 18(14), 14685–14704 (2010)

[493] Williams, R.J., Peng, J.: An efficient gradient-based algorithm for on-line training of recurrent network trajectories. Neural Computation 2(4), 490–501 (1990)

[494] Williamson, T., Cheung, W., Roberts, S.K., Chauhan, S.: Ultrasound-based liver tracking utilizing a hybrid template/optical flow approach. International Journal of Computer Assisted Radiology and Surgery 13(10), 1605–1615 (2018)

[495] Wilson, D.R., Martinez, T.R.: The general inefficiency of batch training for gradient descent learning. Neural Networks 16(10), 1429–1451 (2003)

[496] Wojtkowski, M.: High-speed optical coherence tomography: Basics and applications. Applied Optics 49(16), D30–D61 (2010)

[497] Wojtkowski, M., Leitgeb, R., Kowalczyk, A., Bajraszewski, T., Fercher, A.F.: In vivo human retinal imaging by Fourier domain optical coherence tomography. Journal of Biomedical Optics 7(3), 457–463 (2002)

[498] Wojtkowski, M., Srinivasan, V.J., Ko, T.H., Fujimoto, J.G., Kowalczyk, A., Duker, J.S.: Ultrahigh-resolution, high-speed, Fourier domain optical coherence tomography and methods for dispersion compensation. Optics Express 12(11), 2404–2422 (2004)

[499] Wu, C., Fu, T., Wang, Y., Lin, Y., Wang, Y., Ai, D., Fan, J., Song, H., Yang, J.: Fusion Siamese network with drift correction for target tracking in ultrasound sequences. Physics in Medicine & Biology 67(4), 045018 (2022)

[500] Wu, Y., He, K.: Group normalization. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 3–19 (2018)

[501] Wu, Z., Jiang, Y.G., Wang, X., Ye, H., Xue, X., Wang, J.: Fusing multi-stream deep networks for video classification. arXiv preprint arXiv:1509.06086 (2015)

[502] Wu, Z., Wang, X., Jiang, Y.G., Ye, H., Xue, X.: Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In: Proceedings of the 23rd ACM International Conference on Multimedia. pp. 461–470 (2015)

[503] Wu, Z., Yao, T., Fu, Y., Jiang, Y.G.: Deep learning for video classification and captioning. In: Frontiers of Multimedia Research, pp. 3–29 (2017)

[504] Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. pp. 5987–5995. IEEE (2017)

[505] Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: Proceedings of the European conference on computer vision (ECCV). pp. 305–321 (2018)

[506] Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In: Advances in Neural Information Processing Systems. pp. 802–810 (2015)

[507] Xu, J., Ishikawa, H., Wollstein, G., Kagemann, L., Schuman, J.S.: Alignment of 3-D optical coherence tomography scans to correct eye movement using a particle filtering. IEEE Transactions on Medical Imaging 31(7), 1337–1345 (2012)

[508] Xu, Z., Yang, Y., Hauptmann, A.G.: A discriminative CNN video representation for event detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1798–1807 (2015)

[509] Xue, L.Y., Jiang, Z.Y., Fu, T.T., Wang, Q.M., Zhu, Y.L., Dai, M., Wang, W.P., Yu, J.H., Ding, H.: Transfer learning radiomics based on multimodal ultrasound imaging for staging liver fibrosis. European Radiology 30, 2973–2983 (2020)

[510] Yamada, H., Evans, F.G., et al.: Strength of biological materials (1970)

[511] Yanagihara, R.T., Lee, C.S., Ting, D.S.W., Lee, A.Y.: Methodological challenges of deep learning in optical coherence tomography for retinal diseases: A review. Translational Vision Science & Technology 9(2), 11–11 (2020)

[512] Yang, H., Shao, L., Zheng, F., Wang, L., Song, Z.: Recent advances and trends in visual tracking: A review. Neurocomputing 74(18), 3823–3831 (2011)

[513] Yang, H., Yuan, C., Li, B., Du, Y., Xing, J., Hu, W., Maybank, S.J.: Asymmetric 3D convolutional neural networks for action recognition. Pattern Recognition 85, 1–12 (2019)

[514] Yang, J., Wang, C., Qiu, W., Zheng, H.: Comparative study on shear wave speed estimation algorithms in ARFI for improving its reliability. In: 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. pp. 226–229. IEEE (2014)

[515] Yang, Y.P., Xu, X.H., Guo, L.H., He, Y.P., Wang, D., Liu, B.J., Zhao, C.K., Chen, B.D., Xu, H.X.: Qualitative and quantitative analysis with a novel shear wave speed imaging for differential diagnosis of breast lesions. Scientific Reports 7(1), 1–11 (2017)

[516] Yaqoob, Z., Wu, J., McDowell, E.J., Heng, X., Yang, C.: Methods and application areas of endoscopic optical coherence tomography. Journal of Biomedical Optics 11(6), 063001 (2006)

[517] Yi, J., Shin, Y., Hahn, S., Lee, Y.H.: Deep learning based sarcopenia prediction from shear-wave ultrasonographic elastography and gray scale ultrasonography of rectus femoris muscle. Scientific Reports 12(1), 1–8 (2022)

[518] Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Computing Surveys (CSUR) 38(4), 13–es (2006)

[519] You, W., Zhang, H., Zhao, X.: A Siamese CNN for image steganalysis. IEEE Transactions on Information Forensics and Security 16, 291–306 (2020)

[520] Yu, Y., Si, X., Hu, C., Zhang, J.: A review of recurrent neural networks: LSTM cells and network architectures. Neural Computation 31(7), 1235–1270 (2019)

[521] Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: Deep networks for video classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4694–4702 (2015)

[522] Yun, S.H., Tearney, G.J., de Boer, J.F., Iftimia, N., Bouma, B.E.: High-speed optical frequency-domain imaging. Optics Express 11(22), 2953–2963 (2003)

[523] Zawadzki, R.J., Fuller, A.R., Choi, S.S., Wiley, D.F., Hamann, B., Werner, J.S.: Correction of motion artifacts and scanning beam distortions in 3D ophthalmic optical coherence tomography imaging. In: Ophthalmic Technologies XVII. vol. 6426, p. 642607. International Society for Optics and Photonics (2007)

[524] Zbontar, J., LeCun, Y.: Computing the stereo matching cost with a convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1592–1599 (2015)

[525] Zeng, Q., Honarvar, M., Schneider, C., Mohammad, S.K., Lobo, J., Pang, E.H., Lau, K.T., Hu, C., Jago, J., Erb, S.R., et al.: Three-Dimensional Multi-Frequency Shear Wave Absolute Vibro-Elastography (3D S-WAVE) With a Matrix Array Transducer: Implementation and Preliminary In Vivo Study of the Liver. IEEE Transactions on Medical Imaging 40(2), 648–660 (2020)

[526] Zha, S., Luisier, F., Andrews, W., Srivastava, N., Salakhutdinov, R.: Exploiting image-trained CNN architectures for unconstrained video classification. arXiv preprint arXiv:1503.04144 (2015)

[527] Zhang, C., Benz, P., Argaw, D.M., Lee, S., Kim, J., Rameau, F., Bazin, J.C., Kweon, I.S.: Resnet or densenet? Introducing dense shortcuts to resnet. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3550–3559 (2021)

[528] Zhang, H.M., Dong, B.: A review on deep learning in medical image reconstruction. Journal of the Operations Research Society of China 8(2), 311–340 (2020)

[529] Zhang, H.B., Zhang, Y.X., Zhong, B., Lei, Q., Yang, L., Du, J.X., Chen, D.S.: A comprehensive survey of vision-based human action recognition methods. Sensors 19(5), 1005 (2019)

[530] Zhang, J., Duan, C., Duan, X., Hu, Y., Liu, J., Chen, W.: Quantitative evaluation of real-time shear-wave elastography under deep learning in children with chronic kidney disease. Scientific Programming 2022 (2022)

[531] Zhang, L., Lu, L., Wang, X., Zhu, R.M., Bagheri, M., Summers, R.M., Yao, J.: Spatio-temporal convolutional LSTMs for tumor growth prediction by learning 4D longitudinal patient data. IEEE Transactions on Medical Imaging 39(4), 1114–1126 (2019)

[532] Zhang, M., Ma, Y., Zheng, L., Wang, Y., Liu, Z., Ma, J.: Comparison of neural networks' performance in early screening of autism spectrum disorders under two MRI principles. In: 2019 International Conference on Networking and Network Applications (NaNA). pp. 338–343. IEEE (2019)

[533] Zhang, S., Guo, S., Huang, W., Scott, M.R., Wang, L.: V4d: 4D convolutional neural networks for video-level representation learning. arXiv preprint arXiv:2002.07442 (2020)

[534] Zhang, X., Liang, M., Yang, Z., Zheng, C., Wu, J., Ou, B., Li, H., Wu, X., Luo, B., Shen, J.: Deep learning-based radiomics of B-mode ultrasonography and shear-wave elastography: Improved performance in breast mass classification. Frontiers in Oncology p. 1621 (2020)

[535] Zhang, Y., Wörn, H.: Optical coherence tomography as highly accurate optical tracking system. In: 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. pp. 1145–1150. IEEE (2014)

[536] Zhao, H., Song, P., Manduca, A., Kinnick, R.R., Urban, M.W., Greenleaf, J.F., Chen, S., Catheline, S.: Two-dimensional shear elasticity imaging using external mechanical vibration. In: 2013 IEEE International Ultrasonics Symposium (IUS). pp. 1256–1259. IEEE (2013)

[537] Zhao, Y., Li, X., Zhang, W., Zhao, S., Makkie, M., Zhang, M., Li, Q., Liu, T.: Modeling 4D fMRI data via spatio-temporal convolutional neural networks (ST-CNN). In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 181–189. Springer (2018)

[538] Zhao, Z., Chen, Z., Voros, S., Cheng, X.: Real-time tracking of surgical instruments based on spatio-temporal context and deep learning. Computer Assisted Surgery 24(sup1), 20–29 (2019)

[539] Zheng, X., Yao, Z., Huang, Y., Yu, Y., Wang, Y., Liu, Y., Mao, R., Li, F., Xiao, Y., Wang, Y., et al.: Deep learning radiomics can predict axillary lymph node status in early-stage breast cancer. Nature Communications 11(1), 1236 (2020)

[540] Zhi, H., Ou, B., Luo, B.M., Feng, X., Wen, Y.L., Yang, H.Y.: Comparison of ultrasound elastography, mammography, and sonography in the diagnosis of solid breast lesions. Journal of Ultrasound in Medicine 26(6), 807–815 (2007)

[541] Zhou, H., Wang, K., Tian, J.: The accurate non-invasive staging of liver fibrosis using deep learning radiomics based on transfer learning of shear wave elastography. In: Medical Imaging 2020: Ultrasonic Imaging and Tomography. vol. 11319, p. 113190A. International Society for Optics and Photonics (2020)

[542] Zhou, Y.T., Chellappa, R.: Computation of optical flow using a neural network. In: IEEE International Conference on Neural Networks. vol. 1998, pp. 71–78 (1988)

[543] Zhou, Y., Xu, J., Liu, Q., Li, C., Liu, Z., Wang, M., Zheng, H., Wang, S.: A radiomics approach with CNN for shear-wave elastography breast tumor classification. IEEE Transactions on Biomedical Engineering 65(9), 1935–1942 (2018)

[544] Zitova, B., Flusser, J.: Image registration methods: A survey. Image and Vision Computing 21(11), 977–1000 (2003)

[545] Zolfaghari, M., Singh, K., Brox, T.: Eco: Efficient convolutional network for online video understanding. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 695–712 (2018)

[546] Zvietcovich, F., Gary, R.G., Mestre, H., Giannetto, M., Nedergaard, M., Rolland, J.P., Parker, K.J.: Longitudinal shear waves for elastic characterization of tissues in optical coherence elastography. Biomedical Optics Express 10(7), 3699–3718 (2019)

[547] Zvietcovich, F., Larin, K.V.: Wave-based optical coherence elastography: The 10-year perspective. Progress in Biomedical Engineering 4(1), 012007 (2022)

[548] Zvietcovich, F., Nair, A., Singh, M., Aglyamov, S.R., Twa, M.D., Larin, K.V.: Dynamic optical coherence elastography of the anterior eye: Understanding the biomechanics of the limbus. Investigative Ophthalmology & Visual Science 61(13), 7–7 (2020)

[549] Zvietcovich, F., Pongchalee, P., Meemon, P., Rolland, J.P., Parker, K.J.: Reverberant 3D optical coherence elastography maps the elasticity of individual corneal layers. Nature Communications 10(1), 1–13 (2019)

[550] Zvietcovich, F., Singh, M., Ambekar, Y.S., Aglyamov, S.R., Twa, M.D., Larin, K.V.: Micro air-pulse spatial deformation spreading characterizes degree of anisotropy in tissues. IEEE Journal of Selected Topics in Quantum Electronics 27(4), 1–10 (2020)

List of Figures

List of Tables

List of Abbreviations

2P two-path

aCC average correlation coefficient

ARFI acoustic radiation force impulse

BPTT backpropagation through time

CNN convolutional neural network

conv3D 3D convolution

conv4D 4D convolution

ConvRNN convolutional recurrent neural network

CT computed tomography

DAS delay-and-sum

DenseNet densely connected convolutional network

F-ST factorized spatio-temporal convolution

FC fully connected network

fMRI functional magnetic resonance imaging

FOV field of view

GAP global average pooling

GRU gated recurrent unit

I incremental

LR linear regression

LSTM long short-term memory

MAE mean absolute error

MLP multi-layer fully connected neural networks

MRI magnetic resonance imaging

MSE mean squared error

NCC normalized cross-correlation

NI non-incremental

nP n-path

OCE optical coherence elastography

OCT optical coherence tomography

p.p. percentage points

PCC pearson correlation coefficient

RBF Gaussian kernel

ReLU rectifying nonlinearities

ResNet residual network

RF-data radio-frequency data

rMAE relative mean absolute error

RNN recurrent neural network

ROI region of interest

SGD stochastic gradient descent

SiamFC fully-convolutional siamese network

SIFT scale-invariant feature transform

ST spatio-temporal convolution

SVR support vector regression

SWEI shear wave elasticity imaging

TC time-channel

TE tracking error

ToF time of flight

TSN temporal segment networks

US ultrasound

US-SWEI ultrasound shear wave elasticity imaging

w/o without temporal pooling

List of Symbols

U	Weight matrix of an RNN w.r.t. the input
V	Weight matrix of an RNN for the output layer
W	Weight matrix of an RNN w.r.t. the previous state
α	Significance level
ΔL_E	Relative change in length
ΔL	Path difference
$\Delta\varphi$	Phase difference
ϵ_{BN}	A small constant for numerical stability
ϵ_E	Strain
γ	Rotation angle
γ_{max}	Maximal rotation angle
λ	wavelength
$\mathcal{B}$	Mini-batch
$\mathcal{D}_{test}$	A supervised test dataset
$\mathcal{D}_{train}$	A supervised training dataset
$\mathcal{D}_{val}$	A supervised validation dataset
$\mathcal{L}$	Loss function
L_d	Sample distance
L_r	Reference path length
$\mathcal{D}$	A dataset
N_{LRF}	Number of overlapping but non-equal local receptive fields
w_0	Bias-term
$\mathscr{L}$	Per-example loss
μ	Mean

ν	Poisson's ratio
$\phi(x)$	Transformed version of an input
ρ	Mass density
σ	Standard deviation
σ_E	Stress
τ	An arbitrary point in time
θ	Weighting parameter
θ_E	A scaling factor
$\tilde{T}_{0,n}$	Estimated translation of a target between two time points
$\widetilde{y}$	Output of a machine learning algorithm
ξ	Parameter vector
A	Area
a	Activation value of a neuron
b	sequence length considered for backward propagation
F	Force
$f^{[\tau]}$	Forget gate of an LSM
f_q	frequency
$g^{[\tau]}$	external input gate of an LSM
h	A nonlinear activation function
h_σ	A sigmoid activation function
h_{out}	Output activation function
h_{RL}	A ReLu activation function
I	Template Image
k_d	Kernel size along the dephts axis
k_h	Kernel size along the height axis
k_t	Kernel size along the temporal axis
k_w	Kernel size along the width axis
L	Number of layers of a network

l	A layer number of a network
L_E	Original length
m_b	Batch size
m_{tr}	Number of examples of the training dataset
m_t	Number of examples of the test dataset
m_{val}	Number of examples of the validation dataset
$n^{(l)}$	The number of neurons of a layer
n_c	Number of input channels
n_d	Image depths
n_h	Image height
n_{in}	Number of input feature maps
n_{out}	Number of output feature maps
n_t	Sequence length
n_w	Image width
n_x	Number of features
n_y	Number of targets
n_z	Number of neurons of a layer
p_{dist}	Probability that a B-scan is shifted
p_{drop}	Dropout rate
p_h	Prediction horizon
p_t	Number of past image volumes
p_y	Number of past labels
$q^{[\tau]}$	Output gate of an LSTM
R	Rotation matrix
r	Sequence length considered for forward propagation
$r^{[\tau]}$	Reset gate of GRU
$T_{0,n}$	Translation of a target between two time points
V	Local receptive field of a CNN

v	Shear wave velocity
v_a	Acoustic wave velocity
$x^{[i]}$	A single 2D/3D image
x_t	A sequence of 2D/3D images
y	A learning target
Z	Acoustic impedance
z	Output of a neuron
$z_{end}^{[\tau]}$	Output of a ConvGRU module at the end of an architecture
$z_{front}^{[\tau]}$	Output of a ConvGRU module at the front of an architecture
$z_{middle}^{[\tau]}$	Output of a ConvGRU module at the middle of an architecture
K	Kernel
lr	Learning Rate
p	P-value
s	Stride of a convolution
w	Learnable weights