Deep Learning Methods for Automated Segmentation of Medical Ultrasound Images

Vom Promotionsausschuss der
Technischen Universität Hamburg

zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

von
Lennart Holstein

aus
Stade

2024

: https://orcid.org/0000-0003-0610-0347

DOI: https://doi.org/10.15480/882.9021

Creative Commons Lizenzvertrag
Der Text steht, soweit nicht anders gekennzeichnet, unter der Creative-Commons-Lizenz Namensnennung 4.0 (CC BY 4.0). Das bedeutet, dass er vervielfältigt, verbreitet und öffentlich zugänglich gemacht werden darf, auch kommerziell, sofern dabei stets der Urheber, die Quelle des Textes und o. g. Lizenz genannt werden. Die genaue Formulierung der Lizenz kann unter https://creativecommons.org/licenses/by/4.0/legalcode.de aufgerufen werden.

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

Tag der mündlichen Prüfung: 18. Dezember 2023

Abstract

Methods for automatic analysis of medical images are beginning to find their way into everyday clinical practice. However, this development has just started, and much further research is required to fully unlock the potential of modern image analysis tools like deep learning. A common task in image analysis is segmentation, i.e., delineating particular objects of interest. Usually, this is done manually by well-trained experts in a time-consuming process. Automating this process could streamline the clinical workflow. Deep learning methods are promising regarding their potential to reach human-level performance in segmentation. However, they need much training data to achieve practically relevant results. This is particularly a problem in the medical field, where annotated data is usually scarce. The main reason is the need for well-trained experts to create high-quality annotations in a rather labor-intensive annotation process. Moreover, some pathologies or disease patterns occur very rarely and thus do not allow for the creation of large datasets. Therefore, developing and examining deep learning methods suitable for smaller datasets is meaningful. This is the purpose of this work. Since ultrasound is the most used medical imaging modality worldwide, we focused on developing methods that are adapted to this modality. However, some of the presented approaches can be readily transferred to other modalities as well.

In this work, we develop and investigate multiple methods to improve the filters that a convolutional neural network (CNN) learns during training. The first method combines CNNs with wavelet scattering to enhance the texture information drawn from ultrasound images. The second and third methods incorporate domain knowledge to guide CNN training. The second method employs tissue shape priors via an independent component analysis, while the third method uses a loss function that ensures a particular tissue is completely surrounded by other tissues. The fourth method enables synthetic ultrasound image generation for small ultrasound datasets by inserting a new layer, the speckle layer, into a generative adversarial network (GAN). These resulting synthetic images are then used to augment the underlying dataset of real images to improve the downstream segmentation task.

We tested our methods on 4 different datasets for ultrasound segmentation. These include two intravascular ultrasound datasets, one for lumen and vessel wall segmentation and another for calcium segmentation. Furthermore, a cardiac dataset for endocardium, myocardium, and atrium segmentation, and finally, a neck muscle dataset for the segmentation of 8 different tissues. To investigate the methods' behaviors for different training dataset sizes, we systematically decreased the number of training images in our experiments. All methods were tested with two CNN architectures: U-Net and DeepLabV3. The results show that different methods benefit different tissues, dataset sizes, and CNN architectures. No method led to improvements in all cases. However, the improvements in synthetic data augmentation were the largest, on average.

The methods developed and investigated in this work can be seen as a toolbox that provides deep learning methods for improving ultrasound segmentation performance. This work reveals some tendencies in which individual methods can lead to performance gains. However, in a practical setting, suitable methods have to be found through rigorous experiments. In summary, our work is a valuable step toward making deep learning methods applicable to small datasets prevalent in clinical practice.

Contents

1 Introduction

In this first chapter, we outline the importance of ultrasound imaging in today's clinical practice. We discuss why correct image interpretation is critical and how automated image analysis tools can provide substantial benefits. We then discuss image feature extraction and the differences between conventional methods and deep learning for medical image analysis. This leads us to the problem statement, which is rooted in the amount of training data required to train neural networks effectively. Finally, we propose our solution to this problem and present the research questions we investigate in this work.

1.1 Medical Ultrasound

The discovery of the piezoelectric effect in 1880 by Jacques and Pierre Curie paved the way for the development and advancement of ultrasound imaging. Paul Langevin, who had worked as a PhD student in Pierre Curie's laboratory, and Robert William Boyle developed the first hydrophones for underwater object detection during World War I [65]. However, this technology was not applied in warfare. It was Langevin's idea to use piezoelectric quartz plates as ultrasound transducers. The research on ultrasound physics was continued by Boyle, who published 25 papers on this topic between 1922 and 1932. In the years that followed, much research was done to develop methods that use ultrasound for material testing and medical applications. In the 1950s and 1960s, ultrasound imaging began to take hold in hospitals [247]. Nowadays, ultrasound is the most frequently used imaging modality in hospitals and medical offices worldwide [247]. The reasons for this development are manifold. Compared to other imaging systems like computed tomography (CT) or magnetic resonance imaging (MRI), ultrasound devices are inexpensive and require less space. Moreover, unlike the ionizing radiation of CT, ultrasound waves do not damage tissue. Finally, the ultrasound images are acquired in real-time, which makes this modality very versatile and adaptive.

Ultrasound systems are highly adapted to the underlying use case. This includes the excitation frequency and, thus, the penetration depth of the ultrasound beam, the type and shape of the probe, and the image processing strategy. A frequently occurring use case is abdominal ultrasound. Here, organs like the liver, kidney, spleen, gallbladder, pancreas, and bladder, but also fetuses, can be examined by placing the probe at different positions of the abdomen. Other applications are cardiac and vascular ultrasound. Endoscopic ultrasound devices are used to examine hollow organs from the inside. This includes endobronchial, vaginal, and rectal ultrasound. Intravascular ultrasound (IVUS) is used to examine blood vessels from the inside by inserting a catheter probe into the vessel. Furthermore, ultrasound imaging can be used for guidance during interventions like needle biopsies or brachytherapy.

In summary, ultrasound is an essential and versatile imaging modality in clinical practice. However, ultrasound images exhibit strong noise (speckle) and often quite severe imaging artifacts (compare the exemplary images in Chapter 5). Correctly interpreting ultrasound images

is therefore rather difficult.

1.2 Automated Medical Image Analysis

Since the interpretation of ultrasound images is difficult, the conclusions drawn from the images depend heavily on the physician's experience. Therefore, physicians need much training to increase their knowledge and experience in image interpretation. Untrained physicians may have difficulty interpreting images and do not always have access to colleagues with more experience. Moreover, even experts are not immune to overlooking minor image features. For example, diseases or lesions with very low prevalence, and therefore not well known, can be easily missed. This is especially true if the physician is looking for something specific, thus overlooking anything small that is or seems to be unrelated to the patient's symptoms. Another aspect is image annotation. Physicians often have to mark or delineate structures inside the image to derive quantitative information, e.g., the volume of an organ or distance ratios. This process is rather time-consuming and requires experience.

Tools for automated image analysis can support physicians with additional information and help interpret and annotate images. Such tools are quite versatile and can solve various tasks based on features extracted from the images. An important task is image classification, e.g., classifying a tumor as benign or malignant. Another task is object localization, e.g., localizing lesions in the thyroid or liver. Quantification of specific tissues can be facilitated by image segmentation, i.e., delineation of particular objects of interest.

Automated image analysis addresses the problems with image interpretation by humans mentioned above. First, the results are independent of the physician's experience. Second, random findings which are not directly related to the patient's actual problem can be found. Third, the quantification of objects can be performed instantly without requiring physicians to annotate objects manually, which saves time.

Over the last decades, many hand-crafted image features for different applications have been developed. Their primary purpose is to draw information from images and distinguish between different image contents. Features can be calculated from a large variety of quantities, e.g., pixel value histograms, pixel value gradients, optical flow, autocorrelation, or entropy. Such quantities can also be derived from transformed images, e.g., by Fourier or wavelet transformation. Presenting such approaches in detail is not in the scope of this work. However, further information can be found in a large variety of books, e.g., Petrou and Petrou [203], Jähne [119], and Gonzalez and Woods [82] as well as review articles, e.g., Tian [257], Kumar and Bhatia [143], Wang et al. [272], and Humeau-Heurtier [111].

Extracted features can be used to train classifiers like support-vector machines (SVMs, Cortes and Vapnik [49]), random forests (RFs, Breiman [31]), or simple neural networks (perceptrons, Rosenblatt [218]). For image segmentation, image features are used for methods like thresholding, region growing, or active contours. Other approaches extract features from small image patches to classify these into the underlying segmentation classes. The review articles by Cheng et al. [45] and Vantaram and Saber [266] provide extensive overviews of conventional image segmentation methods for natural images.

Many approaches that work for natural images are also applicable to medical images [47]. However, every imaging modality has its own properties that have to be considered when designing conventional segmentation approaches. This applies to MRI images [87, 289, 270,

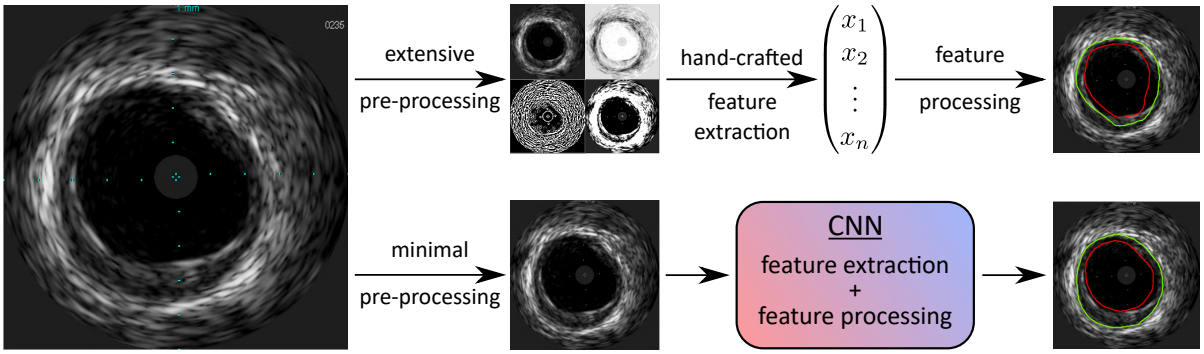

Figure 1.1: Illustration of a conventional image analysis procedure vs. deep learning. An exemplary intravascular ultrasound (IVUS) image is segmented into three regions: lumen (red), vessel wall (green), and background. The features for conventional image analysis (top) must be designed by hand and then processed with explicit algorithms or machine learning methods. In deep learning (bottom), a convolutional neural network (CNN) is trained using annotated data, thus, learning feature extraction and processing by itself.

232], CT images [168, 288, 151], PET images [293, 71] and ultrasound images [187, 234, 186, 167, 40]

All conventional methods for image analysis which are based on the extraction of hand-crafted features suffer from the same drawbacks. First, the suitability of chosen features for a particular problem has to be assessed by the developer and is thus dependent on the developer's experience. Second, it is not possible to prove that the features are optimal or optimally tuned for the given problem. Only exhaustive evaluation on various test data can provide evidence that the features are meaningful for the underlying task. Third, hand-crafted features are usually based on theoretical considerations and are therefore bound to the capabilities of the human mind. Or in other words, features have to be developed. Hypothetical features, whose calculation does not follow a reasonable procedure, can not be developed, although they could be highly suitable for a specific task.

Deep learning has the potential to solve these issues. Since the image features are learned by training a convolutional neural network (CNN), no manual definition and selection of hand-crafted features are needed. Figure 1.1 illustrates the differences between a conventional image analysis pipeline and a deep learning pipeline. In fact, the features that have been learned during training are tailored to the underlying problem. This is possible because the network parameters are adjusted by minimizing a loss function that measures the difference between the actual network output and the desired output, given the input.

In recent years, deep learning has also been used to analyze medical images and has produced state-of-the-art results in tasks like classification, segmentation, localization, regression, and registration. Datasets on which contemporary CNN architectures do (almost) reach human expert-level performance are quite large. Examples for classification and localization are CheXpert [115] with 224k chest radiographs, ChestX-ray: [273] with 112k chest radiographs, MURA [213] with 40k musculoskeletal radiographs, the ISIC challenge dataset [220] with 33k dermoscopic images, or the Kaggle diabetic retinopathy challenge with 35k eye fundus images. One of the largest datasets for image segmentation is the Medical Segmentation Decathlon dataset [235]. It consists of ten subsets depicting different anatomical regions in MRI and CT. The amount of cases per subset ranges from 30 to 700, making a total of 2633 cases. So far, a

dataset for ultrasound image segmentation with comparable size does not exist.

In conclusion, automated image analysis, especially deep learning, provides excellent opportunities to streamline the clinical workflow and maximize the information content drawn from medical images. However, deep learning also has drawbacks that prevent extensive use of automated image analysis methods in clinical practice. Tackling these drawbacks leads us to the purpose of this work.

1.3 Purpose of this Work

As mentioned in Section 1.1, ultrasound is highly versatile and the most frequently used imaging modality around the world [247]. We therefore present the following problem statement and our proposed solutions in the context of ultrasound image segmentation. However, some suggested methods could likely also be readily transferred to other imaging modalities. We have seen that deep learning is a quite potent approach to image analysis and achieves state-of-the-art results in many different applications considering natural and also medical images. However, deep learning also faces some severe drawbacks.

Problem statement The most prominent problem in data-driven medical image analysis, besides model explainability, is data scarcity. Several causes lead to a rather unfavorable data situation in the medical domain. Since high quality and reliability of the data are required, images have to be annotated by very well-trained experts (a rather tedious task that might be rather unappealing for most physicians). This is a significant difference compared to most natural image datasets, since these usually don't need much expertise for annotation. In particular, creating segmentation labels is highly labor-intensive, as manually delineating multiple objects per image takes much time. Moreover, despite being acquired with the same modality, images from different imaging systems differ regarding appearance, structure, and texture. This means that a neural network trained on images from machine A usually does not perform very well on images from machine B. This is called domain shift. Both of these aspects lead to the disadvantage that image datasets in the medical field are usually relatively small, especially when considering segmentation. This problem gets more severe when considering diseases or lesions that are quite rare and thus not well known [69, 285, 23, 135, 2]. For these cases, CNNs would be particularly beneficial by scanning through large amounts of image data and searching for random findings. In this sense, such deep-learning methods would significantly advance global healthcare.

As explained in Section 1.2, neural networks need to be trained on large datasets to reach acceptable performance. This is especially important in the medical domain, where results must reach human expert performance and thus be precise and reliable. As shown in Section 1.2, the largest medical imaging datasets deal with the classification of rather frequently occurring diseases [273, 213, 115, 220]. Since providing an image with a single (or a few) labels is usually much less time-consuming than manually delineating objects, classification datasets are often much larger than segmentation datasets (compare Section 1.2). Moreover, the largest classification datasets like CheXpert [115] are labeled automatically by processing corresponding text reports. In contrast, potential datasets of extremely rare lesions [69, 285, 23, 135, 2] can likely not exceed sizes of 50 to 200 images very easily. An example of relatively rarely occurring cases in intravascular ultrasound (IVUS) is spontaneous coronary artery dissection

(SCAD) [170, 120, 2]. According to Nishiguchi et al. [185], only about 4 % of patients with acute coronary syndrome suffer from SCAD. Hence, finding SCAD during IVUS examination and thus collecting large datasets would not be feasible in a reasonable amount of time.

The poor data situation in the medical field stands in stark contrast to the demand for large datasets for CNN training. Networks trained on relatively small datasets often perform poorly or are extremely sensitive to domain shift. It is therefore necessary to develop and study deep learning methods that are modified to perform better on smaller datasets. The results of such investigations could lead the way for further research and development.

Heuristically, the problem of small datasets develops as follows. When only a small amount of training data is available, the learned convolutional filters turn out to be rather inefficient. Such filters are not able to extract meaningful image features, which, in turn, leads to poor generalizability and a lack of robustness in terms of correct tissue topology. Finally, this results in bad performance on unseen data.

Solution Proposal It is therefore crucial to find ways to improve the informative value of extracted features and thus improve generalizability when dealing with limited data. This work proposes four different deep learning approaches to tackle this problem for ultrasound segmentation. These methods are based on wavelet scattering and ultrasound domain knowledge.

Wavelet scattering networks can be interpreted as CNNs with pre-defined but rather efficient filters. We integrated wavelet scattering transformations into CNNs and investigated, whether the corresponding filters could be able to extract better features and help the CNN to improve generalizability.

Domain knowledge for deep learning is a vast field. In this work, we focus on specific properties of ultrasound images, i.e., speckle noise as well as shape and topological constraints, and integrate this knowledge into CNNs. Incorporating domain knowledge can help to regularize network training, e.g., by restricting filters to more beneficial configurations. This again can improve generalizability (compare Section 4.2, especially Figure 4.7).

Data augmentation is often used to increase the dataset size artificially. This means mainly random transformations during training to alter image appearance without changing the semantic content. This includes flips, rotations, crops, elastic transformations, and color or contrast transformations. Generative adversarial networks (GANs, Goodfellow et al. [84]) have also been proposed to generate synthetic images for data augmentation. However, such artificial images are usually of poor quality when dealing with limited data. In this case, this approach is not helpful for data augmentation. We experienced that generating speckle noise requires much network capacity, so often only a single speckle pattern is generated for all images (compare Subsection 6.5.5). We developed a speckle layer that helps regularize training by adding speckle noise to feature maps adaptively in the generator of a GAN. This approach could help to generate visually appealing synthetic ultrasound images for data augmentation despite a small training dataset.

We evaluated our methods on 4 different ultrasound segmentation datasets and investigated whether differences in behavior occur between datasets. Since no ultrasound segmentation datasets of rarely occurring lesions were publicly available, we chose larger datasets and systematically lowered their sizes during our experiments. This also allows us to gain insight into how the methods behave with different amounts of training images. The datasets include intravascular ultrasound (IVUS) lumen and vessel wall segmentation, IVUS calcium segmenta-

tion, cardiac segmentation, and neck muscle segmentation. We will approach our proposal by answering the following research questions.

RQ 1 What kinds of extensions to vanilla deep learning methods could have the potential to improve segmentation results on smaller ultrasound image datasets?

- **RQ 1.1** How could wavelet scattering help?
- **RQ 1.2** How could incorporating domain knowledge help?
- **RQ 1.3** How could data augmentation with synthetic images generated by GANs be made feasible?

RQ 2 Under what conditions do the presented methods lead to segmentation performance improvements?

- **RQ 2.1** Which CNN architectures benefit from which methods?
- **RQ 2.2** How do the presented methods perform as a function of dataset size?
- **RQ 2.3** Which tissues benefit from the presented methods?
- **RQ 2.4** What types of segmentation errors are reduced by the presented methods?

1.4 Organization

We now want to outline the structure of this work and describe the content of the following chapters. This thesis follows the usual structure, starting with theoretical considerations, presentation of methods and datasets, results, discussion, and conclusion. Parts of this work have been published in Bargsten and Schlaefer [22], Bargsten et al. [21], Bargsten et al. [18], and Bargsten et al. [20].

Chapter 2 All information about ultrasound imaging needed to follow this thesis is provided. The theoretical backgrounds of ultrasound physics and ultrasound technology are briefly introduced. Afterward, we present typical methods for ultrasound image processing and image analysis.

Chapter 3 We discuss the fundamentals of deep learning including network flavors like convolutional neural networks (CNNs) and generative adversarial networks (GANs). We further introduce the basics of CNN training and evaluation, focusing on image segmentation. Finally, we give a short overview of wavelet scattering.

Chapter 4 This chapter presents extensions to ordinary deep learning methods we developed for ultrasound image segmentation. Every section starts with an overview of previous work on that specific topic, followed by a problem statement and our proposed solution. Section 4.1 deals with combining wavelet scattering and CNNs. In Section 4.2, we present our methods of incorporating domain knowledge for ultrasound segmentation into CNNs. Finally, in Section 4.3, we introduce our approach to generate synthetic ultrasound images with GANs to facilitate data augmentation for small datasets.

Chapter 5 Here, we present all scenarios and corresponding datasets we use to evaluate our proposed methods. The datasets include intravascular ultrasound (IVUS), cardiac ultrasound, and neck muscle ultrasound. Besides describing the datasets, we discuss the role of each scenario in clinical practice.

Chapter 6 This chapter presents and summarizes the results of our experiments. Additionally, we answer **RQ 1**.

Chapter 7 In this chapter, we discuss and interpret the results of each method and scenario. After that, we compare methods and scenarios and finally derive an overall interpretation and assessment of our work with respect to **RQ 2**.

Chapter 8 We recapitulate our methods and contributions in the context of our motivation and research questions. Finally, we discuss remaining challenges and recommend further research directions.

2 Fundamentals of Ultrasound Imaging

In this chapter, we will briefly introduce the fundamental physics and technology of medical ultrasound systems so that the data used to evaluate the proposed methods, as well as its acquisition, can be understood. Instead of the common approach of introducing ultrasound physics first and then the technological realization, we will follow the typical ultrasound signal path: the excitation of a piezo transducer, the propagation of acoustic waves in tissue, reflection, and scattering of waves, recording, and processing of the backscattered signal. This chapter's content is based on Szabo [247], Dössel [63], Hangiandreou [97], and Handee and Ritenour [96]. Further details on ultrasound physics and technology can be found in these books, especially the very comprehensive book by [247].

In general, ultrasound technology is facilitated by the piezoelectric effect. Some materials, such as certain crystals or ceramics, induce electric polarization when deformed. This effect can also be reversed: when applying a voltage to those materials, they elastically deform. Such materials are referred to as "piezo materials". A component that is built from a piezo material is often simply called "piezo". If an alternating voltage with appropriate frequency is used to excite the piezo, acoustic waves are emitted, analogous to a loudspeaker. In ultrasound imaging, such a piezo is called a "transducer". The transducer is excited such that it emits short acoustic pulses. The length of these pulses usually corresponds to a multiple of the wavelength. Figure 2.1 depicts a sketch of a piezo transducer with an emitted acoustic pulse with a length of 2 wavelengths. The front of a transducer is often built concavely to help focus the pulse. Since, in practice, pulses are emitted with a considerably high pulse frequency, one usually refers to the emitted signal as a "beam".

The majority of ultrasound probes comprise multiple transducer elements. These can be arranged into different kinds of arrays. Common arrays are linear and curved arrays. Figure 2.2 illustrates the field of views of a linear (**a**) and a curved (**b**) ultrasound probe. A special arrangement is the circular arrangement (**c**). This arrangement can be interpreted as a curved array with an acquisition angle of 360 ∘. It is used for catheter probes that are employed to investigate vessels from the inside (intravascular ultrasound (IVUS), compare Section 5.1).

In order to increase the quality of the acquired images, multiple elements can be excited simultaneously instead of only a single element at once. Figure 2.3 **a** illustrates this behavior.

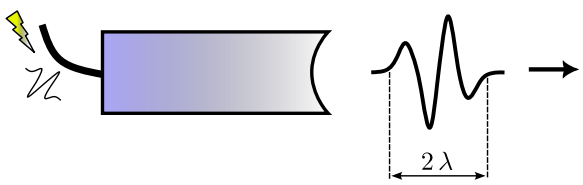

Figure 2.1: Sketch of a piezo transducer. The transducer is excited with an electric pulse. The piezo vibrates according to the applied voltage and emits an acoustic pulse, i.e., a short wave packet. Usually, the length of the pulse is a multiple of the wavelength. The concave shape of the transducer tip improves beam focusing.

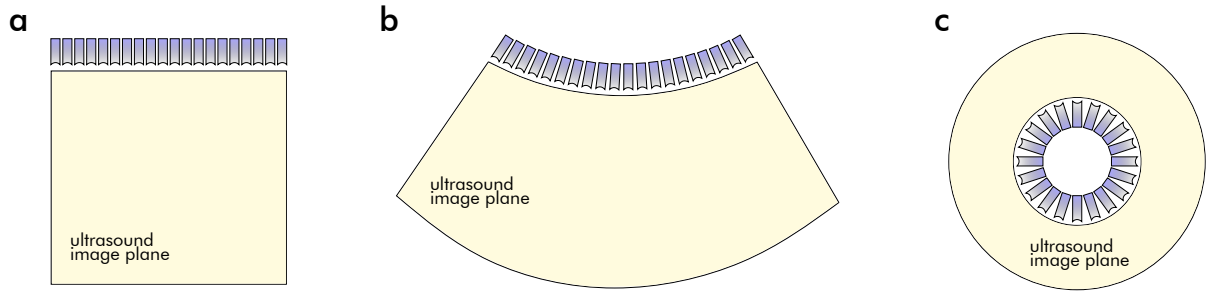

Figure 2.2: Sketches of different transducer arrays. Each array comprises 20 transducer elements. **a:** A linear array. The resulting image field is a rectangle. **b:** A curved array. The resulting image field is a circular sector. Thus, the imaged region is larger compared to the linear array. However, the resolution decreases with increasing penetration depth. **c:** A circular array is used for catheter probes. It is a special case of the curved array allowing for an observation angle of 360 °.

Here, only 5 elements are excited at once. After the acquisition of the signals from this beam is finished, the next beam is generated by shifting the active elements by 1 to the right. The one-dimensional backscattered signals by the individual beams, the scan lines, are stitched together to obtain the final two-dimensional ultrasound image. We further discuss signal processing below. The beam focus can be improved by array delays (Figure 2.3 **b**). Delays in the electrical excitation signals account for different travel times of the incident wave between the individual transducer elements and the focal zone. The delays can be applied during emission and also during receiving. In the case of receiving focusing, the signals by the individual transducer elements are combined with a delay to obtain the scan line. The same principle can be used to steer the beam in different directions. This is achieved by applying a ramped delay configuration. Arrays that operate in this way are referred to as "phased arrays".

How an emitted acoustic pulse behaves depends on the medium in which it propagates. The 3 physical quantities that describe the motion of a wave are frequency f, phase velocity c, and wavelength λ. Their relationship is expressed with the following equation:

$$c = \lambda \cdot f. \tag{2.1}$$

The velocity of the pulse can be assumed to be approximately equal to the phase velocity. The velocity c varies for different tissues. In practice, a value of $c = 1540 \frac{\mathrm{m}}{\mathrm{s}}$ is assumed for soft tissue. Typical frequency values depend on the application but usually lie somewhere between 0.5 MHz and 50 MHz. Thus, estimated values of the wavelength lie between 3 mm and 30 μm.

We can differentiate between three different types of interaction between a wave and the objects it hits:

1. the object is much larger than the wavelength (specular scattering)
2. the object is much smaller than the wavelength (diffusive scattering)
3. the object has about the same size as the wavelength (diffractive scattering)

The third type of interaction is rather complex and not discussed in this work. Further information on diffractive scattering in the context of ultrasound imaging can be found in Szabo [247]. Scattering facilitates acoustic signals, which contain information about the objects that

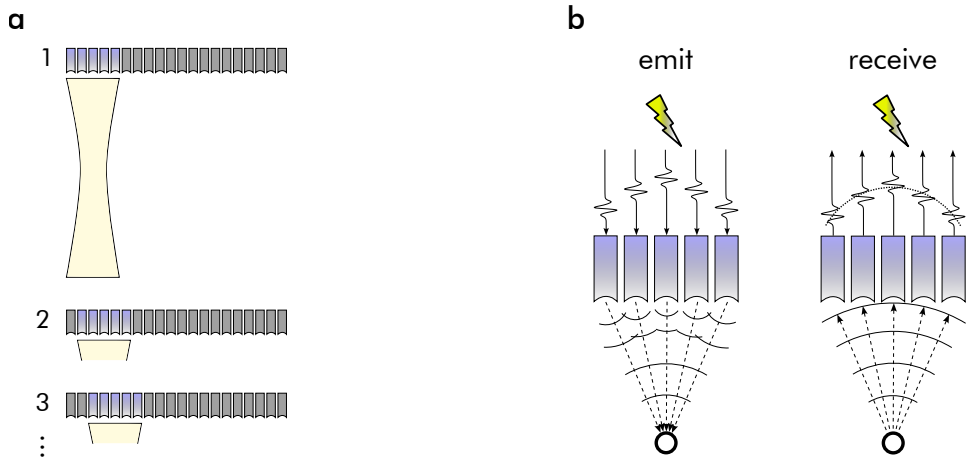

Figure 2.3: Sketches illustrating ultrasound beam geometry. a: Individual beams are generated by only exciting a small number of adjacent transducer elements. After the echo of the first beam has been acquired, the second beam is generated by shifting the active elements one step to the side. This procedure is repeated until the last element in the row has been activated. The individual echoes of each beam are later combined to generate an image. **b:** Focusing during emission and receiving through array delays. In the case of emission, delays in the electrical excitation signals facilitate the focusing of the incident wave by considering different travel times between transducer elements and the focal zone. The same principle can be applied to the received signal by combining delayed signals (dotted line).

were hit, to travel back to the ultrasound probe. To understand specular scattering, we first introduce the acoustic impedance Z. Z is a material parameter and depends on the material's density ρ and sound velocity c:

$$Z = \rho \cdot c \tag{2.2}$$

In the case of specular scattering, we can consider the large object to be another medium in which the wave can propagate. Both media are thus separated by an interface. The incident energy of the wave is therefore split into a reflected and a transmitted part. The corresponding ratios are the reflection coefficient R and the transmission coefficient T. For a perpendicular incidence of the wave upon a large object, R can be calculated with

$$R = \frac{(Z_1 - Z_2)^2}{(Z_1 + Z_2)^2} \tag{2.3}$$

with Z_1 and Z_2 being the impedances of both media. The transmission coefficient results in

$$T = 1 - R = \frac{4\,Z_1\,Z_2}{(Z_1 + Z_2)^2}. \tag{2.4}$$

We can see that the reflected energy increases if the deviation of the impedances of both involved media increases. It is irrelevant whether the wave travels from large to small impedance or vice versa. Specular scattering is the cause of an ultrasound-typical imaging artifact. Acoustically dense objects, like calcifications inside vessel walls (compare Subsection 5.1.4), almost completely reflect the incident wave. Therefore, only very weak backscattered signals can be received from behind such objects. In corresponding images, regions behind acoustically dense objects appear very dark. This effect is referred to as "acoustic shadowing". Acoustic shad-

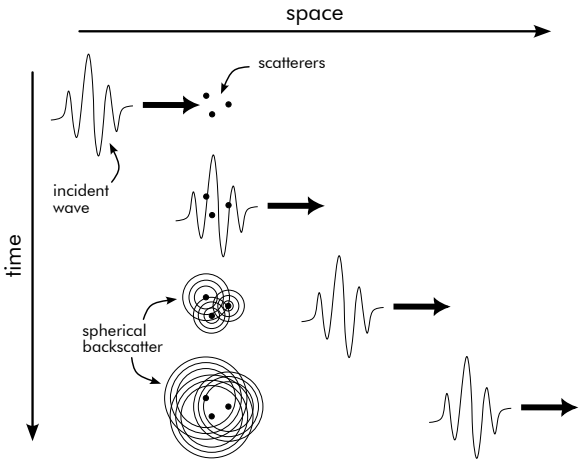

Figure 2.4: Sketch illustrating diffusive scattering. A incident wave with a wavelength much larger than the scatterers propagates from left to right and passes the scatterers. Due to the interaction with the incident wave, the scatterers vibrate and thus emit spherical waves. Note that the incident wave and the secondary spherical waves have the same velocity. For the sake of clarity, we have omitted a realistic depiction of this fact.

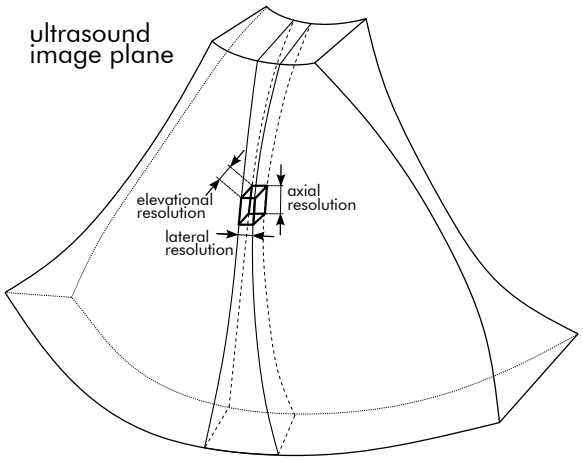

Figure 2.5: Sketch illustrating the ultrasound resolution cell. The ultrasound resolution cell is defined by axial, lateral, and elevational resolution and mainly depends on the transducer geometry and the wavelength. Since the penetration depth of the beam also depends on the wavelength, a compromise between these two factors has to be found for each use case. The resolution cell affects the formation of speckle noise. Figure based on Hangiandreou [97].

owing is intensified by attenuation, which is quite strong in acoustically dense objects (see below).

Figure 2.4 illustrates diffusive scattering. The objects hit by the incident wave are considered as points emitting spherical waves after the interaction. Diffusive scattering is the reason why ultrasound images also show textures between interfaces. These textures usually exhibit a special kind of noise referred to as "speckle noise" [34]. The phenomenon of speckle noise is related to the resolution of an ultrasound image. The resolution of an image is dependent on the ultrasound beam geometry. The beam geometry defines the resolution cell. The resolution cell is composed of the axial resolution, the lateral resolution, and the elevational resolution. Figure 2.5 **b** depicts a sketch that illustrates the resolution cell. The axial resolution mainly depends on the length of the ultrasound pulse and, thus, on the wavelength. The axial and elevational resolutions depend on the wavelength as well as the piezo and probe geometry. There is a trade-off between the size of the resolution cell and the penetration depth of the ultrasound beam. Reducing the wavelength decreases the resolution cell but, at the same time, reduces the penetration depth (compare the effect of attenuation described below). This trade-off has to be considered when designing probes for specific use cases.

Speckle emerges if the mean distance between scatterers is much smaller than the resolution cell. In this case, the individual backscattered signals can interfere constructively or destructively, depending on the propagation times of the receiving transducer elements. The intensity of backscattered signals from adjacent resolution cells can therefore deviate largely.

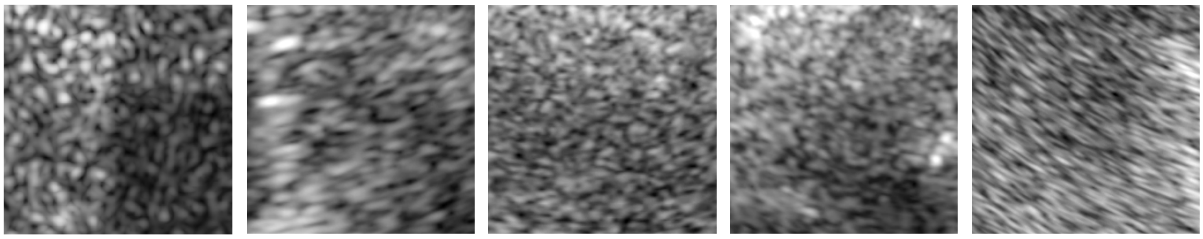

Figure 2.6: Exemplary images of speckle noise. The leftmost image contains simulated speckle noise (compare Subsection 4.3.3). The other images show real speckle noise. We can see that speckles may vary in size, shape, and contrast depending on the underlying tissue. Furthermore, speckles can appear deformed when the image is acquired with a curved array (rightmost image).

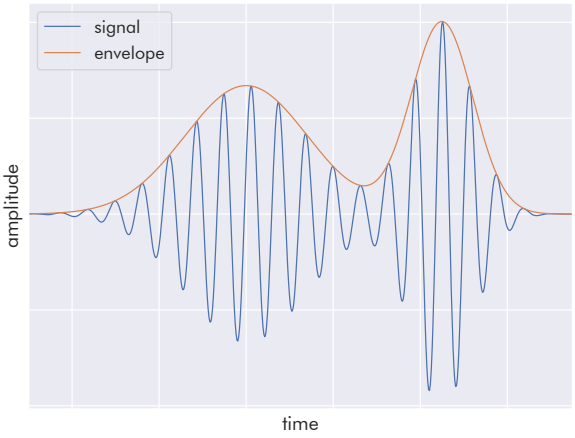

Figure 2.7: Exemplary signal with envelope. Since the carrier frequency of the ultrasound echoes does not provide useful information, it is removed to obtain the signal's envelope. This is done by means of the Hilbert transformation. The envelope then measures the amount of backscatter from different depths.

The resulting noise is characteristic of ultrasound signals. Exemplary images of speckle noise are depicted in Figure 2.6.

The last effect that we want to talk about is attenuation. Traveling through a medium, the intensity I of an acoustic wave decreases. A simple model for attenuation, subject to penetration depth z, is

$$I(z) = I_0 \cdot \exp(-\alpha\, z) \tag{2.5}$$

with the absorption coefficient α, which is a function of the medium and the wavelength. α usually increases for denser media and shorter wavelengths. Hence, the backscattered signals from deeper locations are much weaker compared to signals that are echoed from locations near the probe.

The backscattered acoustic waves finally hit the transducer elements. The resulting vibrations are transformed into electric signals that can be processed afterward. Since the echoes of different pulses should not mix, the pulse frequency cannot be set arbitrarily high. Instead, a new pulse is emitted only after the echoes of the previous pulse have been recorded. The backscattered signals $u(t)$ exhibit a frequency according to the excitation frequency. However, the information about the tissue is stored in the signal envelope. The envelope $v(t)$ can be extracted via a Hilbert transformation $\mathcal{H}$:

$$v(t) = |u(t) + j\,\mathcal{H}[u(t)]| \tag{2.6}$$

with j being the imaginary unit and

$$\mathcal{H}[u(t)] = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{u(\tilde{t})}{t - \tilde{t}} \, d\tilde{t} = u(t) \, * \, \frac{1}{\pi \, t} \tag{2.7}$$

with $*$ being the convolution operator. Figure 2.7 depicts an exemplary signal with its envelope. Attenuation effects are usually tackled by employing time gain compensation. This is, signal parts from deeper locations of the tissue are amplified such that attenuation effects are reversed. The electric signals are log-compressed and stitched together to form a two-dimensional array. This array can then be processed in different ways, e.g., smoothing or contrast enhancement. Finally, the array is transformed into color/brightness space, usually resulting in 256 bit gray scale images.

3 Fundamentals of Deep Learning

In this chapter, we will focus on deep feedforward neural networks and convolutional neural networks in the context of image analysis, particularly segmentation. Parts of this chapter are based on the comprehensive introduction to deep learning by Goodfellow et al. [83]. Further details on many topics can be found in that book. We limit this chapter's scope to those aspects essential to understanding the methods we used in this work.

In contrast to conventional image analysis methods that usually rely on manually defined features, deep learning methods are founded on the ability to learn features during a training process. However, training requires data. Two important categories of learning are supervised and unsupervised learning. The goal of supervised learning is that the neural network provides a preferably correct output $\hat{y}$, given a certain input $\boldsymbol{x}$. This includes classification or segmentation (see Section 3.3) of images. This type of training requires ground truth data, from which the network can learn a function $f(\cdot\,;\theta)$ by adapting parameters θ, such that

$$\hat{y} = f(\boldsymbol{x}\,;\theta). \tag{3.1}$$

The goal of supervised learning is to learn $f(\cdot\,;\theta)$ such that it approximates the ideal function f^* as well as possible. The function f^* defines the relationship between input $\boldsymbol{x}$ and correct output y:

$$y = f^*(\boldsymbol{x}). \tag{3.2}$$

The goal of unsupervised learning is to derive structural information from a dataset without having input-output pairs. Examples are clustering or learning the probability distribution that generates the data. We will return to this topic when introducing generative adversarial networks in Section 3.4.

3.1 Neural Network Components

This section briefly introduces all important network components essential for the methods we investigated in this work.

3.1.1 Densely Connected Layers

Neural networks usually comprise multiple layers, so the function $f(\cdot\,;\theta)$ is a composition ($\circ$) of multiple sub-functions. In the case of a neural network with L layers, we have

$$f(\cdot\,;\theta) = f^L(\cdot\,;\theta_L) \circ f^{L-1}(\cdot\,;\theta_{L-1}) \circ ... \circ f^2(\cdot\,;\theta_2) \circ f^1(\cdot\,;\theta_1) \tag{3.3}$$

with $\theta = \{\theta_1, \theta_2, ...\theta_{L-1}, \theta_L\}$ and $f^\ell(\cdot\,;\theta_\ell)$ the function defining the ℓth layer. The final layer $f^L(\cdot\,;\theta_L)$ is usually referred to as the "output layer". The other layers are called "hidden layers" since their outputs are not directly observable.

In the simplest form of a neural network, a multilayer perceptron [218, 219], the individual layers comprise an affine operation followed by a non-linear activation function. Without non-linear activation functions, the multilayer perceptron would be entirely linear and thus trivial. We can write the ℓth layer as

$$\boldsymbol{h}^\ell = g((\boldsymbol{h}^{\ell-1})^\top \boldsymbol{W}^\ell + \boldsymbol{b}^\ell) \tag{3.4}$$

with $\boldsymbol{h}^\ell$ being the output of the ℓth hidden layer, $\boldsymbol{W}^\ell = (w_{ij}^\ell)_{i,j}$ being the weight matrix, $\boldsymbol{b}^\ell$ being the bias term and g being the non-linear activation function. The parameters $\boldsymbol{W}^\ell$ and $\boldsymbol{b}^\ell$ are learned during training. Due to historical reasons, the individual elements of the layer outputs are called "neurons". One of the following functions usually serves as an activation function and is applied elementwise: sigmoid σ

$$\sigma(x) = \frac{1}{1 + \mathrm{e}^{-x}}, \tag{3.5}$$

hyperbolic tangent

$$\tanh(x) = \frac{\mathrm{e}^{x} - \mathrm{e}^{-x}}{\mathrm{e}^{x} + \mathrm{e}^{-x}}, \tag{3.6}$$

and the rectified linear unit (ReLU)

$$\mathrm{ReLU}(x) = \max(0, x). \tag{3.7}$$

Another version of the ReLU function is the leaky ReLU:

$$\mathrm{LReLU}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x <= 0 \end{cases} \tag{3.8}$$

with $\alpha \ll 1$ being a constant parameter.

3.1.2 Convolutional Neural Networks

Equation 3.4 shows that every element of a layer input $\boldsymbol{h}^{\ell-1}$ is used to calculate an element of the layer output $\boldsymbol{h}^\ell$. If we assume that input and output have m and n elements, respectively, the weight matrix comprises $m \cdot n$ elements. This shows that densely connected layers are unsuitable for analyzing images since the number of parameters would exceed feasible limits. Furthermore, the network would be prone to overfitting (see Subsection 3.2.4). A strategy to overcome these problems is weight sharing. Instead of using an individual weight for each connection between an input and output element, multiple connections share the same weight. The resulting operation can be described as a convolution. Neural networks built from convolutional layers are referred to as "convolutional neural networks" or CNNs for short. Important contributions in the development of CNNs are Fukushima [76], Waibel et al. [268], and LeCun et al. [146, 147]. If we assume that the layer input and output are two-dimensional, which we call "feature maps", we can write

$$\tilde{h}_{i,j}^\ell = (\boldsymbol{k}^\ell * \boldsymbol{h}^{\ell-1})_{i,j} + b_{i,j}^\ell = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} k_{m,n}^\ell \, h_{i-m,j-n}^{\ell-1} + b_{i,j}^\ell \tag{3.9}$$

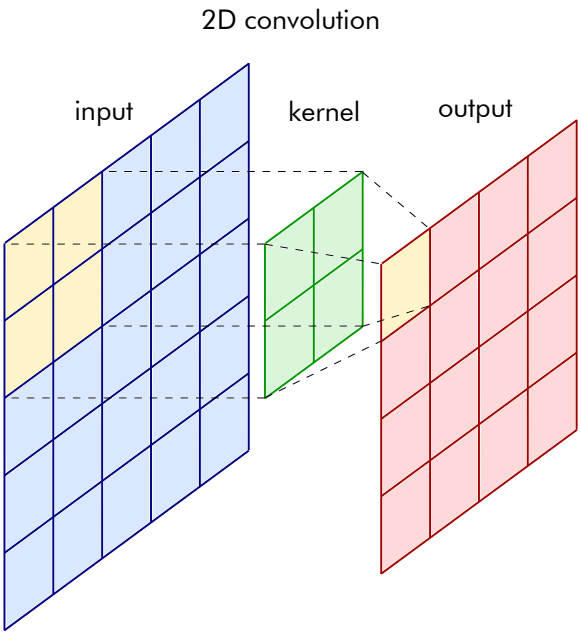

Figure 3.1: Sketch of an ordinary convolution. The 2×2 kernel (green) strides across the input (blue) and linearly combines the affected values of the input (yellow) with its weights to yield a single value (yellow) of the output (red).

with $*$ being the convolution operator and $\boldsymbol{k}^\ell$ being the convolution kernel, or filter, with a size of $M \times N$. To obtain the layer output $\boldsymbol{h}^\ell$, we then have to apply an activation function $\boldsymbol{h}^\ell = g(\tilde{\boldsymbol{h}}^\ell)$. Since the values of the kernel $\boldsymbol{k}^\ell$ are learned during training, the convolution is usually implemented as a cross-correlation that omits flipping one of the operands:

$$\tilde{h}^\ell_{i,j} = (\boldsymbol{k}^\ell \star \boldsymbol{h}^{\ell-1})_{i,j} + b^\ell_{i,j} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} k^\ell_{m,n}\, h^{\ell-1}_{i+m,j+n} + b^\ell_{i,j}. \tag{3.10}$$

Here, $\star$ denotes the cross-correlation operator. Since this operation reduces the spatial dimension of the output feature map by one pixel less than the kernel size, one can pad the input to keep the feature map size constant. The most prominent padding approach is zero padding. Figure 3.1 depicts a sketch of a 2D convolution (or cross-correlation) with a 2×2 kernel.

When processing images with CNNs, it is reasonable to add another dimension to the feature maps: the channel dimension. As color images comprise 3 channels, the feature maps can also have multiple channels. The convolutional layer can thus be written as

$$\tilde{h}^\ell_{s,i,j} = (\boldsymbol{k}^\ell \star \boldsymbol{h}^{\ell-1})_{s,i,j} + b^\ell_{s,i,j} = \sum_{r=0}^{R-1} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} k^\ell_{s,r,m,n}\, h^{\ell-1}_{r,i+m,j+n} + b^\ell_{s,i,j}. \tag{3.11}$$

Therefore, a single channel of the output feature map receives information from all input feature maps.

The cross-correlation can be viewed as the kernel or filter moving across the input feature map generating a single output value for each position. Usually, the filter moves in steps of a single pixel or with a stride of 1. When using larger strides, e.g., a stride of 2, the output feature map's size is reduced. With a stride of $s \in \mathbb{N}$, we have (omitting the channel dimension)

$$\tilde{h}^\ell_{i,j} = (\boldsymbol{k}^\ell \star \boldsymbol{h}^{\ell-1})_{i\cdot s,j\cdot s} + b^\ell_{i,j} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} k^\ell_{m,n}\, h^{\ell-1}_{i\cdot s+m,j\cdot s+n} + b^\ell_{i,j}. \tag{3.12}$$

Strided convolutions are often used to decrease the feature map size in deeper layers of the CNN while simultaneously increasing the number of feature map channels.

3.1.3 Other Components

Pooling

Besides densely connected layers, convolutional layers, and non-linear activation functions, neural networks can contain other components. Strided convolutions are not the only method to reduce feature map size. Another prominent method is pooling. We can think of pooling as special convolutional layers with pre-defined kernels often used with a stride of 2, thus halving the feature map size. Frequently used pooling layers are

- average pooling (average of values in the pooling window),
- max pooling (maximum of values in the pooling window),
- sum pooling (sum of values in the pooling window).

A unique form of pooling is global pooling which outputs a single value for each input feature map, e.g., by averaging or summing the whole feature map. This can be used to transform a stack of feature maps into a vector in cases where densely connected layers follow some convolutional layers.

Normalization

Applying a form of normalization after individual layers was found to be a rather useful tool for improving network performance [114, 108]. The most prominent normalization technique is batch normalization [114]. So far, it is not completely understood why batch normalization often improves the performance. However, it does. Usually, inputs are fed into CNNs in batches. Hence, a feature map tensor $\boldsymbol{h}$ exhibits four dimensions: batch, channel, height, and width

$$\boldsymbol{h} \in \mathbb{R}^{N \times C \times H \times W} \tag{3.13}$$

with N, C, H, and W being the batch size, the number of channels, the height, and the width of the feature map tensor, respectively. We use corresponding lowercase letters to index the respective dimensions (don't confuse h as a feature map element with h as a height index). Batch normalization (BN) is defined as

$$\mathrm{BN}(\boldsymbol{h})_{n,c,h,w} = \gamma_c \frac{h_{n,c,h,w} - \mu_c}{\sigma_c} + \beta_c \tag{3.14}$$

with

$$\mu_c = \frac{1}{NHW} \sum_{n,h,w} h_{n,c,h,w} \quad \text{and} \quad \sigma_c = \sqrt{\frac{1}{NHW} \sum_{n,h,w} (h_{n,c,h,w} - \mu_c)^2 + \varepsilon} \tag{3.15}$$

being the mean and standard deviation of channel c calculated by summing over the batch, height, and width dimensions. The scaling parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are vectors with a length of c and

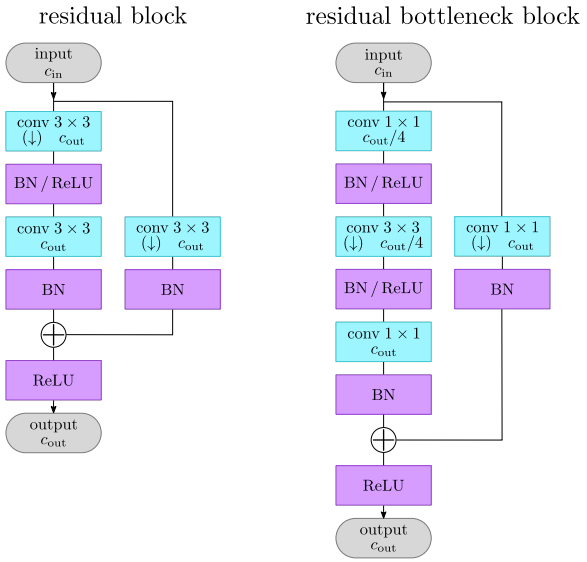

Figure 3.2: Diagrams of residual blocks. The basic residual block (left) contains 2 convolutions in the main path and an optional convolution in the parallel path. If the number of channels and the spatial size of input and output feature maps are the same, the parallel path is just an identity. If the spatial size of the feature maps should be halved, the first convolution in the main path and the convolution in the parallel path are applied with a stride of 2 (downward error). The bottleneck version (right) reduces the number of feature maps with a 1×1 convolution and increases it again in the third convolution of the main path.

are learned during training. They allow batch normalization to preserve the identity function. When adding batch normalization to a convolutional layer, the bias term in Equation 3.11 can be removed since they are redundant with the scaling factors of batch normalization.

Another form of normalization is instance normalization [261]. In contrast to batch normalization, instance normalization (IN) does not calculate the mean and standard deviation over the batch dimension. Instead, both statistics are calculated over the spatial dimensions only:

$$\text{IN}(\boldsymbol{h})_{n,c,h,w} = \gamma_c \frac{h_{n,c,h,w} - \mu_{n,c}}{\sigma_{n,c}} + \beta_c \tag{3.16}$$

Note that the scaling parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ have the same size as in the case of batch normalization. However, they are often omitted. Instance normalization is usually employed for generative tasks, not for predictive tasks. We will return to instance normalization when introducing generative adversarial networks (Section 3.4).

The residual block

Convolutions and other network components can be arranged into modules or blocks, which are then used to build a whole CNN. One of the most prominent blocks is the residual block [101]. Besides a main path with 2 or 3 convolutions, a parallel path is added - sometimes containing a single 1×1 convolution to change the number or the spatial size of feature maps. The outputs of both paths are summed to obtain the block output. Figure 3.2 shows sketches of the two most common types of residual blocks: the basic residual block (left) and the residual bottleneck block (right).

Again, convolution and normalization in the residual path are optional and are usually omitted if not necessary. As the basic residual block comprises 2 convolutions in the main path, the bottleneck version comprises 3 convolutions. The first one reduces the number of feature maps with a 1×1 convolution, the second one is an ordinary 3×3 convolution that does not alter the number of feature maps, and the third one increases the number of feature maps again. Convolutions with a kernel size of 1×1 are often used to alter the number of feature maps. Mathematically speaking, they simply linearly combine pixels of the same spatial

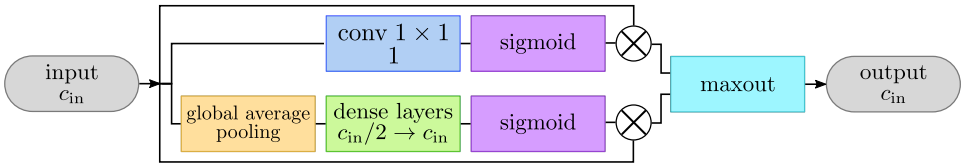

Figure 3.3: Diagram of a squeeze and excitation block. The upper path applies spatial attention, and the lower path applies channel attention. The sigmoid functions push the values into the range [0, 1]. Values near 0 indicate unimportant regions that are diminished, while values near 1 indicate important regions that are enhanced.

location across feature maps. Each convolution is followed by batch normalization. ReLUs are inserted at the shown positions. The downward arrows indicate an optional reduction of the spatial feature map size by employing a kernel stride of 2. One of the main theories why residual blocks often improve performance is the idea that gradients can propagate easier through the residual paths. Thus, the network training suffers less from the vanishing gradient problem (compare Section 3.2).

Squeeze and excitation

Another block that is sometimes used in CNNs is the squeeze and excitation block [106, 221]. It aims at enhancing important regions and diminishing unimportant regions of a feature map tensor by multiplying all pixels with values between 0 and 1. This approach is called "attention" and gained much importance in the context of neural language models, e.g., neural machine translation [14]. However, also image processing can improve by employing attention mechanisms such as squeeze and excitation.

The squeeze and excitation block can be inserted anywhere into a CNN since the input and output are of equal size. A sketch of a squeeze and excitation block is depicted in Figure 3.3. We can divide the block into a spatial attention path (top) and a channel attention path (bottom). For spatial attention, the input feature maps are convoluted with a 1×1 kernel to yield a single feature map. The sigmoid function scale the values into the range [0, 1]. The resulting spatial attention map is multiplied with each input feature map yielding spatially enhanced feature maps. For channel attention, global average pooling is used to transform the input feature maps into a feature vector with a length that equals to the number of input feature maps c_{in}. A few densely connected layers, which diminish the feature vector size in the first hidden layer, yield a feature vector, again, with a length of c_{in}. As with the spatial attention path, the sigmoid function scale the values into the range [0, 1]. The resulting channel attention coefficients are multiplied with the input feature maps, thus enhancing specific channels and diminishing others. Finally, the results of the spatial and channel attention paths are combined by taking the elementwise maximum of both tensors. The parameters of the convolutional layer and the dense layers are learned during training.

3.2 Network Training

So far, we have covered the architectural basics of neural networks, particularly convolutional neural networks. Now, we want to introduce the basics of neural network training. Only through proper training can networks learn efficient filters that enable the extraction of mean-

ingful features. Meaningful features, in turn, improve the networks' generalizability. This is the ability to perform well on data that was not used for training.

3.2.1 Loss Function

Neural network training is based on stochastic gradient descent. The network weights θ are adapted to minimize a cost or loss function $\mathcal{L}$, given the training data $\boldsymbol{x}$. This process is also called "optimization". In supervised learning, we want the network $f(\cdot\,;\theta)$ to yield a prediction $\hat{\boldsymbol{y}}$ that is similar to the ground truth $\boldsymbol{y}$. Loss functions therefore usually measure the deviation between the prediction and the ground truth and yield a scalar value. In general terms

$$\mathcal{L}(\boldsymbol{x};\theta) = \operatorname{dist}\left(f(\boldsymbol{x};\theta), \boldsymbol{y}\right) = \operatorname{dist}\left(\hat{\boldsymbol{y}}, \boldsymbol{y}\right) \tag{3.17}$$

with $\operatorname{dist}(\cdot\,,\cdot)$ being an arbitrary distance function, not necessarily a metric in strictly mathematical terms. The actual shape of the distance function is dependent on the underlying problem. Below in Section 3.3, we will define a loss function for image segmentation.

Since the whole neural network is made up of differentiable operations, we can calculate the gradient of each of the learnable parameters, or weights, with respect to the loss function. We can use these gradients to update the network weights by taking a step in parameter space towards the direction of the largest descent of $\mathcal{L}(\cdot\,;\theta)$. This method is called "gradient descent". CNN training requires a large amount of data. Hence, the data is fed into the CNN in small proportions called "batches". During training, loss calculation and updating the weights are done for each batch successively. The whole training set is fed multiple times into the CNN. A complete pass through the training set is referred to as an "epoch". The number of epochs to train is a hyperparameter and thus cannot be learned during training. A common practice is to define the number of epochs and the batch size beforehand.

The method with which the individual gradients are calculated is referred to as "backpropagation" [222]. It uses the chain rule of differentiation which essentially boils down the gradient calculation to successive multiplications. One factor for each operation between the parameter of interest and the loss function. The whole process comprises 4 steps:

1. forward propagation of the input through the network
2. calculation of the loss
3. backpropagation of the loss and thus calculation of the weight gradients
4. update of weights with the calculated gradients

Since gradients are calculated by numerous multiplications, it is obvious that gradients for shallow layers (early in the network) can get very small if the gradients of the individual layers are smaller than 1. This is called the "vanishing gradient problem". Above, we stated that this is precisely the problem that residual blocks try to circumvent (see Subsection 3.1.3).

3.2.2 Optimizers

Different optimizers have been developed for updating the weights θ, given their gradients with respect to the loss $\nabla_\theta \mathcal{L}$. The most simple one is the ordinary stochastic gradient descent

optimizer. The update rule for time step t looks as follows:

$$\theta_t = \theta_{t-1} + \Delta\theta_{t-1} = \theta_{t-1} - \eta \cdot \nabla_{\theta_{t-1}} \mathcal{L}. \tag{3.18}$$

The hyperparameter η is the learning rate. It scales the step sizes and is usually defined before training. However, it can also be changed at any point during training. Other optimizers make use of a momentum term. Optimizers with momentum not only take the current gradient into account but accumulate the gradients of past update steps to consider long-term optimization behavior. An update rule with momentum looks as follows:

$$\theta_t = \theta_{t-1} + \Delta\theta_{t-1} \tag{3.19}$$

$$= \theta_{t-1} + \gamma \cdot \Delta\theta_{t-2} - \eta \cdot \nabla_{\theta_{t-1}} \mathcal{L}. \tag{3.20}$$

Here, γ is another hyperparameter that controls the balance between the relative impacts of the current and past gradients.

Another quite prominent optimizer is Adam [138], a name derived from "adaptive moment estimation". In contrast to the ordinary momentum optimizer, it introduces bias-corrected moment estimates that are used to calculate the momentum. The update rule for time step t looks as follows:

$$\theta_t = \theta_{t-1} + \Delta\theta = \theta_{t-1} - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon} \tag{3.21}$$

with

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \qquad m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot \nabla_{\theta_{t-1}} \mathcal{L} \tag{3.22}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \qquad v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot (\nabla_{\theta_{t-1}} \mathcal{L})^2 \tag{3.23}$$

Adam comprises 4 hyperparameters. The learning rate η, exponential decay rate parameters β_1 and β_2, and ε, a small number for avoiding singularities of the fraction. Adam stores two values for each learnable weight, namely, the exponential moving averages of the gradient m_t and squared gradient v_t. It therefore requires more memory than simple optimizers like the ordinary stochastic gradient descent optimizer. However, this is usually only a small fraction of the amount of memory that is occupied by layer inputs and outputs and is thus often negligible. Adam is quite robust in terms of the values one chooses for its hyperparameters compared to other optimizers. Hence, it does not require extensive hyperparameter tuning and often facilitates fast convergence [138]. It is therefore quite frequently used for training neural networks.

3.2.3 Data Augmentation

Since a sufficiently large amount of data is crucial for successfully training a CNN, data augmentation methods are used whenever possible to artificially enlarge the underlying dataset [233]. In the case of image analysis, the most common approach is to transform images without changing their labels. The set of transformations that do not alter the labels depends on the task to be solved by the network. Common transformations are rotation, translation, contrast and brightness transformation, skewing, scaling, random cropping, and elastic transformation. For defining an image augmentation pipeline, it is essential to know which transformations are

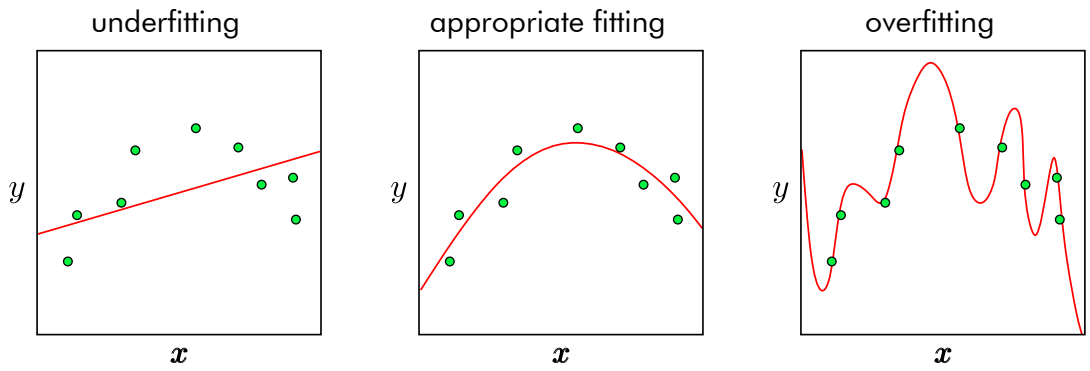

Figure 3.4: Different grades of fitting. Underfitting (left) indicates that the model cannot approximate the data-generating function because its capacity is insufficient. Overfitting (right) indicates that the model recognizes the noise in the data as important features that have to be considered. This usually happens if the model capacity is way too large. Appropriate fitting (center) indicates that the model has enough capacity to approximate the data-generating function while not considering the noise as important features. Figure based on Goodfellow et al. [83].

useful regarding the data that is likely seen by the network during deployment. For example, applying 180 ° rotations to images of a dog classification dataset is not very reasonable. However, smaller rotations will certainly be useful, as one cannot expect perfectly straight images during deployment.

Usually, the various image transformations are randomly applied online, i.e., during training. Across multiple training epochs, the CNN is fed with differently transformed instances of the same image. Thus, the dataset is artificially enlarged. However, completely new images often provide more valuable information compared to transformed instances of images that already belong to the dataset. Nevertheless, data augmentation is a comparably cheap method to improve CNN performance. This also holds for image segmentation (see Section 3.3). Here, data augmentation via spatial transformations is applied to both the input images and the corresponding segmentation masks. Corresponding images and masks must be transformed identically.

3.2.4 Overfitting

Overfitting is an important effect that has to be considered for all types of predictive models that learn from data. To evaluate the performance of a trained model, one calculates the prediction error of a set of examples that has not been used for training. In contrast to the training set, this other set is referred to as the "test set". The test set aims to simulate a potential model deployment, as, during deployment, the model sees data that was not used for training. If the performance on the test set is much worse than the performance on the training set, we often have a case of overfitting. This means that the model is "specialized" on the training set (or "memorizes" it) but cannot generalize to data that was not used for training. However, also the opposite behavior, i.e., underfitting, might occur. Figure 3.4 depicts underfitting (left), overfitting (right), and appropriate fitting (center) in the case of a simple regression task. The green dots indicate noisy values of the response variable y, which depends on the explanatory variable $\boldsymbol{x}$. While underfitting yields a regression function that does not

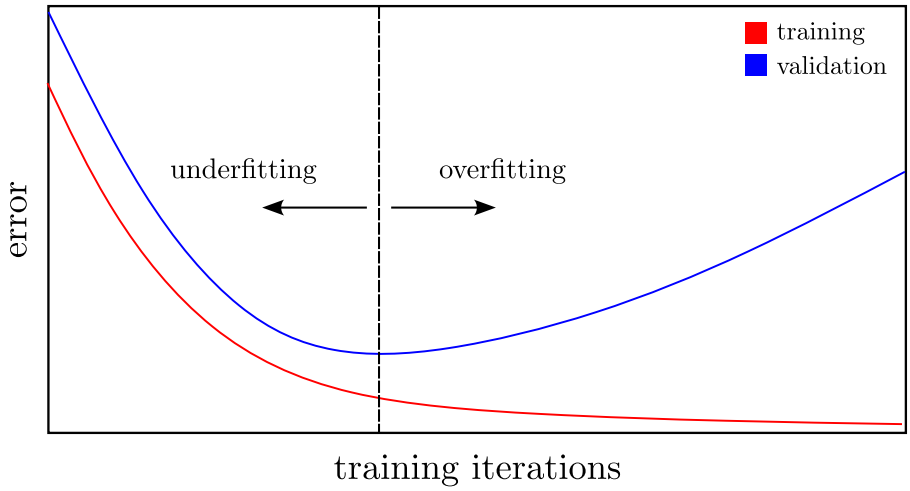

Figure 3.5: Training and validation error. While the training error usually decreases with training iterations, the validation error reaches a minimum and then increases again. The reason for this behavior is overfitting. When validating during training, the goal is to find this minimum and stop training. Figure based on Goodfellow et al. [83].

follow the values of y at all, overfitting yields a regression function that touches nearly all values of y, and thus oscillates strongly. However, both underfitting and overfitting yield models with quite poor predictive performance.

The tendency of a model to overfit depends on its capacity, i.e., the cardinality of the set of functions that the model could potentially learn. We can alter the capacity of a CNN by changing the number of layers or the convolutional kernel size, thus, changing the number of learnable parameters. Another critical factor is the number of training iterations. During the first training epochs, the training and test error usually decrease concurrently. At a certain point during training, the test error reaches its minimum. After that, the test error rises again while the training error continues to decrease. From this point on, overfitting begins, and the effect increases as training continues. Since computing the test error during training is not allowed, one usually defines a third set: the validation set. The validation set is not used for training or testing but to evaluate the model during training. Since the model is not trained with the validation set, it cannot overfit on it. The validation set is thus suitable for estimating the model's generalizability during training. Figure 3.5 depicts a typical training procedure in terms of the training and validation error.

An approach to prevent overfitting is early stopping. One monitors the validation error and stops training if the validation error increases for a certain number of steps in a row. The model can then be evaluated on the test set. The number of steps for early stopping is a hyperparameter and has to be defined before training. This approach is unfavorable if the training is inconsistent and not "smooth". In this case, one could stop way too early, not reaching a performance level that would be achievable if one had trained somewhat longer. Another possibility to prevent overfitting includes training for a sufficiently large number of epochs and simultaneously calculating the validation error. During training, one iteratively saves the model checkpoint that performs best on the validation set. After training, one loads the checkpoint with the best validation performance and evaluates its performance on the test set. This method requires that one is able to predict a sufficiently large number of

epochs. A drawback is the often longer training time compared to early stopping, as one usually overestimates the number of required epochs.

In order to further increase the significance of validation, one can perform cross-validation. Multiple types of cross-validation exist. The type that is often used for deep learning is k-fold cross-validation. Here, the training set is split into k parts. One part is used for validation, and the other k-1 parts are used for training. This procedure is repeated for each of the k splits of the training set yielding k different models (compare the sketches of the cross-validation splits of the datasets used in this work in Chapter 5). Each of these models is tested with the test set leading to k measures of performance. These can be used to calculate statistics and test for statistical significance, e.g., when comparing multiple models or methods. Alternatively, the k predictions of the k individual models can be combined by majority voting to yield a more stable prediction and, thus, even better performance compared to a single model.

3.3 Image Segmentation

3.3.1 Segmentation CNNs

After covering the architectural and training basics of neural networks, we now want to take a closer look at CNNs for image segmentation. The methods in this work unexceptionally deal with semantic image segmentation. Segmentation means that each pixel of an image is classified to belong to a particular segmentation class (see Chapter 5 for examples). A segmentation CNN therefore outputs a segmentation mask with the same spatial size as the input image.

One of the most commonly used CNNs for image segmentation, which we also use in this work, is U-Net [217]. It contains an encoding (contractive, downsampling) and a decoding (expansive, upsampling) path (see Figure 3.6). The encoding path contains layers that halve the size of the feature map, like pooling layers, strided convolutions, or simply downsampling layers. The decoding path contains layers that double the feature map size. These can be interpolation layers or transposed convolutions (more on this below). The layers that output feature maps with a given spatial size are referred to as "levels" of the U-Net. Another essential ingredient to U-Net is skip-connections between the encoding and the decoding path. They enhance the flow of gradients to shallow (early) layers during backpropagation and facilitate efficient information flow on all levels. Despite these properties, the block structure of a U-Net is rather arbitrary.

The original U-Net implementation comprised blocks with two convolutions and max pooling for downsampling [217]. In this work, we employ a U-Net architecture as depicted in Figure 3.6. The encoding and decoding blocks are residual blocks (see Subsection 3.1.3). We therefore call this architecture "U-Net-Res". Using residual blocks in U-Net was proposed by Milletari et al. [172]. Downsampling is performed with strided convolutions (indicated by a downward arrow), however, only in the first encoding block of a given level. The last number in each block indicates the number of output feature maps.

Upsampling is performed with transposed convolution (or informally deconvolutions) [66]. A corresponding sketch is depicted in Figure 3.7. Upsampling is achieved by a stride of 2. However, in contrast to ordinary convolutions, where striding is performed on the input, striding in transposed convolutions is performed on the output. Only in this case is upsampling possible. For more details, consult [66]. The skip-inputs are added elementwise to the upsampled feature

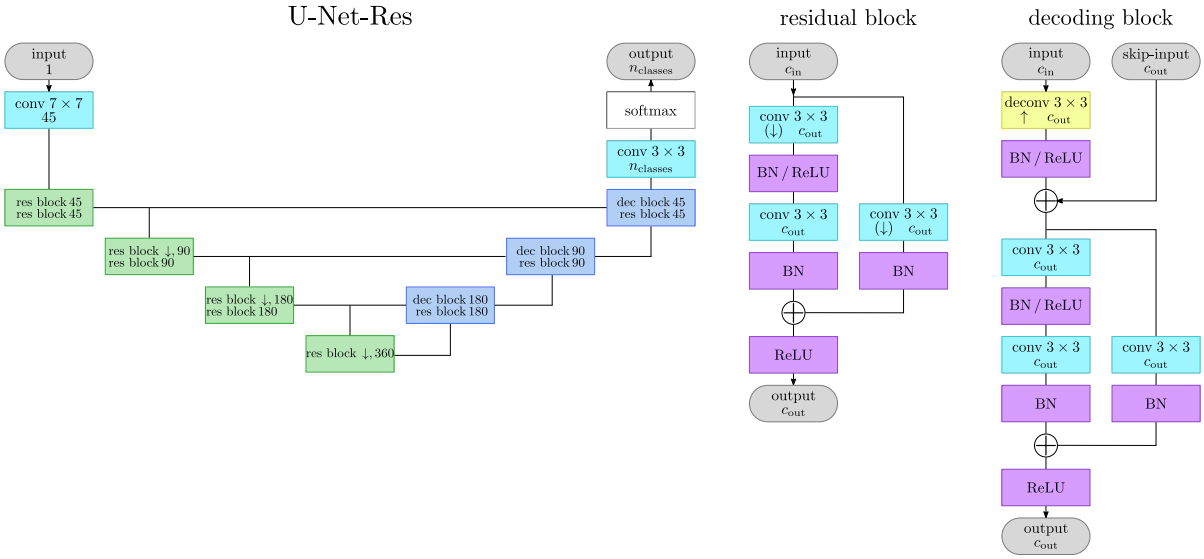

Figure 3.6: Sketch of the U-Net-Res architecture. Images with a single channel, i.e., grayscale images, are fed into the network. The output has the same spatial size as the input, and the number of channels corresponds to the number of segmentation classes. The softmax function ensures that the values of pixels with the same spatial location across the output sum up to 1. The encoding path comprises residual blocks, while the decoding path comprises decoding blocks as well as residual blocks. Skip-inputs are added elementwise to the deconvolved feature maps of the previous level. The number of feature maps is doubled (45, 90, 180, 360) each time a lower level is reached. Accordingly, the spatial size is halved, indicated by downward pointing arrows. Symbols in parentheses (like the downward pointing arrows) are optional and only considered if the symbol occurs in the corresponding blocks.

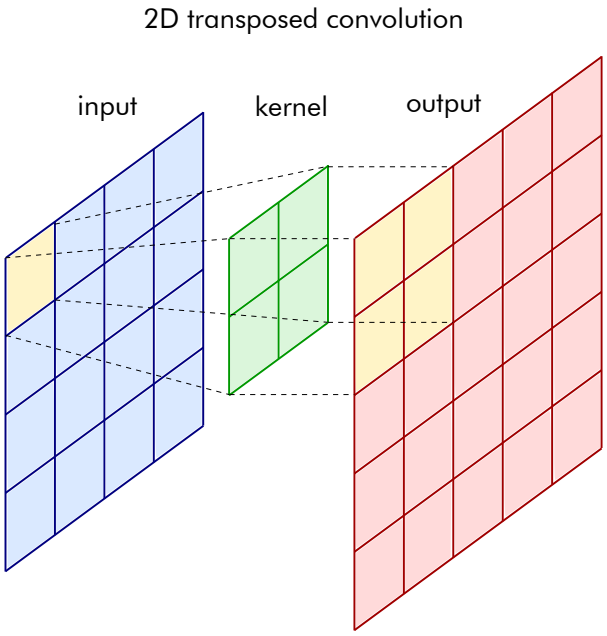

Figure 3.7: Sketch illustrating transposed convolution. Each individual pixel of the input is multiplied with the kernel weights. The resulting values are added according to the kernel location in order to produce the output. Compare to the ordinary convolution in Figure 3.1 to notice the parallels.

maps of the level below (Figure 3.6,right).

If K is the number of segmentation classes, U-Net-Res yields K unnormalized output maps, one for each class. These are normalized with the softmax function to obtain the final output that is used for loss calculation. The softmax is calculated over pixels with the same spatial location in order to find the segmentation class with the highest value at all spatial locations. The softmax function is a differentiable approximation of the maximum function and is used to push the values of all segmentation masks to the range [0, 1]. It is defined as

$$\text{softmax}(\boldsymbol{x})_j = \frac{\exp(x_j)}{\sum_{k=1}^{K} \exp(x_k)} \quad \text{for} \quad j = 1, ..., K. \tag{3.24}$$

Thus, the values of output pixels with the same spatial location across output maps add up to 1, yielding a discrete probability distribution for each spatial location. The resulting tensor is used to compute the loss (see Subsection 3.3.3).

Another prominent segmentation CNN we also use in this work is DeepLabV3 [42, 43]. It is composed of an encoder followed by atrous spatial pyramid pooling [43] and bilinear upsampling. See Figure 3.8 for a corresponding sketch. In general, any kind of CNN can be used as a backbone for the encoder. Initially, different kinds of ResNets [101] were employed [43]. The architecture we use in this work is based on ResNet50. However, since ResNets were designed for large natural image datasets, we diminished the number of convolutional channels to account for the smaller datasets of grayscale images we investigate in this work. In contrast to U-Net-Res, which comprises residual blocks, DeepLabV3 contains residual bottleneck blocks.

A distinctive feature of DeepLabV3 is atrous (or dilated) convolutions. To better understand the purpose of atrous convolutions, we have to discuss the receptive field. The receptive field of a particular neuron in a given layer is defined as the set of neurons in previous layers that contribute to the value of this specific neuron. A sketch illustrating the receptive field with an example of 3×3 convolutions is shown in Figure 3.9. We see that the blue neuron in layer 2 only receives information from blue neurons in layer 1. Likewise, the orange neuron in layer 3 receives information only from orange neurons in layer 2 and layer 1. The receptive field therefore grows with kernel size - 1 per layer in the backward direction. The receptive field implies that information cannot cross the boundaries of the receptive field. Hence, to consider long-range dependencies in images, the receptive fields of neurons near the output must cover large parts of the input image.

The goal of the DeepLabV3 developers was to use CNNs that were pre-trained on large image classification datasets like ImageNet [223] as a backbone encoder for segmentation. However, classification CNNs usually output a single vector, so the spatial feature map size gets very small in deeper layers. This is often done by strided convolutions or pooling. To prevent the backbone encoder from decreasing the feature map size, one can omit the stride of 2 in some convolutional layers. However, this drastically diminishes the receptive field of neurons in the final layers. Thus, long-range dependencies are not considered in the output but are crucial for successful segmentation. To increase the receptive field, Chen et al. [42] employed atrous (or dilated) convolutions. Atrous convolution kernels exhibit vacancies between their values. Their number depends on the atrous rate r. A value of $r = 2$ implies that a single vacancy is inserted between all kernel values. With this approach, the receptive field can be greatly enlarged. A corresponding sketch of a receptive field with 2×2 atrous convolutions and $r = 2$ is depicted in Figure 3.10.

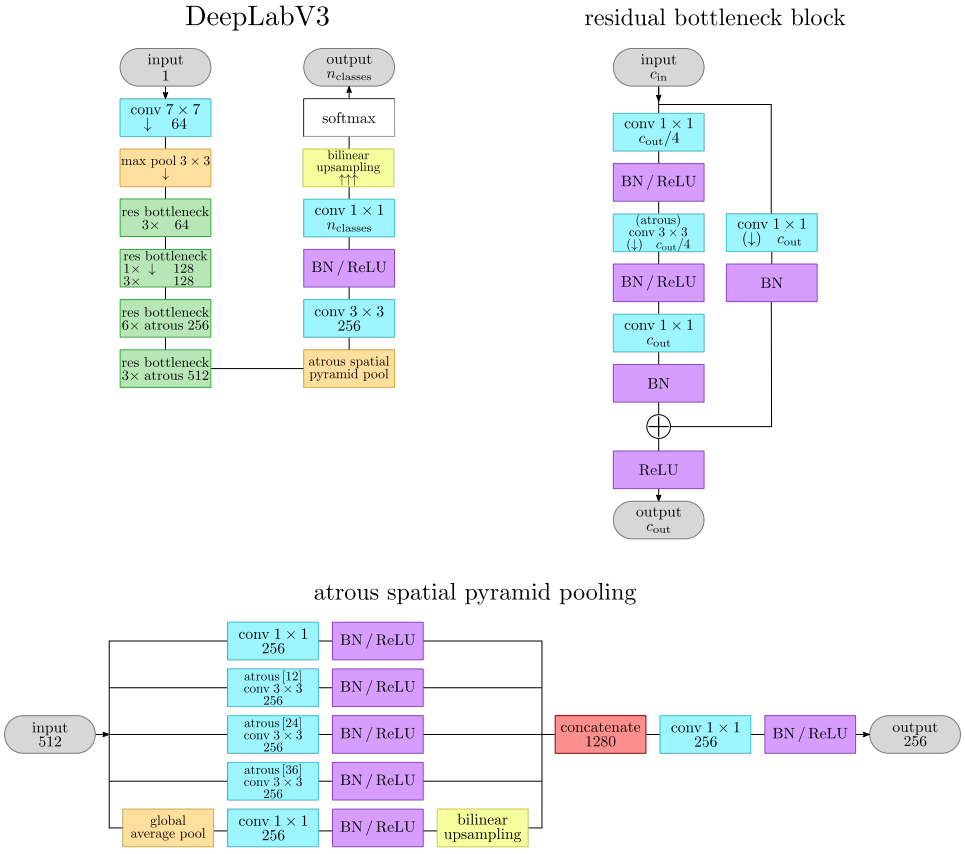

Figure 3.8: Architecture of DeepLabV3. It comprises multiple residual bottleneck blocks with and without atrous convolutions. Atrous convolutions ensure that the receptive fields of deeper neurons are large enough to consider long-range dependencies. The same goal is pursued with atrous spatial pyramid pooling. Since the encoder contains 3 downsamplings, bilinear upsampling is used to enlarge the feature maps with a scaling factor of 8.

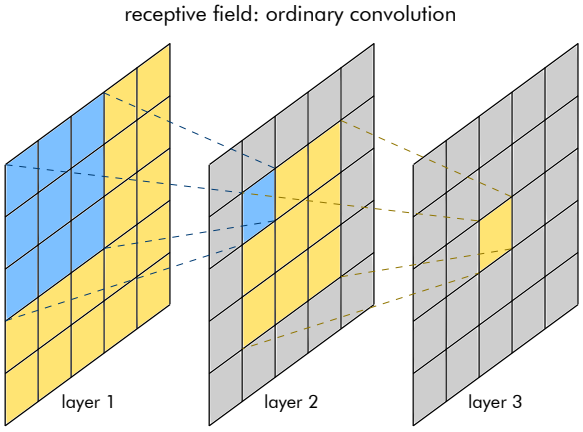

Figure 3.9: Receptive field with ordinary convolutions. The underlying kernels have a size of 3×3. The receptive field of a neuron increases with kernel size - 1 per previous layer.

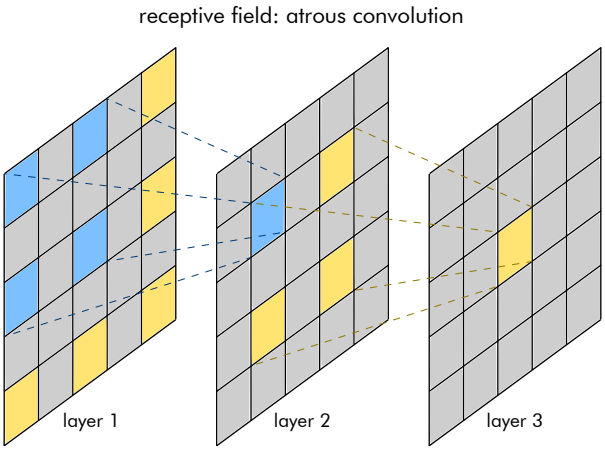

Figure 3.10: Receptive field with atrous convolutions. The underlying kernels have a size of 2×2 but are dilated with $r = 2$. Hence, there is a single vacancy between adjacent kernel elements. The resulting receptive fields have the same size as if ordinary convolutions with 3×3 were used. However, instead of 9 weights per kernel, only 4 weights have to be learned. Nevertheless, the vacancies do not provide any information.

Another important ingredient to DeepLabV3 is atrous spatial pyramid pooling [43, 42]. A corresponding diagram is shown in Figure 3.8 (bottom). The rationale behind atrous spatial pyramid pooling is to capture and combine information at different scales. This is accomplished by combining 5 branches. One branch contains an ordinary 1×1 convolution, 3 branches contain atrous convolutions with atrous rates of 12, 24, and 36, and the last branch contains global average pooling with a 1×1 convolution and bilinear upsampling. This approach aims at improving multi-scale object recognition.

Since DeepLabV3 does not comprise transposed convolutions or other layers to gradually increase the feature map size after encoding, the resulting segmentation maps have to be upsampled by bilinear interpolation. As the encoder contains 3 downsampling steps, the bilinear interpolation increases the spatial feature map size by a factor of 8. This implies that DeepLabV3 cannot resolve fine details smaller than 8 pixels. At the same time, the predictions of DeepLabV3 tend to lack small-scale noise.

3.3.2 Segmentation Metrics

A large variety of metrics have been developed for evaluating segmentation performance. Taha and Hanbury [251] provide a detailed overview of metrics for medical image segmentation. Important types of metrics are overlap-based and distance-based metrics. Overlap-based metrics measure the proportion of correctly classified pixels. Distance-based metrics can often be interpreted as measures of edge alignment. A rigorous evaluation of segmentation performance requires both types of metrics. A commonly used overlap-based metric is the Dice coefficient DC, or Dice score [58, 242]. For two sets A and B, it is defined as

$$DC(A, B) = \frac{2 \cdot |A \cap B|}{|A| + |B|}, \tag{3.25}$$

thus, ranging between 0 (no overlap) and 1 (perfect overlap). For binary segmentation problems, one can write

$$DC(A, B) = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{3.26}$$

with TP, FP, and FN being true positives, false positives, and false negatives, respectively. An important measure of edge alignment is the Hausdorff distance d_H [100]. For 2 sets A and

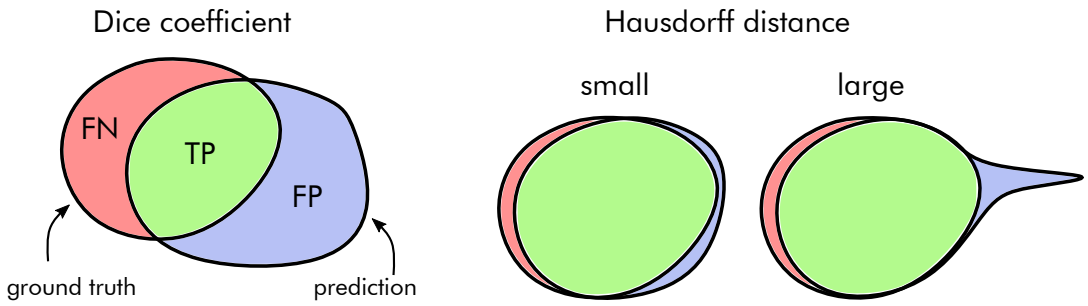

Figure 3.11: Sketch illustrating Dice coefficient and Hausdorff distance. The different parts for calculating the Dice coefficient are shown on the left. On the right side, examples with the same Dice coefficient but largely deviating Hausdorff distances are shown.

B, it can be defined as

$$d_H(A,B) = \max\left(\max_{a\in A}\min_{b\in B} d(a,b), \quad \max_{b\in B}\min_{a\in A} d(a,b)\right)$$

with a distance function $d(\cdot,\cdot)$, usually the euclidean distance. The Hausdorff distance is defined to fulfill the axioms of a metric in a mathematical sense. However, it is extremely sensitive to outliers. Therefore, the significance in terms of image segmentation is rather low since just a single pixel that deviates far from the ground truth completely spoils the result. Dubuisson and Jain [64] proposed a modified version of the Hausdorff distance that is more suitable for object matching. In this work, we refer to this modified Hausdorff distance as the "average Hausdorff distance". It is defined as

$$d_H^{ave}(A,B) = \max\left(\operatorname*{mean}_{a\in A}\min_{b\in B} d(a,b), \quad \operatorname*{mean}_{b\in B}\min_{a\in A} d(a,b)\right).$$

Here, the inner maximum operations of the ordinary Hausdorff distance are replaced with mean operations. Thus, the metric drastically reduces its sensitivity to outliers. This makes it an appropriate metric for image segmentation. Illustrating sketches for both metrics are shown in Figure 3.11.

3.3.3 Loss Function

Common loss functions for segmentation are often based on the types of metrics mentioned above: overlap-based and distance-based. An overlap-based loss function is the Dice loss [172], a differentiable version of the Dice coefficient. For a binary segmentation task with ground truth segmentation mask S and predicted, non-thresholded segmentation mask $\hat{S}$, it is defined as

$$\mathcal{L}_{DC} = 1 - 2\,\frac{\sum_n S_n\,\hat{S}_n}{\sum_n S_n^2 + \hat{S}_n^2} \tag{3.27}$$

with n indexing pixels. Sudre et al. [245] extended the ordinary Dice loss for multi-class segmentation with class imbalances. The generalized Dice loss $\mathcal{L}_{gDC}$ is hence defined as

$$\mathcal{L}_{gDC} = 1 \;-\; 2\,\frac{\sum_c w_c \sum_n S_n^c\,\hat{S}_n^c}{\sum_c w_c \sum_n S_n^c + \hat{S}_n^c} \tag{3.28}$$

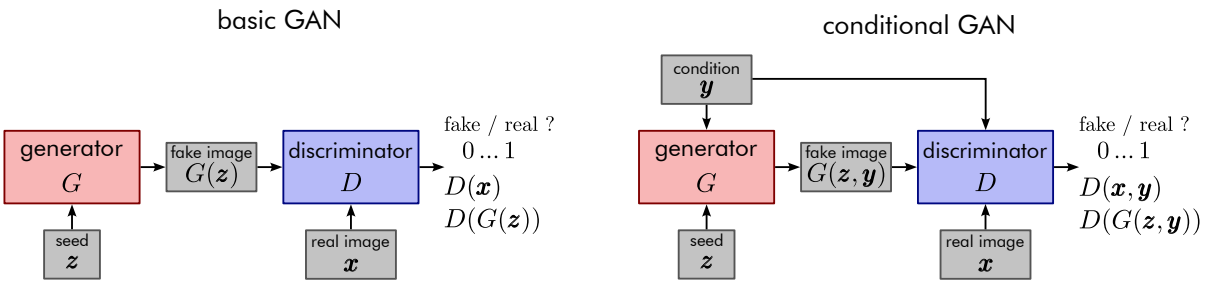

Figure 3.12: Diagrams of a GAN and a conditional GAN. The conditional GAN receives an additional input $\boldsymbol{y}$ that is considered by both generator and discriminator. The images generated by the generator therefore have to correspond to $\boldsymbol{y}$, since the discriminator rates pairs images and conditions.

with c indexing segmentation classes. The weights w_c are calculated with

$$w_c = \left(\sum_n S_n^c \right)^{-2} . \tag{3.29}$$

They correct the contribution of each class by its inverse area, giving more weight to smaller classes.

3.4 Generative Adversarial Networks

Parts of this section have been published in Bargsten and Schlaefer [22].

We already mentioned that learning methods can roughly be divided into supervised and unsupervised learning. So far, we have focused on supervised learning, i.e., training the CNN with a given ground truth, such as classification labels or segmentation masks. We now want to present a method that aims at learning the data distribution implicitly. If the data distribution is known, one can generate new synthetic data from it. Generative adversarial networks (GANs) were proposed by Goodfellow et al. [84]. The ways in which GANs can be used in the medical domain are manifold. We discuss examples in Section 4.3. Additionally, the review papers by Yi et al. [290], Kazeminia et al. [131], and Singh and Raza [238] provide a comprehensive overview.

In its basic form, a GAN consists of 2 sub-networks: a generator, which generates synthetic images from a random seed, and a discriminator, which predicts whether an image is an actual image from the data distribution or a fake image by the generator. The generator tries to fool the discriminator by generating images that the discriminator cannot distinguish from real images. Ideally, both sub-networks improve concurrently. While the discriminator improves in distinguishing real and fake images, the generator improves in generating images similar to real ones. An extension to the basic GAN is the conditional GAN (cGAN) proposed by Mirza and Osindero [174]. Here, a conditional input is provided to both the generator and discriminator. The goal is that the synthetic images correspond to the conditional input. Conditional inputs can be classification labels but also segmentation masks. Figure 3.12 illustrates the principle of a GAN (left) and a cGAN (right).

3.4.1 Spatially Adaptive Normalization

A rather elegant way to process segmentation masks as conditional inputs of a generator is spatially-adaptive normalization (SPADE) for semantic image synthesis [199]. SPADE layers transform segmentation masks (encoded as images with integer pixel values in $[0, ..., n_{\text{classes}} - 1]$ or as a stack of binary images, each corresponding to a segmentation class) into feature maps $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ by feeding them through 2 convolutional layers. The segmentation masks are resized before feeding them into SPADE in order to have the same size as the feature maps, which should be normalized. Values $h_{n,c,h,w}$ of feature maps $\boldsymbol{h}$ to be normalized are transformed as follows (also compare the normalization approaches in Subsection 3.1.3):

$$\text{SPADE}(\boldsymbol{h})_{n,c,h,w} = \gamma_{c,h,w} \frac{h_{n,c,h,w} - \mu_c}{\sigma_c} + \beta_{c,h,w}, \tag{3.30}$$

where the multi-index (n, c, h, w) refers to (sample in batch, channel, height, width). The parameters μ_c and σ_c denote the channel-wise mean and standard deviation of $\boldsymbol{h}$ (similar to batch normalization). SPADE ensures that the generated images correspond to the conditional segmentation maps.

3.4.2 Loss Function

Many loss functions for GAN training have been proposed in recent years [276]. However, the original loss functions proposed by Goodfellow et al. [84] are still used quite often. The loss comprises two parts: $\mathcal{L}_D$ for updating the weights of the discriminator D and $\mathcal{L}_G$ for updating the weights of the generator G. The discriminator rates an input image with a value between 0 and 1. A value near 0 indicates a fake image, while a value near 1 indicates a real image. Therefore, the generator's goal is to generate images rated as real (a value near 1) by the discriminator. This behavior can be transformed into two loss functions

$$\mathcal{L}_D(\boldsymbol{x}, \boldsymbol{z}) = -\left(\log(D(\boldsymbol{x})) + \log(1 - D(G(\boldsymbol{z}))\right), \tag{3.31}$$

$$\mathcal{L}_G(\boldsymbol{z}) = -\log(D(G(\boldsymbol{z}))), \tag{3.32}$$

with $\boldsymbol{x}$ being a real image and $\boldsymbol{z}$ being the random seed of the generator. These loss functions can be readily transferred to cGANs by simply considering the conditional input $\boldsymbol{y}$:

$$\mathcal{L}_D(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = -\left(\log(D(\boldsymbol{x}, \boldsymbol{y})) + \log(1 - D(G(\boldsymbol{z}, \boldsymbol{y}))\right), \tag{3.33}$$

$$\mathcal{L}_G(\boldsymbol{z}, \boldsymbol{y}) = -\log(D(G(\boldsymbol{z}, \boldsymbol{y}))). \tag{3.34}$$

Conditional segmentation masks can be concatenated to the input images along the channel dimension and fed into the discriminator. The generator can be conditioned to a segmentation mask by using SPADE. More details on cGANs for generating synthetic images for segmentation are presented below in Section 4.3.

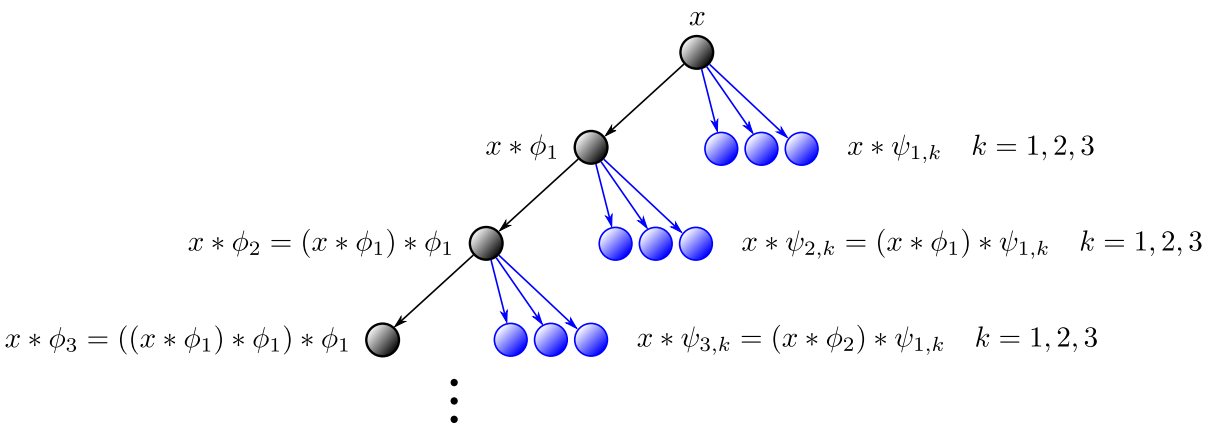

Figure 3.13: Diagram of the basic wavelet transformation. Shown is the fast wavelet transform algorithm. It is composed of downscalings with ϕ_1 and subsequent filterings with $\psi_{1,k}$ to obtain the wavelet coefficients $x * \psi_{j,k}$.

3.5 Wavelet Scattering

Parts of this section have been published in Bargsten et al. [20].

Wavelet scattering, as introduced by Mallat [165], is strongly related to CNNs. Wavelet scattering is based on wavelet transformation, which is widely used for image processing and analysis. The wavelet transformation provides a change of data representation analogous to the Fourier transformation. For $u \in \mathbb{R}^2$, a family of two-dimensional wavelet filters is defined as

$$\psi_{j,k}(u) = 2^{-j}\psi(2^{-j}R_{\theta_k}u) \tag{3.35}$$

with the mother wavelet $\psi(u)$, $j \in \{1, ..., J\} \subset \mathbb{N}$ and R as a matrix defining a rotation of angle θ_k with $k \in \{1, ..., K\} \subset \mathbb{N}$. The individual wavelets are therefore obtained by scaling and rotating the mother wavelet. In the same way, one can define a family of scaling functions

$$\phi_j(u) = 2^{-j}\phi(2^{-j}u) \tag{3.36}$$

that serve as low pass filters with scaling 2^{-j}. Analogous to Bruna and Mallat [32], we denote $\lambda = (j, k)$ and $\Lambda_{J,K} = \{\lambda = (j, k) : j \in \{1, ..., J\}, k \in \{1, ..., K\}\}$. The wavelet transform $Wx(u)$ of a two-dimensional signal $x(u)$ can be written as a set of convolutions

$$Wx(u) = \{x(u) * \phi_J,\ x(u) * \psi_\lambda(u)\}_{\lambda \in \Lambda_{J,K}} \tag{3.37}$$

with the convolution operator $*$. Figure 3.13 depicts a sketch of the wavelet transformation and how it is computed in practice with the fast wavelet transform algorithm. This algorithm is based on successive downsampling with ϕ_1 and subsequent filtering with $\psi_{1,k}$ to iteratively compute convolutions with higher order wavelets $\psi_{j,k}$ with $j > 1$.

The scattering transformation can be described as a cascade of complex wavelet transformations with beneficial properties. These properties include translation invariance, stability to noise, stability to deformations, and fast energy decay [165]. Energy decay means that the scattering coefficients decay to zero quite fast with increasing scattering order. An order of two usually proved to be sufficient [32].

To derive the corresponding equations, one defines an operator $U[\lambda]x = |x * \psi_\lambda|$ and a path

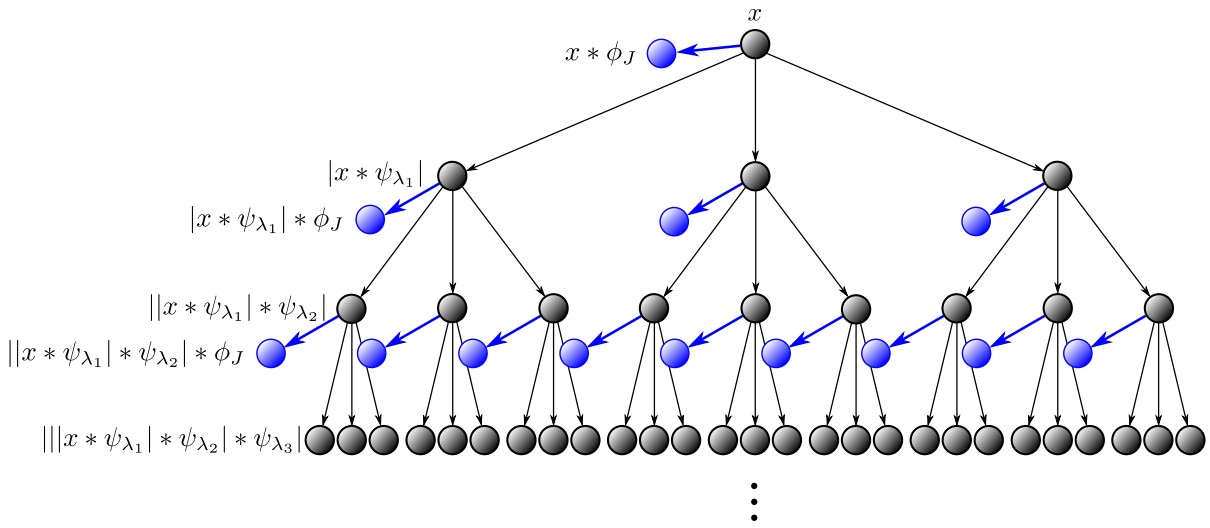

Figure 3.14: Diagram of the wavelet scattering transformation. It basically resembles a CNN with fixed filters ψ_{λ_i} and the modulus as a non-linearity.

$p = (\lambda_1, \lambda_2, ..., \lambda_m)$ representing a sequence of λ's. U can now be applied to x with respect to the path p via composition:

$$U[p]x = U[\lambda_m] \dots U[\lambda_2]\, U[\lambda_1]\, x \tag{3.38}$$

$$= || \dots |x * \psi_{\lambda_1}| * \psi_{\lambda_2}| \dots * \psi_{\lambda_m}|. \tag{3.39}$$

The resulting descriptors $U[p]x$ are processed with the scaling function ϕ_J yielding the scattering coefficients $S[p]x$:

$$S[p]x = U[p]x * \phi_J. \tag{3.40}$$

We now want to get the scattering coefficients of all possible paths. Let $\Lambda^m_{J,K}$ denote the set of all paths $p = (\lambda_1, \dots \lambda_m)$ of length m. Hence, $U[\Lambda^m_{J,K}]x = \{U[p]x\}_{p \in \Lambda^m_{J,K}}$ and $S[\Lambda^m_{J,K}]x = \{S[p]x\}_{p \in \Lambda^m_{J,K}}$. If one now defines the wavelet modulus propagator $\tilde{W}x(u)$ as (compare Equation 3.37)

$$\tilde{W}x(u) = \{x(u) * \phi_J(u),\ |x(u) * \psi_\lambda(u)|\}_{\lambda \in \Lambda_{J,K}}, \tag{3.41}$$

the scattering coefficients can finally be calculated with

$$\tilde{W}U[\Lambda^m_{J,K}]x = \{\tilde{W}U[p]x\}_{p \in \Lambda^m_{J,K}} \tag{3.42}$$

$$= \{S[\Lambda^m_{J,K}]x,\ U[\Lambda^{m+1}_{J,K}]x\}. \tag{3.43}$$

Figure 3.14 depicts a sketch of the wavelet scattering process. Comparing with Figure 3.13 shows the differences to the ordinary wavelet transformation. Instead of only convolving the downscaled versions $x * \phi_j$ of the input x with the wavelet filters $\psi_{1,k}$, the scattering transformation consists of composed filterings $||...|x * \psi_{\lambda_1}| * \psi_{\lambda_2}|... * \psi_{\lambda_m}|$. Thus, the wavelet scattering transformation is very similar to a CNN. While CNNs employ ReLU, sigmoid, or other non-linear activation functions, scattering transformations comprise the modulus as a non-linearity. However, instead of learning the filters, fixed wavelet filters are used. Originally, scattering

transformations were introduced with the Morlet wavelet as the underlying wavelet function. Performing scattering transformations with biorthogonal complex wavelets via a dual-tree complex wavelet transformation (DTCWT) [140] was proposed by Singh and Kingsbury [237]. In DTCWTs, the real and imaginary parts of the wavelets form a Hilbert pair and are processed in two separate trees with individual filters.

4 Deep Learning Methods for Ultrasound Image Segmentation

As mentioned in Chapter 1, image datasets in the medical domain are usually quite small. We identified several reasons for this, such as the need for well-trained experts to provide high-quality annotations in a rather time-consuming process, especially for segmentation labels. Additionally, we often face domain shift or want to detect diseases or lesions with only a small prevalence. The last point, in particular, is essential since recognizing rare diseases that are not very well known could be facilitated by neural networks that scan large amounts of image data for random findings. Therefore, deep learning methods that work well even when trained on relatively small datasets would greatly benefit global healthcare.

Traditionally, data augmentation and transfer learning are preferred methods when dealing with small datasets. More recent approaches exploit large unlabeled datasets with self-supervision. However, other methods have also been proposed, sometimes even considering dataset-specific knowledge. In the following, we will present some publications in these fields, focusing on medical image analysis. However, all these methods are rather versatile regarding the underlying dataset. Later on, we will also discuss ultrasound-specific approaches.

As explained in Section 3.2, traditional data augmentation techniques that do not change labels are usually used by default when training CNNs. We can divide these techniques roughly into geometrical and color transformations. Geometrical transformations include rotations, flips, crops, scaling, shearing, and elastic deformations. Color transformations include brightness, contrast, saturation, adding noise, and filtering. It depends on the situation which transformations are helpful and which are not. However, traditional data augmentation should be used whenever possible. The paper by Shorten and Khoshgoftaar [233] provides a good overview of many data augmentation approaches. Besides conventional spatial transformation-based data augmentation, also other approaches exist. As a rather unusual approach, we want to briefly take a look at mixup [296] that was originally introduced for image classification. With mixup, two input images and corresponding labels are linearly combined and fed into the network. Noguchi et al. [188] investigated mixup for improving bone segmentation in CT images. However, the results with mixup tend to be worse than the baseline.

Transfer learning is heavily used for natural but also for medical image analysis. With transfer learning, the goal is to improve the performance on a target task by initializing the CNN with weights that have already been learned on another task, the source task. One can also say that the CNN is "pre-trained" on the source task. Well-known datasets for pre-training in the natural image domain include ImageNet [223] for classification or COCO [156] and ADE20K [305] for segmentation. However, pre-training models for medical image analysis with natural image datasets only sometimes lead to improvements [211] since image contents, structures, and textures usually differ vastly. Wang et al. [269] provide an extensive review on transfer learning for medical image analysis showing mixed success. Pre-training with a medical

image dataset instead of a natural image dataset seems to be more promising. In the case of segmentation, Karimi et al. [126] found that improvements via transfer learning generally only occur if the target dataset is comparatively complex and small. A recent study on transfer learning for prostate segmentation in MR images was provided by Saunders et al. [227]. The authors systematically varied the number of training images and compared the results when employing pre-training, dataset fusion, or no transfer learning. The positive impact of transfer learning with another MRI dataset was most significant for the smallest training set of 5 cases and decreased with increasing training set size until it vanished for 40 cases. In the case of image classification, Matsoukas et al. [166] showed that the success of transfer learning from ImageNet to medical images depends on the similarity between the source and target images as well as the CNN architecture. However, they just focused on microscopic and X-ray images.

If a large amount of unlabeled data is available in addition to a labeled data set, self-supervision could be an option to improve performance. Self-supervised learning exploits unlabeled data for learning features that are beneficial for the downstream supervised task. Prominent methods include auto-encoding, in which the input image is encoded into a latent space and then reconstructed to its original appearance, or image inpainting, in which parts of the input image are removed manually and reconstructed by the CNN. CNNs or parts of CNNs that have been pre-trained in this fashion can then be fine-tuned with supervised learning. The papers by Jing and Tian [122] and Ohri and Kumar [189] provide extensive reviews on this topic. Instead of removing image patches, Chen et al. [41] swapped multiple patches in a single image and trained CNNs to predict their relative positions or reconstruct the image. The downstream tasks like classification, localization, and segmentation did only improve when using smaller datasets. Zhou et al. [308] proposed a similar approach but swapping three-dimensional cubes instead of two-dimensional patches. Nguyen et al. [184] tried to improve multi-organ segmentation in CT slices by employing a self-supervision task in which the CNN should detect manual replacements of patches across multiple image slices. Predicting the distance between the affected slices should help the CNN to extract contextual features better. The authors reported minor improvements compared to other methods. Improvements increased with decreasing dataset size of the downstream supervised task.

Besides the methods mentioned above, data augmentation, transfer learning, and self-supervision, other approaches for improving results on small datasets have been developed. Mainly, they involve some kind of regularization, often facilitated with domain knowledge. The idea of incorporating domain knowledge into CNN training will occupy us in a later section. In the following, we want to present two other approaches that have been used for ultrasound image segmentation, but are generally transferable to other modalities. [89] proposed a method that refines the output of a CNN with continuous kernel cuts [253] for cardiac MRI segmentation. The largest improvements over the baseline were reported for a small training dataset of only 5 subjects. Also, on larger datasets of 50 cases, improvements of over 40 % were achieved with respect to Hausdorff distance. However, the Dice score did not improve. For abdominal organ segmentation of CT images, Wang et al. [271] proposed intra- and inter-patient image pairs as CNN inputs. Besides the ground truth segmentation masks, the network also predicts the intersection and exclusive union of the two masks. The authors report minor, likely not significant, improvements when training with 80 % of the datasets (161 cases for liver segmentation and 32 cases for multi-organ segmentation). However, improvements get significant in extreme cases of using 5 % or 1 % of the datasets for training.

Summarizing, established methods like data augmentation, transfer learning, and self-supervision are frequently used in deep learning. Whereas (conventional) data augmentation can and should be used in almost any case, the applicability of transfer learning and self-supervision is often limited, especially in the medical domain. Transfer learning requires a large annotated auxiliary dataset that should be more or less similar to the target dataset. In our experiments with ultrasound segmentation, we did not find any advantage of transfer learning [19]. Self-supervision only requires a large unlabeled dataset, but the benefits are usually smaller compared to transfer learning. Moreover, in the medical field, it is often difficult to even obtain unlabeled data due to ethical or privacy restrictions. Therefore, developing and investigating methods that further exploit dataset-specific properties is crucial. Usually, CNNs need large training datasets with sufficient variety to learn filters that can extract meaningful features. These features, in turn, allow for appropriate generalizability and robustness to unseen data. Hence, possible methods to improve performance on small datasets could act upon enhancing the explanatory power of extracted features.

This chapter will present and explain our developed deep learning methods to tackle the aforementioned challenges in the context of ultrasound image segmentation. These methods are based on wavelet scattering (Section 4.1), domain knowledge for regularization (Section 4.2), and domain knowledge for synthetic image generation (Section 4.3).

4.1 Combining Wavelet Scattering and CNNs

Parts of this section have been published in Bargsten et al. [20].

As mentioned in Section 3.5, discrete wavelet transformations (DWTs) are usually calculated by performing a multiresolution analysis via a filter bank. The wavelet scattering transformation extends this idea, i.a., by introducing the modulus operator as a non-linearity. Such a filter cascade is quite similar to a CNN, which itself is basically a filter cascade with activation functions in between. Hence, we can interpret wavelet scattering transformations as CNNs with pre-defined filters. This allows wavelet scattering transforms and CNNs to be combined in many different ways.

This section presents previous work on wavelet approaches for medical data analysis. We motivate when and why combining wavelet or scattering transformations with CNNs could be beneficial and discuss limitations. We then present a novel CNN block incorporating wavelet scattering to increase performance on ultrasound image analysis.

4.1.1 Previous Work

Wavelet transformations have a long tradition in automated analysis of medical data. One-dimensional signals like electrocardiograms and electroencephalograms can efficiently by filtered and analyzed with wavelets [262, 258]. But wavelets have also been used extensively for processing medical images. Amajor application is denoising. The division of (two-dimensional) signals into multiple sub-bands covering different frequency ranges makes denoising quite flexible and thus customizable [195].

In ultrasound imaging, denoising usually means despeckling. Speckle reduces the visibility of boundaries between different tissues and diminishes image contrast resolution (compare Chapter 2). However, experienced radiologists can extract meaningful information from tissue texture like speckle [141, 240]. Hence, speckle reduction should be applied with caution to ensure no relevant information is lost. Nonetheless, in some cases, speckle can severely impede image interpretation. By removing or smoothing specific wavelet sub-bands, speckle noise can be reduced [207, 123]. Since despeckling enhances boundaries between tissues, it is an essential pre-processing step for conventional ultrasound image segmentation methods [244, 231, 13, 59].

For classifying different tissues, extracted wavelet features are typically used to train a classifier like a random forest [246, 214, 228], a support-vector machine (SVM) [295, 244, 228] or a simple neural network [60, 38, 292, 99]. This approach can also be used to segment images by combining the classification of wavelet features with clustering [295, 209], morphological operations [209] or probability models [5].

In recent years, some attempts have been made to enhance convolutional neural networks (CNNs) with wavelet pre-processing. This means that instead of feeding only raw images into the CNNs, a stack of wavelet transforms is used as a network input. Fujieda et al. [73] performed texture classification by feeding different wavelet sub-bands into different locations of a CNN with corresponding feature map size. A similar approach was used by Khatami et al. [133] for X-ray image classification. Liu et al. [158] integrated a discrete wavelet transformation (DWT) and an inverse DWT into a CNN such that the CNN learns most of its filters in wavelet space. Similar approaches have been studied for image segmentation [239, 163]. Very recently, Zhao et al. [300] presented a method for cardiac ultrasound segmentation that employs DWTs for downscaling and inverse DWTs for upscaling in a U-Net. They furthermore combined the

U-Net with another encoder-decoder network and an attention mechanism. Besides our work [20] and Cotter and Kingsbury [51], this is the only published approach so far that integrates wavelet or scattering transformations into a CNN. The method by Zhao et al. [300] and ours [20] have been developed independently.

In addition to ordinary wavelet transformations, wavelet scattering transformations have also been investigated as a pre-processing step for CNNs [236, 237, 196, 51]. Cotter and Kingsbury [51] fused dual-tree complex wavelet transformation (DTCWT) scattering transformations and CNNs by splitting the scattering orders into convolution-like layers - termed *invariant layers* - and adding mixing terms which can be learned during training. The authors provided an implementation for Pytorch [50], which we also used for our studies. In their work, Cotter and Kingsbury [51] inserted invariant layers before and after the first ordinary convolutional layer of a CNN but were unable to outperform other state-of-the-art CNNs on CIFAR-10. Nevertheless, they were able to reduce the number of network weights substantially. So far, approaches involving scattering transformations have mainly been tested on natural image benchmark datasets like MNIST or CIFAR-10. An extensive analysis regarding the performance on medical image data is still pending.

4.1.2 Squeeze and Excitation with Scattering Transform

It remains an open question in which cases CNNs can benefit from a preceding wavelet or scattering transformation. Most of the mentioned publications only reported minor or no improvements compared to a CNN-only baseline or evaluated their methods on relatively small natural images (32×32 pixels in the case of CIFAR-10). Wavelet and scattering transformations are basically carried out by convolving the input image with pre-defined filters yielding rather meaningful features. A CNN trained with a large amount of data usually learns considerably efficient filters to solve the underlying objective. In this case, a preceding wavelet or scattering transformation would not provide any benefit. Therefore, we hypothesize that combining wavelet or scattering transformations with CNNs only leads to performance improvements if the training dataset is rather small. In this case, the CNN is unable to learn efficient filters and yields features that are not very meaningful. Features of integrated wavelet or scattering transformations could therefore provide valuable additional information.

As we discussed in Chapter 1, datasets are often rather small in the medical domain. Furthermore, some applications inherently allow for only small amounts of available data, e.g., the detection of diseases with marginal prevalence. Hence, combining wavelet or scattering transformations with CNNs could benefit medical image analysis. We therefore want to investigate whether incorporating scattering transformations into CNNs can increase their ability to extract meaningful features when trained with small ultrasound image datasets. In preliminary experiments, we observed that the existing method of using scattering transformations as a pre-processing step to CNNs did not improve performance. We therefore propose a new method of fusing scattering transformations and neural networks. As mentioned previously, using DTCWT scattering transformations as a first or second layer of a CNN did not outperform state-of-the-art CNNs on CIFAR-10. Nevertheless, the scattering transformation is an efficient feature extractor and could therefore also be beneficial in other parts of a CNN. Based on the squeeze and excitation block [106, 221], we developed an attention module for CNNs that uses the DTCWT scattering transform. Figure 4.1 depicts a corresponding sketch. The number of input feature maps c_{in} is first reduced to one via convolution with kernel size 1×1 yielding

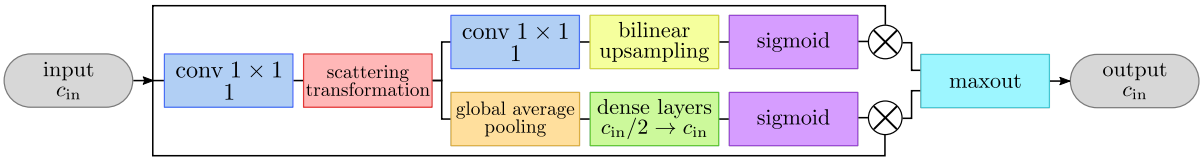

Figure 4.1: Diagram of the SEST block. Compared to the ordinary squeeze and excitation block (Subsection 3.1.3), the input feature maps are aggregated into a single channel by a 1×1 convolution. This feature map is then scattering transformed and fed into the spatial attention (top) and channel attention path (bottom). Since the scattering transformation reduces the spatial feature map size, bilinear upsampling is performed to obtain the spatial attention map.

a condensed representation of the input. After that, the scattering transformation is applied. The result goes into two separate paths, one for spatial and the other for channel attention. The spatial attention path results in a single feature map with values between zero and one, indicating unimportant and important regions of the input. The channel attention path results in c_{in} neurons with values between zero and one, indicating unimportant and important channels of the input. The results of both paths are multiplied with the input, respectively. Finally, both results are combined by an elementwise max operation leading to the output with c_{in} feature maps.

We call this block *squeeze and excitation with scattering transform* (SEST). It can be integrated into any CNN, basically everywhere. In general, one can increase the number of output feature maps of the first convolution, e.g., to c_{in}/k with $k \in \mathcal{N}$, but this can drastically increase the number of output feature maps of the scattering transformation layer. For c_{in} input channels, six different orientations of the underlying wavelet (which is the standard value for biorthogonal DTCWTs), and a 2nd order scattering, the number of output feature maps is $c_{out} = c_{in} \cdot 7^2 = c_{in} \cdot 49$, which can usually get very large in modern CNN architectures. We therefore restricted the number of output feature maps of the initial convolution of the SEST block to one. All scattering transforms in this work are of 2nd order.

The images in Figure 4.2, Figure 4.3, Figure 4.4, and Figure 4.5 show the 49 output features of a 2nd order scattering transformation of an example image from each of the datasets investigated in this work (see Chapter 5 for introductions to the datasets). The scattering transformation is calculated with the implementation by Cotter [50], which includes a mixing operation after the actual scattering transformation. The mixing operation is simply a convolution with a 1×1 kernel that linearly combines pixels with the same spatial location from each feature map. The kernels are learned during training. The image at the top left is the input of the scattering transformation. One can see that different regions are highlighted in different output feature maps. However, the way in which the CNN combines these in subsequent convolutions is not transparent. The experiments in Section 6.2 will reveal which kind of attention maps can be generated with SEST.

4.1.3 SEST Network Architectures

We integrated SEST blocks into the 2 baseline CNNs we presented in Subsection 3.3.1. Both resulting network architectures are depicted in Figure 4.6. In the corresponding baseline networks, scattering transformations were replaced with equivalent ordinary learnable convolutional layers. The baseline CNNs thus have up to 4 k more parameters.

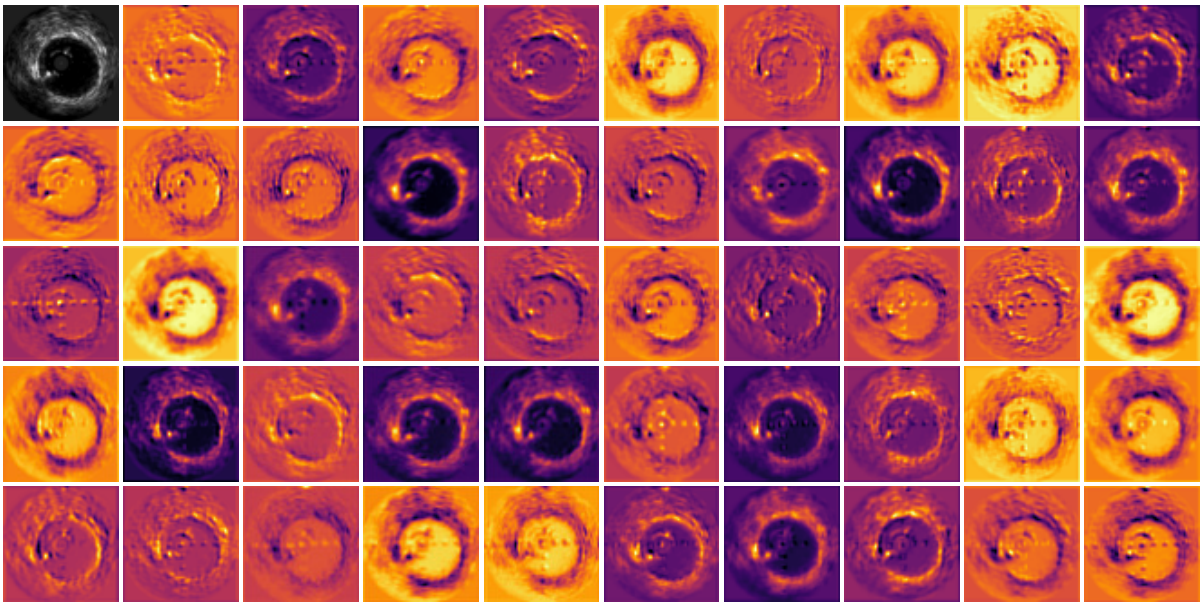

Figure 4.2: Scattering transformation of an image from the IVUS lumen and vessel wall dataset. The input image is depicted in the top left corner. The resulting 49 feature maps highlight different regions and show different features. See Subsection 5.1.4 for an introduction to the dataset.

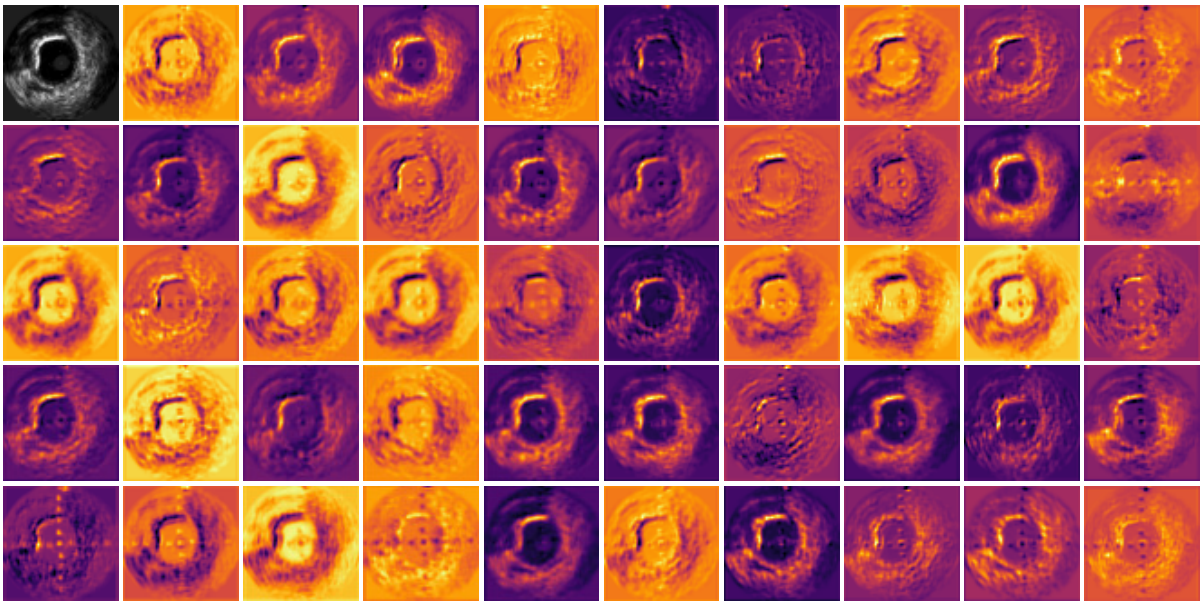

Figure 4.3: Scattering transformation of an image from the IVUS calcium dataset. The input image is depicted in the top left corner. The resulting 49 feature maps highlight different regions and show different features. See Subsection 5.1.4 for an introduction to the dataset.

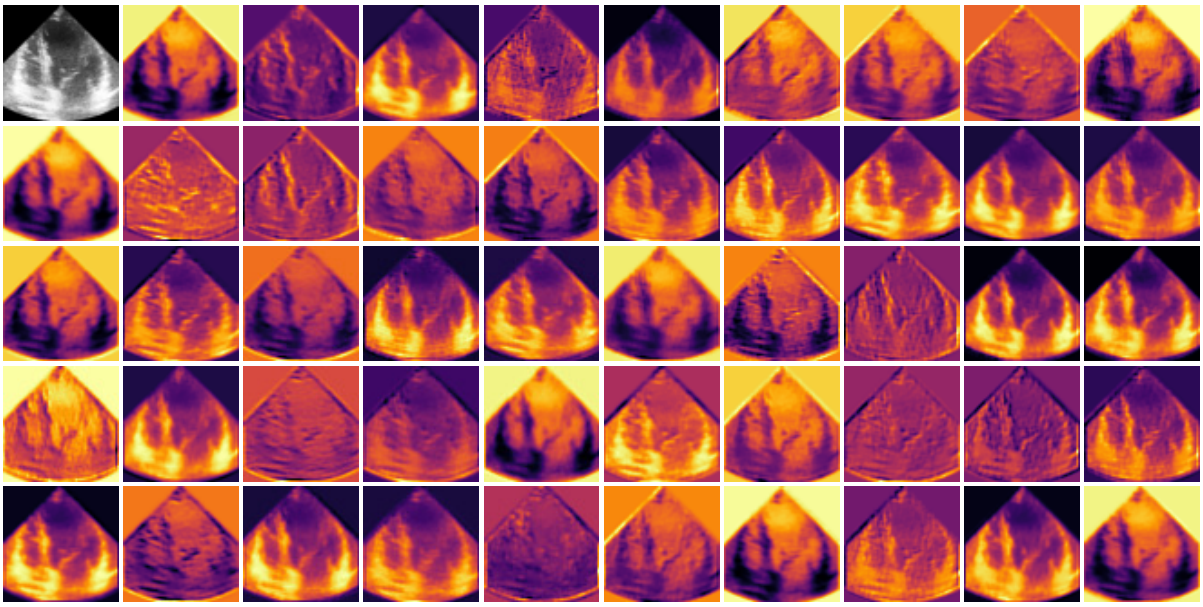

Figure 4.4: Scattering transformation of an image from the cardiac dataset. The input image is depicted in the top left corner. The resulting 49 feature maps highlight different regions and show different features. See Subsection 5.2.3 for an introduction to the dataset.

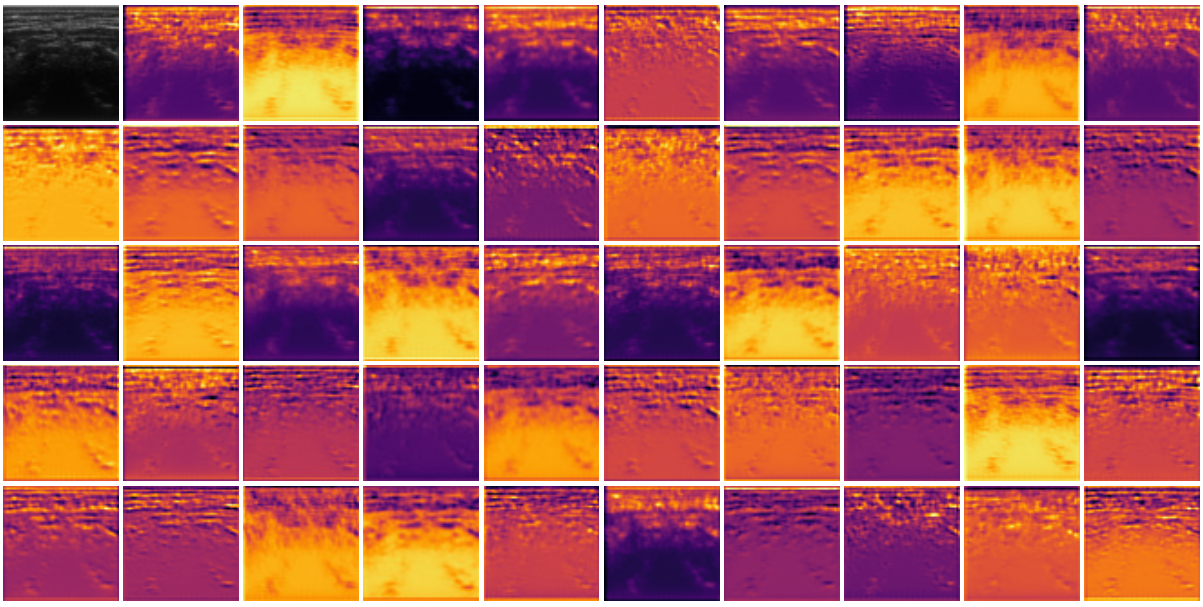

Figure 4.5: Scattering transformation of an image from the neck muscle dataset. The input image is depicted in the top left corner. The resulting 49 feature maps highlight different regions and show different features. See Subsection 5.3.3 for an introduction to the dataset.

4.1.4 Summary

We have seen that previous work on combining DWTs with CNNs is pretty much limited to using DWTs for pre-processing. Inserting wavelet scattering into CNNs as a first or second layer was only done by Cotter and Kingsbury [51], but despite a reduction of network parameters, no performance improvements on CIFAR-10 could be reported. We now want to answer **RQ 1.1**.

RQ 1.1: How could wavelet scattering help improve segmentation performance on small ultrasound datasets?

Since CNNs can learn efficient filters by themselves if enough training data is available, we assume that combining wavelet or scattering transformations with CNNs is only beneficial when the underlying datasets are small. SEST combines the effectiveness of wavelet scattering to extract meaningful texture features with an attention mechanism that guides the CNN to focus on regions that are important for segmentation. Rather than employing wavelet scattering only for pre-processing, our SEST block can be inserted into any CNN at any location. This approach could improve the extraction of meaningful features and thus improve segmentation performance, even for smaller datasets. The results of corresponding experiments are presented in Section 6.2.

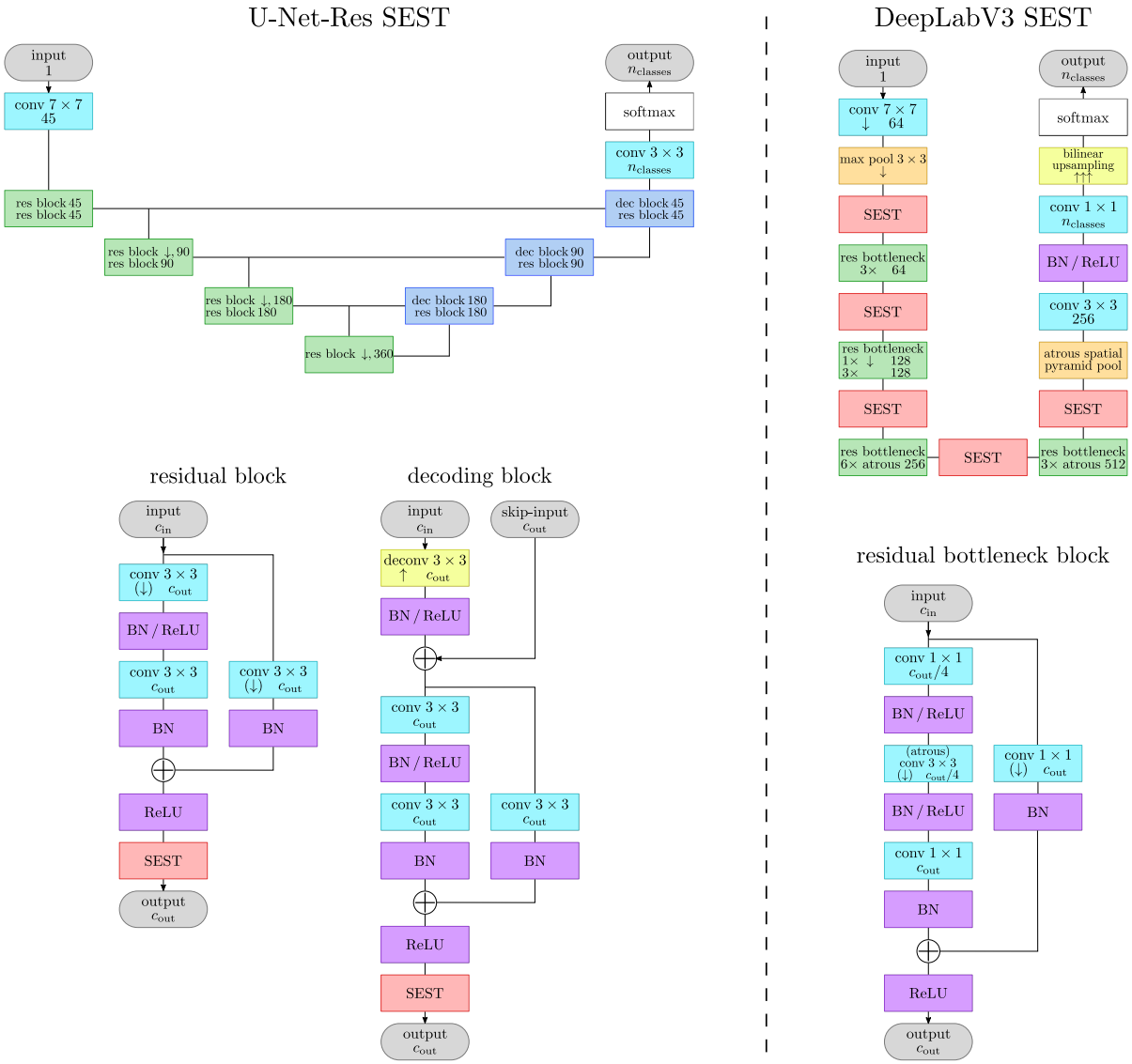

Figure 4.6: Diagrams of U-Net Res and DeepLabV3 extended with SEST blocks. Besides the inserted SEST blocks, the architectures resemble the basic ones (see Subsection 3.3.1). While U-Net-Res showed improved performance when inserting SEST blocks in each residual block, DeepLabV3 benefited more from inserting SEST blocks only between chunks of residual blocks.

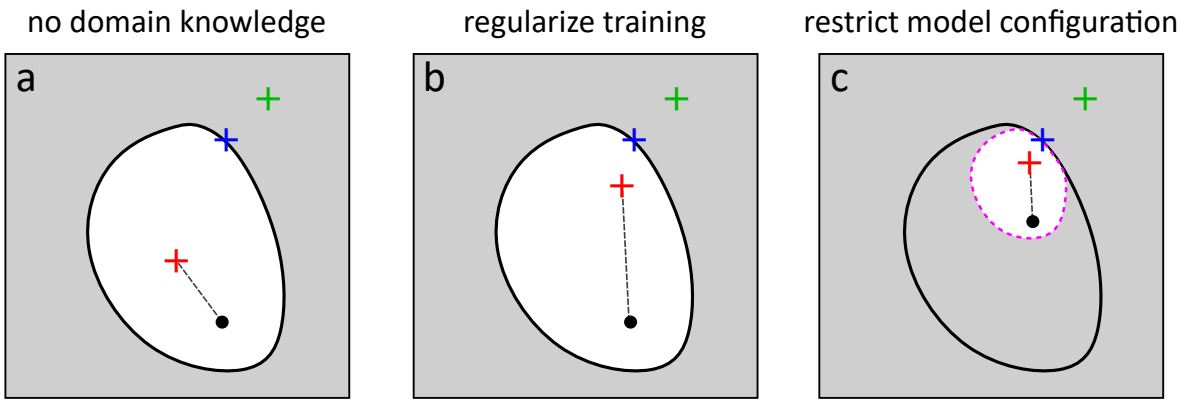

Figure 4.7: Sketch illustrating the impact of different domain knowledge approaches on model configuration space. Gray regions indicate model configurations that the underlying CNN architecture can not reach. The black dot indicates the model state after random initialization. The red cross marks the model configuration after training, the blue cross marks the best reachable model configuration, and the green cross marks the optimal model configuration (not reachable). **a:** Situation without any adaptations. **b:** By applying regularization for network training, the direction of the trajectory in configuration space, as well as the final position after training, could be optimized. **c:** An advantageous situation after initialization can be attained by appropriately restricting the configuration space.

4.2 Incorporating Domain Knowledge Into CNNs

The primary hypothesis motivating domain knowledge is that incorporating domain knowledge into deep learning can regularize network training. Figure 4.7 illustrates theoretical considerations about how regularization might positively influence model training. However, the positive effect of regularization vanishes with larger dataset sizes because the CNN will reach decent minima in the potential landscape and thus learn to extract meaningful features without any regularization by domain knowledge. Nonetheless, many approaches in this field have been proposed to solve different tasks (Dash et al. [56] provides an overview). After presenting previous work, we introduce our own contributions to this field with respect to ultrasound image analysis.

4.2.1 Previous Work

The field of domain knowledge in deep learning for medical image analysis is vast and cannot be discussed exhaustively at this point. However, the work of Xie et al. [283] provides a broad overview of different approaches. Here, we will discuss only a few selected types of methods, leading us to our contributions to this field.

One category of methods uses auxiliary attention map labels, which are fed into the CNN as additional information. These maps usually correspond to the areas that physicians focus on when examining the images and have to be generated before training. Chai et al. [35] used such attention map labels to let the CNN focus on the optic disc in eye fundus images for glaucoma diagnosis. Li et al. [153] enhanced this method by generating attention maps from eye-tracking recordings of physicians inspecting real images. Vakanski et al. [265] computed attention maps via quadratic programming optimization based on region connectedness and integrated these into a U-Net encoder for breast tumor segmentation in ultrasound images. To improve the

performance on breast cancer diagnosis with contrast-enhanced ultrasound images, Chen et al. [37] extended a CNN to transform specific examination patterns by physicians into temporal and channel attention blocks.

Another category of methods deals with combining learned and hand-crafted features to enhance the CNN's generalizability. Yang et al. [287] employed the established "Thyroid Imaging Reporting and Data System"-features [256] for seeding a GAN that was used for thyroid nodule classification in ultrasound images. These features were also used by Chen et al. [44]. Similar approaches have been developed for brain tumor detection in MR images [224], lung nodule detection in CT images [252] as well as melanoma classification [91].

Also, text reports can be used to boost the performance of different medical image analysis tasks. Usually, the text is encoded into some latent space and then combined with extracted image features. Shi et al. [230] used encoded text descriptions of lesions as auxiliary features for thyroid nodule classification in ultrasound images. Zhang et al. [299] employed descriptive texts about the disease state of cancerous bladder tissue for classification. The chest X-ray classification model by Wang et al. [274] did not only learn from images and text descriptions but was also capable of generating preliminary reports.

A comparably large area of research regarding domain knowledge for medical image segmentation is anatomical shape priors. Although exhibiting small local changes from patient to patient, the global shapes of organs and their topology are usually consistent among patients. However, the integration of such shape priors into CNNs can be realized in many different ways. An extensive review on this topic can be found in [67]. A rather classic way to consider anatomical knowledge in image analysis tasks are atlases, pre-defined templates of organ shape and topology, that are transformed by an algorithm to match the input image. Iglesias and Sabuncu [113] provide a detailed review of non-deep learning atlas segmentation methods. However, atlases can also be integrated into a deep learning framework. Vakalopoulou et al. [264] trained multiple CNNs to learn deformations of corresponding atlases to segment interstitial lung disease in CT images. A single probabilistic atlas prior for CT liver segmentation was used by [301] to increase the loss weights of harder training samples analogous to the focal loss [155]. The probabilistic atlas prior is calculated by averaging the ground truth segmentation masks of the training set. Zotti et al. [312] concatenated a probabilistic atlas prior to CNN feature maps near the network output to improve cardiac segmentation in MR images.

Shape priors can also be encoded into latent space. For these types of shape priors, an autoencoder is usually trained on the ground truth segmentation masks to obtain a preferably disentangled latent space. During the training of the segmentation network, an auxiliary loss is calculated that measures the difference between encodings of the predicted and ground truth segmentation masks. Such and similar methods have been investigated for cardiac image segmentation in ultrasound and MR images [190, 291], scapula segmentation in MR images [29], segmentation of neck and head in CT images [255, 79] as well as liver segmentation in CT images [177]. Tappeiner et al. [255] additionally combined their encoder with a principal component analysis for encoded segmentation masks. Boutillon et al. [29] combined their shape prior with an adversarial GAN loss to further regularize CNN training. Finally, Gao et al. [79] did not pre-train an autoencoder but trained an adversarial autoencoder in conjunction with the segmentation network.

Another subgroup of shape prior methods is incorporating topological or geometrical constraints via an additional term in the loss function. For segmenting gland histology images,

BenTaieb and Hamarneh [25] exploited the condition that lumen and goblet cells are always surrounded by epithelial boundaries, which are contained in a matrix of stromal nuclei. Therefore, they introduced a loss function that rewards the correct topological hierarchy. Reddy et al. [216] applied this idea to brain tumor segmentation in MR images. When segmenting brain MR or whole-body CT images with many different classes, regularizing CNN training by a loss function that ensures correct topology can be beneficial. Ganaye et al. [78] recognized this and defined a loss function that penalizes forbidden class combinations of adjacent pixels by multiplying their corresponding softmax outputs. To improve dental CT segmentation performance, Zheng et al. [304] formulated topological constraints regarding neighboring pixel classes as a loss term via mean-field approximation. Mirikharaji and Hamarneh [173] employed a loss function that penalizes segmented regions that are not star-shaped for boosting skin lesion segmentation. The method by Olender et al. [191] takes a unique role since the geometrical constraint is not encoded in the loss function. Instead, it is considered in preprocessing. For classifying atherosclerosis from intravascular ultrasound (IVUS) images, they first extracted the vessel wall region. Then they defined a pathological thickness threshold to categorize all parts of the vessel wall as pathological or non-pathological. Afterward, only small crops of pathological tissue were classified by a CNN.

Finally, we want to present some methods incorporating physical constraints into deep learning. Such approaches are termed "physics-informed", "physics-guided" or "theory-guided" in literature and can be used to force scientific consistency of data-driven models or to discover scientific knowledge by using data. Karpatne et al. [127] provide a comprehensive introduction to this topic. Neural networks can be used to solve and discover partial differential equations [212] via automatic differentiation [24]. Such methods can accelerate rather CPU-intensive and time-consuming simulations. Physical constraints can also be incorporated into neural networks through a loss function. Already in 1988, Zhou and Chellappa [306] improved optical flow computation with a loss function taking local rigidity and smoothness into account. Buoso et al. [33] personalized left-ventricular biophysical models with physics-informed neural networks without needing actual labels. The network output was restricted to deformations characteristic for left ventricles by only allowing a specific subset of radial bases functions. Furthermore, an energy potential as loss function penalized non-realistic predicted left-ventricular mechanics. For coarse-graining turbulent flows, Mohan et al. [178] inserted non-trainable layers into a CNN, which enforced incompressibility of the fluid and boundary conditions.

Encouraged by these works, we developed two methods incorporating domain knowledge into CNNs to improve ultrasound image segmentation. The first method incorporates shape priors in the form of independent components into the CNN. These are linearly combined with learned coefficients in a secondary CNN branch. The second method, a new loss function, is applicable to IVUS lumen and vessel wall segmentation and cardiac segmentation. It favors a particular tissue to be completely surrounded by another one. Parts of the following sections have been published in Bargsten et al. [21].

4.2.2 Independent Component Analysis as a Shape Prior

As explained in the previous section, organs usually follow a strong shape prior. In the case of IVUS, the lumen appears as round discs, whereas the vessel wall should have the shape of a circular ring (compare Subsection 5.1.4). In the case of cardiac segmentation, the endocardium is surrounded by the myocardium and atrium (compare Subsection 5.2.3).

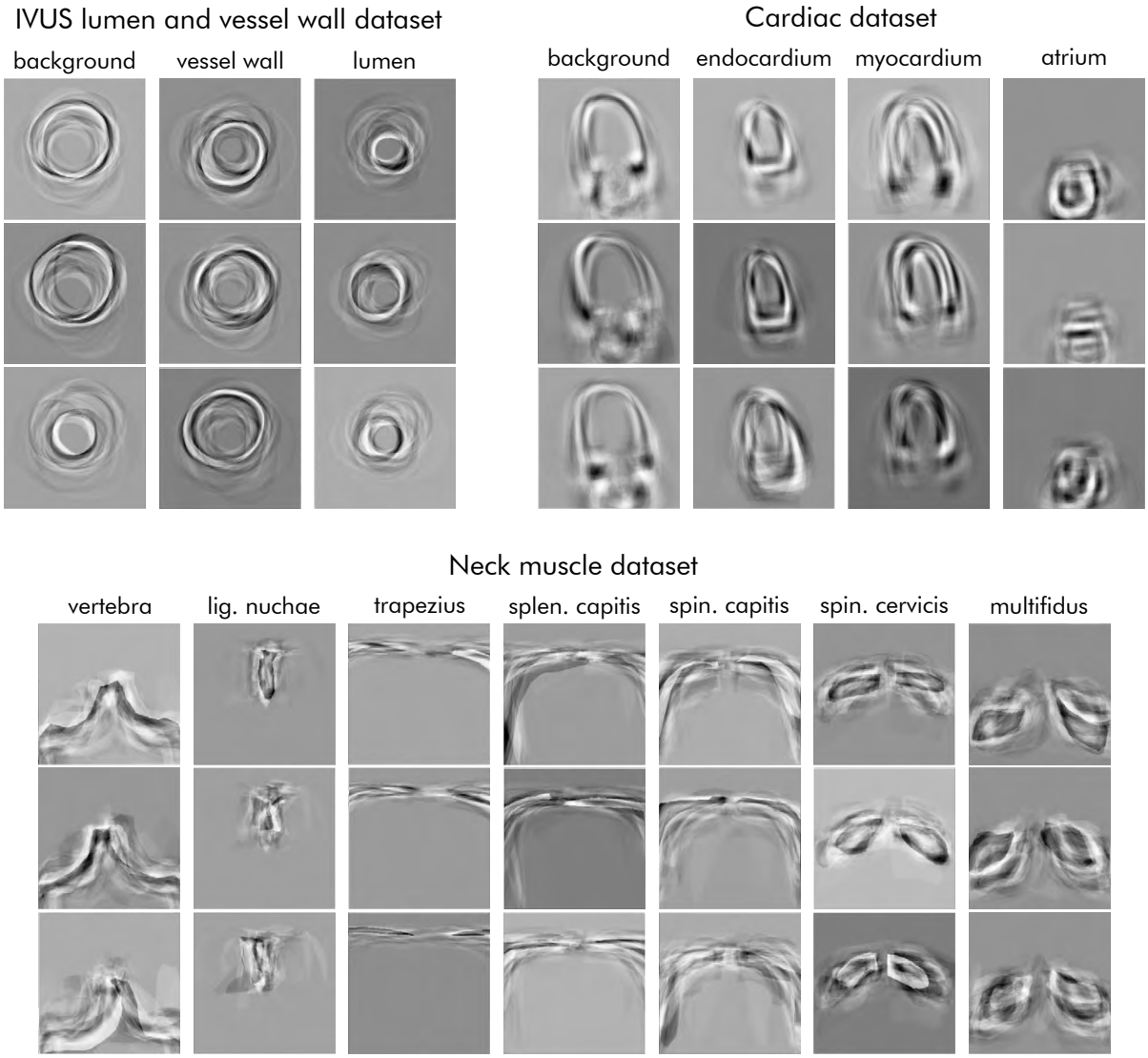

Figure 4.8: Exemplary independent components from appropriate datasets. One can identify the various tissue shapes. Different independent components from the same tissue exhibit noticeable differences.

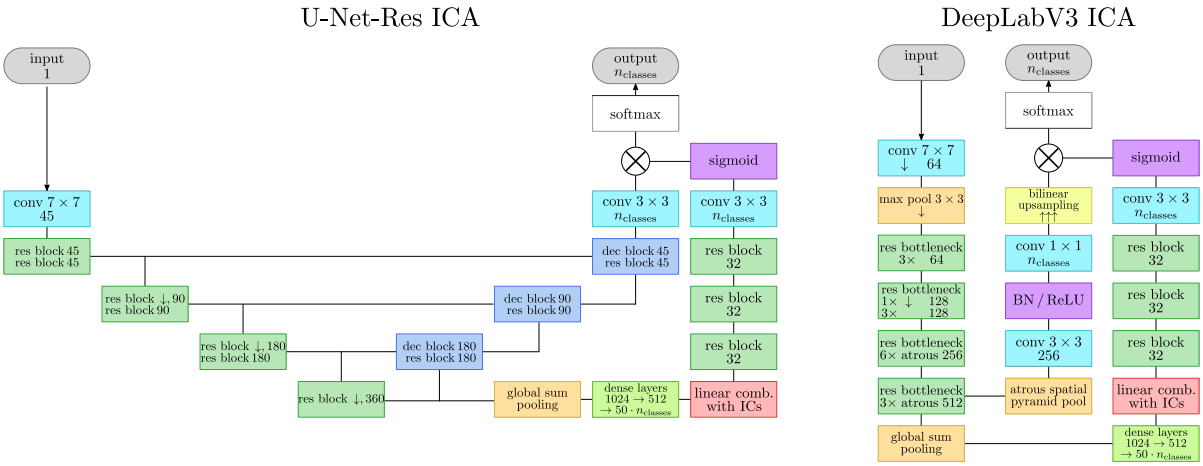

Figure 4.9: CNN architectures that incorporate ICA shape priors. The ICA block is a network branch running parallel to the main branch. The feature maps of the deepest encoding are fed into a global sum pooling layer and afterwards in some dense layers with batch normalization and ReLU, resulting in $n_{\text{components}} \cdot n_{\text{classes}}$ coefficients that are used to linearly combine the ICs. Three residual blocks with 32 output feature maps follow. Lastly, a convolution with n_{classes} output channels and a subsequent sigmoid function are applied, resulting in an attention map that is multiplied by the output of the main branch.

We developed a new method for defining and integrating shape priors into CNNs by performing independent component analysis (ICA) with all ground truth masks from the training set. Figure 4.8 depicts three exemplary independent components (ICs) for each segmentation class of the IVUS lumen and vessel wall dataset, the cardiac dataset, and the neck muscle dataset we will present in Chapter 5. The first 50 components (or less, if the number of training samples is smaller) from every segmentation class are taken to define a prediction model with 50 weights per image, which yields the best approximation of the desired target shape. The CNN learns 50 coefficients for every segmentation class from the latent code of a CNN. These are used to linearly combine the ICs. The result is fine-tuned with subsequent layers in a secondary network branch and squashed to values between 0 and 1 with a sigmoid layer. The resulting attention map is multiplied by the output of the main branch of the segmentation CNN. Figure 4.9 shows diagrams of U-Net-Res and DeepLabV3 equipped with the ICA branch. Using 50 components in preliminary experiments turned out to yield the best performance.

We hypothesize that this approach guides the CNN towards learning correct shapes since the independent components provide additional information that can be processed by the network and integrated into its predictions. The CNN can still ignore this additional information by setting all IC coefficients to 0. However, we will see in Section 6.3 that this is not the case.

4.2.3 Topological Constraints

In IVUS images, the lumen is always surrounded by a vessel wall. The same is true for the endocardium in cardiac images, which is surrounded by the myocardium and atrium. The predicted segmentation masks sometimes violate this condition. We designed the containment loss function to penalize the network if this condition is not met. In the case of IVUS lumen and vessel wall segmentation, this approach only works if the vessel wall is segmented as a disc such that it also covers the lumen area. This requires performing the lumen and vessel wall

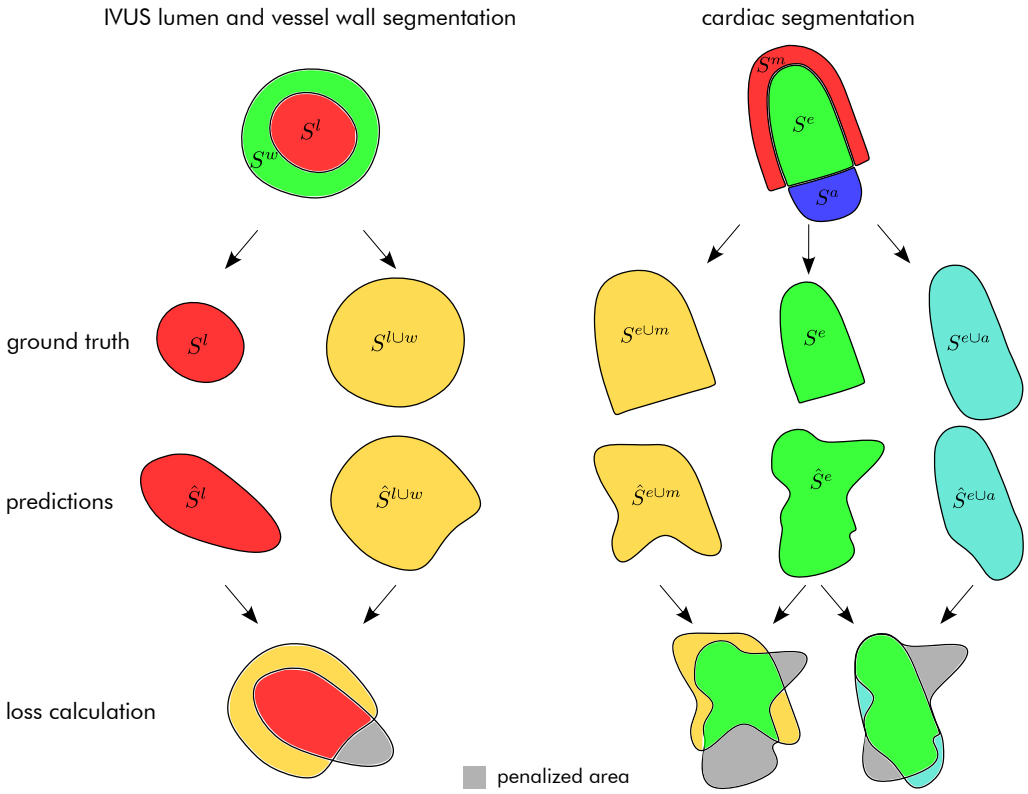

Figure 4.10: Sketch illustrating the containment loss. The containment loss is defined for IVUS lumen and vessel wall segmentation (left) as well as for cardiac segmentation (right). If the inner tissue protrudes through the outer tissues, the protruding area is penalized (gray).

segmentation via 2 binary segmentations. In the case of cardiac segmentation, the approach is quite similar. We perform 3 binary segmentations. One for the endocardium, another for the union of the endocardium and myocardium, and the last for the union of the endocardium and atrium. Figure 4.10 clarifies the approach.

In order to push the CNN outputs after softmax even further to values near 0 or 1, the smooth Heaviside function $\mathcal{H}$ is applied:

$$\hat{S} = \mathcal{H}(\hat{S}_{\text{out}} - 0.5) \tag{4.1}$$

Here, $\hat{S}_{\text{out}}$ depicts the CNN outputs after softmax. The smooth Heaviside function is implemented as

$$\mathcal{H}(x) = \frac{1}{2}\left(1 + \tanh\left(\frac{x}{\varepsilon}\right)\right)$$

with a scaling parameter set to $\varepsilon = 1/32$.

The resulting predictions are then used to calculate the containment loss. For IVUS lumen and vessel wall images, the containment loss is defined as:

$$\mathcal{L}_c^{\text{IVUS}} = \text{mean}\left(\mathcal{H}(-\hat{S}^{l\cup w} + \hat{S}^{l} - 0.1)\right).$$

Here, $\hat{S}^{l\cup w}$ and $\hat{S}^{l}$ are the predicted non-thresholded segmentation masks for the union of lumen and vessel wall on the one hand and lumen on the other. Figure 4.10 (left) depicts a corresponding sketch. In contrast to BenTaieb and Hamarneh [25] and Reddy et al. [216], our approach does not reward correct topology but penalizes incorrect topology.

For cardiac segmentation, the loss function reads as follows:

$$\mathcal{L}_c^{\text{cardiac}} = \text{mean}\left(\mathcal{H}(-\hat{S}^{e\cup m} + \hat{S}^{e} - 0.1)\right) + \text{mean}\left(\mathcal{H}(-\hat{S}^{e\cup a} + \hat{S}^{e} - 0.1)\right) \tag{4.2}$$

with $\hat{S}^{e\cup m}$ being the predicted binary segmentation mask for the union of endocardium and myocardium, $\hat{S}^{e}$ being the predicted segmentation mask of endocardium, and $\hat{S}^{e\cup a}$ being the predicted segmentation mask for the union of endocardium and atrium. Figure 4.10 (right) depicts a corresponding sketch.

For evaluation, we need to get a single segmentation mask with values in the range $[0, n_{\text{classes}} - 1]$ from the multiple masks we received from multi-binary segmentation. To accomplish this, we have to recombine the thresholded binary segmentation masks. In the case of IVUS lumen and vessel wall segmentation, we add the masks for vessel wall and lumen to obtain a single mask with 0 indicating background, 1 indicating vessel wall, and 2 indicating lumen. In the case of cardiac segmentation, this approach does not work. Instead, we insert the different binary segmentation masks into a new mask initialized with zeros in the following order: $\hat{S}^{e\cup a}$ with value 3, $\hat{S}^{e\cup m}$ with value 2, and $\hat{S}^{e}$ with value 1.

4.2.4 Summary

We have seen that the field of domain knowledge for deep learning is vast and diverse. The most prominent approaches are attention mechanisms, the combination of hand-crafted and learned features, text reports, shape priors, or physical constraints. Resulting improvements over ordinary CNN baselines are often marginal or even nonexistent. Therefore, we want to extend the toolbox of domain knowledge approaches for deep learning focusing on ultrasound

segmentation. We can now answer **RQ 1.2**.

RQ 1.2: How could incorporating domain knowledge help improve segmentation performance on small ultrasound datasets?
Our methods presented in this section employ shape priors via ICA and topological constraints in the form of the containment loss for IVUS lumen and vessel wall and cardiac segmentation. The ICA shape prior aims to guide the CNNs to predict segmentations that comply with the usual tissue shapes obtained from the training data. This guidance is mediated by an attention mechanism that rules out predicted shapes that are not similar to the shape prior. The containment loss regularizes network training by penalizing predictions that are not consistent with the topological condition, i.e., vessel wall surrounds lumen in IVUS segmentation, as well as myocardium and atrium surround endocardium in cardiac segmentation.

Both methods required a comparatively high effort to develop and implement, and the question remains whether this effort was worth the improvements achieved. We will answer this question in Section 6.3 and Section 6.4.

4.3 Generative Adversarial Networks for Data Augmentation

Parts of this section have been published in Bargsten and Schlaefer [22].

As shown in Section 3.4, GANs are quite versatile, especially in the context of medical imaging. Via domain transfer, images of a particular modality, e.g., computed tomography (CT), can be mapped to other modalities, e.g., ultrasound. The adversarial loss can be used to regularize CNN training and thus improve results on segmentation, detection, image reconstruction, or denoising [131]. However, GANs have been initially developed to generate synthetic images by implicitly modeling and then sampling from the data distribution [84]. In medical imaging, labeled datasets are usually relatively small since annotation is labor-intensive and requires trained experts to ensure adequate quality. Hence, generating synthetic data could help to boost small datasets. However, GANs themselves have to be trained with large datasets in order to generate realistically appearing images. With our contribution, *speckleGAN*, we try to break this vicious circle for ultrasound imaging.

4.3.1 Previous Work

So far, generative adversarial networks (GANs) have been used for various tasks in medical imaging. Most work focuses on applications involving computed tomography (CT), magnetic resonance imaging (MRI), or positron emission tomography (PET). Extensive reviews of these applications can be found in Singh and Raza [238], Sorin et al. [243], Yi et al. [290], and Osuala et al. [193]. The extent of publications dealing with GANs for ultrasound imaging is far smaller. However, some progress in this field was made in the previous few years.

An important application of GANs in ultrasound imaging is image translation. While Mishra et al. [175] use a GAN for despeckling, Goudarzi et al. [88] train GANs for transforming single focus to multi-focus images in order to improve acquisition time. Other works focus on improving the quality of images from less potent handheld ultrasound devices [308, 307] or image enhancement in general [68]. For speeding up image acquisition, Nair et al. [181] employed a GAN to transform plane wave radio frequency data to corresponding B-mode images.

Another way of using GANs for ultrasound imaging is to employ the adversarial loss for boosting other tasks like segmentation, as was done for breast lesions by Xing et al. [284], Negi et al. [183], and Han et al. [95]. A similar approach was used by Tuysuzoglu et al. [260] to detect landmarks in transrectal ultrasound images of the prostate. Pavlov et al. [200] employed the adversarial loss for reconstructing speed of sound maps from simulated plane wave ultrasound images.

Originally, GANs were developed to generate new synthetic images similar to those of the training set [84]. This has also been done for ultrasound in terms of images of the breast [75, 74], brain resections [62], muscle fibers [52], fetus phantoms [107] simulated data [202] and vessels [259]. The GAN by Hu et al. [107] was conditioned on spatial locations via the concatenation of coordinates to different features of the generator and the discriminator input. [259] and Peng et al. [202] conditioned their GANs to segmentation maps that define the position and type of different tissues. Tom and Sheet [259] started their image generation pipeline by transforming an echogenicity map into a speckle map using a pseudo-B-mode ultrasound image simulator [16]. A two-stage GAN then transforms this speckle map into the final output.

An obvious application for synthetic images is data augmentation. As explained in Chap-

ter 1, CNNs need large amounts of data to reach human-level performance in image analysis tasks. Hence, labeled fake images generated by GANs could potentially help to improve the performance of a downstream task like classification or segmentation. However, GANs themselves need much training data with sufficient variability in order to converge and generate realistically appearing samples. This raises the question of whether data augmentation with artificial GAN images is helpful at all in the medical domain. A group of methods that employ GANs for data augmentation is image-to-image translation. Architectures like pix2pix GAN [116] or cycleGAN [309] can transform images from a source into a target domain. Gadermayr et al. [77] trained a cycleGAN for transforming MR image slices of healthy human thighs into corresponding images with fatty infiltration. With this data augmentation approach, the performance in the downstream segmentation task could only be improved when not training with real images of fatty infiltrated thighs at all (large domain shift). In this case, the Dice coefficient improved from about 64 % to 87 %. The underlying dataset consisted of 41 image volumes. Another cycleGAN was employed by Sandfort et al. [225] for converting contrast CT images into non-contrast CT images. However, a large database of more than 11k cases was available for GAN training. This approach turned out to be rather efficient when augmenting an external dataset from another hospital (large domain shift) for a downstream segmentation task. Here, an improvement from 9 % to 66 % in terms of the Dice coefficient could be reported. Other authors proposed GAN methods for removing or adding lesions into CT [121] or ultrasound [298] images for reducing class imbalance. Jin et al. [121] trained a GAN on a database of 1018 image volumes to insert nodules into CT images of healthy lungs. Downstream lung segmentation was only slightly improved when artificial images with nodules near the lung borders were selected. Furthermore, it seems that traditional data augmentation methods (see Chapter 3) were not used. Zhang et al. [298] went the opposite way and trained a GAN to remove breast tumors from ultrasound images to decrease severe class imbalance (404 images vs. 8232 and 9966 images). This approach led to large performance improvements of up to 6.5% in the downstream classification task. No statements about traditional data augmentation techniques were made.

If image translation is not an option, one can also try to train a GAN with all available data and hope that the resulting synthetic images add valuable information to improve the downstream task. A heuristic explanation of the circumstances under which this approach could lead to success follows later. First, we take a look at some publications that investigate this approach in different non-ultrasound settings. Finlayson et al. [70] trained a GAN to generate synthetic radiographs of human pelvises to improve a downstream binary classification task (fractured vs. non-fractured hip). The dataset included 11.7k images with a size of 1024×1024 pixels. However, combining real and synthetic images for training a classifier did not improve the performance. Frid-Adar et al. [72] generated synthetic CT images of liver lesions with a size of 64×64 pixels using a training dataset of 120 small lesion crops. It should be noted that the small images exhibit quite simple textures. Due to the small training set size, the quality of synthetic images varies and artifacts are visible. Nonetheless, the classification accuracy increases when augmenting with synthetic images from 79 % to 86 %. Han et al. [93] employed GANs for generating synthetic MR images of brain metastases using a training dataset of 126 MR volumes. The fake images of size 256×256 partially include severe artifacts. Augmenting the training dataset of the downstream localization task increased the sensitivity but did also increase the number of false positive bounding boxes per image. The usefulness of this approach

is therefore at least questionable. In a subsequent publication, Han et al. [94] applied a similar approach to a larger dataset of 154 training volumes and reported minor improvements in the downstream classification task. For improving pulmonary nodule classification accuracy, Onishi et al. [192] trained a GAN on pre-cropped 2D nodule images of size 64×64 from 60 CT scans. They reported major improvements of up to 14.8 % for the downstream classification task. However, it is not stated that traditional data augmentation methods like geometrical or color transformations were used as a baseline.

Other publications study similar approaches for augmenting segmentation datasets. Mok and Chung [179] trained a pix2pix-like GAN with a two-path generator for transforming a segmentation mask into corresponding image slices of four different MR modalities (Flair, T1, T1c, and T2). The training set included 274 MR scans. It is not stated how many slices per scan were used. The authors reported relative Dice improvements of 3.5 % in the downstream segmentation task. However, standard traditional augmentation methods like elastic geometrical or contrast transformation were not used for the baseline model. Bowles et al. [30] employed a progressive growing GAN [128] to generate synthetic brain MR image patches with a size of 128×128. The training set consisted of 500 slices with corresponding segmentation masks. The downstream segmentation task could be marginally improved when using 50 % or 10 % of the training data. Only random rotations were used for traditional data augmentation. For generating synthetic CT image patches of lymph nodes, Tang et al. [254] used pix2pix on a data basis of 124 training volumes. The performance of the downstream segmentation task could be improved by 2.2 % in terms of the Dice coefficient. Unfortunately, no information about the underlying 2D image patch size is given. Traditional data augmentation included random rotations and flips.

More recent publications do also investigate similar approaches applied to ultrasound image data. Montero et al. [180] tried to improve image plane classification (trans-thalamic vs. trans-ventricular) in ultrasound recordings of fetal heads. They trained a GAN with 4274 images of size 128×128. The downstream classification accuracy could be improved by about 1.5 % when augmenting real images with 600 % to 800 % of synthetic images. For improving breast tumor classification performance, Pang et al. [198] combined the training of GAN and classifier. The training set included 1158 ultrasound images of benign and malignant lesions with a size of 128×128 pixels. The proposed approach outperformed other GANs in terms of classification accuracy. However, no experiments with traditional data augmentation were conducted.

A few recent articles deal with GAN data augmentation for ultrasound image segmentation. Used datasets include bone surface segmentation [294, 9] and thyroid segmentation [154]. Zaman et al. [294] trained pix2pix with 3104 images of size 512×512 and reported an improvement of 4.88 % in terms of the Dice coefficient for the downstream segmentation task. However, the standard deviation across images is still larger than the improvement. Moreover, the bone surface is usually clearly visible in ultrasound images but only occupies a small curved line. Therefore, some distance measures like the Hausdorff distance would be more meaningful for evaluating this task than an overlap measure like the Dice coefficient. A similar dataset for bone surface segmentation was used by Alsinan et al. [9]. 400 images with a size of 256×256 served for GAN training, 300 for training the segmentation CNN. The authors report large improvements in segmentation metrics when adding synthetic training images, e.g., Dice coefficients of 86.42 %, and 95.80 %, respectively. However, the approach of training GAN and segmentation CNN with different subsets of the data is problematic since the baseline segmentation CNN was

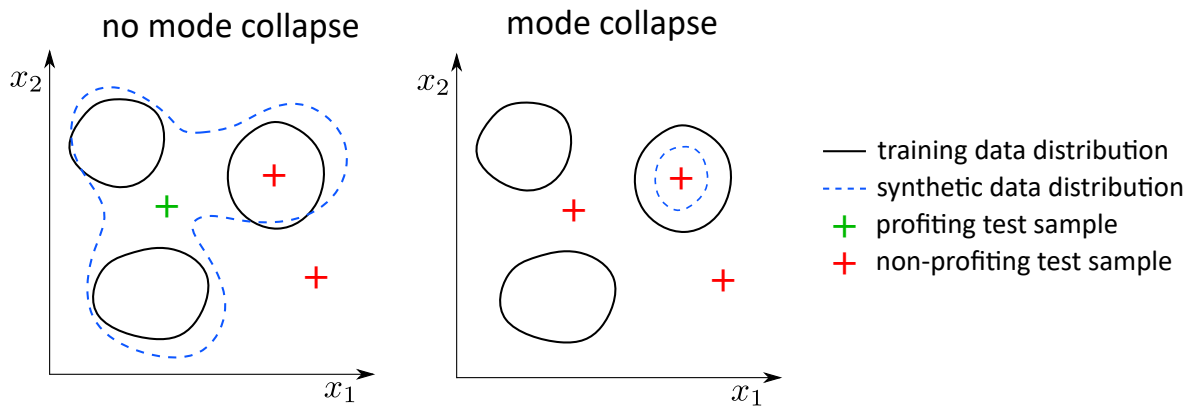

Figure 4.11: Sketch illustrating cases in which GAN data augmentation could improve the results of the downstream task. Both diagrams depict an extremely oversimplified two-dimensional data space. For images with a size of 256×256, the space would have $2^{16} = 65536$ dimensions. The training data distribution has three modes (solid black). In case of no occurring mode collapse (left diagram), the synthetic data distribution spans the three modes of the training data distribution. Test samples that lie outside the training and synthetic data distribution (red cross, bottom) do not benefit from GAN data augmentation, since their correct classification or segmentation requires information not covered by both distributions. Test samples that lie in the training data distribution (red cross, top) do not benefit from GAN data augmentation either since samples from the synthetic data distribution do not add necessary information. Only test samples that lie inside the synthetic data distribution and outside the training data distribution (green cross) benefit from GAN data augmentation. The right diagram shows the case of mode collapse. Here, GAN data augmentation is completely pointless.

not trained with the GAN training images. The baseline is therefore drastically disadvantaged. A realistic scenario would be to train the segmentation network with all available labeled data. This would likely lead to performances comparable with the results of the proposed method. Finally, Liang and Chen [154] proposed a combination of GAN and variational autoencoder [139] for generating synthetic ultrasound images of the thyroid. No significant improvements in the downstream segmentation task could be reported. Furthermore, no traditional data augmentation was performed for the baseline, and the synthetic images were manually annotated by non-experts. The validity of this approach is therefore highly debatable.

4.3.2 On the Usefulness of GAN Data Augmentation

Summarizing the previous paragraphs, it is hardly possible to systematically identify scenarios in medical image analysis in which data augmentation with GANs leads to substantial improvements in the downstream task. Heuristically, a condition for improvement is that generated images add valuable information to the training set with respect to the test set. Figure 4.11 illustrates this based on a strongly simplified example of a two-dimensional data space. It turns out that an improvement in the downstream task can only be expected if the synthetic data distribution covers regions not occupied by the training data distribution and if samples of the test data lie in this region. This implies that mode collapse, in any case, does not lead to improvement. Conditions and sub-conditions for proper GAN data augmentation are, therefore

1. Convergence of GAN and prevention of mode collapse.

   a) Hyperparameters have to be tuned carefully.
   b) Sufficient capacity of GAN architecture.
   c) Enough training data.
   d) Smaller images with less detail are advantageous.

2. The synthetic data distribution has to cover regions outside the training data distribution.
   a) Not too much training data. Otherwise, the discriminator gets too strong and forces the generator to solely generate images from the most salient modes of the training data distribution.

3. Some test samples must be located outside the training data distribution.
   a) The training dataset must not be "exhaustive" (usually the case).

4. Some test samples must be located inside the synthetic data distribution.
   a) Test samples must not be located too far from the training data distribution. This means no (large) domain shift.
   b) Coincidence.

The most eye-catching aspect is the trade-off between too little and too much data. Too small training datasets will lead the GAN into non-convergence or mode collapse, whereas too large training datasets result in the GAN not generating "new" data with respect to the test set. Additionally, the test data distribution must deviate from the training data distribution but, at the same time, must not deviate too much. That is another trade-off to consider. Of course, GAN architecture and hyperparameters have to be adequately adapted to the underlying problem. Lastly, we must consider that smaller images with less detail are usually easier to generate but, at the same time, easier to classify or segment in the downstream task. Now, all these statements are qualitative. The quantitative values must be determined individually for each application through rigorous experiments.

If we take a look at the publications presented above, we see that the largest improvements in the downstream tasks were reported for data augmentation via image-to-image translation, i.e., domain adaptation. This is not surprising since significant domain shifts can thus be overcome (see the last bullet point in the previous enumeration). As for the other scenarios, we can only speculate as to why improvements are seen in some cases and not in others.

In practice, one usually deals with a given amount of data. Therefore, dataset size is a constant parameter (at least in the short term). The test data or the data that the model sees during deployment is usually unknown. The application specifies the required image size. Hence, the only variables we can directly influence are the hyperparameters, as well as model architecture and capacity. Since hyperparameters have to be searched rigorously anyway to optimize the results, options for improving GAN data augmentation are restricted to the GAN itself. Developing efficient and versatile GAN architectures is a field of active research. However, another possibility would be to incorporate domain knowledge into GAN training. Comparing Figure 4.7 in Section 4.2, regularizing GAN training or restricting the model configuration space could help to find a generator that is capable of generating realistic images without the need for large training datasets. In this case, we could obtain a synthetic data distribution

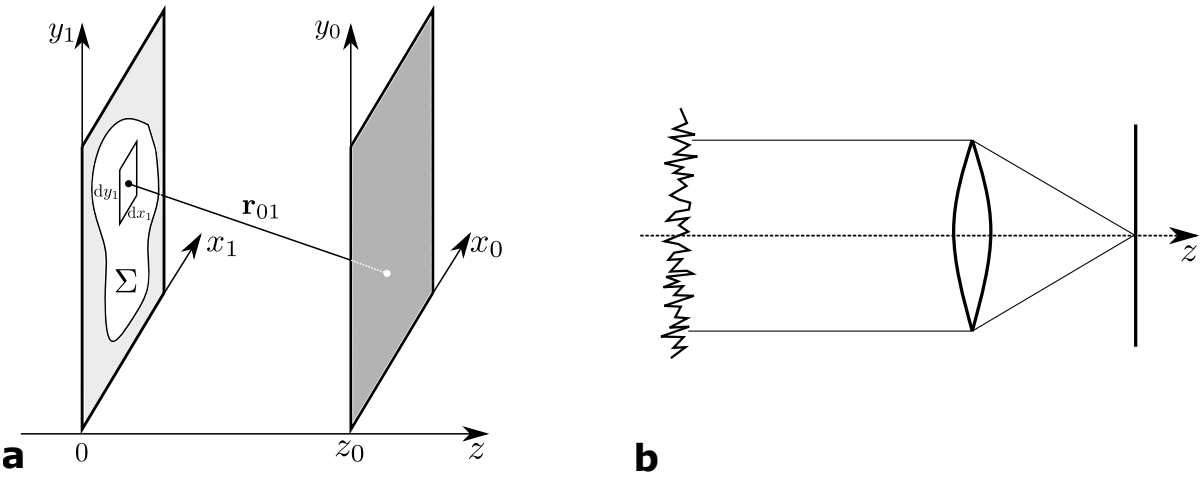

Figure 4.12: Sketches illustrating diffraction. **a**: Diffraction at an aperture. The naming of variables corresponds to Equation 4.3. **b**: Sketch of a simple imaging system with a rough object and a converging lens. Due to the roughness, the object's signal exhibits a spatial distribution of random phases, which leads to speckle patterns in the focal plane of the lens.

that does not suffer from mode collapse but does also not perfectly resemble the training data distribution.

In preliminary experiments, we observed that the generation of speckle noise is typically an issue for GANs if training sets are comparatively small. Either no crisp speckle patterns can be generated, but only blurry regions without texture or all images exhibit the same speckle pattern, indicating mode collapse (compare Subsection 6.5.5). Speckle noise and barely visible borders between tissues are usual properties of ultrasound images. It therefore seems plausible that generating such images with GANs tends to require more network capacity than images from other modalities that often appear less textually complex, such as CT or X-ray. However, larger networks with much capacity are prone to overfitting on small datasets. To overcome the mentioned issues, we developed and investigated a method that restricts the model configuration space and simultaneously regularizes GAN training: the speckle layer [22].

4.3.3 SpeckleGAN

Speckle Layer. Speckle is an interference phenomenon in imaging systems and occurs if the mean distance between scatterers is smaller than the resolution cell defined by the imaging methodology [34]. The size of the resolution cell is determined mainly by the wavelength of the carrier (or excitation) signal. Another condition for developing speckle noise is the presence of independent random phases of the scattered waves at the point of observation, usually generated by surface roughness (optics) or inhomogeneous volumes like tissue (ultrasound). Interference of these signals leads to characteristic speckle patterns.

The algorithm for the speckle layer resembles the one found in the appendix of Goodman [86] and is based on the principles of Fourier optics explained in Goodman [85]. In Fourier optics, one takes advantage of the fact that, under certain simplifications, the propagation and diffraction of wave signals can be expressed as Fourier transformations. Although the process of speckle formation differs in ultrasound systems, the effect on the gray values is similar. We illustrate the approach in the context of a simple optical system.

The algorithm is based on an imaging system comprised of an illuminated rough object and a converging lens (see Figure 4.12). Two consecutive Fourier transformations can represent the propagation and focusing of the wave signal emitted by the object. This is possible if some approximations are applied to the following general form of the diffraction integral. It describes how wave signals are diffracted at apertures and is defined as

$$U(x_0, y_0) = \frac{1}{j\lambda} \iint_{\Sigma} \frac{\exp(jkr_{01})}{r_{01}} \cos(\mathbf{n}, \mathbf{r}_{01})\, U(x_1, y_1)\, dx_1\, dy_1. \tag{4.3}$$

Here, $U(x_0, y_0)$ denotes the field amplitude in the observation plane, $U(x_1, y_1)$ the field amplitude in the aperture plane, and Σ the aperture. The vector $\mathbf{n}$ represents the normal of the aperture plane, k is the wave number, $\mathbf{r}_{01}$ the vector between a point on the aperture plane and another point on the plane of observation and r_{01} its norm. See Figure 4.12 for a corresponding sketch. Further details regarding the derivation of the formula and its application to the imaging system of Figure 4.12 can be found in Goodman [85].

The speckle layer imitates the optical system of Figure 4.12 and can be described by the following equation:

$$I_{sp}(x, y) = \left| \mathcal{F}^{-1} \left\{ \mathcal{F} \left\{ I(x, y) \cdot e^{j\,\varphi(x,y)} \right\} \cdot \mathrm{rect}_d(f_x, f_y) \right\} \right|, \tag{4.4}$$

where $I(x, y)$ and $I_{sp}(x, y)$ denote the source and speckled image respectively. $\mathcal{F}$ represents the Fourier transformation and $\mathrm{rect}_d(f_x, f_y)$ the rectangular window function with edge length d and spatial frequencies f_x and f_y. For the sake of simplicity, we did not use a circular window function indicated by the lens in Figure 4.12. On the one hand, we did not observe any difference in the visual appearance of the resulting speckle. On the other hand, the calculation of a circular mask function is computationally more expensive because the distance between every pixel to the image center has to be calculated in every training step. Equation 4.4 can be interpreted as a low-pass filter of the source image, which is multiplied pixel-wise with random phases and is thus equivalent to

$$I_{sp}(x, y) = \left| I(x, y) \cdot e^{j\,\varphi(x,y)} \ast \mathcal{F}^{-1} \{ \mathrm{rect}_d(f_x, f_y) \} \right| \tag{4.5}$$

$$I_{sp}(x, y) = \left| I(x, y) \cdot e^{j\,\varphi(x,y)} \ast \mathrm{sinc}_d(x, y) \right|. \tag{4.6}$$

Here, $\ast$ is the convolution operator and $\mathrm{sinc}_d(x, y)$ the sinc-function with scaling $1/d$. The edge length d of the rectangular window function defines the mean size of the resulting speckles and can be learned during the network training. Smaller windows lead to larger speckle patches. We note that the runtime complexity of a convolution operation scales with n^2 while the fast Fourier transform scales with $n \cdot \log(n)$. It is thus computationally more efficient to implement Equation 4.4. In order to generate warped speckles that occur in ultrasound systems with radial fields of view, e.g., in the case of IVUS or curved transducers for cardiac imaging (see Chapter 5), coordinate transformations from polar to Cartesian coordinates and vice-versa can be added to the pipeline. An exemplary speckle transformation process for IVUS images is depicted in Figure 4.13.

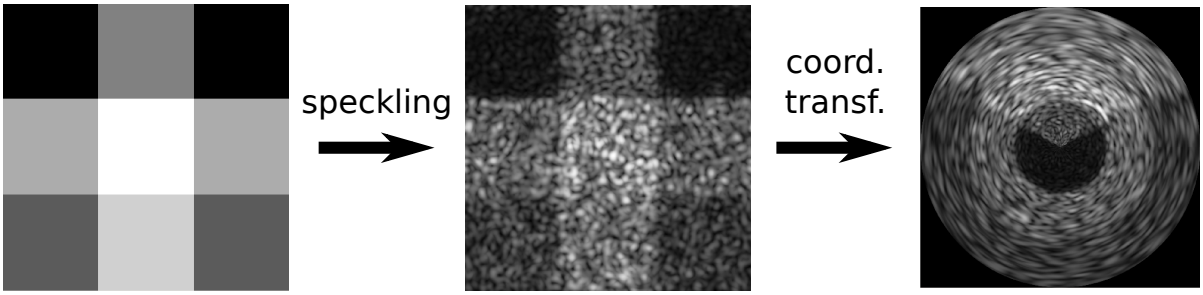

Figure 4.13: Speckle transformation of a test image. In the first step, the image is provided with speckles. An optional subsequent coordinate transformation yields warped speckles typical for IVUS images.

SpeckleGAN architecture. A segmentation mask has to be used as a conditional input to the generator and discriminator to generate ultrasound images with a defined geometry and tissue distribution. In the case of the generator, we employed SPADE, as introduced in Section 3.4.

Figure 4.14 gives an overview of the overall GAN architecture. The generator consists of multiple residual blocks [101], while the patchGAN discriminator [116] comprises ordinary convolutional layers. In the generator, SPADE [199] layers are used to condition the generated image to a given one-hot-encoded segmentation mask. The first convolutions in all SPADE layers have 64 output channels. Batch normalization precedes the affine transformation by SPADE. The discriminator employs instance normalization. Upscaling in the generator is performed by nearest-neighbor interpolation, while downscaling in the discriminator is performed by convolutions with a stride of 2. The generator is seeded with a 128-dimensional random vector sampled from a standard multivariate Gaussian distribution. Spectral normalization [176] was applied to the generator and the discriminator.

The speckle layer follows the penultimate residual layer of the generator. Here, the feature maps have already reached the output image size. Inserting the speckle layer into a deeper part of the network led to poor results. This is likely caused by the feature maps in deeper layers not yet having reached the original image size. The speckle layer adds speckle noise with 4 different speckle sizes to all input feature maps, respectively. This means that 8 input feature maps are transformed into 32 output feature maps, whereby 4 feature maps each exhibit the same morphology but with different speckle sizes. These hyperparameters were found by grid search and stayed the same for all experiments. The input feature maps of the speckle layer are also used to compute channel attention coefficients by applying global sum pooling and two linear layers. The output feature maps of the speckle layer are weighted with these coefficients to filter out unimportant combinations of input feature maps and speckle sizes. A spatial attention approach led to massive checkerboard artifacts and was therefore discarded. The resulting synthetic ultrasound images have a size of 256×256 pixels.

Other than in our paper [22], we employed a multi-scale discriminator in this work (see Figure 4.14). This means that we have 2 discriminators, one processing input in the original image size and the other with half the image size. The two resulting losses are added. The down-scaled input is computed by applying average pooling with a 3×3 kernel and a stride of 2. This approach significantly improved synthetic image quality as well as training stability compared to our paper [22], also for smaller datasets. However, this approach also improved the baseline GAN performance, which was previously unable to generate realistic synthetic

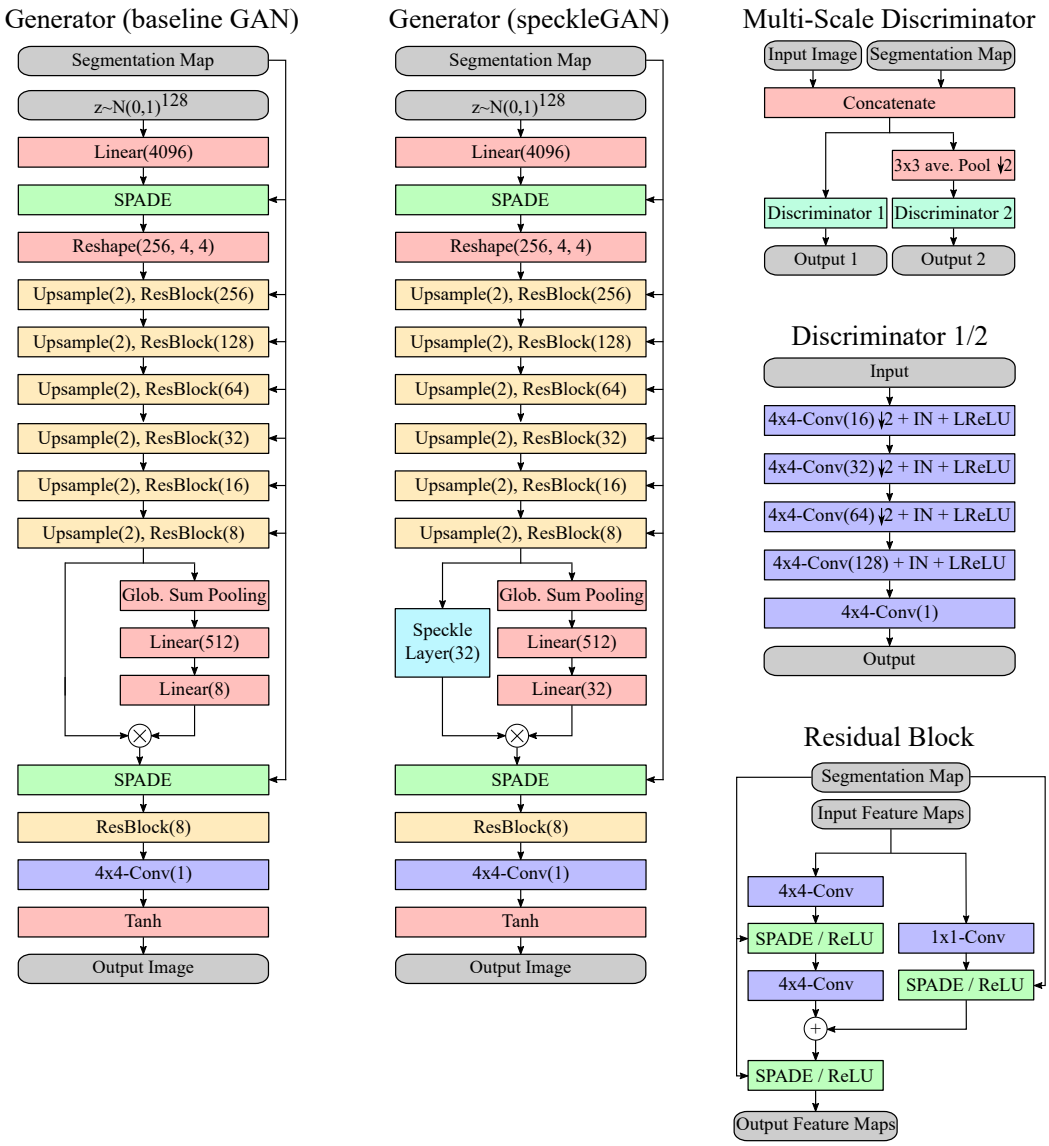

Figure 4.14: Sketch of the speckleGAN architecture. Numbers in round parentheses depict the numbers of output channels. Exceptions are *Upsample* (number depicts the scaling factor) and *Reshape* (number depicts the output's channel and spatial dimensions). (L)ReLU indicates the (leaky) rectified linear unit activation function, and IN indicates instance normalization. A downward pointing arrow in conjunction with the number 2 indicates downsampling with strided convolutions. Upsampling is performed with nearest-neighbor interpolation. The baseline GAN resembles speckleGAN but lacks the speckle layer.

images from smaller datasets. Section 6.5 will show whether speckleGAN still provides some advantages.

4.3.4 Summary

We have seen GANs used in medical imaging in many different ways, e.g., domain adaptation and style transfer, employing an auxiliary adversarial loss to solve other tasks, or synthetic image generation. The generation of synthetic images works well if training datasets are large. Small datasets, on the other hand, usually result in poor synthetic images. In both cases, data augmentation with generated images only provides little or no benefit. We can therefore answer **RQ 1.3**.

RQ 1.3: How could data augmentation with synthetic images generated by GANs be made feasible?
With speckleGAN, we try to overcome the aforementioned problem by incorporating domain knowledge in the form of the speckle layer into the generator. Since none or only little of the generator's capacity is needed to generate different speckle textures, we hypothesize that speckleGAN can generate high-quality ultrasound images without mode collapse of the speckle pattern, even when trained on comparatively small datasets. We investigate this hypothesis in Section 6.5.

5 Application Scenarios and Datasets

5.1 Intravascular Ultrasound

5.1.1 Fundamentals and Clinical Practice

Intravascular ultrasound (IVUS) is widely used in cardiac catheter laboratories, e.g., for examining coronary arteries from within. In contrast to coronary angiography, IVUS allows for a more in-depth assessment of stenoses and plaques, especially for complex cases around bifurcations. Moreover, IVUS guides percutaneous coronary interventions and thus helps optimize stent implantation and subsequent evaluation of stent apposition. Previous research found that IVUS and other intravascular imaging methodologies, e.g., intravascular optical coherence tomography, substantially improve the outcome of percutaneous coronary interventions [210, 124, 55, 164]. Exemplary IVUS images are presented below in Subsection 5.1.4.

Benefits of automated IVUS image analysis

During percutaneous coronary interventions, physicians need to take as much information as possible into account in order to maximize the treatment outcome. Here, choosing the correct stent and balloon size for angioplasty is crucial for a successful treatment. The lumen and vessel wall must be manually delineated in multiple IVUS frames to estimate the degree of stenosis and the correct stent size. The same holds in principle for calcifications, which also have to be considered. Their shape, volume, and location decisively influence the treatment. For example, cutting balloons can be used to cut up large calcium deposits. For estimating calcium burden, calcifications have to be manually delineated in multiple IVUS frames.

Delineating the lumen, vessel wall, and calcifications in multiple IVUS frames is rather time-consuming. The quality, and thus the reliability, of the resulting delineations, depends heavily on the annotator's experience and therefore varies among physicians. Hence, automated segmentation of the structures mentioned above can streamline the clinical workflow and standardize the quality of annotations.

5.1.2 Lumen and Vessel Wall Segmentation

Since 1994 [241], many approaches for lumen and vessel wall segmentation in IVUS images have been developed. The review paper by Katouzian et al. [129] provides an exhaustive overview of conventional methods published until 2012. The authors divide IVUS segmentation approaches into four categories: edge tracking and gradient-based, active contour-based, statistical- and probabilistic-based, and multi-scale expansion-based. Please see the paper for more details. In addition to providing a publicly available IVUS segmentation dataset (see also Subsection 5.1.4), Balocco et al. [15] presented several IVUS segmentation approaches by participants of a related IVUS segmentation workshop held at the medical image computing and computer assisted intervention conference in 2011. More recent work is also based on the

aforementioned types of conventional approaches [39, 132, 46, 280, 110, 161, 92, 275].

Deep learning methods for IVUS image segmentation have also been studied in previous years. Most of the presented CNN architectures exhibit an encoder-decoder structure based on U-Net [217]. Nandamuri et al. [182] employed an ordinary U-Net without notable modifications. Yang et al. [286] customized their U-Net by inserting Inception-like blocks [250]. Kim et al. [136] employed a U-Net with multi-scale input and output. This approach allows calculating multiple portions of the loss function, all considering different scales. Xia et al. [281] used a similar approach but added bidirectional convolutional long short-term memory (LSTM) cells [104] after each skip-connection. Other works proposed concatenating meshgrids, i.e., pixel coordinate matrices, to the input image [61] or intermediate feature maps [248] for providing the network information about the location of different image contents. Li et al. [152] employed two separate U-Nets for segmenting the lumen and vessel wall. Both predicted segmentation masks are combined and used as an auxiliary input for subsequent segmentation of calcifications. Gao et al. [80] combined IVUS segmentation with intravascular optical coherence tomography (IVOCT) segmentation to boost performance on both tasks. Also Ziemer et al. [311] made use of a U-Net but stacked several input images in the channel domain to provide the network contextual information. Moreover, they fine-tuned the predicted segmentation mask with a Gaussian process regression in order to obtain smooth contours. The only approach for IVUS segmentation to date that combines wavelet decomposition and CNNs was proposed by Sinha et al. [239]. Here, the input images are wavelet-decomposed into different sub-bands. The individual sub-bands are then fed into multiple CNNs, which predict radii, centers, and orientations of two ellipses representing the lumen border and external elastic membrane.

5.1.3 Calcium Segmentation

Many conventional methods for calcium segmentation in IVUS images have been reported. Especially thresholding was extensively used since calcifications usually appear very bright compared to other (but not all) image regions. Otsu's thresholding method [194] is often used iteratively to segment calcifications [11, 226]. Lee et al. [148] and Santos Filho et al. [226] additionally took shadows, which usually appear behind calcifications, into account. Another thresholding approach based on one-dimensional radial brightness profiles was presented in Zheng and Bing-Ru [302]. Plissiti et al. [208] and Zhang et al. [297] employed active contour models, with Plissiti et al. [208] minimizing the contour energy via a neural network. Araki et al. [10] proposed fuzzy clustering and Markov random fields to segment calcifications. However, an already segmented vessel wall is required for the algorithms to work.

Other publications focus on classifying individual pixels from a multitude of hand-crafted image features. Classification based on these features was performed with random forests [12, 134], support-vector machines [159] or ensembles [112]. Giannoglou et al. [81] employed a genetic fuzzy rule-based classification system and Lee et al. [149] used hand-crafted features to train a deep belief net [103].

Literature on calcium segmentation by means of deep neural networks is extremely sparse. Apart from our own work on this topic [20, 19], only [152] presented an approach so far. Here, they employed two separate CNNs for lumen and vessel wall segmentation. The predicted segmentation masks were then used as auxiliary inputs for a third CNN, which was trained to segment calcifications. Their paper was published at about the same time as ours.

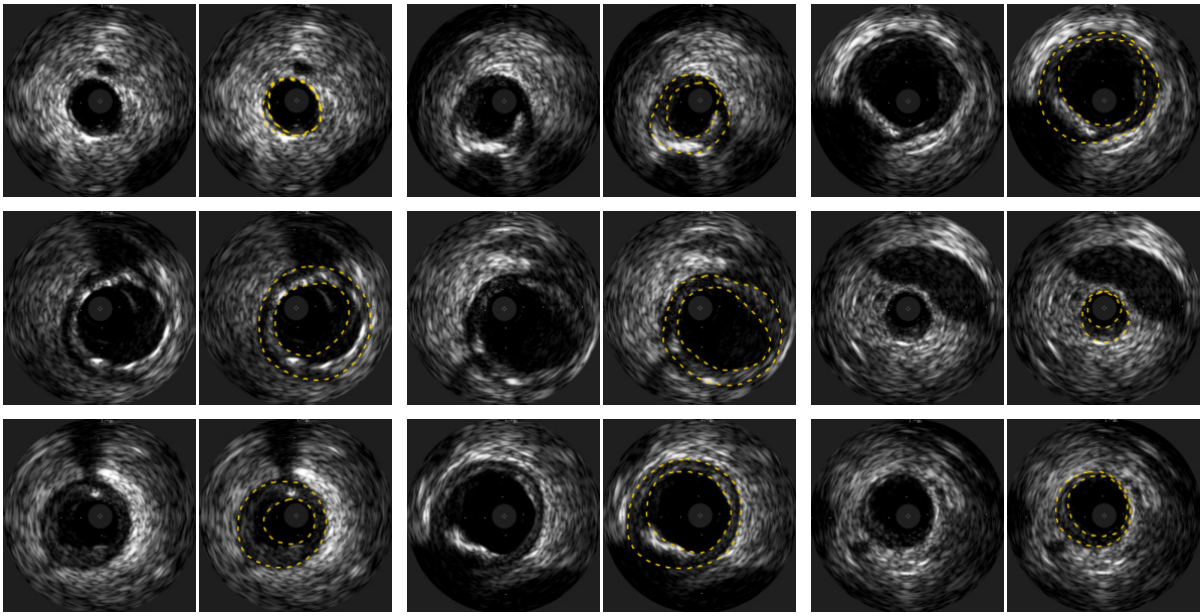

Figure 5.1: Exemplary samples from the IVUS lumen and vessel wall dataset. Ground truth annotations are depicted with yellow dashed lines. The inner circle indicates the lumen border. The outer circle indicates the external elastic membrane, i.e., the transition between the vessel wall and surrounding tissue.

5.1.4 IVUS Datasets

We used 2 different IVUS datasets that were acquired as part of the research collaboration MALEKA (German: *Maschinelle Lernverfahren für die kardiovaskuläre Bildgebung*, English: *Machine learning for cardiovascular imaging*) between Hamburg University of Technology, the University Medical Center Hamburg-Eppendorf and Philips Healthcare. The local institutional review board approved our retrospective single-center study and waived the requirement for informed consent. All images were acquired in a non-gated fashion with a 20 MHz Eagle Eye Platinum phased array probe (Philips Healthcare, San Diego, USA) at the University Medical Center Hamburg-Eppendorf and annotated by experienced cardiologists.

IVUS lumen and vessel wall dataset

The first IVUS dataset contains 400 images from 23 patients. The number of images per patient ranges from 1 to 53. Annotated are the lumen border and the external elastic membrane, i.e., the transition between the vessel wall and surrounding tissue (see Figure 5.1). The area between the lumen border and the external elastic membrane is denoted as "vessel wall" in this work. The area inside the lumen border is referred to as "lumen".

The dataset includes images showing stents, plaque, bifurcations, neighboring vessels, guidewires, and shadow artifacts. Different images of the same patient often correlate largely. This has to be considered when defining the training, validation, and test set. To prevent data leakage and to define a test set completely independent of the training and validation set, one has to ensure not inserting images from the same patient into different sets. This tends to reduce the variability in all sets and hampers the CNN from gaining large generalizability.

Figure 5.2 shows how the patient data is split into the different sets we use for CNN

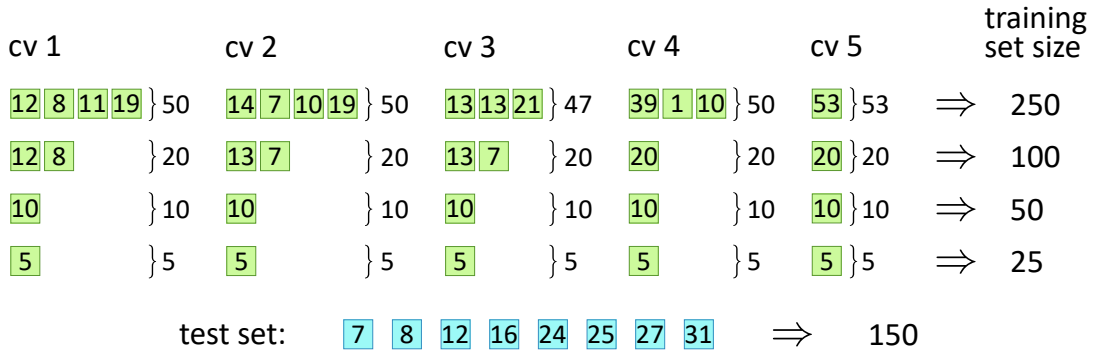

Figure 5.2: Cross-validation splits for the IVUS lumen and vessel wall dataset. Each square corresponds to a subject. The numbers inside the squares indicate the amounts of images that are used per subject.

training with 5-fold cross-validation. Figure 5.2 also depicts the training datasets for smaller amounts of training data since we will systematically reduce the training dataset size for our experiments. Each square corresponds to a subject. The amount of images per subject is indicated with the numbers inside the squares. Each column represents an individual subject. This means that the number of training images in a cross-validation split can be reduced by removing subjects completely and by removing images from an individual subject. We can see that the numbers of different patients in the individual cross-validation splits are quite small, especially when reducing the training set size. The resulting lack of variability in the sets will play a significant role when discussing the results in Chapter 6 and Chapter 7. The test set comprises 150 images from 8 subjects.

A typical pitfall in this dataset regarding automated segmentation is dark areas (usually neighboring vessels or shadow artifacts, see Figure 5.1) that are easily incorrectly recognized as lumen.

We saw that none of our proposed methods (see Chapter 4) were previously applied to IVUS lumen and vessel wall segmentation. While SEST has the potential to draw valuable texture information from all tissue classes, the ICA shape prior could stabilize the shape of the predicted lumen since the lumen varies less across images than the vessel wall. The same holds for the containment loss. Synthetic images generated with speckleGAN could be helpful to increase the variability of the training dataset, thus improving the CNN's generalizability.

IVUS calcium dataset

The second IVUS dataset includes 693 images from 31 patients. The number of images per patient ranges from 1 to 75. Annotated are calcifications that occur inside the vessel wall. Therefore, we perform binary segmentation on this dataset. Calcifications can take many shapes in 2D images, from small rounded dots to long, curved snake-like shapes that nearly completely encircle the lumen. Figure 5.3 depicts some images from the dataset with corresponding annotations.

Besides calcifications, the dataset contains images that show other types of plaque, bifurcations, neighboring vessels, guidewires, and shadow artifacts. Like the other IVUS dataset, this dataset lacks variability due to the limited number of different patients. The variability is even smaller than the other IVUS dataset since the correlation between images of the same

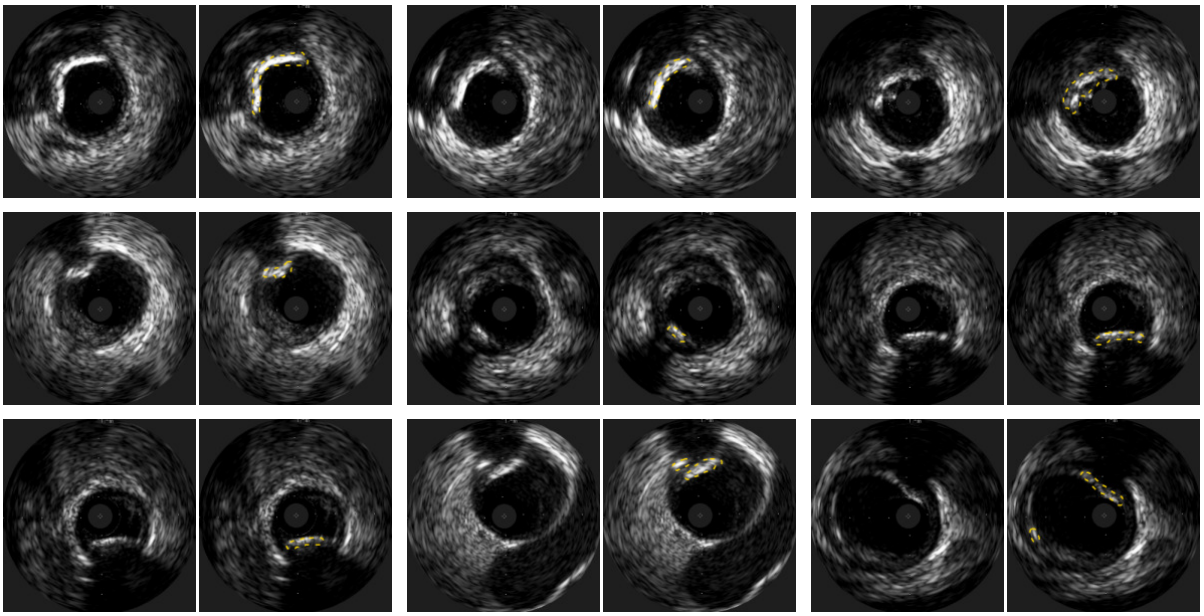

Figure 5.3: Exemplary samples from the IVUS calcium dataset. Since this is a binary segmentation dataset, only calcifications are delineated by yellow dashed lines indicating the ground truth.

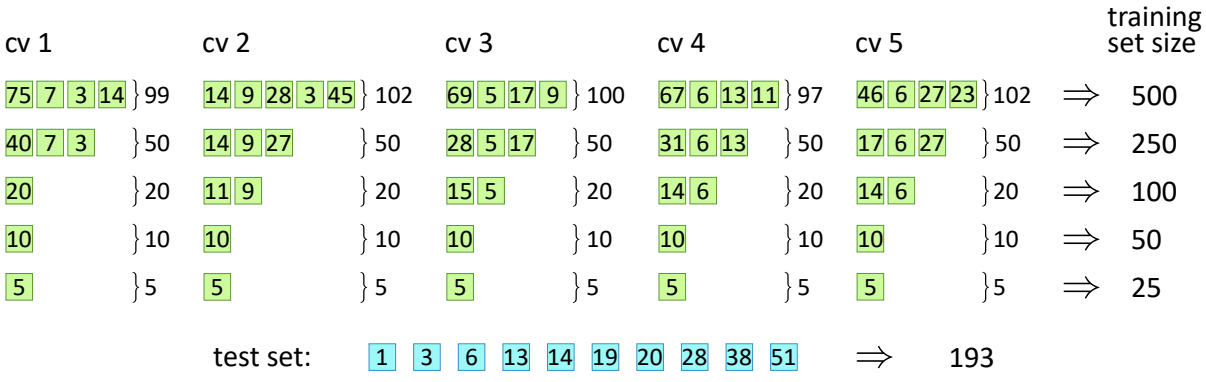

Figure 5.4: Cross-validation splits for the IVUS calcium dataset. Each square corresponds to a subject. The numbers inside the squares indicate the amounts of images that are used per subject.

patient tends to be larger. Most patients only have a single calcification, and different images of the same calcification appear quite similar. Again, this observation will be important in Chapter 6 and Chapter 7.

Figure 5.4 depicts the cross-validation splits for each training set size that will be investigated. As with the IVUS lumen and vessel wall dataset, we see that the sets' variabilities decrease largely when their sizes are reduced. The test set comprises 193 images from 10 patients.

The largest pitfall in this dataset regarding automated segmentation is ambiguities between calcifications and other bright spots that often emerge in the surrounding tissue near the vessel wall leading to false positive segmentations (see Figure 5.3). Such confusion can also happen between calcium and bright artifacts, e.g., from guidewires. In the case of automated segmentation in clinical practice, the most crucial ambiguity would be between calcium and stent struts since both occur in the vessel wall. However, this dataset does not include images that contain stent struts. A method to distinguish stent struts from other similarly appearing structures like calcifications is presented in Wissel et al. [279].

We saw above that none of our proposed methods (see Chapter 4) were previously applied to IVUS calcium segmentation. Since calcifications do not follow a strong shape prior, ICA shape priors are not applicable in this case. The same holds for the containment loss. SEST could probably draw texture information from the calcifications. However, calcifications appear rather plain and homogeneous, which may hamper extracting meaningful features through SEST. As with the IVUS lumen and vessel dataset, synthetic data augmentation with speckleGAN could increase the variability in the training dataset and thus improve the CNN's generalizability.

5.2 Cardiac Ultrasound

5.2.1 Fundamentals and Clinical Practice

Cardiac ultrasound is a standard procedure carried out by cardiologists worldwide. It enables assessing the condition of the heart in terms of geometry and movement. Many metrics can be estimated through cardiac ultrasound: ventricle diameter, ventricle shape, wall thickness, aorta root shape, atrium size, heart valve movement, wall movement, and ventricle movement. Additionally, more complex metrics such as ejection fraction and global longitudinal strain can be derived. We will discuss both of these below. All these metrics provide indications of whether the heart operates correctly or whether any dysfunctions have to be treated. Exemplary cardiac ultrasound images are presented below in Subsection 5.2.3.

Benefits of Automated Cardiac Ultrasound Image Analysis

Important cardiac metrics can be derived from ultrasound sequences to assess the morphology and function of the heart. An example is the ejection fraction of the left ventricle. It is defined as the ratio between stroke volume and end-diastolic volume (maximum expansion). The stroke volume is the difference between the end-diastolic and the end-systolic volume (maximum contraction). The ejection fraction can therefore be estimated by manually delineating the left ventricle in multiple ultrasound frames. Other important metrics like global longitudinal strain [1] can be estimated by segmenting the myocardium.

As explained in the previous section on IVUS, manual delineation of cardiac structures for estimating important metrics is rather time-consuming. The quality, and thus the reliability of the resulting delineations, depend heavily on the annotator's experience and therefore varies among physicians. Hence, automated segmentation of cardiac structures can streamline the clinical workflow and standardize the quality of annotations.

5.2.2 Cardiac Segmentation

To date, many conventional methods for cardiac ultrasound segmentation have been published. The paper by Bernard et al. [26] presents the results of a conference challenge on left ventricular segmentation from 2014. Quite prominent in literature are active contour models [171, 117, 229, 17, 3, 105, 201, 109, 7]. Different image properties were proposed to define energy functionals. Ahn et al. [3] and Ali et al. [7] relied on gray value distributions that speckle follows in different models, such as Rayleigh or Nakagami distributions. Other active contour models were combined with wavelet features [117], template matching [229], or B-splines [17, 201]. Level set methods are another type of deformable models and have also been used for cardiac ultrasound segmentation [310, 267, 303]. Zhu et al. [310] proposed an approach based on the Nakagami distribution for speckle noise and an incompressibility constraint for the myocardium. Veni et al. [267] used segmentation masks provided by a trained CNN as shape priors for a level set method. Other approaches for cardiac ultrasound segmentation use statistical methods like Bayesian models combined with Gibbs sampling [98] or Markov random fields based on Rayleigh distribution [282]. While thresholding methods have frequently been used to solve the calcium segmentation task in IVUS images, they are not very common for cardiac ultrasound segmentation. Only a recent study by Kulkarni and Madathil [142] presented an iterative thresholding approach based on wavelet features. Binder et al. [27] extracted handcrafted features for small crops around every pixel of an image and classified these with a neural network.

In recent years, also CNNs have been employed to segment cardiac ultrasound images. The paper by Chen et al. [36] provides a sound review of deep learning methods for cardiac image segmentation up to 2019. In addition to MRI and CT, also for ultrasound images. Most methods presented in [36] employ deep belief nets [103] or U-Nets [217]. More recent papers also contain approaches based on U-Net-like encoder-decoder CNNs with skip-connections [118, 145, 144, 8, 137, 160]. To regularize CNN training, Jafari et al. [118] and Kim et al. [137] added a discriminator with a corresponding adversarial loss to their pipelines. Predicting regions of interest as an auxiliary task was also proposed to improve segmentation results [144, 150]. Guo et al. [90] and Liu et al. [157] incorporated attention mechanisms into their CNNs to let them focus on more relevant regions. Painchaud et al. [197] presented an approach to incorporate a shape prior, which is independent of the segmentation algorithm. Instead, a variational autoencoder generates latent encodings of ground truth segmentation maps. These are used to fine-tune erroneous segmentation maps generated by an arbitrary segmentation method. Finally, Zhao et al. [300] combined an encoder-decoder CNN with a U-Net-like network consisting of DWTs for downscaling and inverse DWTs for upscaling feature maps. Low-frequency features are additionally used for an attention mechanism.

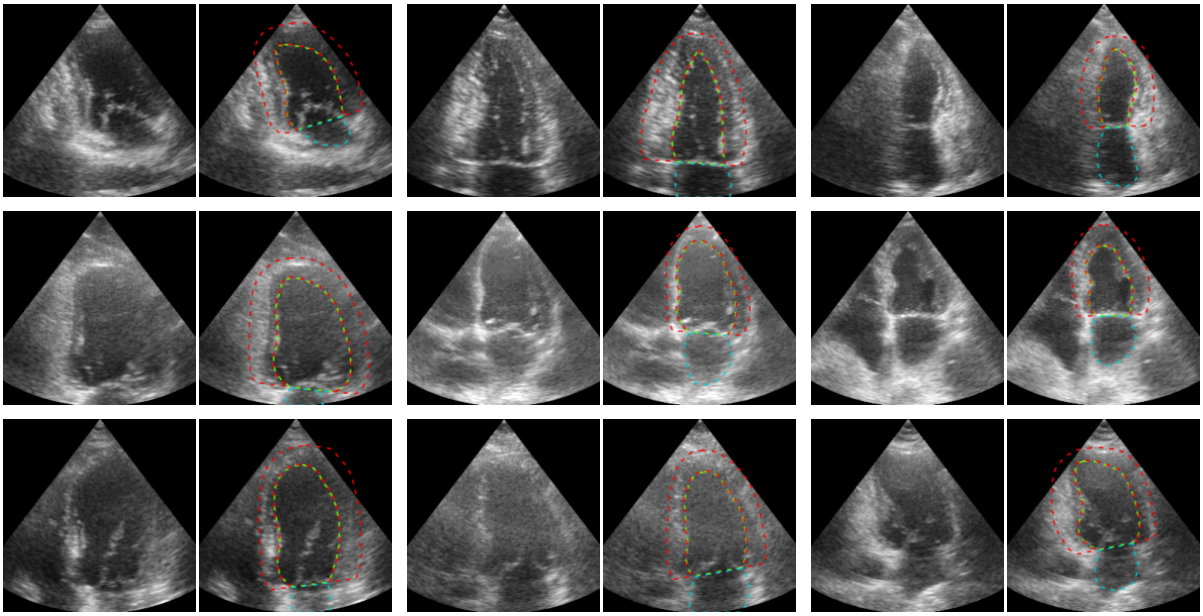

Figure 5.5: Exemplary samples from the cardiac dataset. The ground truth annotations are depicted with dashed lines. The endocardium is in green, the myocardium in red, and the atrium in blue. While some images show clearly visible boundaries, others lack quality and are much harder to interpret. The dataset was provided by Leclerc et al. [145].

5.2.3 Cardiac Dataset

The cardiac dataset that is used in this work was published by Leclerc et al. [145]. It comprises 2000 images from 500 patients, with 4 images of each patient. The 4 images include 2- and 4-chamber views of the left ventricle's maximum expansion (end-diastolic) and maximum contraction (end-systolic). All images were acquired with a GE M5S probe (GE Healthcare, Chicago, USA) and a frequency between 1.5 and 4.5 MHz. Since no particular acquisition protocol was defined, all images were recorded with arbitrary settings. Moreover, the dataset was not curated in a special way. It therefore comprises images with various appearances and levels of quality, ranging from high-quality images with very distinct boundaries to low-quality images with only barely visible boundaries and tissue classes in general. Hence, the dataset is quite diverse and does not suffer the problem of small variability in the individual sets as the IVUS datasets do. Further details on the dataset can be found in Leclerc et al. [145]. Figure 5.5 depicts some exemplary images with corresponding annotations. Annotated are the left ventricle endocardium, the corresponding epicardium, and the left atrium. In this work, we define the area between the endocardium and epicardium as "myocardium".

Figure 5.6 depicts the cross-validation splits for varying amounts of training images. Compared to the IVUS datasets, the number of different patients in the individual sets is substantially larger, indicating a larger variability and, thus, generalizability of the trained CNN. The test set comprises 500 random but fixed samples from the data of 180 subjects. We limited the number of test samples because evaluation includes calculating the average Hausdorff distance. The average Hausdorff distance is computed with a runtime complexity of n^2, n being the number of pixels per image. Although the calculation can be optimized for binary pixel masks, the time needed for the calculation is still enormous. Since the number of trainings we performed for this work is also quite enormous, we decided to reduce the number of test

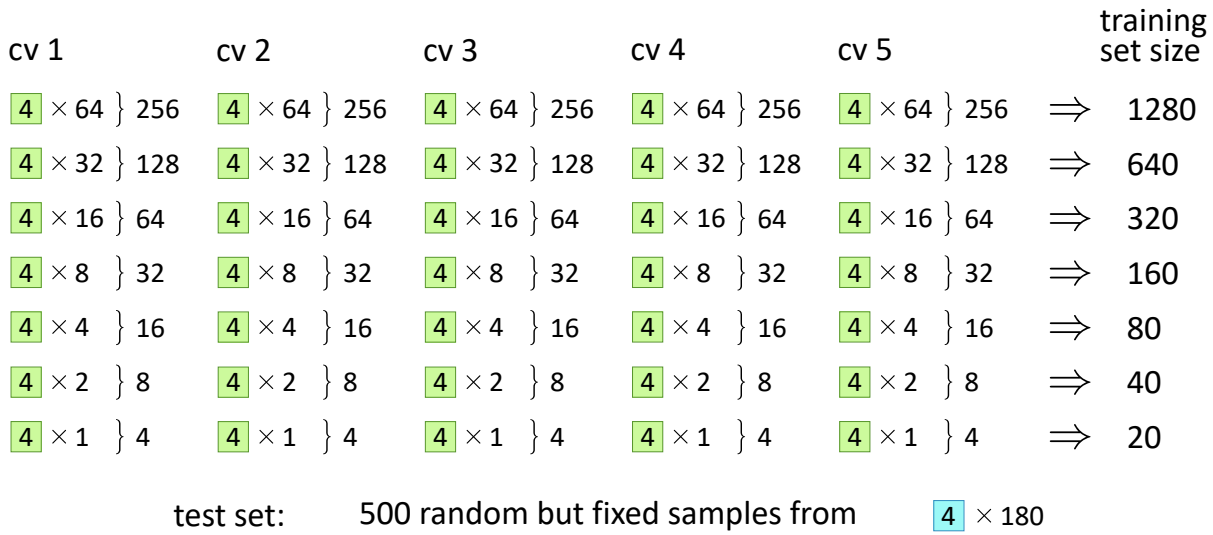

Figure 5.6: Cross-validation splits for the cardiac dataset. Each square corresponds to a subject. The numbers inside the squares indicate the amounts of images that are used per subject. Since each subject provides 4 images, all resulting numbers are based on 4.

samples to a reasonable amount. Nevertheless, the variability of the test set is still very large and thus representative because we sample individual images and not subjects.

The major pitfall in this dataset concerning automatic segmentation is that roughly 19 % of the images exhibit very poor quality (see Figure 5.5) with only barely visible boundaries and tissue classes in general [145]. Here, the CNNs likely completely fail at segmentation. We will investigate this hypothesis in Chapter 6.

We saw above that none of our proposed methods (see Chapter 4) were previously applied to cardiac segmentation. While SEST has the potential to draw valuable texture information from all tissue classes, the ICA shape prior could stabilize the shape of predicted tissues. However, the shapes of all tissue classes vary much more across images compared to the lumen in IVUS. It is therefore likely that ICA shape priors have a smaller positive impact on cardiac segmentation than on IVUS lumen and wall segmentation. The containment loss for cardiac segmentation is more complex than the one for IVUS lumen and vessel wall segmentation. The two parts of the loss function process the endocardium and myocardium on the one hand and the endocardium and atrium on the other. Hence, there is the possibility that both training signals cancel each other, leading to no improvement. Although the variability of the training dataset is much larger compared to the IVUS datasets, synthetic data augmentation could still provide a large benefit, especially for smaller dataset sizes. Due to the comparatively large variance of the training dataset, the GAN is likely to generate images of better quality and variability than the IVUS datasets. However, this effect likely vanishes for larger datasets.

5.3 Neck Muscle Ultrasound

5.3.1 Fundamentals and Clinical Practice

Neck muscle ultrasound is not very common in clinical practice. Besides the assessment of cervical dystonia, which we will discuss below, ultrasound of the neck muscles is often used to measure muscle and fascia thickness in the context of sports and movement science [206, 28,

215, 4].

Skeletal muscle ultrasound, in general, is conducted for assessing neuromuscular disorders that entail muscle atrophy (reduction of muscle tissue) or infiltration of adverse structures like fibrous or fatty tissue [204, 278]. Furthermore, skeletal muscle ultrasound can be used to investigate muscular trauma or inflammatory myopathies [205]. Nevertheless, in this work, we focus on neck muscle ultrasound in the context of cervical dystonia.

Benefits of automated neck muscle segmentation in ultrasound images Dystonic muscles are usually identified by observing the involuntary spastic movements of the patient. This procedure is prone to errors due to complex interactions between different neck muscles. Hence, ultrasound imaging was proposed to support the identification of affected muscles [48]. Automatic segmentation of neck muscles in ultrasound images can therefore help identify dystonic muscles by analyzing changes in muscle thickness.

Cervical dystonia treatment usually includes botulinum neurotoxin injections into the affected neck muscles [6]. Reaching deeper muscles like multifidus or spinalis cervicis is challenging, even with ultrasound guidance [130]. Therefore, automatic segmentation of neck muscles in ultrasound images can provide valuable information to the physician and increase the precision and efficacy of injections (compare the exemplary images in Subsection 5.3.3). Moreover, it would allow monitoring muscle thickness and thus changes in muscle tension to systematically assess treatment success [125, 48].

5.3.2 Neck Muscle Segmentation

Literature on automated segmentation of neck muscles is highly sparse. Cunningham et al. [53] proposed an approach based on principal component analysis to define a shape dictionary as well as a texture dictionary. An active shape model then estimates muscle boundaries by matching its contours to dictionary elements. Loram et al. [162] employed a U-Net [217] for predicting muscle boundaries and fine-tuned the results by median filtering and application morphological operations.

5.3.3 Neck Muscle Dataset

The neck muscle dataset used in this work was provided by Loram et al. [162]. It comprises 2127 images of 61 patients, 35 with cervical dystonia. The dataset is divided into a portion recorded with different postures (147 images) and another portion recorded during head motion (1980 images). For this work, we chose the head motion dataset comprising 1980 images from 51 patients. The amount of images per patient ranges between 11 and 54. All images were captured with a 7.5 MHz linear probe (Ultrasonix, Vancouver, Canada). Figure 5.7 shows some samples from the dataset. Annotated are 5 muscles (multifidus, spinalis cervicis, spinalis capitis, splenius capitis, and trapezius), vertebra, ligamentum nuchae, and skin yielding 8 classes. Originally, the annotations differentiate between left and right muscles, which yields 13 classes. However, for this work, we combined both sides into a single class.

Figure 5.7 reveals that the deeper muscles are often only barely or not visible. To facilitate annotation despite the poor visibility, MR images of the individual patients were used to guide the annotators [162, 54]. Multiple markers were attached to the patients' necks to register the MR images with the ultrasound images. The annotators used this additional

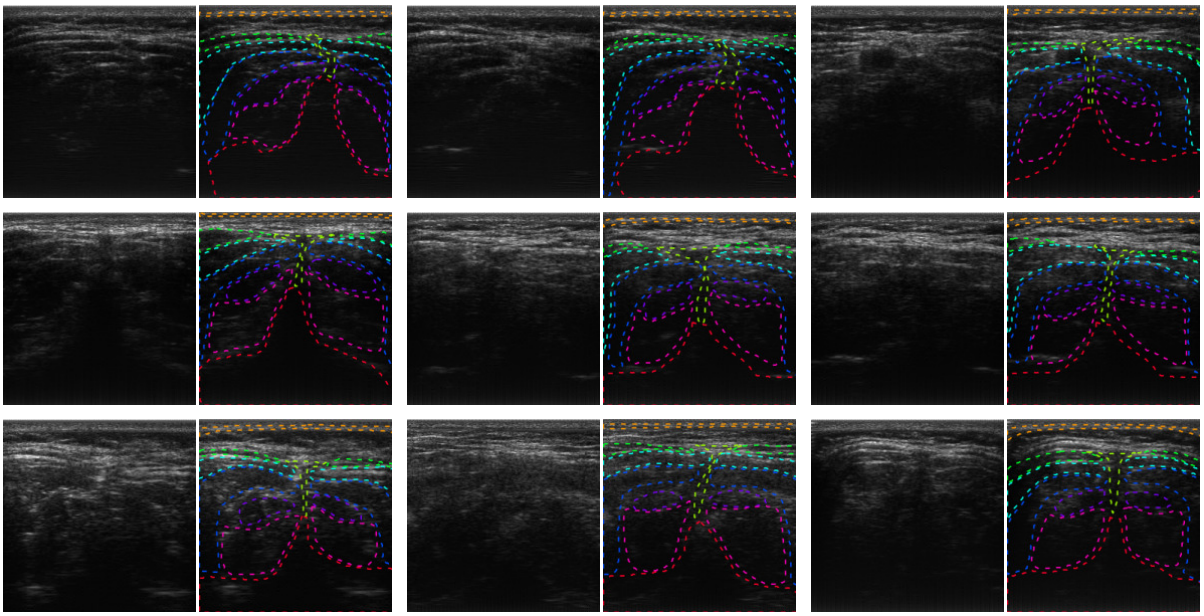

Figure 5.7: Exemplary samples from the neck muscle dataset. Ground truth annotations are depicted with dashed lines. Shown are the vertebra (red), cervical muscles from pink to green: multifidus, spinalis cervicis, spinalis capitis, splenius capitis, and trapezius, ligamentum nuchae (lime), and skin (orange). The dataset was provided by Loram et al. [162].

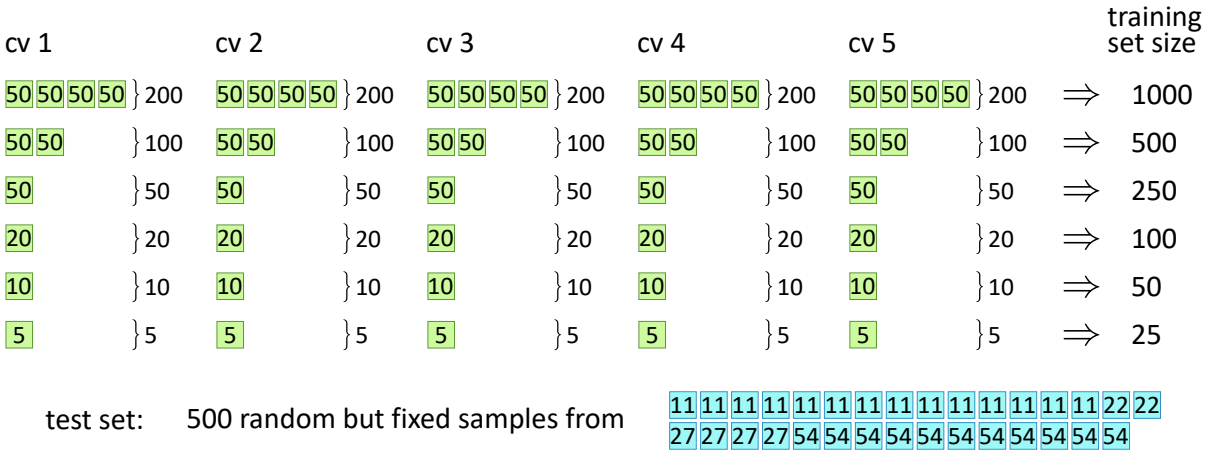

Figure 5.8: Cross-validation splits for the neck muscle dataset. Each square corresponds to a subject. The numbers inside the squares indicate the amounts of images that are used per subject.

information to adapt the delineations of barely visible tissues. However, no estimates regarding the registration error are given. It is therefore debatable whether the ground truth annotations are sufficiently accurate and consistent. Given these facts, it is questionable whether the CNNs can systematically detect different tissues, especially in the deeper parts of the images. We will give an answer in Chapter 6 and Chapter 7.

The distribution of subjects into cross-validation splits is shown in Figure 5.8. The correlation between images of a single patient is large because moving the head does not change the image content very much. Hence, similar to the IVUS datasets, the image variability in the training set is relatively small, especially when reducing the dataset size. Like the cardiac dataset, we created the test set by sampling 500 images randomly from 31 subjects. The required runtime for calculating the average Hausdorff distance is even larger for the neck muscle dataset because it has to be calculated for 8 classes.

The sparse literature on neck muscle segmentation implies that our proposed methods (see Chapter 4) were not previously applied to this problem. Apart from the containment loss, all methods are basically applicable to neck muscle segmentation. Since SEST is meant to draw texture information from the images, it is not clear whether it provides any benefit to this problem, as we saw that the many textures of the individual tissues are only barely or even not visible. Whether ICA shape priors are helpful in this case is not predictable. Maybe they do not provide additional information because the shapes of the tissues do not vary much across images of a single patient anyway. However, they could guide the CNNs in segmenting structures that are not or just barely visible. Lastly, synthetic data augmentation could be beneficial if the generated images are realistic enough and resemble the real images in terms of the barely visible deeper muscles. However, if the real images do not provide enough information for consistent segmentation, then adding synthetic images will not change this.

5.4 Summary and Contribution

As shown above, much work has already been done on IVUS and cardiac ultrasound segmentation employing deep learning methods. Neck muscle segmentation has also already been tackled with a CNN. However, none of these works tried to combine wavelet scattering and CNNs for segmentation (see Section 4.1). Sinha et al. [239] used the ordinary wavelet transformation as a preprocessing step for CNNs, and Zhao et al. [300] constructed a U-Net-like architecture by employing DWTs for downscaling and inverse DWTs for upscaling feature maps. Also, data augmentation by generating synthetic ultrasound images with a GAN has not been investigated (see Section 4.3). And finally, none of these works tried to incorporate domain knowledge according to Section 4.2 into their CNNs.

What is also missing is a systematic investigation of CNN performance when the dataset size is varied. Knowing the behavior of different approaches for analyzing ultrasound images as the amount of data decreases could provide insight into which methods should be preferred when only small datasets are available, e.g., for recognizing rare diseases, lesions, or objects in general. The following chapter provides the results of a systematic study of the proposed methods (see Chapter 4) on the presented datasets with varying sizes.

6 Evaluation of Proposed Methods

This chapter reports the results of all experiments conducted with the presented methods and datasets. But before going into the numbers, we want to explain how the experiments were conducted and how the evaluation was done.

All networks were trained with 5-fold cross-validation. The final result was obtained by majority vote of the 5 models from cross-validation. This means that the predictions of all models were added. For each pixel, the segmentation class with the highest value was selected for the final result. Since a single result would not allow us to test for statistical significance, we repeated each experiment 10 times, resulting in 10 different majority votes. Thus, variations of results due to different network parameter initializations could be averaged out. That is particularly important for smaller datasets where results are more dependent on network initialization. To test for statistically significant improvements, we employed Welch's t-test with $n = 10$. This test is appropriate since we have unpaired samples, and we have to assume unequal variance of the samples.

For assessing the segmentation performance, we employed the Dice coefficient as a measure of overlap and the average Hausdorff distance as a measure of edge alignment. Furthermore, we defined typical segmentation errors for each dataset (except the neck muscle dataset) and measured whether one of the presented methods reduced the frequencies of these errors compared to the baseline. Counting the errors was done manually by the author. To make this feasible in terms of time, we counted the errors not for all ten repetitions of each experiment but for the majority vote with (upper) median performance only. This approach can further be justified by the fact that majority voting already greatly reduces the variance of occurring errors.

To gain insight into the methods' behaviors on different dataset sizes, we varied the amounts of training images. The corresponding cross-validation splits have been defined in Chapter 5. The results in terms of segmentation metrics are given for every tissue (segmentation class) separately. An exception is the neck muscle dataset, whose results are presented as the mean of all nine classes.

We conducted all segmentation experiments with two types of CNNs: a U-Net-Res type and a DeepLabV3 type Subsection 3.3.1. Both U-Net-Res and DeepLabV3 are designed to have roughly the same amount of 6.2 M trainable parameters. Furthermore, both include three downsampling layers such that the smallest feature map size is 1/8th of the input image size. The vanilla versions of both networks were used to define baseline performances on all datasets for all dataset sizes. All examined methods were adopted to U-Net-Res and DeepLabV3 such that the number of network parameters was still the same for both network types.

All input images were resized to 256×256 pixels, and fed into the networks with a batch size of 12. For all CNNs we found that the Adam optimizer with a learning rate of $\mathrm{lr} = 2 \cdot 10^{-4}$ led to the best results. We investigated which conventional data augmentation techniques like horizontal and vertical flips, rotations, shifts, crops, and elastic transformations work best for

the individual datasets. We found the following configurations:

- IVUS lumen and vessel wall
  - horizontal flips
  - vertical flips
  - rotations
- IVUS calcium
  - horizontal flips
  - vertical flips
  - rotations
- Cardiac
  - elastic transformations
- Neck muscle
  - elastic transformations

The training evaluation procedures for GANs are explained in Section 6.5.

6.1 Baseline Results

The segmentation performances by the vanilla versions of U-Net-Res and DeepLabV3 serve as a baseline. In this section, we will present these baseline results and show which types of segmentation errors typically emerge for every dataset.

6.1.1 IVUS Lumen and Vessel Wall Segmentation

The baseline results on lumen and vessel wall segmentation are depicted in Figure 6.1. Corresponding exemplary images can be found in Figure 6.2. Ground truth annotations for the lumen and vessel wall are depicted with yellow dashed lines. The predicted lumen (red) and vessel wall (green) are marked with solid lines. As expected, the performance increases as the size of the data set increases. U-Net-Res outperforms DeepLabV3 regarding the Dice score of the vessel wall for dataset sizes up to 100 images. Vice versa for 250 training images. Dice scores for lumen are quite similar. Regarding the average Hausdorff distance, DeepLabV3 pretty much outperforms U-Net-Res for all numbers of training images. When examining the exemplary images in Figure 6.2, one can clearly see how incorrectly segmented areas decrease with increasing dataset size. One can also see that some types of errors occur repeatedly. Figure 6.3 shows examples of these typical segmentation errors. The severity increases towards the right side. The errors are

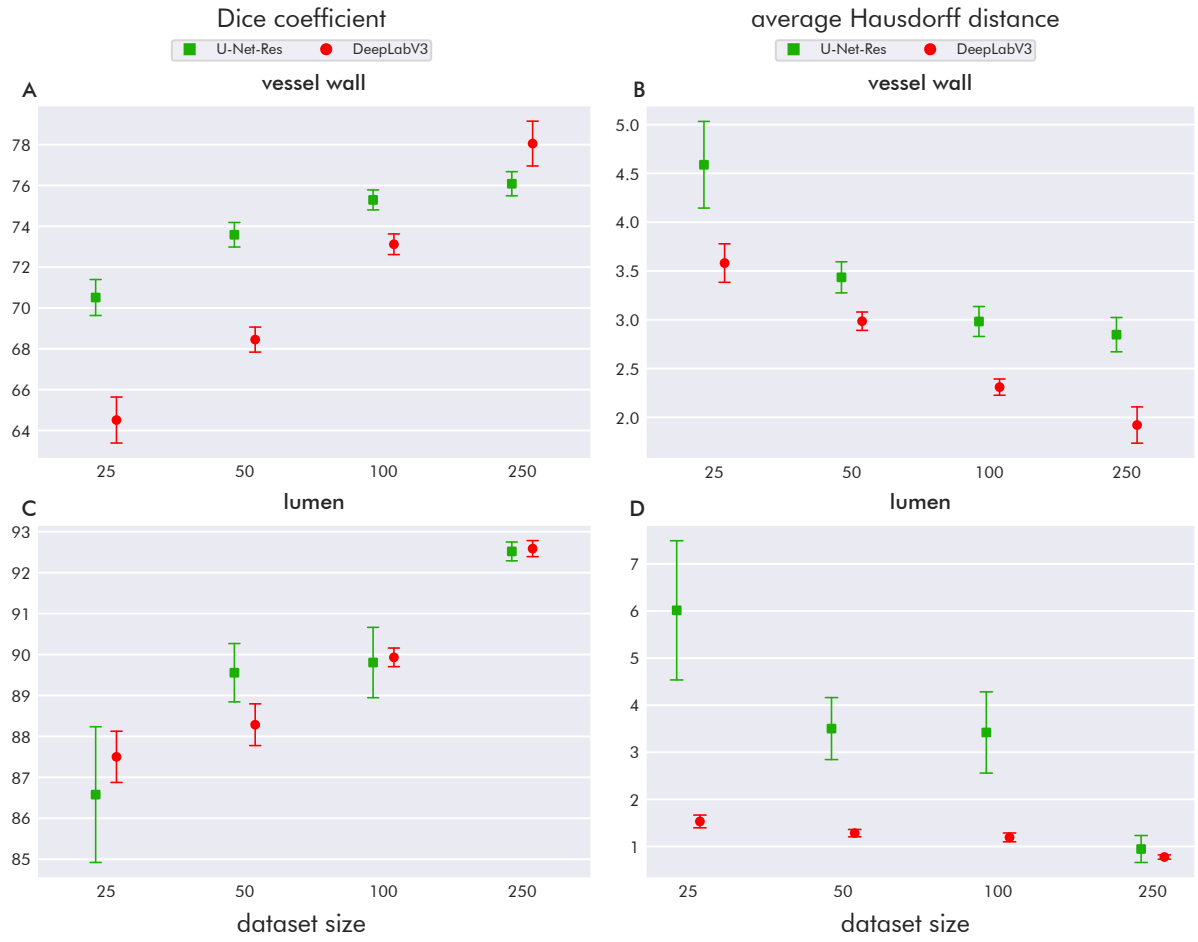

Figure 6.1: Baseline results of IVUS lumen and vessel wall segmentation. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

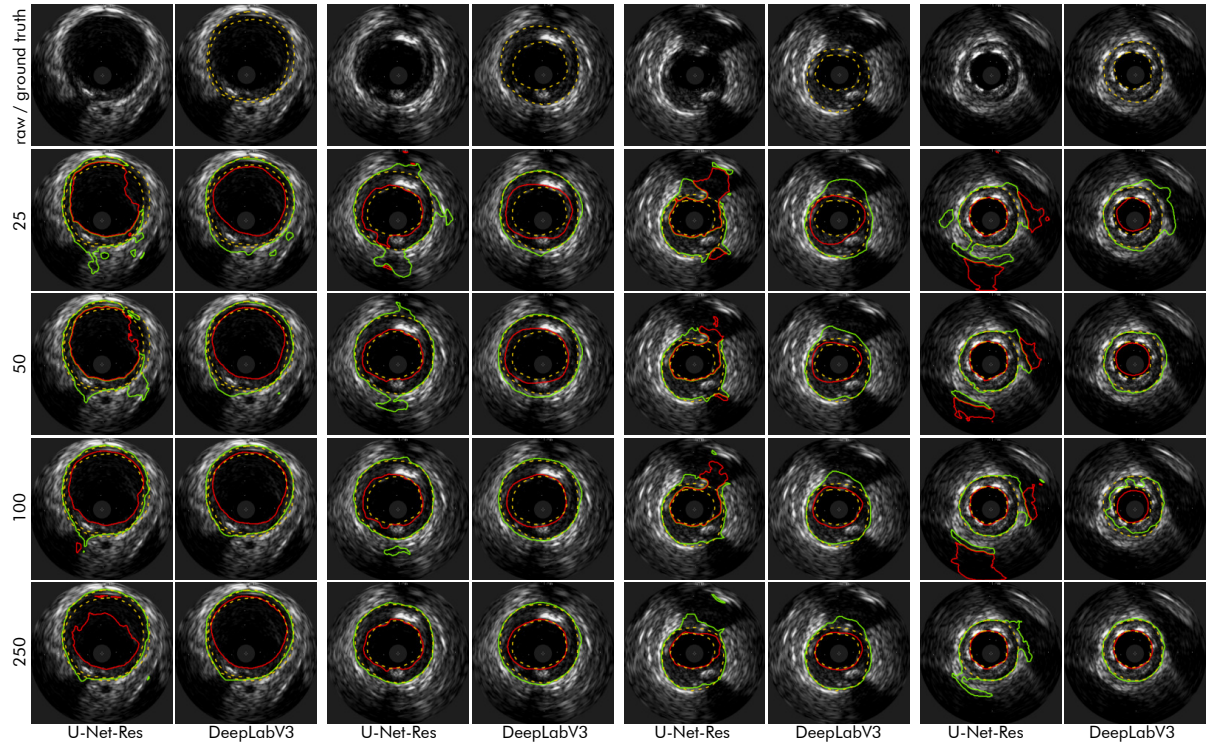

Figure 6.2: Exemplary baseline segmentations of the IVUS lumen and vessel wall dataset. The first row depicts raw images and corresponding ground truth segmentations while the other rows denote the dataset size. Columns correspond to CNN architectures.

error 1: completely meaningless topology of tissues

error 2: small incorrect patches at arbitrary spots

error 3: vessel wall is not continuous (no closed ring)

error 4: darker areas in the background are incorrectly identified as lumen or vessel wall, often leading to bulges protruding into the background

error 5: brighter areas in the vessel wall are incorrectly identified as background, often leading to notches in the vessel wall

When employing a segmentation algorithm in clinical practice, the different errors would lead to different problems. An algorithm producing error 1 is certainly not appropriate to be used in clinical practice. At least, an algorithm should recognize making error 1 and warn the user. Mild cases of error 2 (columns 1 to 3) could be reduced with appropriate post-processing, like morphological erosion, whereas severe cases would make the result useless. Mild cases of error 3 (columns 1 and 2) could be rectified by post-processing like ellipsis fitting. However, more severe cases would make the calculation of vessel wall metrics impossible. Severe cases of error 4 (columns 3 to 6) and error 5 (columns 3 to 6) would lead to incorrect values for vessel wall thickness and must be prevented for clinical practice. In all these errors, the lumen shape is not directly considered. The reason is that we did not observe any typical errors in lumen shape alone. It either fitted or not.

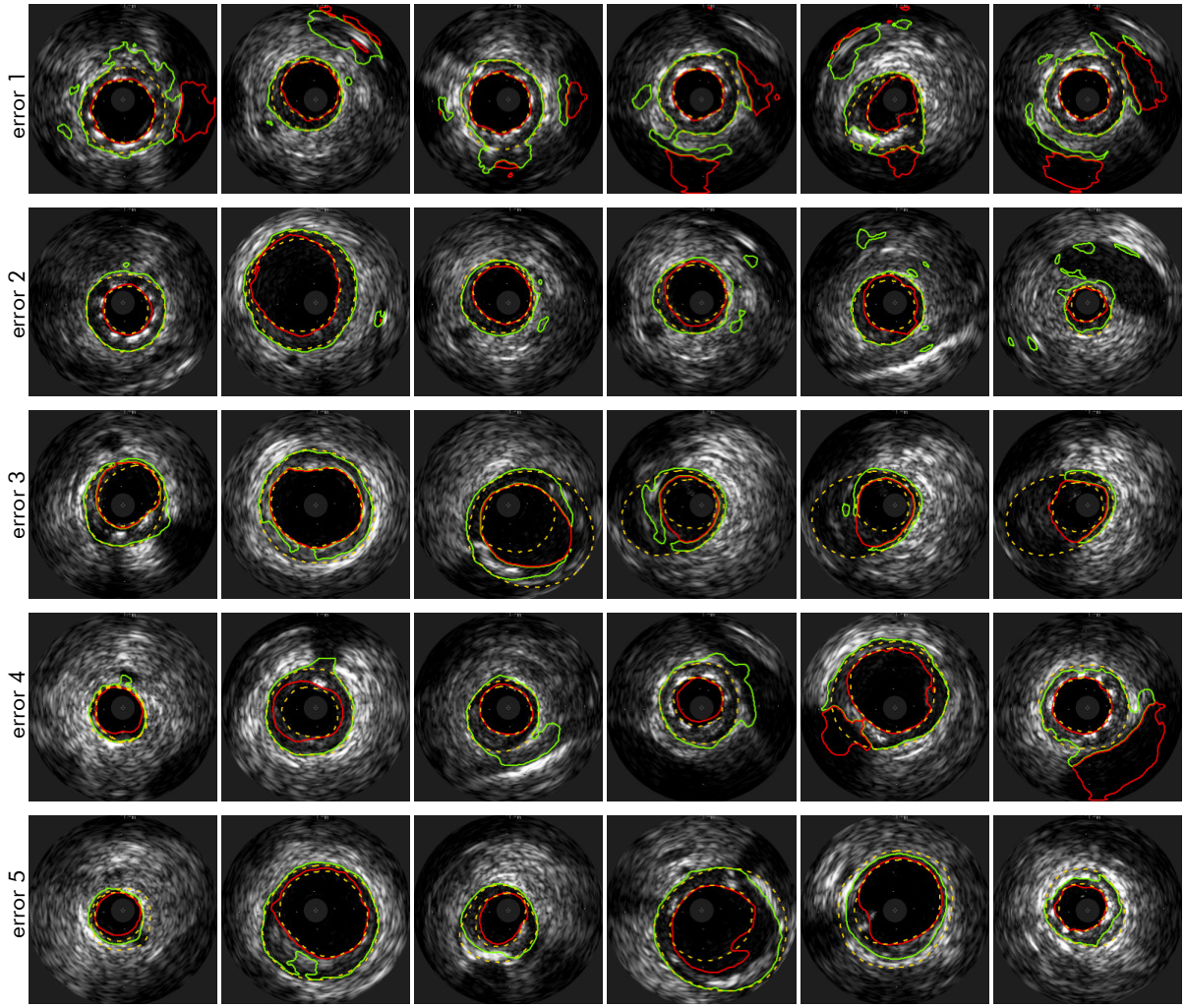

Figure 6.3: Error types that appear in IVUS lumen and vessel wall segmentation. Rows correspond to different error types, whereas columns correspond to different images. The error types are **error 1**: topological disorder, **error 2**: small incorrect patches, **error 3**: discontinuous vessel wall, **error 4**: lumen or vessel wall marked in the background, and **error 5**: background marked in the lumen or vessel wall.

Table 6.1 shows the relative frequencies (in %) of every error as a function of dataset size and CNN. The color coding is as follows: white: $v = 0$, green: $0 < v < 5$, light red: $5 \leq v < 15$, medium red: $15 \leq v < 30$, dark red: $v \geq 30$. We see that error 1 vanishes completely when using DeepLabV3 on larger datasets. On average, DeepLabV3 seems to produce fewer erroneous predictions than U-Net-Res. However, both networks produce results with many errors 2, errors 4, and errors 5, even for larger datasets. Hence, the models are not appropriate for clinical applications. It should be noted that severe and mild cases of errors 1 to 5 are counted equally for 6.1. This means that although the segmentation metrics improve, the error frequencies do not have to decrease necessarily. However, both ways of measuring segmentation performance together provide more insight than relying only on a single one.

Table 6.1: Frequencies of error types that appear in IVUS lumen and vessel wall segmentation. Values are given as percentages.

error	net	25	50	100	250
error 1	U-Net-Res	2.7	2.0	1.3	0.7
	DeepLabV3	0.7	0.7	0.0	0.0
error 2	U-Net-Res	53.3	22.7	22.7	20.7
	DeepLabV3	14.0	8.0	8.7	8.0
error 3	U-Net-Res	16.0	12.7	9.3	9.3
	DeepLabV3	2.0	4.7	4.0	0.7
error 4	U-Net-Res	69.3	54.7	47.3	43.3
	DeepLabV3	51.3	33.3	31.3	24.0
error 5	U-Net-Res	23.3	21.3	20.0	16.7
	DeepLabV3	25.3	26.7	18.7	10.0

6.1.2 IVUS Calcium Segmentation

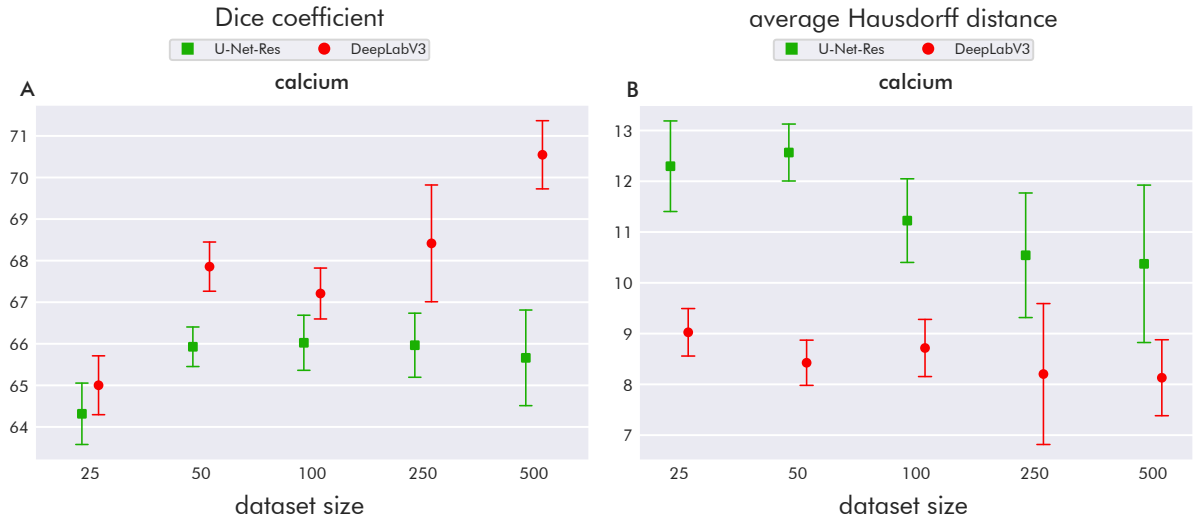

Figure 6.4: Baseline results of IVUS calcium segmentation. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

The baseline performances on the IVUS calcium dataset can be seen in Figure 6.4. Corresponding exemplary images are found in Figure 6.5. Here, ground truth segmentations are drawn with dashed yellow lines. Predictions are depicted with solid red lines. DeepLabV3 outperforms U-Net-Res in almost all cases. Whereas in the case of U-Net-Res, the Dice score does not increase for more than 50 training images, DeepLabV3 can increase the Dice score gradually with increasing dataset size. However, there is a small dip for the 100 images dataset. The Hausdorff distance achieved by DeepLabV3 does not improve much with increasing dataset size but is still smaller than the values achieved by U-Net-Res. Interestingly, the standard deviation increases with increasing amounts of training data. A trend that we do not observe in the other datasets.

Figure 6.6 depicts examples of the two typical segmentation errors we observed in our experiments. The severity increases with the column index. The errors are

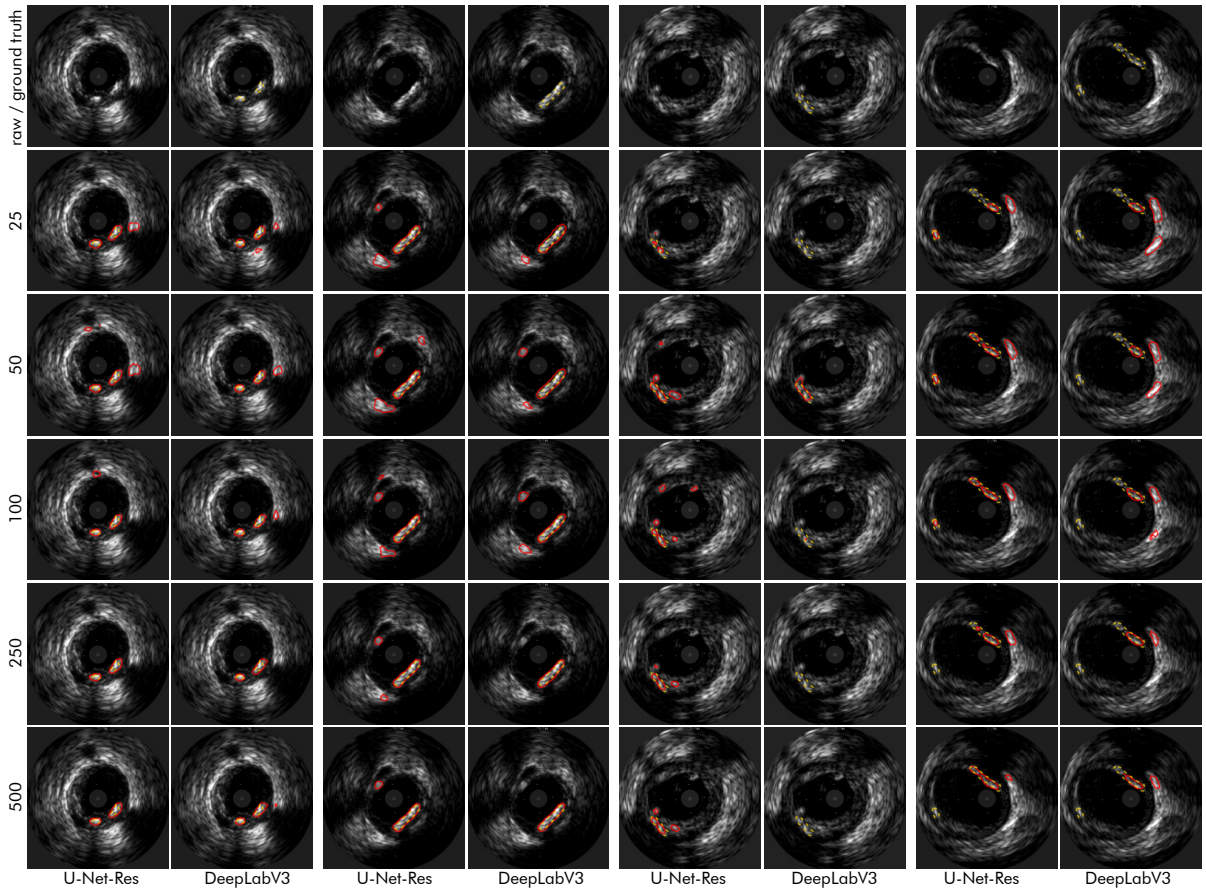

Figure 6.5: Exemplary baseline segmentations of the IVUS calcium dataset. The first row depicts raw images and corresponding ground truth segmentations while the other rows denote the dataset size. Columns correspond to CNN architectures.

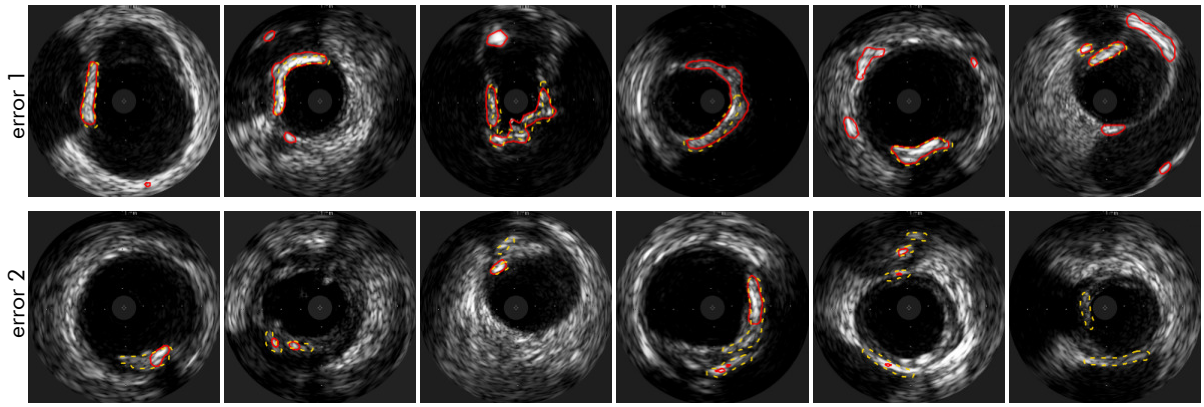

Figure 6.6: Error types that appear in IVUS calcium segmentation. Rows correspond to different error types, whereas columns correspond to different images. The error types are **error 1**: false positive patches, **error 2**: false negative patches.

error 1: bright spots (often in the background) are incorrectly identified as calcium

error 2: actual calcifications are not segmented (often because they appear less bright than usual)

If an algorithm in clinical practice would produce such errors, we would run into multiple problems. Error 1 would lead to false positive findings or overestimated calcium sizes. Post-processing by removing small patches would not make sense since calcifications can appear as those. Error 2 implicates the underestimation of lateral calcium sizes as well as axial calcium sizes. This would increase the risk of selecting a stent that is too short. So, in any case, both errors must be completely avoided in case of a practical clinical application.

The frequencies of occurring errors are given in Table 6.2. One can clearly see that the error rates are pretty high. Error 1 rate decreases with increasing dataset size, whereas error 2 is nearly constant. More training data does not seem to help the CNNs reduce false negatives. This is also reflected in the Hausdorff distance, which does not decrease substantially. This behavior can also be seen in Figure 6.5.

Table 6.2: Frequencies of error types that appear in IVUS calcium segmentation. Values are given as percentages.

error	net	25	50	100	250	500
error 1	**U-Net-Res**	62.2	58.0	49.7	42.0	37.3
	DeepLabV3	56.5	50.8	42.0	39.9	31.6
error 2	**U-Net-Res**	33.7	36.3	31.6	37.8	36.8
	DeepLabV3	37.8	35.8	32.1	36.3	35.8

6.1.3 Cardiac Segmentation

Figure 6.7 depicts the baseline result for cardiac segmentation. Corresponding exemplary images are shown in Figure 6.8. The ground truth is drawn with dashed lines and predictions

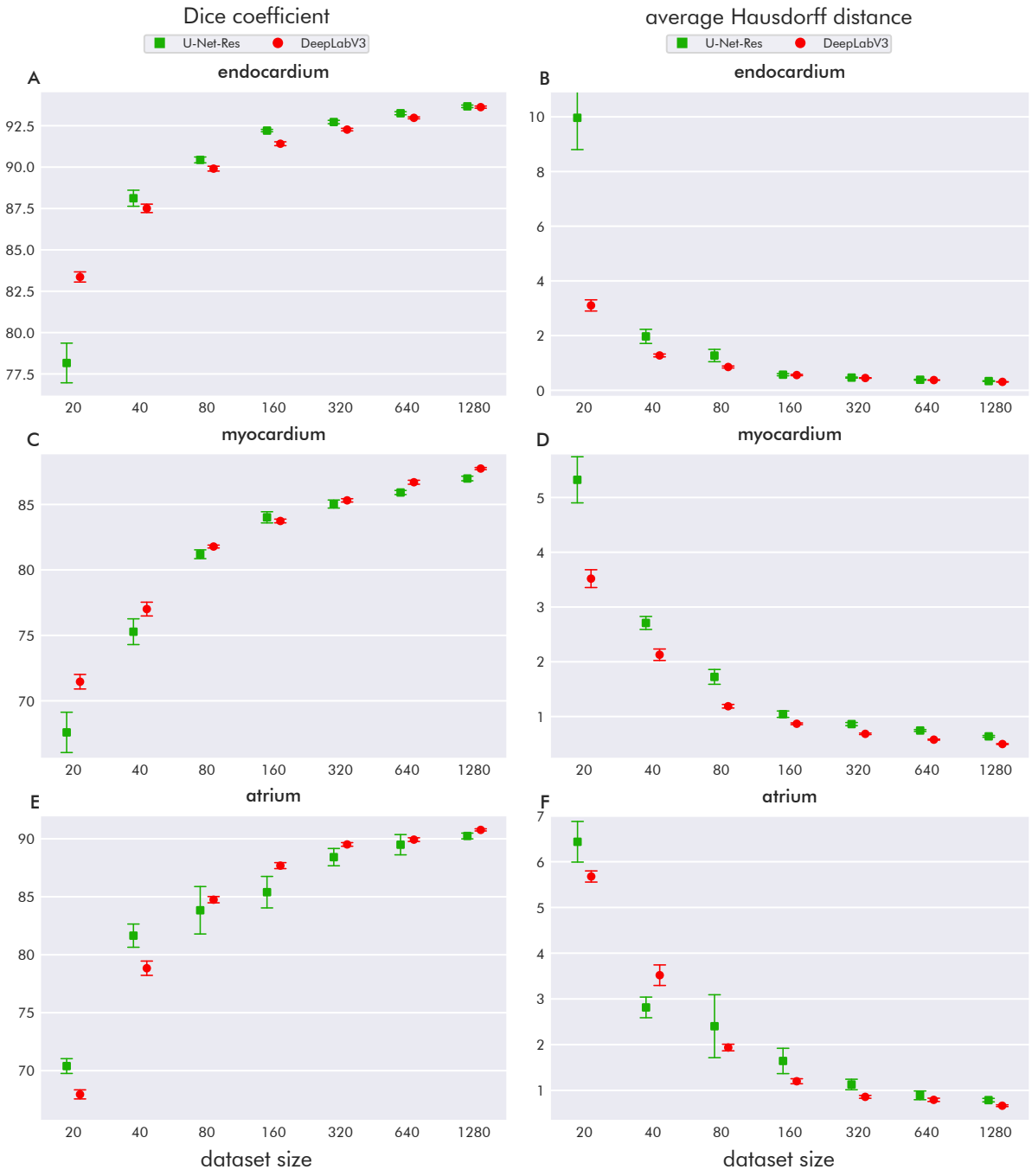

Figure 6.7: Baseline results of cardiac segmentation. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

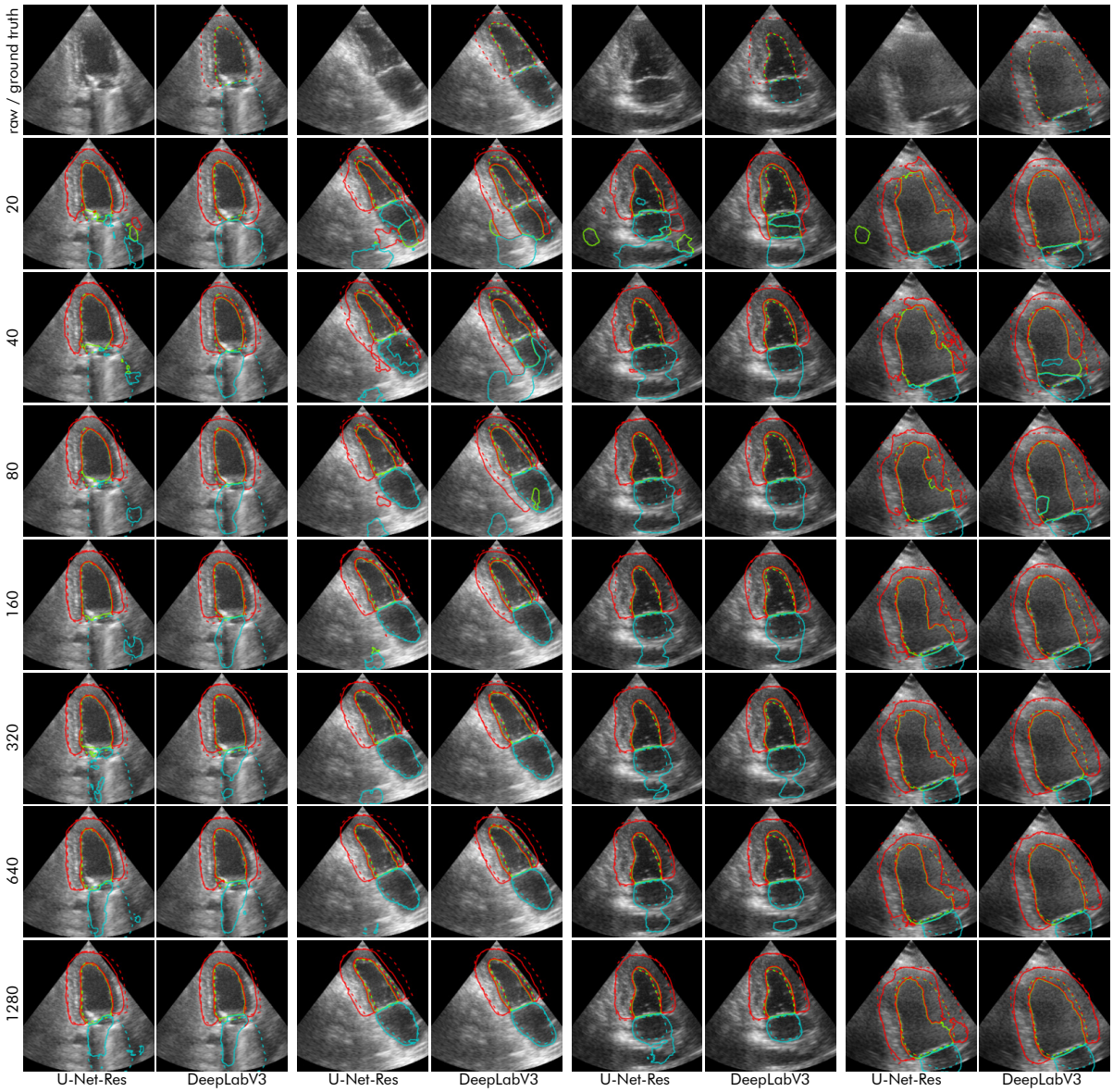

Figure 6.8: Exemplary baseline segmentations of the cardiac dataset. The first row depicts raw images and corresponding ground truth segmentations while the other rows denote the dataset size. Columns correspond to CNN architectures.

with solid lines. The endocardium is depicted as green, the myocardium red, and the atrium blue. We can see that the performance already begins to saturate for 1280 training images, especially regarding the endocardium. When comparing U-Net-Res and DeepLabV3, there does not seem to be a clear tendency about who is in the lead. Larger dataset sizes are dominated by DeepLabV3. DeepLabV3 outperforms U-Net-Res on myocardium and atrium for larger dataset sizes. U-Net-Res performs better regarding the atrium Dice score for smaller datasets and the endocardium Dice score for mid-sized datasets. Figure 6.8 reveals that images with blurry borders between tissues (image 4) lead to worse results, even for larger datasets.

As for the IVUS datasets, we observed typical segmentation errors that occurred regularly. Figure 6.9 depicts some examples. The errors are

error 1: completely meaningless topology of tissues

error 2: small incorrect patches at arbitrary spots

error 3: myocardium is not continuous or too short (does not reach atrium)

error 4: myocardium placed around atrium

error 5: endocardium and atrium partially confused

As in the case of IVUS lumen and wall segmentation, error 1 has to be avoided entirely for any clinically employed algorithm. Mild cases of error 2 (columns 1 to 3) could potentially be resolved by post-processing since the tissues to segment usually do not appear as small patches (in contrast to calcifications in IVUS images). Error 3 would lead to incorrect measures of the myocardium resulting in erroneous values for, e.g., global longitudinal strain (5.2). The same applies to error 4. Confusing endocardium and atrium (error 5) would yield rather erroneous estimations for stroke volume and thus ejection fraction (5.2).

Table 6.3 provides frequencies for individual segmentation errors. One can see how the error rates decrease with increasing amounts of training images. DeepLabV3 is utterly resistant to error 1, unlike U-Net-Res, which produces some completely failed segmentations even with 640 training images. Furthermore, DeepLabV3 seems much more resistant to producing incorrect patches (error 2) or discontinuous myocardia (error 3). All in all, DeepLabV3 generates fewer or the same number of errors as U-Net-Res. When trained with 1280 training images, DeepLabV3 yields acceptably low error rates.

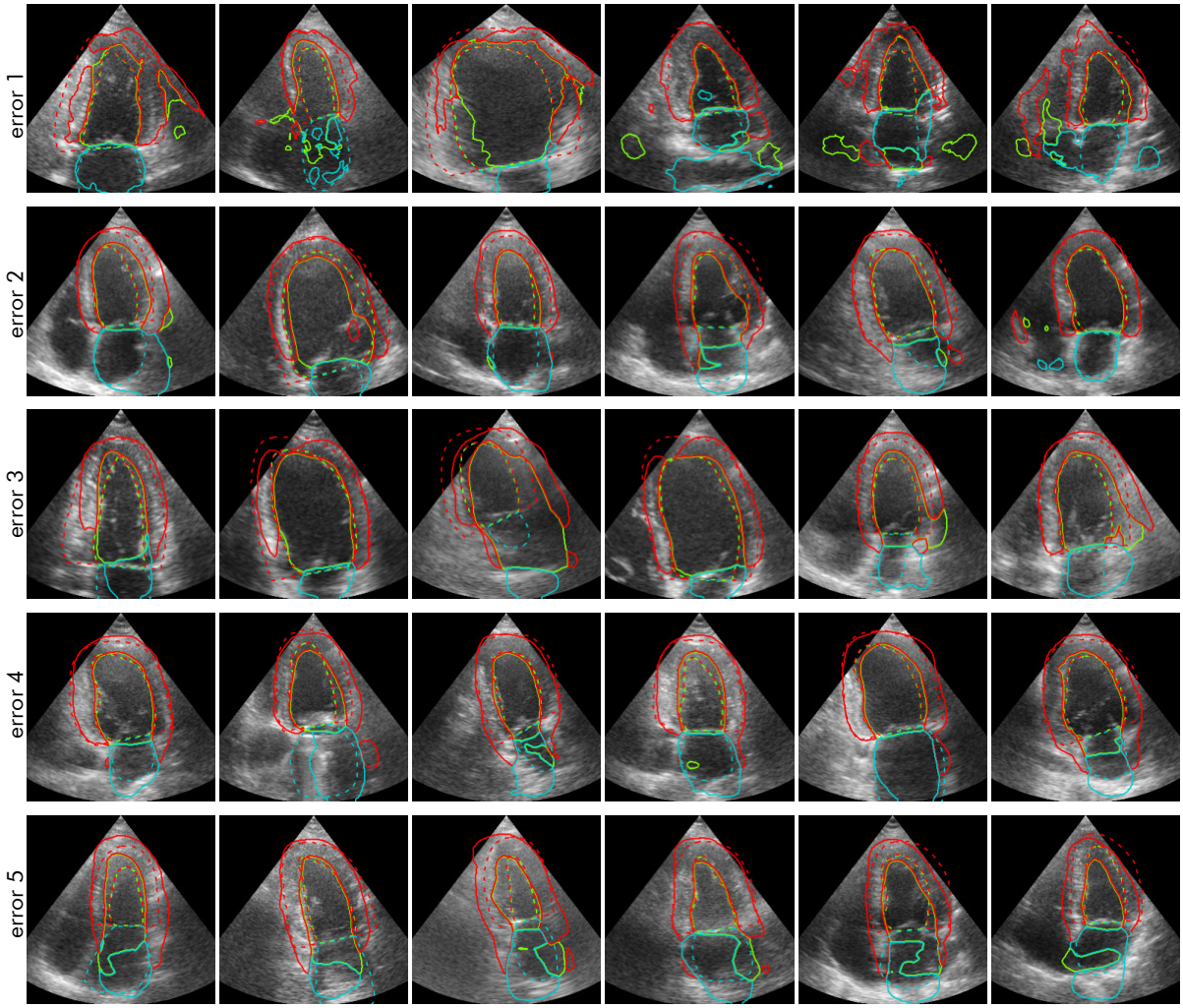

Figure 6.9: Error types that appear in cardiac segmentation. Rows correspond to different error types, whereas columns correspond to different images. The error types are **error 1**: topological disorder, **error 2**: small incorrect patches, **error 3**: discontinuous myocardium, **error 4**: myocardium placed around the atrium, and **error 5**: endocardium and atrium partially confused.

Table 6.3: Frequencies of error types that appear in cardiac segmentation. Values are given as percentages.

error	net	20	40	80	160	320	640	1280
error 1	**U-Net-Res**	26.0	12.6	7.2	3.8	1.2	0.6	0.0
	DeepLabV3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
error 2	**U-Net-Res**	67.0	50.6	39.8	36.4	20.2	14.4	11.0
	DeepLabV3	18.6	14.0	11.6	6.2	3.4	1.6	1.0
error 3	**U-Net-Res**	42.0	31.0	20.2	13.2	9.0	6.2	4.6
	DeepLabV3	18.0	14.0	13.0	6.2	5.0	1.8	2.0
error 4	**U-Net-Res**	22.2	19.2	17.4	10.4	8.0	3.0	0.8
	DeepLabV3	27.0	18.2	17.6	9.8	7.0	2.0	0.8
error 5	**U-Net-Res**	25.0	21.0	17.2	8.8	4.6	2.4	0.6
	DeepLabV3	13.0	14.0	8.2	6.6	3.6	1.2	0.2

6.1.4 Neck Muscle Segmentation

The baseline results for neck muscle segmentation are depicted in Figure 6.10. When considering only the tissue means (A and J), two salient characteristics can be seen immediately. First, the values remain more or less constant as dataset size increases, and second, DeepLabV3 largely outperforms U-Net-Res. But even for 1000 training images, the results are extremely poor. Some classes improve slightly with increasing amounts of training images, like skin (B, K) and trapezius (D, M). However, the majority of classes do not improve substantially.

Exemplary segmentations for varying dataset sizes are shown in Figure 6.11. Dashed lines indicate the ground truth, solid lines the predictions. U-Net-Res generates completely useless results, even when trained with 1000 images. Multiple tissues are either forgotten or placed in the wrong positions with incorrect shapes. The smaller the amount of training data, the more fractured the predicted segmentations. DeepLabV3, on the other hand, can only generate a mean segmentation mask with mostly correct topology but with little variation across different images. It does not recognize the borders between tissues. Nevertheless, DeepLabV3 achieves better metrics since it does not produce topological disorder. However, neither CNNs produce practically relevant results, even with 1000 training images.

Since showing and discussing all neck segmentation classes individually is rather pointless and does not provide meaningful information concerning the research questions, we will only present tissue-average results in the following sections. Furthermore, we have refrained from defining typical segmentation errors since U-Net-Res only produces topological disorder, and DeepLabV3 always produces quite flawless but rigid topologies.

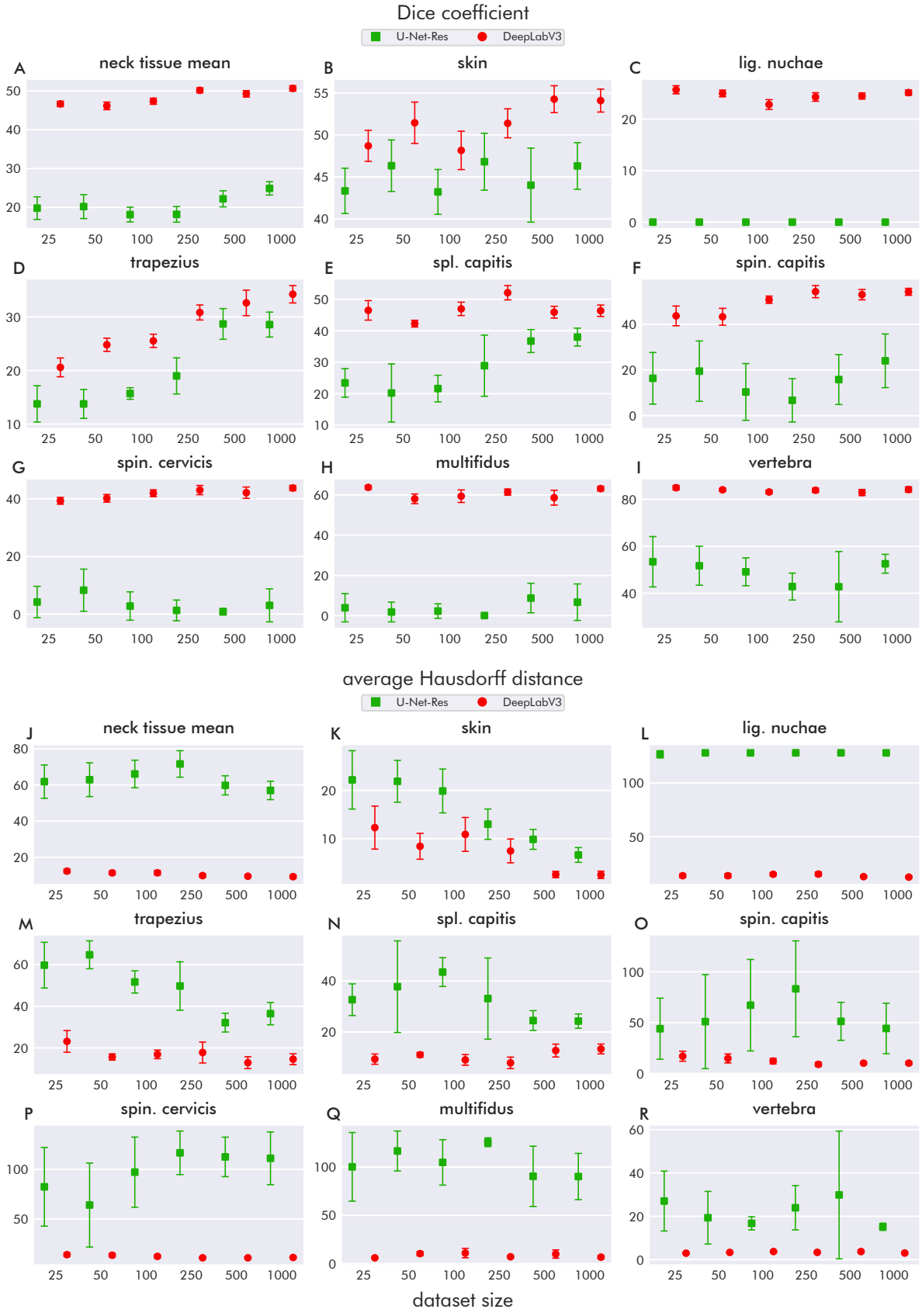

Figure 6.10: Baseline results of neck muscle segmentation. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

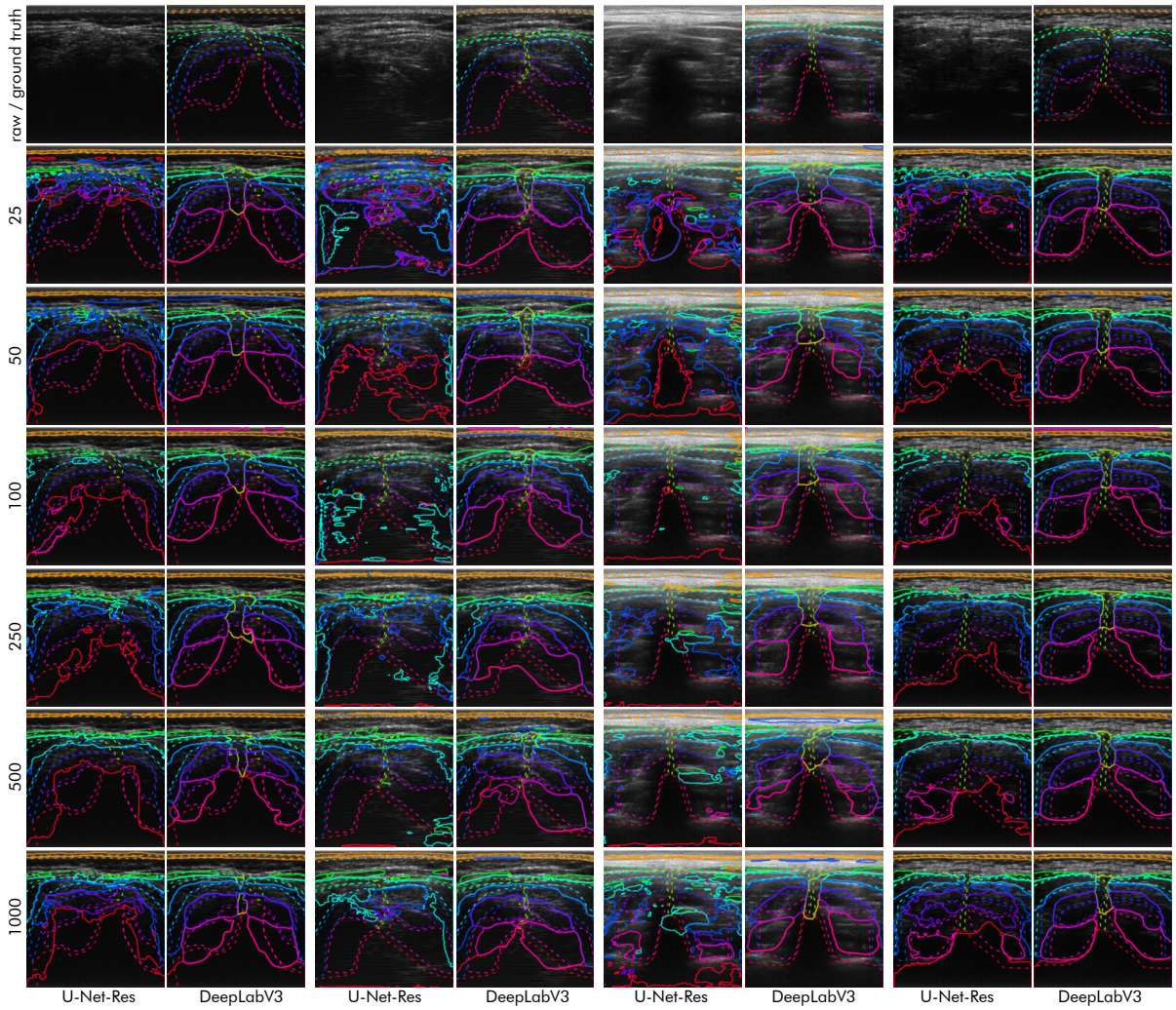

Figure 6.11: Exemplary baseline segmentations of the neck muscle dataset. The first row depicts raw images and corresponding ground truth segmentations while the other rows denote the dataset size. Columns correspond to CNN architectures.

6.1.5 Summary and Discussion

In this section, we have presented the segmentation performances of the baseline CNNs, U-Net-Res, and DeepLabV3, on all datasets. It turned out that all datasets have their own characteristics regarding typical segmentation errors. Datasets with more segmentation classes tend to allow for more different error types. The neck muscle dataset takes a special role. The complexity in terms of the number of segmentation classes (8) and the poor visibility of borders between tissues (also see Section 5.3) lead to odd behaviors of both CNNs, namely, producing topological disorder (U-Net-Res) or just a single average and quite rigid segmentation map (DeepLabV3).

Except for the neck muscle dataset, the segmentation performances increase with more training data. Accordingly, the error rates decrease. However, except for DeepLabV3 on the cardiac dataset, none of the CNNs achieved clinically acceptable performance, even on the largest datasets.

When comparing the exemplary segmentation images, it seems that DeepLabV3 tends to generate predictions with fewer topological errors than U-Net-Res. The error rates support this observation. Here, the rates of DeepLabV3 for topological disorder (error 1 for IVUS lumen and vessel wall, as well as cardiac) and small incorrect patches (error 2 for IVUS lumen and vessel wall, as well as cardiac) are substantially lower. The same holds for error 3, which corresponds to discontinuities in two-dimensional tubular structures (vessel wall, myocardium). A potential reason for this behavior is that DeepLabV3 does not contain deconvolution layers, as opposed to U-Net-Res (see Subsection 3.3.1). Instead, the segmentation masks (pre-softmax) are obtained with a size 8 times smaller than the output size. To reach the output size, the masks are linearly interpolated. Hence, patches with a size of about 8 pixels and smaller can not be generated. Furthermore, the topology tends to be rougher but more stable. U-Net-Res, on the other hand, contains deconvolution layers. Moreover, it comprises skip-connections, even at a level where feature maps have input image size. These aspects affect segmentation maps on a very small scale (at pixel-level). It facilitates more fine-grained and detailed segmentations but also leads to a less coherent topology, especially when datasets are small.

The above also explains why DeepLabV3 generates very rigid mean segmentation masks for the neck muscle dataset. Here, we can see that the lack of clearly visible boundaries in the deeper muscles prevents the networks from detecting them (also compare Section 5.3). Therefore, the barely visible speckle patterns do not seem to add valuable information for identifying the different muscles. This is not surprising, however, as it is likely that speckles of the same type of tissue (muscle) also look quite similar. In addition, we have already discussed that the quality of the annotations is at least debatable (see Section 5.3). All these aspects indicate that generating clinically sufficient results with an end-to-end CNN approach is very hard or even impossible. Nevertheless, in the following sections, we will see if the presented methods can still lead to some improvements.

In summary, the baselines still leave much room for improvement, especially for smaller datasets. In the upcoming sections, we will show whether our developed methods can improve some of the baselines' weaknesses.

6.2 Combining Wavelet Scattering and CNNs

In this section, we present the results of combining wavelet scattering with CNNs and investigate in which cases this approach can improve segmentation performance. Furthermore, we want to examine whether scattering transformations provide useful information in a limited data situation in contrast to ordinary learnable convolutional layers. To make the comparison between SEST CNNs and baseline CNNs fair, we added squeeze and excitation blocks similar to SEST blocks to the baseline CNNs (compare Section 4.1). However, we substituted the scattering transformation layer with ordinary learnable convolutional layers. In this sense, the baseline CNNs even got slightly more parameters than the SEST networks, namely 6.51 M vs. 6.47 M.

6.2.1 IVUS Lumen and Vessel Wall Segmentation

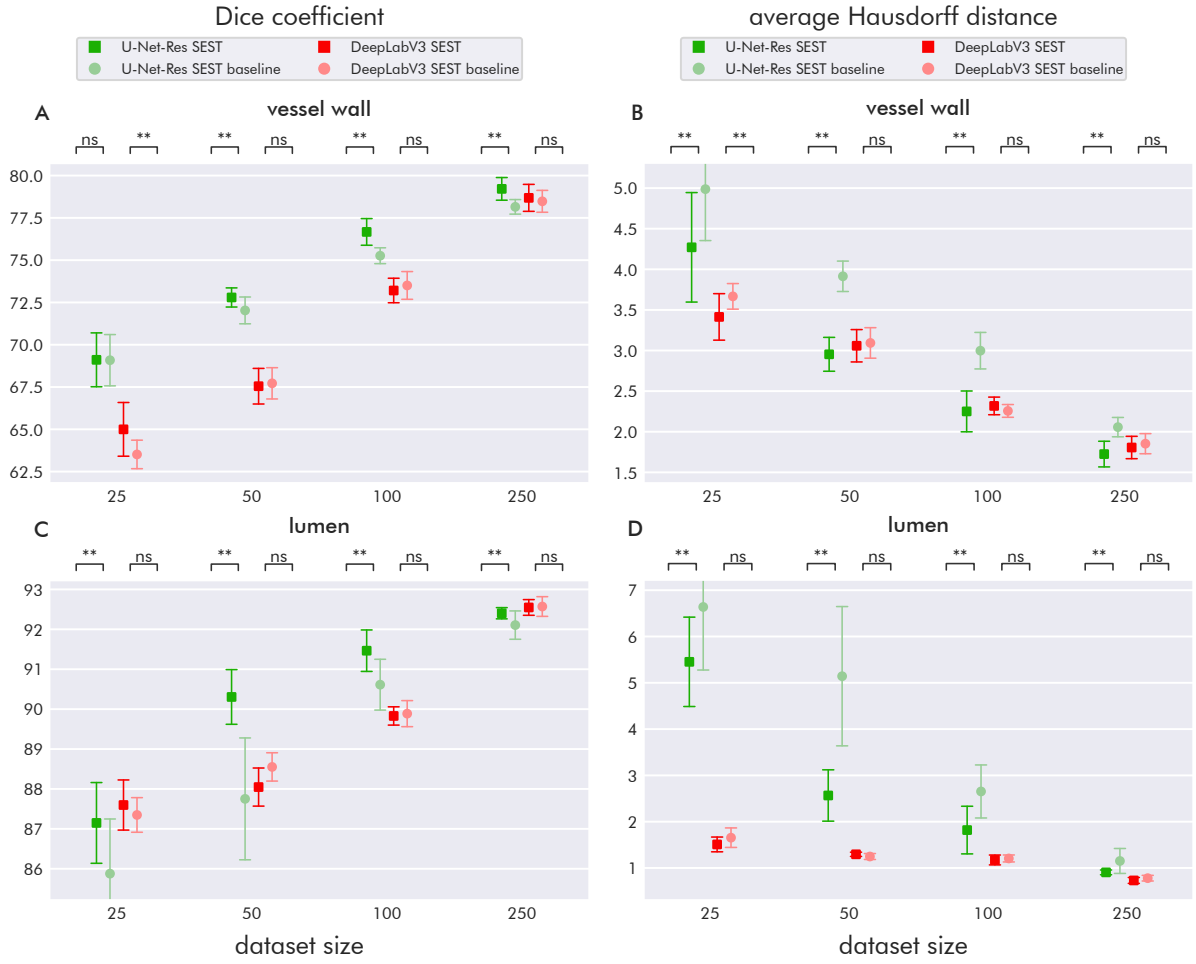

Figure 6.12: IVUS lumen and vessel wall segmentation results using SEST. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

The segmentation results of both SEST CNNs and SEST baseline CNNs are shown in Figure 6.12. The annotations above the markers indicate whether the improvements of SEST CNNs over baseline CNNs are statistically significant (**) or not (ns). Only networks of the same flavor are compared. It turns out that almost all improvements due to SEST are

Table 6.4: IVUS lumen and vessel wall segmentation error rates using SEST. For each error, the rates achieved by SEST baseline and SEST are given, as well as the relative change of the latter compared to the former. The columns are divided by dataset size (25, 50, 100, and 250 images) and CNN architecture (U: U-Net-Res, D: DeepLabV3). Values are given as percentages.

		25		50		100		250	
error	**method**	U	D	U	D	U	D	U	D
error 1	**SEST baseline**	1.3	0.7	1.3	0.0	0.7	0.0	0.0	0.0
	SEST	0.7	0.0	0.7	0.0	0.0	0.0	0.0	0.0
	rel. change	**−50.0**	**−100.0**	**−50.0**	—	**−100.0**	—	—	—
error 2	**SEST baseline**	46.0	11.3	23.3	10.7	16.0	8.7	15.3	8.7
	SEST	38.0	5.3	26.7	4.7	18.0	6.7	6.7	7.3
	rel. change	**−17.4**	**−52.9**	**14.3**	**−56.2**	**12.5**	**−23.1**	**−56.5**	**−15.4**
error 3	**SEST baseline**	13.3	2.7	8.0	2.7	6.7	1.3	3.3	1.3
	SEST	8.0	2.0	6.0	2.0	4.0	1.3	0.0	1.3
	rel. change	**−40.0**	**−25.0**	**−25.0**	**−25.0**	**−40.0**	**0.0**	**−100.0**	**0.0**
error 4	**SEST baseline**	71.3	46.0	51.3	36.7	58.0	30.7	42.0	24.0
	SEST	74.7	47.3	39.3	37.3	62.7	38.0	36.7	28.0
	rel. change	**4.7**	**2.9**	**−23.4**	**1.8**	**8.0**	**23.9**	**−12.7**	**16.7**
error 5	**SEST baseline**	20.0	20.0	18.0	14.7	11.3	16.0	11.3	8.7
	SEST	14.7	13.3	14.0	10.7	14.0	12.0	13.3	10.0
	rel. change	**−26.7**	**−33.3**	**−22.2**	**−27.3**	**23.5**	**−25.0**	**17.6**	**15.4**

statistically significant for U-Net-Res. SEST has virtually no positive effect on DeepLabV3, except for vessel wall metrics with 25 training images. Both DeepLabV3 flavors achieve much better Hausdorff distances for 25 training images than the U-Net-Res flavors. The values of the metrics equalize as the number of training images increases. When comparing the SEST baseline results with the original baseline results (Subsection 6.1.1), one can see no systematic differences.

Table 6.4 depicts the error rates (in %) of all CNNs for all dataset sizes. See Subsection 6.1.1 for details on the individual error types. Table 6.4 also shows relative changes of error rates between baseline and SEST networks. The more intense the green color, the greater the improvement in the error rate. The more intense the red color, the greater the deterioration. The exact color coding for relative changes is as follows: dark green: $v \leq -50$, medium green: $-50 < v \leq -25$, light green: $-25 < v < 0$ white: $v = 0$ or none, light red: $0 < v < 25$, medium red: $25 \leq v < 50$, dark red: $v \geq 50$. Since no variances are available, we cannot make any statements regarding statistical significance. Therefore, small changes in error rates up to 25 % (depicted in light colors) should be interpreted with caution. Furthermore, relative improvements in error rates that are already quite small can appear quite large. On the other hand, large absolute changes in error rates that are already very large can appear quite small when presented as relative changes. This effect occurs mostly with larger datasets, where error rates are already below 5 % or even 2 %. These aspects have to be taken into account.

Error 1 (topological disorder) could be reduced in all cases. However, the numbers of errors were already quite low (1 or 2), such that the large reductions of 50 % or 100 % are not very momentous. Reducing incorrect patches (error 2) through SEST was more successful for DeepLabV3. Error 3 (discontinuous vessel wall) was decreased or stayed the same in all cases. Interestingly, the rates of error 4 (lumen or vessel wall marked in the background)

were slightly increased by SEST. Exceptions are U-Net-Res for 50 and 250 training images. Error 5 (background marked in lumen or vessel wall) could be reduced for smaller datasets but increased slightly with larger datasets.

6.2.2 IVUS Calcium Segmentation

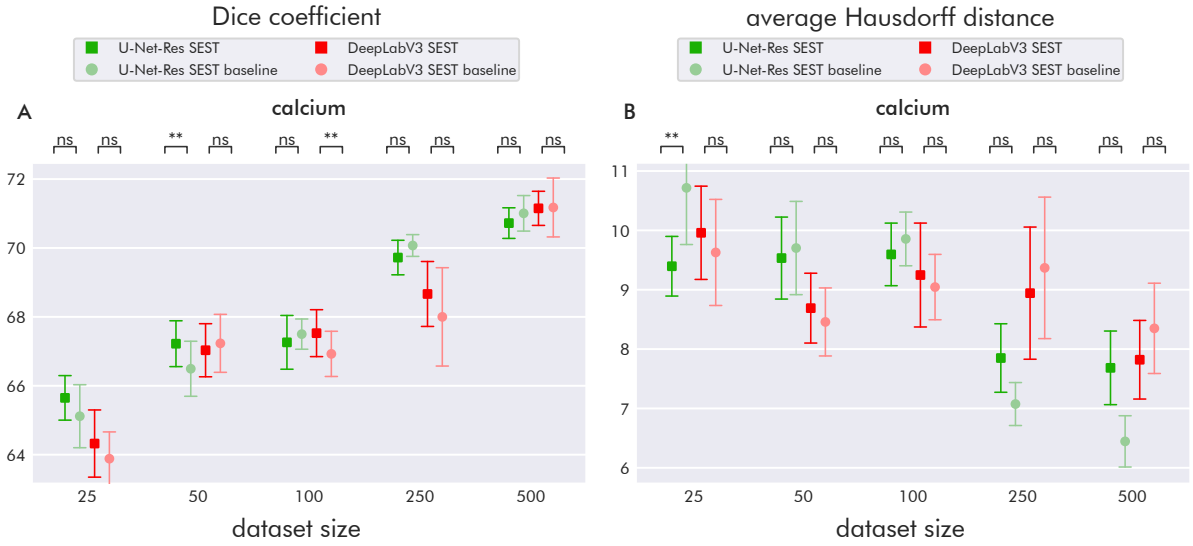

Figure 6.13: IVUS calcium segmentation results using SEST. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

Table 6.5: IVUS calcium segmentation error rates using SEST. For each error, the rates achieved by SEST baseline and SEST are given, as well as the relative change of the latter compared to the former. The columns are divided by dataset size (25, 50, 100, 250, and 500 images) and CNN architecture (U: U-Net-Res, D: DeepLabV3). Values are given as percentages.

		25		50		100		250		500	
error	**method**	U	D	U	D	U	D	U	D	U	D
error 1	**SEST baseline**	62.7	57.0	54.9	47.7	63.2	46.6	45.6	37.8	37.3	30.6
	SEST	59.6	48.7	35.2	49.2	32.1	45.6	48.7	39.9	32.1	32.6
	rel. change	**−5.0**	**−14.5**	**−35.8**	**3.3**	**−49.2**	**−2.2**	**6.8**	**5.5**	**−13.9**	**6.8**
error 2	**SEST baseline**	31.6	34.2	38.9	31.6	35.2	35.2	35.8	29.5	39.9	30.6
	SEST	25.4	30.6	26.9	34.7	25.9	36.8	26.9	34.2	27.5	31.6
	rel. change	**−19.7**	**−10.6**	**−30.7**	**9.8**	**−26.5**	**4.4**	**−24.6**	**15.8**	**−31.2**	**3.4**

The results on the IVUS calcium dataset are depicted in Figure 6.13. Interestingly, adding squeeze and excitation blocks with learnable convolutions to U-Net-Res increases its performance on calcium segmentation compared to not adding these blocks (see Figure 6.4). However, with two exceptions, SEST does not provide any benefit for this dataset. Although the mean metrics sometimes reach better values than the baselines, these improvements are not significant due to the comparatively large variances. The large variances indicate that the actual performance of a network is largely dependent on the parameter initialization. When we compare the SEST baseline results with the original baseline performance (see Subsection 6.1.2),

we can see that the average Hausdorff distance of U-Net-Res for 250 and 500 training images largely improved. This indicates that ordinary squeeze and excitation blocks may boost the performance of U-Net-Res for calcium segmentation when datasets are larger.

If we look at the error rates in Table 6.5, we can see that errors tend to be reduced more successfully in the case of U-Net-Res. However, most of the relative changes are in the range $[-25\,\%, 25\,\%]$ and should therefore be interpreted cautiously.

6.2.3 Cardiac Segmentation

Table 6.6: Cardiac segmentation error rates using SEST. For each error, the rates achieved by SEST baseline and SEST are given, as well as the relative change of the latter compared to the former. The columns are divided by dataset size (20, 40, 80, 160, 320, 640, and 1280 images) and CNN architecture (U: U-Net-Res, D: DeepLabV3). Values are given as percentages.

		20		40		80		160		320		640		1280	
error	method	U	D	U	D	U	D	U	D	U	D	U	D	U	D
error 1	SEST baseline	19.2	0.0	11.4	0.0	6.6	0.0	3.6	0.0	1.8	0.0	1.0	0.0	0.0	0.0
	SEST	7.8	0.0	4.2	0.0	3.2	0.0	2.4	0.0	0.6	0.0	0.6	0.0	0.0	0.0
	rel. change	**−59.4**	—	**−63.2**	—	**−51.5**	—	**−33.3**	—	**−66.7**	—	**−40.0**	—	—	—
error 2	SEST baseline	66.6	18.0	45.8	14.0	33.2	9.8	30.8	5.6	19.2	4.4	11.2	2.2	7.0	0.0
	SEST	56.0	18.2	37.6	16.4	30.2	11.0	19.2	6.2	10.2	3.4	8.8	1.6	2.4	0.0
	rel. change	**−15.9**	**1.1**	**−17.9**	**17.1**	**−9.0**	**12.2**	**−37.7**	**10.7**	**−46.9**	**−22.7**	**−21.4**	**−27.3**	**−65.7**	—
error 3	SEST baseline	33.0	15.4	25.0	13.4	17.0	10.4	11.0	8.2	10.2	6.8	7.4	3.0	4.2	1.0
	SEST	30.2	9.0	23.6	7.2	17.4	7.2	9.2	5.4	3.2	4.8	0.6	1.8	0.8	0.8
	rel. change	**−8.5**	**−41.6**	**−5.6**	**−46.3**	**2.4**	**−30.8**	**−16.4**	**−34.1**	**−68.6**	**−29.4**	**−91.9**	**−40.0**	**−81.0**	**−20.0**
error 4	SEST baseline	19.8	22.6	12.6	17.6	7.8	13.8	7.0	7.8	4.6	6.4	3.4	1.6	0.4	0.6
	SEST	12.2	16.2	10.4	14.6	8.0	12.4	6.6	9.2	3.8	5.6	1.0	1.8	0.4	0.8
	rel. change	**−38.4**	**−28.3**	**−17.5**	**−17.0**	**2.6**	**−10.1**	**−5.7**	**17.9**	**−17.4**	**−12.5**	**−70.6**	**12.5**	**0.0**	**33.3**
error 5	SEST baseline	22.2	15.4	14.8	13.4	8.2	8.0	6.0	7.2	4.2	4.2	2.2	1.4	0.2	0.0
	SEST	12.6	14.2	9.6	12.2	6.4	9.0	3.8	8.4	3.8	4.0	1.0	1.8	0.0	0.0
	rel. change	**−43.2**	**−7.8**	**−35.1**	**−9.0**	**−22.0**	**12.5**	**−36.7**	**16.7**	**−9.5**	**−4.8**	**−54.5**	**28.6**	**−100.0**	—

Figure 6.14 shows the influence of SEST on cardiac segmentation performance. As with the IVUS lumen and vessel wall dataset, the improvements through SEST are mostly restricted to U-Net-Res. U-Net-Res SEST reaches the best performance in almost all cases up to at least 640 training images. Exceptions are Hausdorff distances with 20 training examples. But even in these cases, U-Net-Res SEST clearly outperforms the baseline. When comparing the SEST baseline results with the original baseline results (Subsection 6.1.3), one can see that there are no systematic differences except that U-Net-Res seems to benefit from ordinary squeeze and excitation blocks for atrium Dice performance with mid-sized datasets.

If we look at the error rates in Table 6.6, we see much improvement. Error 1 (topological disorder) decreased largely for U-Net-Res. Error 2 (incorrect patches) decreased slightly with U-Net-Res but increased with DeepLabV3 for up to 160 training images. With one exception, error 3 (discontinuous myocardium) decreased moderately or even largely. Error 4 (myocardium placed around atrium) and error 5 (endocardium and atrium partially confused) rates provide a mixed picture. However, larger improvements tend to be achieved with U-Net-Res, which is also reflected by the segmentation metrics.

6.2.4 Neck Muscle Segmentation

Figure 6.15 reveals that SEST does not systematically improve the mean segmentation results on the neck muscle dataset. All improvements, except one, are restricted to U-Net-Res and tend to occur with larger dataset sizes. However, U-Net-Res still produces useless results with Dice

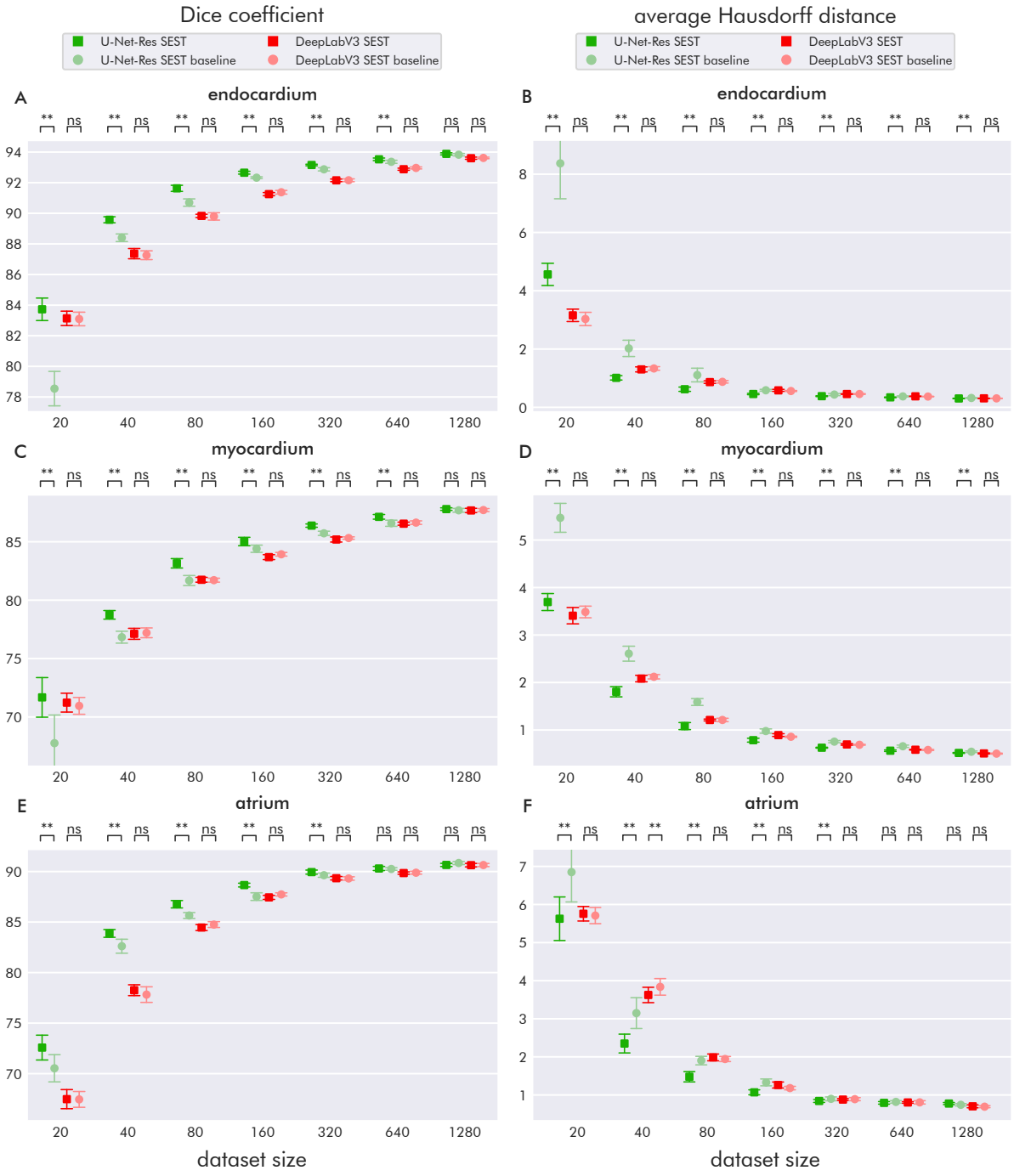

Figure 6.14: Cardiac segmentation results using SEST. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

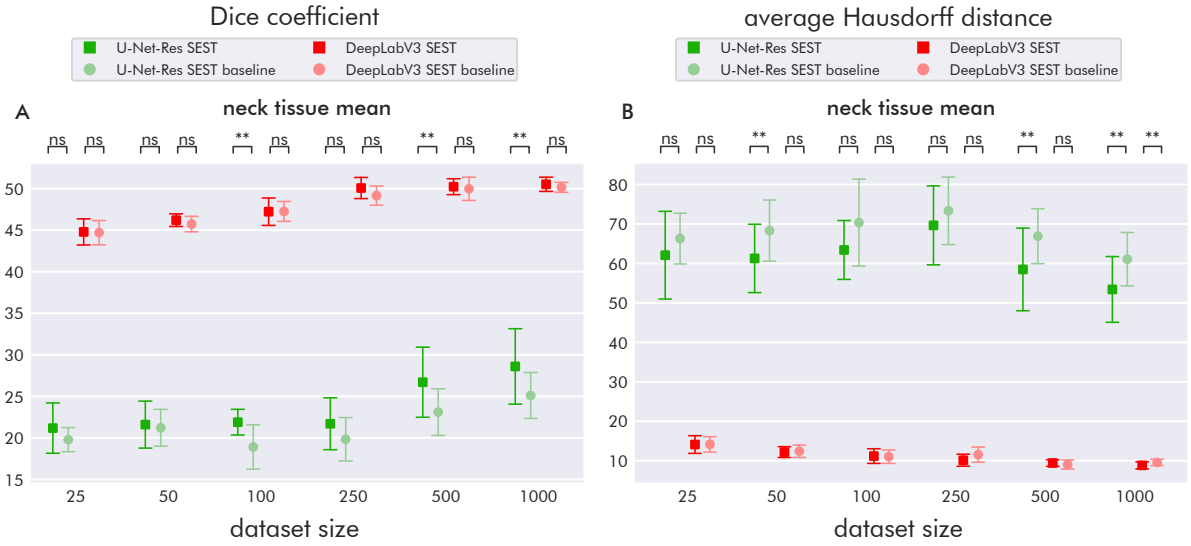

Figure 6.15: Neck muscle segmentation results using SEST. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

scores below 30 % and average Hausdorff distances above 50 pixels. We can see no systematic differences when we compare the SEST baseline results with the original baseline performance (see Subsection 6.1.4).

6.2.5 Summary and Discussion

In this section, we have shown in which cases incorporating wavelet transforms via SEST blocks into CNNs leads to improvements in segmentation performance. Before we go into answering **RQ 2** for this method, we want to point out some other notable findings.

Interestingly, a reduction of certain error rates does not always implicate an improvement in segmentation metrics. For example, a decrease in incorrect patches in the IVUS lumen and vessel wall dataset (DeepLabV3), as well as a reduction of false positives in the IVUS calcium dataset (U-Net-Res), does not lead to respective improvements of the average Hausdorff distance. In the first case, the error 2 frequency of the DeepLabV3 baseline was already quite small ($< 12\,\%$). Furthermore, small patches do not affect the average Hausdorff distance much (in contrast to the ordinary Hausdorff distance). We also see the opposite behavior. SEST slightly raised error 4 of the IVUS lumen and vessel wall dataset, but the metrics of U-Net-Res still improved. However, although still present, the severity of errors could have been reduced. The error frequency does not account for reductions in error severity, but a reduction in error severity improves the segmentation metrics. The results on the cardiac dataset show that reducing topological disorder (error 1) does, indeed, correlate with improved segmentation metrics. The same holds for U-Net-Res and the other errors. However, although reducing the error 1 rate of U-Net-Res for 20 training images, the average Hausdorff distance does still not reach the values by DeepLabV3. This is not surprising since DeepLabV3 does not generate error 1 in any case. This shows again that DeepLabV3 is quite robust in terms of correct topology. In summary, reducing a particular error does not have to correlate with improving segmentation metrics. A reduction of a certain error can be balanced or even outbalanced by another error. Furthermore, removing very mild errors hardly influences the segmentation

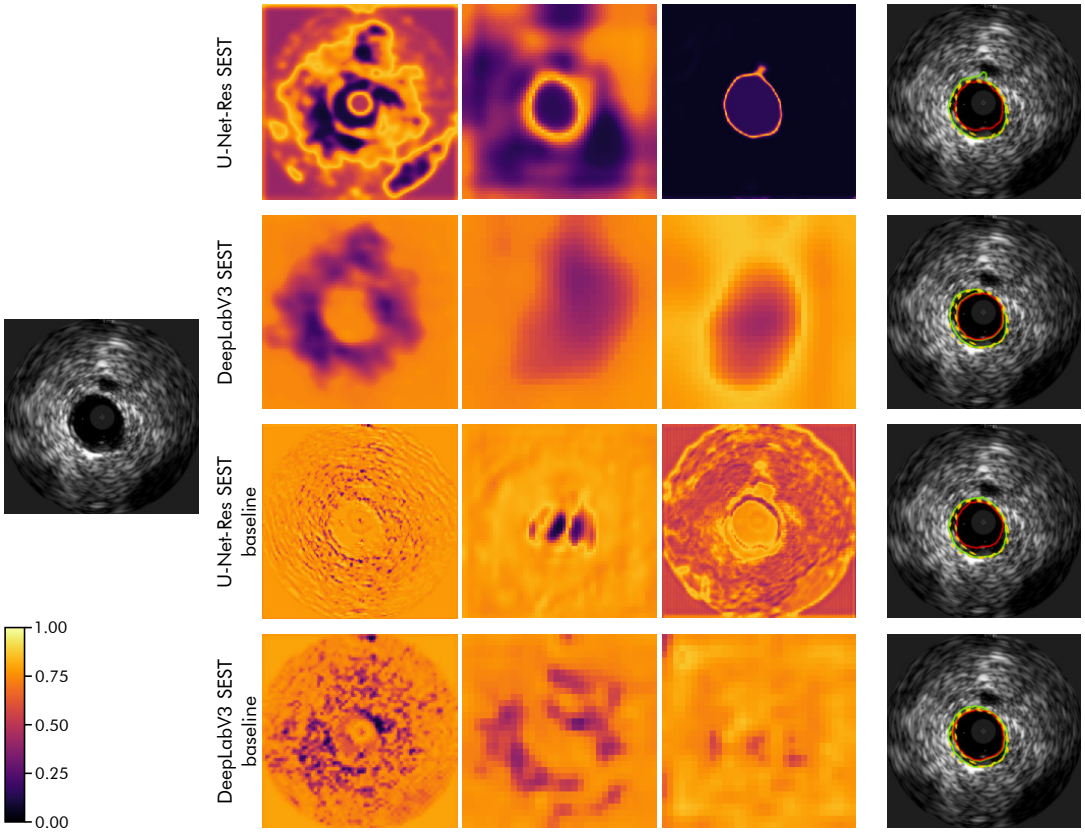

Figure 6.16: Spatial attention maps in IVUS lumen and vessel wall segmentation. The input image is shown on the left side. The rows correspond to different CNN architectures. Depicted are spatial attention maps from a shallow layer (left), a medial layer (center), and a deep layer (right). The predicted segmentation masks are shown on the right.

metrics. Lastly, large relative reductions of baseline error rates that are already quite small can achieve quite large values. But improving only a very small amount of test images does not have a large effect on the overall test metrics.

The results on the neck muscle dataset show that SEST cannot draw valuable information from the images. This supports our hypothesis that this dataset does not provide much information for CNNs. However, we observed some minor improvements for larger dataset sizes. So maybe there is some potential in considering the other methods that will be investigated in the upcoming sections.

We now want to compare different attention maps that individual CNNs generate. Figure 6.16 depicts exemplary attention maps generated by the SEST and baseline versions of U-Net-Res and DeepLabV3 for a medium-sized training set. The left side shows the input images, and the right shows the resulting segmentation masks. The three columns in the center depict the attention maps from deep, medial, and shallow layers of the CNNs, respectively. The values of the attention maps range from 0 to 1 (see the color bar in the bottom left), indicating regions of higher (near 1) and lower (near 0) importance, as the attention maps are multiplied with the corresponding feature maps in the CNNs. First, we notice that U-Net-Res generates attention maps that are far more localized than the ones by DeepLabV3. Second, the attention maps by the SEST CNNs, particularly U-Net-Res SEST, tend to indicate boundaries between the regions of interest and the background, especially in the layers near the output. Third,

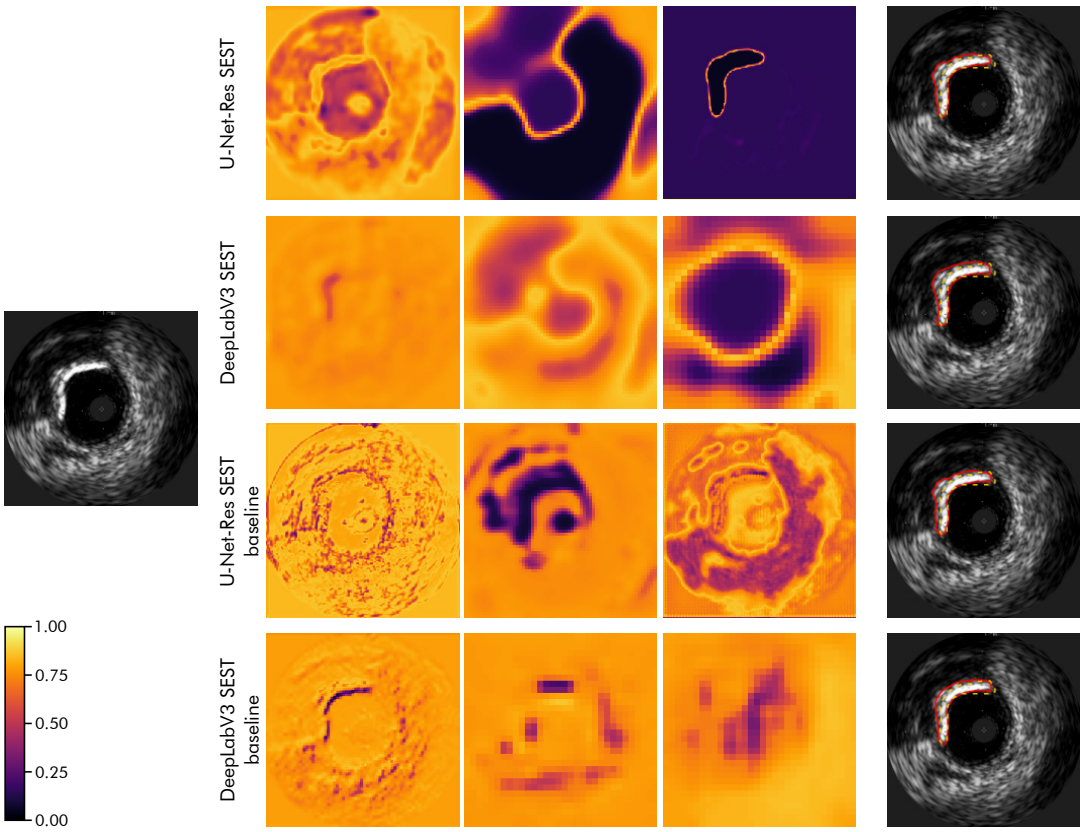

Figure 6.17: Spatial attention maps in IVUS calcium segmentation. The input image is shown on the left side. The rows correspond to different CNN architectures. Depicted are spatial attention maps from a shallow layer (left), a medial layer (center), and a deep layer (right). The predicted segmentation masks are shown on the right.

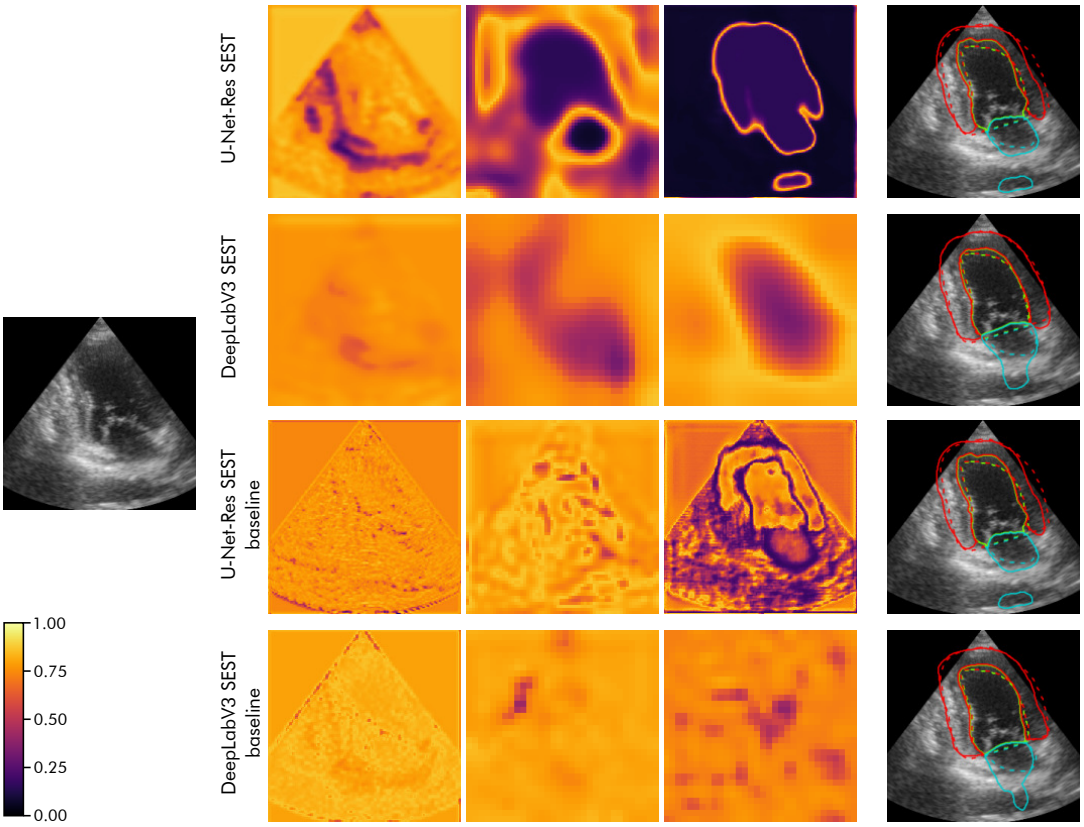

Figure 6.18: Spatial attention maps in cardiac segmentation. The input image is shown on the left side. The rows correspond to different CNN architectures. Depicted are spatial attention maps from a shallow layer (left), a medial layer (center), and a deep layer (right). The predicted segmentation masks are shown on the right.

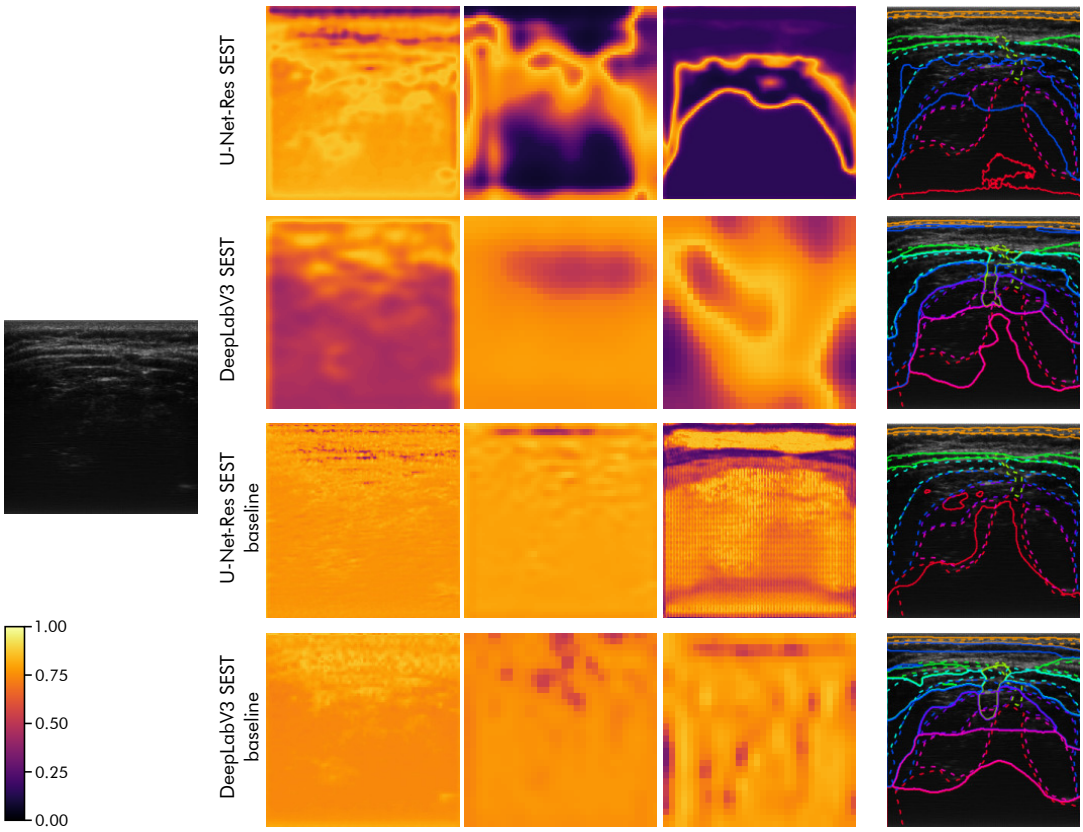

Figure 6.19: Spatial attention maps in neck muscle segmentation. The input image is shown on the left side. The rows correspond to different CNN architectures. Depicted are spatial attention maps from a shallow layer (left), a medial layer (center), and a deep layer (right). The predicted segmentation masks are shown on the right.

the early attention maps of the baseline SEST networks appear rather spotty, highlighting the speckle noise and imitating a kind of edge detection. The corresponding attention maps of the SEST networks exhibit more contiguous regions and do not just recognize edges of the speckle noise. Considering all these aspects indicates that the attention maps by SEST guide the CNNs in a meaningful way. Similar arrangements for the other datasets are depicted in Figure 6.17 (IVUS calcium dataset), Figure 6.18 (cardiac dataset), and Figure 6.19 (neck muscle dataset). These basically all show the same behavior.

When we compare the SEST baseline results with the results of the original baseline, we see that the performances are fairly similar. An exception is the IVUS calcium dataset which seems to benefit quite largely from squeeze and excitation blocks when applied to U-Net-Res, even without scattering transformation. Figure 6.20 compares the aforementioned results and shows that the improvements tend to increase with increasing dataset size. The average Hausdorff distance even reaches values better than DeepLabV3. The latter does not benefit from ordinary squeeze and excitation blocks. On the contrary, the mean values tend to be slightly worse.

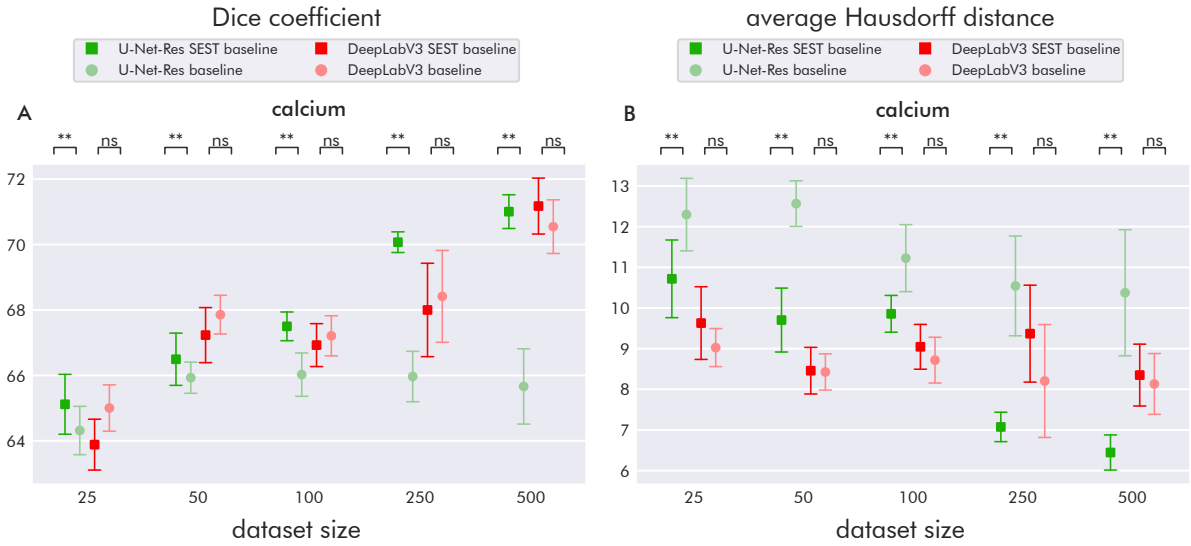

Figure 6.20: Comparison between original baseline and SEST baseline for IVUS calcium segmentation. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

We note that the results on IVUS calcium segmentation in this work differ from those we obtained in our paper on SEST [20]. We see multiple reasons for that. First, the dataset in our paper was different and comprised fewer patients and thus exhibited less variability. Second, we performed only 3-fold cross-validation in our paper and did not employ majority voting of the individual models obtained through cross-validation. Therefore, the results in our paper and this work are not comparable.

We now want to answer **RQ 2** for SEST. As we saw in the results, the behavior of SEST performance depends on the dataset, the considered metric, and the CNN.

RQ 2.1: Which CNN architectures benefit from SEST?

In the case of the IVUS lumen and vessel wall dataset, as well as the cardiac dataset, we observe that U-Net-Res benefits much stronger from SEST than DeepLabV3. In this way, U-Net-Res could even outperform DeepLabV3 in some cases. SEST did not enable systematic

improvements for DeepLabV3. Instead, only a few scattered cases improve but do not show any particular pattern. Therefore, SEST does not seem to be a suitable method for DeepLabV3. The images in Figure 6.16 to Figure 6.19 show that the highlighted regions in the attention maps by DeepLabV3 tend to be rougher and less localized than the ones by U-Net-Res. This also indicates that DeepLabV3 is unable to leverage the information by SEST. Due to the wavelet scattering transformation, SEST is likely to draw information from image textures on a more fine-grained scale. We already explained in the last section (Section 6.1) that DeepLabV3 cannot resolve details smaller than about 8 pixels. Due to its decoding path, U-Net-Res is not restricted in this manner and, thus, can produce more fine-grained segmentations. This is where the strength of SEST could come in, leading to the improvements we have observed. In the following, we limit our considerations to U-Net-Res and ignore DeepLabV3.

RQ 2.2: How does SEST perform as a function of dataset size?
The performance on the IVUS lumen and vessel wall dataset indicates that improvements through SEST are smaller for smaller datasets, increase for medium-sized datasets and decrease for larger datasets. However, in the case of cardiac segmentation, improvements are largest for the smallest datasets and decrease with increasing dataset size. It therefore seems that SEST is not able to unlock its full potential for IVUS lumen and vessel wall segmentation with only 25 training images. It is likely that the information extracted by SEST from the smallest IVUS lumen and vessel wall dataset cannot be meaningfully combined with the information extracted by the learnable filters.

RQ 2.3: Which tissues benefit from SEST?
While we see no particular benefits of SEST for IVUS calcium segmentation, IVUS lumen and vessel wall segmentation can be largely improved by SEST. Especially regarding the average Hausdorff distance for both classes and the lumen Dice score. Moreover, all three classes of cardiac segmentation benefit from SEST. In the case of neck muscle segmentation, the improvements are negligible. Wavelets are basically designed to extract information from image textures. Therefore, we assume that SEST does not perform well on calcium and neck muscle segmentation since these structures do not exhibit textures that provide information for segmentation. The neck muscle images mostly appear rather dark and homogeneous such that even humans struggle to identify any borders between the deeper muscles (that's why MRT images were used to estimate the ground truth, see Section 5.3). Calcifications appear relatively bright and plain and usually do not show typical brightness variations. Hence, we think that the position in the context of the vessel topology is much more important for identifying calcium than its texture. But this seems not to be the strength of SEST.

RQ 2.4: What types of segmentation errors are reduced by SEST?
SEST significantly reduced topological disorder (error 1 for IVUS lumen and vessel wall and cardiac segmentation) of U-Net-Res and thus increased segmentation performance. Moreover, SEST reduced discontinuous vessel walls and myocardia (error 3) and thus improved the corresponding segmentation metrics. But also, the other errors tend to improve, especially for smaller datasets. It is therefore likely that the attention maps by SEST supported the CNN to generate the correct topology.

Summarizing, SEST positively impacted U-Net-Res to generate predictions of the IVUS

lumen and vessel wall dataset, as well as the cardiac dataset. Error rates concerning tissue topology were reduced. The performance improvements with respect to the segmentation metrics tend to vanish for larger datasets.

6.3 Independent Component Analysis as a Shape Prior

In this section, we look at the segmentation results when incorporating a shape prior via independent component analysis (ICA) into the CNNs. Since generating such a shape prior for calcifications is pointless, we omitted the IVUS calcium dataset. In this dataset, the difficulties do not lie in the shape and topology but rather in the location of the small calcified regions. Since the skin class of the neck muscle dataset did not allow for calculating an ICA, we omitted this class from the ICA shape prior. See Subsection 4.2.2 for exemplary independent components of each dataset we investigated in this section.

As for the SEST networks, adding a second branch to the baseline CNNs increases the number of trainable parameters. Furthermore, a parallel branch could improve segmentation results, even without a linear combination of independent components. Therefore, we adjusted the baseline CNNs by adding the same parallel branch as for the ICA networks. However, we replaced the tensors holding the independent components with a random but fixed tensor, i.e., a tensor sampled from a normal distribution once before all experiments and then kept fixed. In that way, the baseline CNNs could adapt to this tensor during training to generate useful attention maps. Adding a parallel branch increased the number of trainable parameters to about 7.2 M. This value slightly varies with different datasets and thus different numbers of segmentation classes.

6.3.1 IVUS Lumen and Vessel Wall Segmentation

Table 6.7: IVUS lumen and vessel wall segmentation error rates using ICA shape priors. For each error, the rates achieved by ICA baseline and ICA are given, as well as the relative change of the latter compared to the former. The columns are divided by dataset size (25, 50, 100, and 250 images) and CNN architecture (U: U-Net-Res, D: DeepLabV3). Values are given as percentages.

		25		50		100		250	
error	**method**	U	D	U	D	U	D	U	D
error 1	**ICA baseline**	3.3	0.0	3.3	0.0	0.7	0.0	0.0	0.0
	ICA	2.7	0.0	2.0	0.0	0.7	0.0	0.0	0.0
	rel. change	**−20.0**	—	**−40.0**	—	**0.0**	—	—	—
error 2	**ICA baseline**	51.3	11.3	32.7	10.7	22.7	6.7	16.7	6.0
	ICA	44.0	12.0	27.3	10.7	25.3	8.0	18.7	7.3
	rel. change	**−14.3**	**5.9**	**−16.3**	**0.0**	**11.8**	**20.0**	**12.0**	**22.2**
error 3	**ICA baseline**	16.7	2.7	11.3	3.3	12.0	2.7	10.0	2.0
	ICA	10.0	1.3	9.3	0.7	6.0	0.7	6.0	0.7
	rel. change	**−40.0**	**−50.0**	**−17.6**	**−80.0**	**−50.0**	**−75.0**	**−40.0**	**−66.7**
error 4	**ICA baseline**	73.3	50.7	64.7	34.7	52.7	26.7	44.0	22.0
	ICA	68.7	54.7	65.3	37.3	47.3	28.0	41.3	24.0
	rel. change	**−6.4**	**7.9**	**1.0**	**7.7**	**−10.1**	**5.0**	**−6.1**	**9.1**
error 5	**ICA baseline**	21.3	26.7	23.3	16.0	18.0	18.0	14.7	10.0
	ICA	18.7	16.7	19.3	12.7	18.7	14.0	16.0	10.7
	rel. change	**−12.5**	**−37.5**	**−17.1**	**−20.8**	**3.7**	**−22.2**	**9.1**	**6.7**

Figure 6.21 shows the results of IVUS lumen and vessel wall segmentation. ICA shape priors do not seem to positively impact vessel wall segmentation, but they largely improve U-Net-

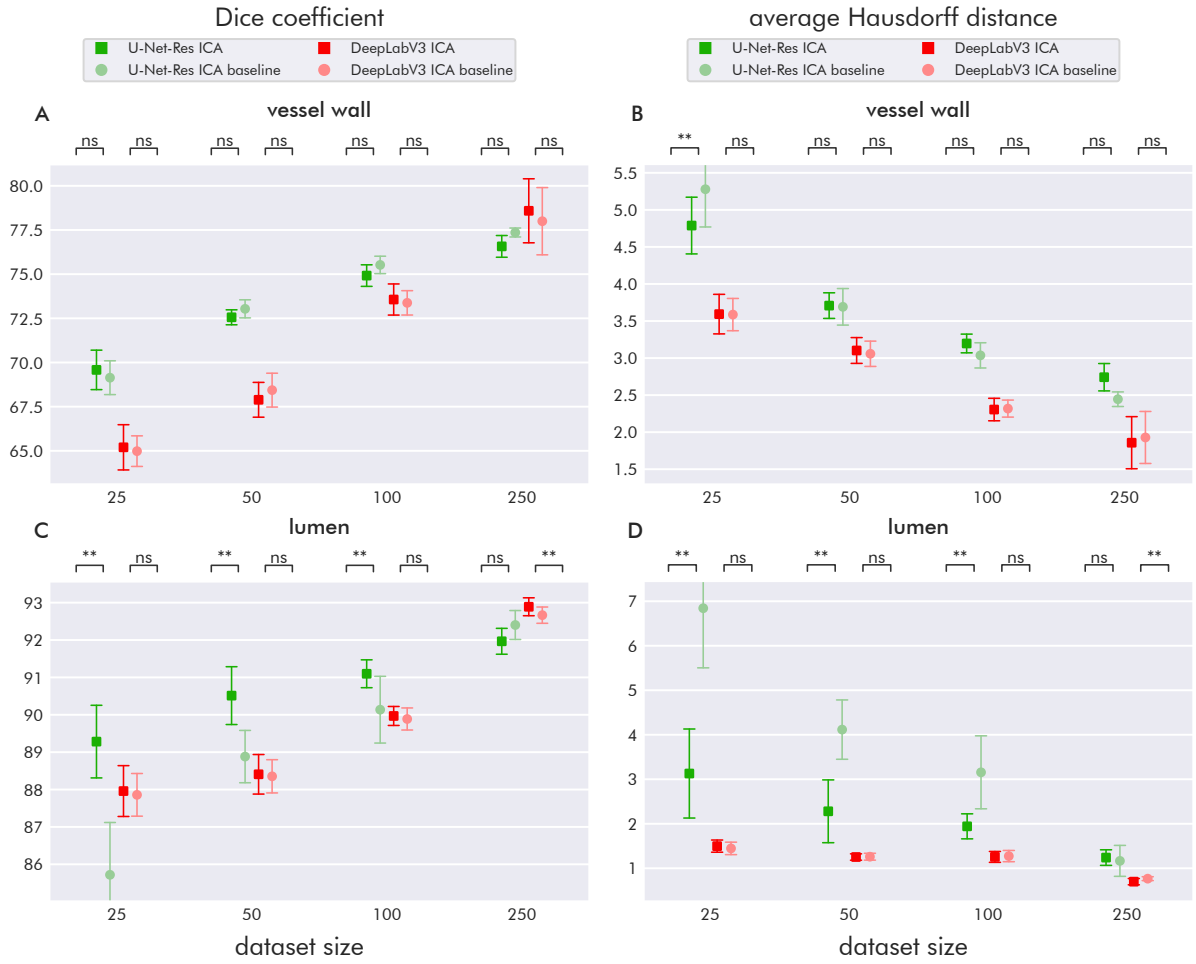

Figure 6.21: IVUS lumen and vessel wall segmentation results using ICA shape priors. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

Res performance on lumen segmentation. This leads to U-Net-Res outperforming DeepLabV3 in terms of Dice score and dataset sizes up to 100 images. ICA shape priors also improve the average Hausdorff distance in this range of dataset size. However, the performance of DeepLabV3 was not surpassed. Comparing the ICA baseline results with the original baseline performance (Subsection 6.1.1) shows that the secondary network branch is not beneficial in this case.

Regarding the error rates in Table 6.7, the improvement of error 3 (discontinuous vessel wall) is quite striking, although the baseline values are already quite small for larger datasets. However, errors 2, 4, and 5 were only slightly reduced or even increased. Again, DeepLabV3 does not produce any topological disorder (error 1).

6.3.2 Cardiac Segmentation

Table 6.8: Cardiac segmentation error rates using ICA shape priors. For each error, the rates achieved by ICA baseline and ICA are given, as well as the relative change of the latter compared to the former. The columns are divided by dataset size (20, 40, 80, 160, 320, 640, and 1280 images) and CNN architecture (U: U-Net-Res, D: DeepLabV3). Values are given as percentages.

		20		40		80		160		320		640		1280	
error	**method**	U	D	U	D	U	D	U	D	U	D	U	D	U	D
error 1	**ICA baseline**	23.0	0.0	14.6	0.0	9.6	0.0	5.0	0.0	2.2	0.0	1.0	0.0	0.0	0.0
	ICA	2.4	0.0	1.6	0.0	1.4	0.0	0.4	0.0	0.8	0.0	0.0	0.0	0.0	0.0
	rel. change	**−89.6**	—	**−89.0**	—	**−85.4**	—	**−92.0**	—	**−63.6**	—	**−100.0**	—	—	—
error 2	**ICA baseline**	73.2	18.4	54.6	14.0	41.0	11.0	38.0	6.2	19.6	4.4	12.2	2.6	9.0	1.2
	ICA	81.0	15.4	63.0	13.6	45.2	12.0	35.0	7.6	22.0	4.2	14.4	3.0	9.6	2.0
	rel. change	**10.7**	**−16.3**	**15.4**	**−2.9**	**10.2**	**9.1**	**−7.9**	**22.6**	**12.2**	**−4.5**	**18.0**	**15.4**	**6.7**	**66.7**
error 3	**ICA baseline**	40.2	17.6	30.0	14.0	22.0	12.4	15.6	5.2	10.6	3.8	6.0	2.0	4.4	0.0
	ICA	36.2	15.6	24.6	11.8	17.8	6.2	11.0	2.8	7.2	2.6	5.6	0.4	3.4	0.0
	rel. change	**−10.0**	**−11.4**	**−18.0**	**−15.7**	**−19.1**	**−50.0**	**−29.5**	**−46.2**	**−32.1**	**−31.6**	**−6.7**	**−80.0**	**−22.7**	—
error 4	**ICA baseline**	21.8	24.4	17.6	18.0	14.0	14.4	13.6	7.2	7.0	6.0	2.2	1.8	0.0	0.8
	ICA	10.0	10.2	8.6	8.4	7.4	6.6	7.0	4.2	5.2	3.6	1.6	1.6	0.0	1.0
	rel. change	**−54.1**	**−58.2**	**−51.1**	**−53.3**	**−47.1**	**−54.2**	**−48.5**	**−41.7**	**−25.7**	**−40.0**	**−27.3**	**−11.1**	—	**25.0**
error 5	**ICA baseline**	23.8	14.2	22.0	13.2	16.6	9.0	8.6	6.2	3.8	4.0	2.0	1.4	0.0	0.0
	ICA	31.0	16.6	23.2	14.6	16.0	8.2	10.4	5.8	4.0	4.2	2.0	2.0	0.0	0.0
	rel. change	**30.3**	**16.9**	**5.5**	**10.6**	**−3.6**	**−8.9**	**20.9**	**−6.5**	**5.3**	**5.0**	**0.0**	**42.9**	—	—

The results on the cardiac dataset are depicted in Figure 6.22. We see that, except for a few cases, ICA shape priors do not lead to any improvements. However, the large variances of some baseline atrium metrics are quite striking. Here, the network completely failed to recognize the atrium in all test images, leading to Dice scores of 0 and average Hausdorff distances of 128 pixels (half the image dimensions). This behavior does not occur with the original baseline. Adding a secondary branch to a U-Net-Res could therefore also lead to instabilities in network training, causing the network to occupy an unfavorable minimum of the loss landscape. Apart from that, the ICA baseline results are comparable to the original baseline results (Subsection 6.1.3).

Another striking aspect shows up in Table 6.8. We can see that the frequency of error 1 (topological disorder) is drastically reduced for U-Net-Res. In the previous sections, a reduction in topological disorder always coincided with improvements in the segmentation metrics. Here, that is not the case. We will give a reasonable explanation for this behavior later in Subsection 6.3.4.

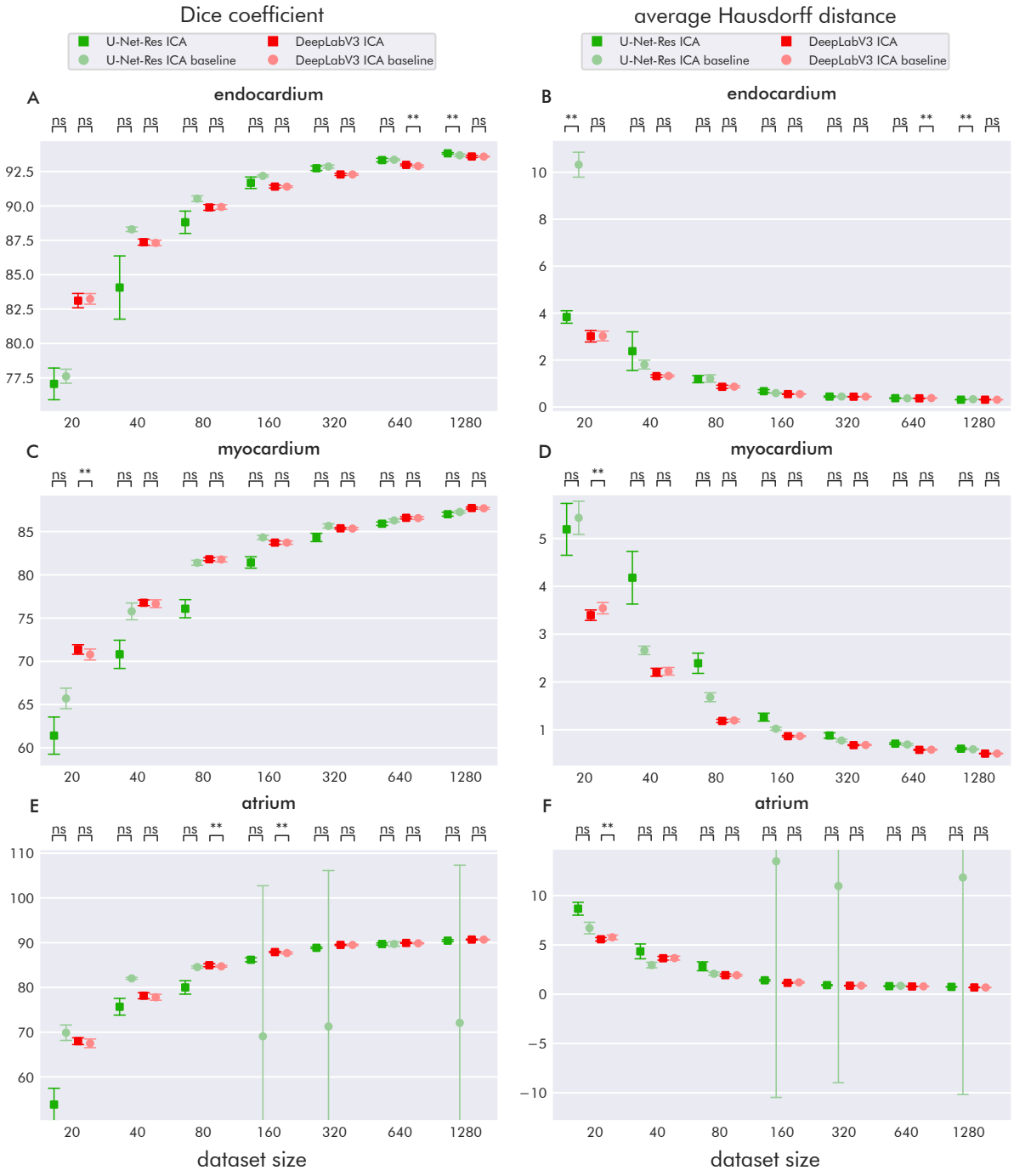

Figure 6.22: Cardiac segmentation resiöts using ICA shape priors. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

6.3.3 Neck Muscle Segmentation

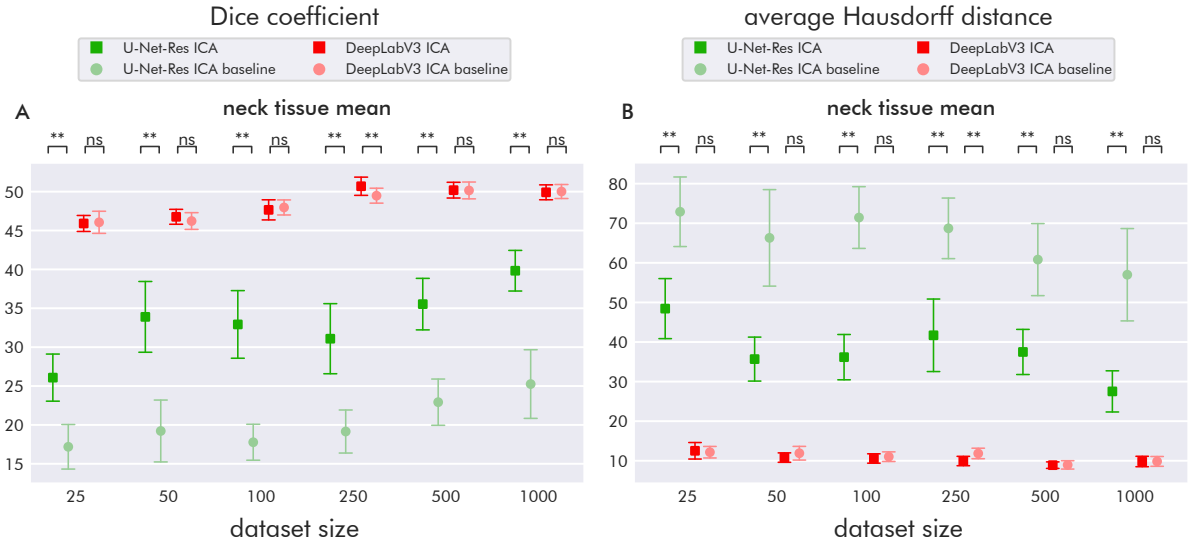

Figure 6.23: Neck muscle segmentation results using ICA shape priors. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

Figure 6.23 depicts the segmentation performance on the neck muscle dataset. Interestingly, the performance of U-Net-Res was largely improved. However, DeepLabV3 still greatly outperforms U-Net-Res. Except for a few cases, DeepLabV3 does not benefit from ICA shape priors. Comparing the ICA baseline results with the original baseline results (Subsection 6.1.4) reveals that a second parallel network branch is also not an appropriate tool for drawing additional information from this dataset.

6.3.4 Summary and Discussion

Summarizing, integrating ICA shape priors into CNNs as a secondary network branch is only partially beneficial. We want to discuss the results by answering **RQ 2.1** to **RQ 2.4** for ICA shape priors.

RQ 2.1: Which CNN architectures benefit from ICA shape priors?
We saw that only U-Net-Res benefited from ICA shape priors in the cases of lumen and neck muscle segmentation. The performance on cardiac segmentation even got worse for smaller datasets. The segmentation metrics by DeepLabV3 were hardly affected by using ICA shape priors. Reasons for this behavior are associated with certain error rates are discussed below in the answer to **RQ 2.4**. In the following, we limit our considerations to U-Net-Res and ignore DeepLabV3.

RQ 2.2: How do ICA shape priors perform as a function of dataset size?
In the case of lumen segmentation, the improvements are largest for the smallest dataset and decrease with increasing dataset size. For neck muscle segmentation, the improvements are rather constant across dataset size. Presumably, the tendency of U-Net-Res ICA to generate predictions with correct topology but less variation helps improve neck muscle segmentation

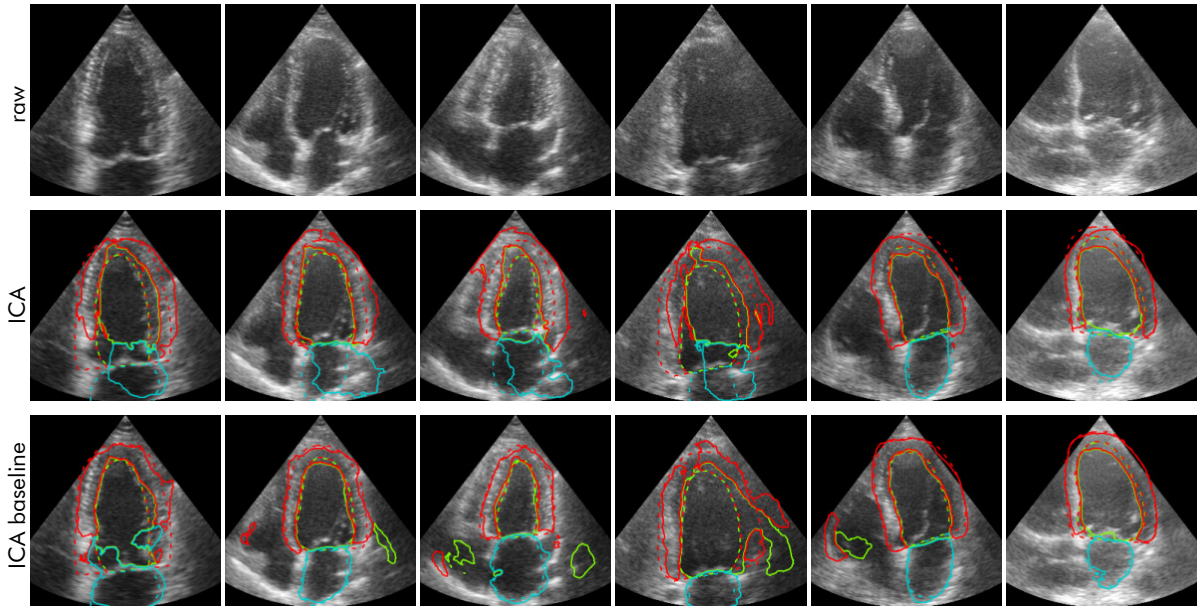

Figure 6.24: Comparison between ICA and ICA baseline for cardiac segmentation. ICA (row 2) reduces topological errors but tends to introduce some bias in edge alignment. This bias is less strong in the ICA baseline (row 3).

metrics. However, since a CNN that consistently produces more or less the same prediction does not provide any information, the results are still useless.

RQ 2.3: Which tissues benefit from ICA shape priors?
Lumen and neck muscle segmentation benefited from employing ICA shape priors. We discuss the reasons below in the answer of **RQ 2.4**.

RQ 2.4: What types of segmentation errors are reduced by ICA shape priors?
ICA shape priors helped reduce the error rates of discontinuous tubular structures (vessel wall and myocardium) and topological disorder (error 1). However, it seems that these improvements come with a cost. Although the error 3 rates were reduced, the corresponding vessel wall and myocardium metrics did not improve. In the case of the IVUS lumen and vessel wall dataset, only the lumen metrics by U-Net-Res improved. Since errors in lumen segmentation are not directly addressed with the presented error types, not observing correlations between lumen segmentation performance and error rates is quite reasonable. An exception could be error 1 (topological disorder), which addresses both the vessel wall and lumen. However, the question remains of why lumen segmentation performance increased for U-Net-Res ICA. Taking a look at cardiac segmentation provides additional insight to answer this question.

Although errors 1, 3, and 4 could be greatly reduced for cardiac segmentation, we do not observe any systematic improvements in the corresponding segmentation metrics, particularly in the case of U-Net-Res. This is particularly confusing regarding error 1, since reducing topological disorder was previously also associated with improving segmentation metrics. However, looking at some predicted segmentations, it becomes clear why this behavior occurs. It seems that ICA shape priors support the CNNs to generate correct topologies but also reduce the variability of possible positions, shapes, and orientations. Consequently, predicted

tissue boundaries move away from the ground truth boundaries. As a result, error rates assessing topology decrease, but metrics remain about the same. Figure 6.24 depicts some examples. This effect increases for smaller datasets. Since the number of ICs is quite small in these cases, linear combinations of the ICs do not allow for much variability in shape priors and, thus, attention maps. In this manner, the predicted segmentations become more rigid across images. This does also explain why U-Net-Res could improve the results of lumen segmentation. Since the variability of lumen shape across different images is comparatively small, the effect of shape priors becoming rigid does not have a very large impact. DeepLabV3 generally does not benefit much from ICA shape priors since it already produces images with only a few topological errors in its baseline version. We explained this matter of fact in Subsection 6.1.5.

If we compare the ICA baseline results with the original baseline results (Section 6.1), we see that adding a secondary branch without the ICs seems to have no benefit on segmentation performance at all. Sometimes, it even worsens the segmentation metrics, as in the case of the observed instabilities with atrium segmentation. Likely, the random but fixed surrogates of the ICs do not allow the ICA baseline CNNs to generate meaningful attention maps. An alternative to random tensors would have been constant ones. However, this would not have allowed the parallel branch to generate a spatially varying attention map since convolutions of a constant feature map would again result in a constant feature map (with small areas of variability at the edges due to zero padding). This shows that linear combinations of fixed feature maps with coefficients generated by latent code are only helpful if these feature maps comprise meaningful content with respect to the dataset. This is the case for our ICA shape prior.

Overall, an ICA shape prior proved to be beneficial for U-Net-Res and segmentation classes that do not vary too much across images, like lumen. It pushes the CNN towards generating a topologically correct prediction. Concurrently, it reduces the variability of position, shape, and orientation of the individual segmentation classes, especially with small datasets. ICA shape priors should therefore be used with caution.

6.4 Topological Constraints

This section presents the results of incorporating topological constraints via loss functions into CNNs. These loss functions are designed to make CNNs consider that certain tissues are surrounded by other tissues. For example, the lumen should always be completely surrounded by the vessel wall, and the endocardium should always be surrounded by the myocardium and atrium (see Subsection 4.2.3 for more details). The corresponding term in the total loss function, the containment loss, can be weighted differently depending on how strongly these constraints are to be enforced. We found that weights of 0.1 for the IVUS lumen and vessel wall dataset, as well as 0.01 for the cardiac dataset, led to the best performances. The weight of the Dice loss term was 1 for all experiments.

6.4.1 IVUS Lumen and Vessel Wall Segmentation

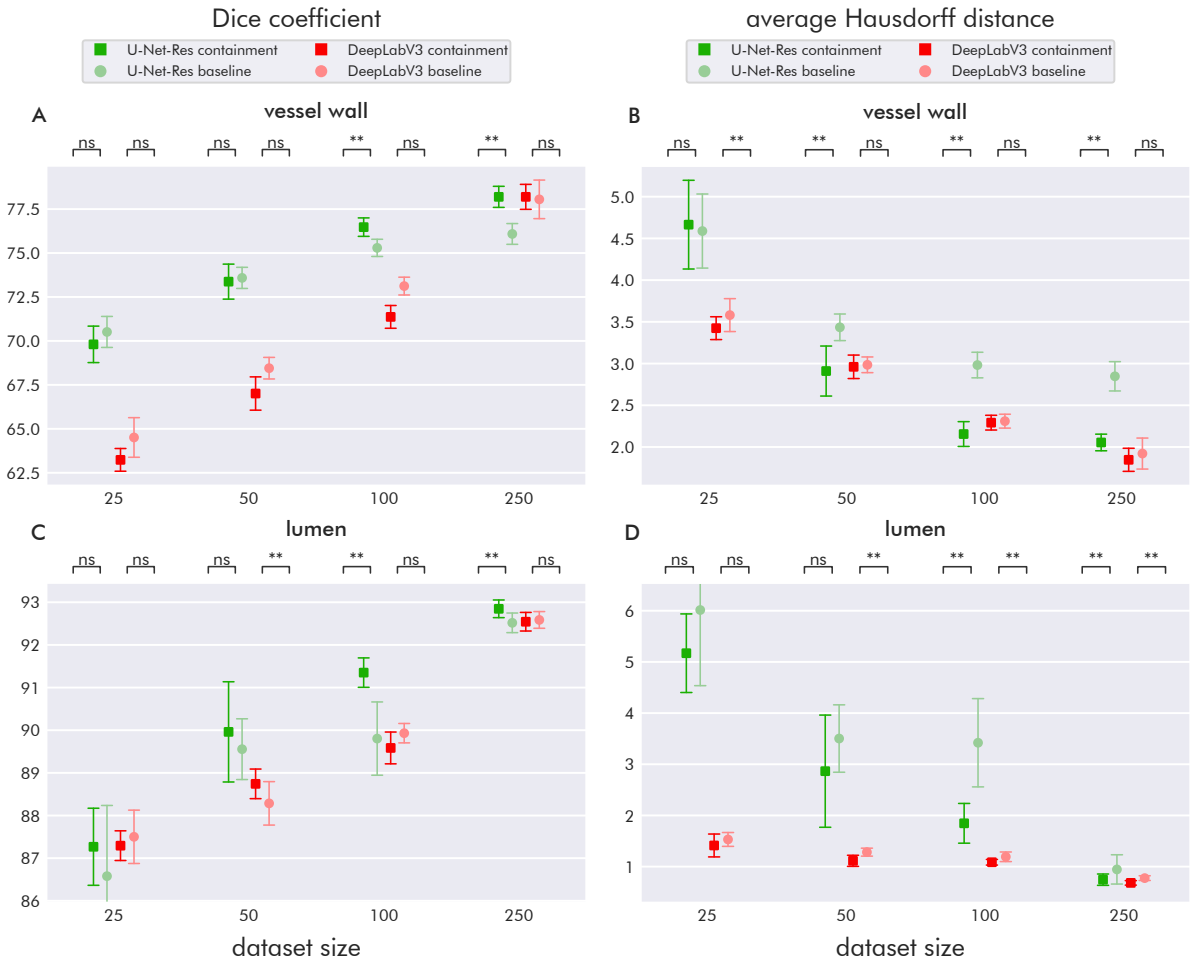

Figure 6.25: IVUS lumen and vessel wall segmentation results using the containment loss. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

The performance on the IVUS lumen and vessel wall dataset is shown in Figure 6.25. Here we can observe a different behavior of improvements compared to the previous methods: the improvements tend to increase with increasing dataset size. Primarily for U-Net-Res, but also

Table 6.9: IVUS lumen and vessel wall segmentation error rates using the containment loss. For each error, the rates achieved by the original baseline and by employing the containment loss are given, as well as the relative change of the latter compared to the former. The columns are divided by dataset size (25, 50, 100, and 250 images) and CNN architecture (U: U-Net-Res, D: DeepLabV3). Values are given as percentages.

		25		50		100		250	
error	**method**	U	D	U	D	U	D	U	D
error 1	**baseline**	2.7	0.7	2.0	0.7	1.3	0.0	0.7	0.0
	containment	2.7	0.0	0.7	0.0	0.0	0.0	0.0	0.0
	rel. change	**0.0**	**−100.0**	**−66.7**	**−100.0**	**−100.0**	—	**−100.0**	—
error 2	**baseline**	53.3	14.0	22.7	8.0	22.7	8.7	20.7	8.0
	containment	60.0	10.7	28.7	8.0	26.0	7.3	23.3	6.0
	rel. change	**12.5**	**−23.8**	**26.5**	**0.0**	**14.7**	**−15.4**	**12.9**	**−25.0**
error 3	**baseline**	16.0	2.0	12.7	4.7	9.3	4.0	9.3	0.7
	containment	4.7	0.7	3.3	0.0	2.7	0.0	2.7	0.0
	rel. change	**−70.8**	**−66.7**	**−73.7**	**−100.0**	**−71.4**	**−100.0**	**−71.4**	**−100.0**
error 4	**baseline**	69.3	51.3	54.7	33.3	47.3	31.3	43.3	24.0
	containment	64.0	49.3	58.0	40.0	44.7	31.3	38.7	27.3
	rel. change	**−7.7**	**−3.9**	**6.1**	**20.0**	**−5.6**	**0.0**	**−10.8**	**13.9**
error 5	**baseline**	23.3	25.3	21.3	26.7	20.0	18.7	16.7	10.0
	containment	20.7	19.3	14.7	16.7	14.7	10.7	14.7	8.0
	rel. change	**−11.4**	**−23.7**	**−31.3**	**−37.5**	**−26.7**	**−42.9**	**−12.0**	**−20.0**

for DeepLabV3 in the case of lumen segmentation with 50 training images. In the case of vessel wall Dice score, U-Net-Res with containment loss greatly outperforms both DeepLabV3 and can catch up with DeepLabV3 for 250 training images. Furthermore, the containment loss slightly decreases the vessel wall Dice score by DeepLabV3 on datasets smaller than 250 images. The average Hausdorff distance of vessel wall by U-Net-Res increases up to almost 50 % for dataset sizes of 100 and 250 images. The largest improvements in lumen segmentation are achieved for 100 training images. The other improvements are minor.

Table 6.9 reveals that the containment loss systematically decreases error 1 (topological disorder), error 3 (discontinuous vessel wall), and error 5 (background marked in lumen or vessel wall). This makes sense since these errors break the desired topology. This is especially the case for error 3. Furthermore, we see that the error 2 rate (incorrect patches) tends to suffer when applying topological constraints to U-Net-Res. However, the improvements in error rates are more or less distributed equally across dataset sizes. That does not correlate with the segmentation metrics improvements, which accumulate at larger dataset sizes.

6.4.2 Cardiac Segmentation

Figure 6.26 depicts the segmentation metrics of cardiac segmentation. As with the IVUS lumen and vessel wall dataset, statistically significant improvements primarily occur for endocardium and myocardium with larger datasets. The atrium metrics tend to worsen for smaller datasets when applying the containment loss.

If we take a look at the error frequencies in Table 6.10, we see that, as in the IVUS lumen and vessel wall dataset, incorrect patches (error 2) were increased. In contrast, discontinuous myocardia (error 3) were greatly reduced. Also, error 4 (myocardium around atrium) was reduced more extensively for DeepLabV3 than for U-Net-Res. Interestingly, error 5 (endo-

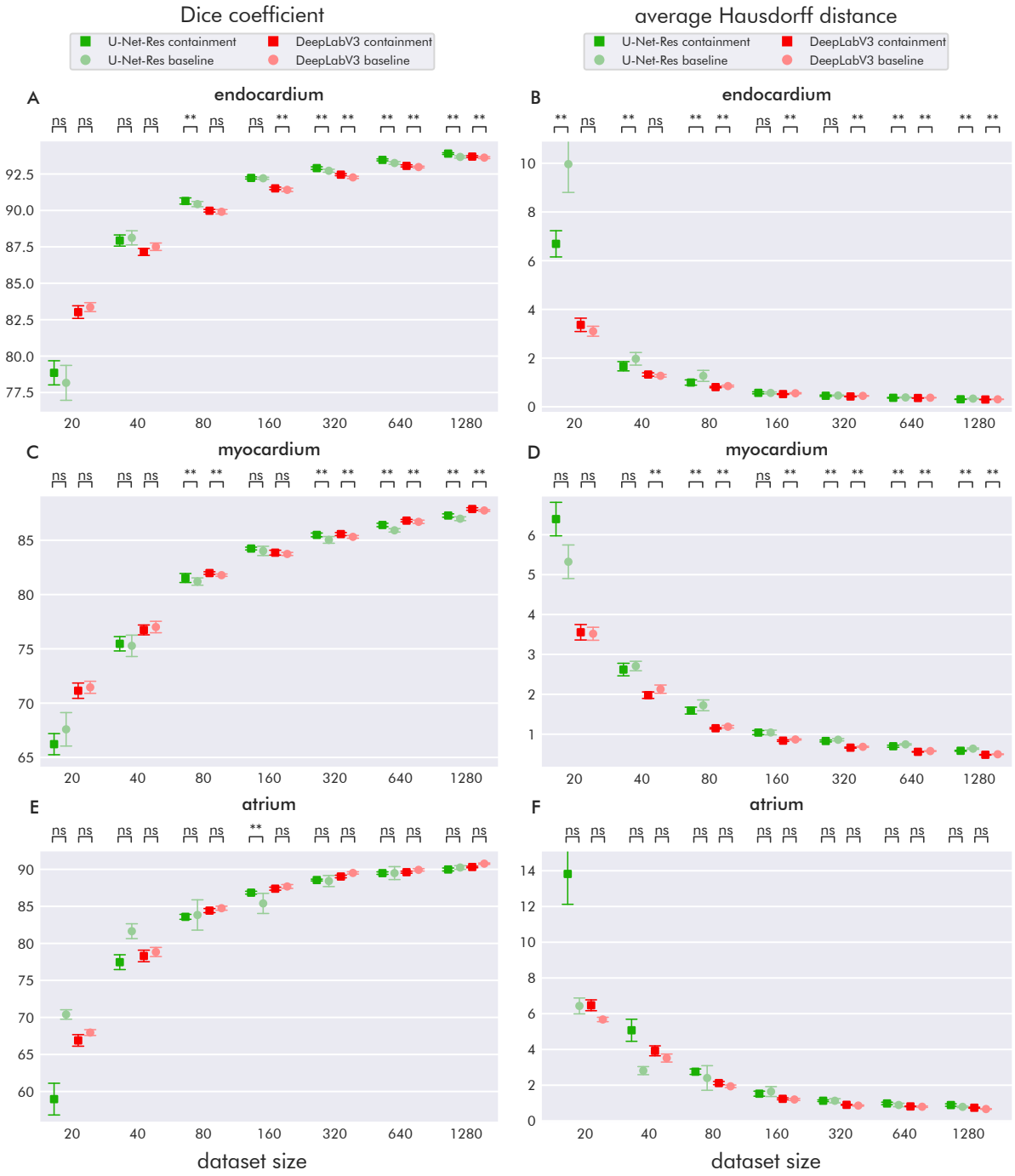

Figure 6.26: Cardiac segmentation results using the containment loss. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

Table 6.10: Cardiac segmentation error rates using the containment loss. For each error, the rates achieved by the original baseline and by employing the containment loss are given, as well as the relative change of the latter compared to the former. The columns are divided by dataset size (20, 40, 80, 160, 320, 640, and 1280 images) and CNN architecture (U: U-Net-Res, D: DeepLabV3). Values are given as percentages.

		20		40		80		160		320		640		1280	
error	**method**	U	D	U	D	U	D	U	D	U	D	U	D	U	D
error 1	**baseline**	26.0	0.0	12.6	0.0	7.2	0.0	3.8	0.0	1.2	0.0	0.6	0.0	0.0	0.0
	containment	20.8	0.0	14.2	0.0	7.8	0.0	3.0	0.0	1.8	0.0	0.4	0.0	0.0	0.0
	rel. change	**−20.0**	—	**12.7**	—	**8.3**	—	**−21.1**	—	**50.0**	—	**−33.3**	—	—	—
error 2	**baseline**	67.0	18.6	50.6	14.0	39.8	11.6	36.4	6.2	20.2	3.4	14.4	1.6	11.0	1.0
	containment	69.0	29.6	54.2	22.2	40.6	17.2	33.8	8.4	18.6	4.6	14.4	2.4	11.4	1.8
	rel. change	**3.0**	**59.1**	**7.1**	**58.6**	**2.0**	**48.3**	**−7.1**	**35.5**	**−7.9**	**35.3**	**0.0**	**50.0**	**3.6**	**80.0**
error 3	**baseline**	42.0	18.0	31.0	14.0	20.2	13.0	13.2	6.2	9.0	5.0	6.2	1.8	4.6	2.0
	containment	10.0	5.4	7.2	4.4	5.4	4.2	3.8	3.2	2.8	1.6	1.0	0.0	0.0	0.0
	rel. change	**−76.2**	**−70.0**	**−76.8**	**−68.6**	**−73.3**	**−67.7**	**−71.2**	**−48.4**	**−68.9**	**−68.0**	**−83.9**	**−100.0**	**−100.0**	**−100.0**
error 4	**baseline**	22.2	27.0	19.2	18.2	17.4	17.6	10.4	9.8	8.0	7.0	3.0	2.0	0.8	0.8
	containment	23.0	15.4	17.6	8.2	14.0	8.0	12.4	6.2	7.0	3.4	2.6	1.4	0.0	0.0
	rel. change	**3.6**	**−43.0**	**−8.3**	**−54.9**	**−19.5**	**−54.5**	**19.2**	**−36.7**	**−12.5**	**−51.4**	**−13.3**	**−30.0**	**−100.0**	**−100.0**
error 5	**baseline**	25.0	13.0	21.0	14.0	17.2	8.2	8.8	6.6	4.6	3.6	2.4	1.2	0.6	0.2
	containment	32.8	19.0	26.4	16.2	19.6	11.8	13.8	8.8	8.2	5.2	4.4	2.2	0.0	0.0
	rel. change	**31.2**	**46.2**	**25.7**	**15.7**	**14.0**	**43.9**	**56.8**	**33.3**	**78.3**	**44.4**	**83.3**	**83.3**	**−100.0**	**−100.0**

cardium and atrium confused) also increased with the containment loss up to dataset sizes of 640 images.

6.4.3 Summary and Discussion

We want to go straight into answering **RQ 2.1** to **RQ 2.4** for the containment loss.

RQ 2.1: Which CNN architectures benefit from ICA shape priors?
While only U-Net-Res tends to benefit from the containment loss on the IVUS lumen and vessel wall dataset, both U-Net-Res and DeepLabV3 may benefit for cardiac segmentation.

RQ 2.2: How do ICA shape priors perform as a function of dataset size?
The containment loss is the first and, to anticipate a little, also the only method that leads to improvements only for larger datasets. This behavior seems rather odd since we expected domain knowledge to only provide additional information if the convolutional filters are inefficient due to less training data. However, we can solve this puzzle by looking at the predicted segmentation masks. For large training datasets, most of the test images are segmented quite well. However, some test images have very poor quality, usually due to low contrast or artifacts producing large shadows. These images are prone to be segmented with discontinuities in the vessel wall or myocardium. The containment loss helps to close these discontinuities leading to improved metrics for these tissues (see Figure 6.27 and Figure 6.28). At the same time, the segmentation metrics of the tissues surrounded by the tubular structures can often benefit as well.

RQ 2.3: Which tissues benefit from ICA shape priors?
As stated in the answer to **RQ 2.2**, tissues with tubular morphology benefit the most from topological constraints, mainly in terms of error rate reduction. In the case of IVUS lumen and vessel wall segmentation, a reduction of error rate 3 also led to improved segmentation

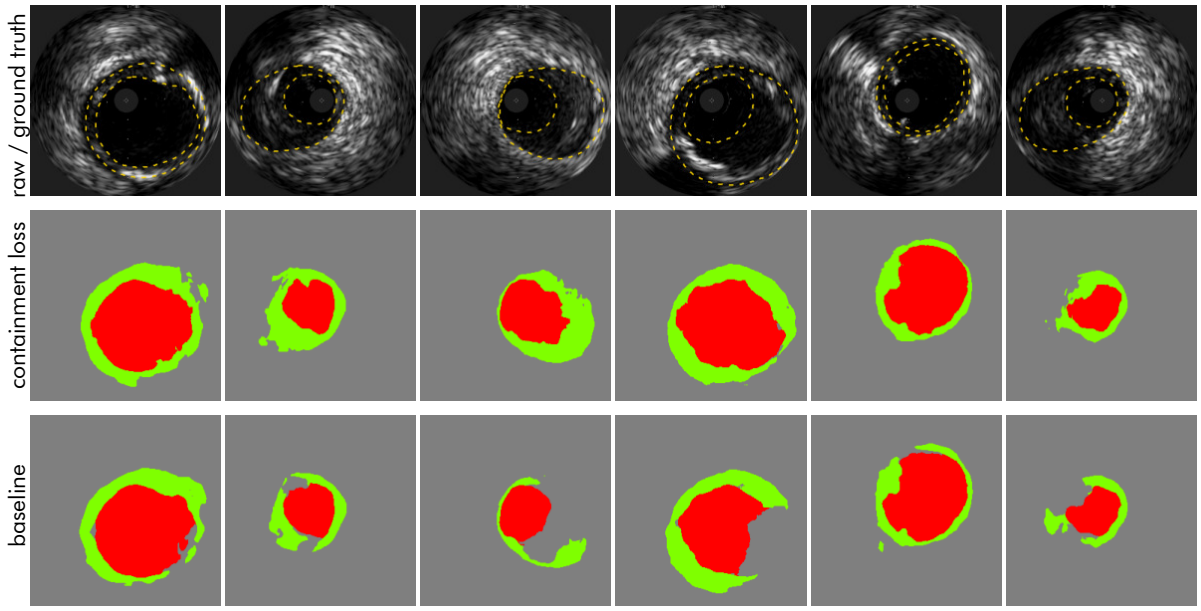

Figure 6.27: The effect of containment loss on vessel topology. The containment loss helps to predict continuous vessel walls (green).

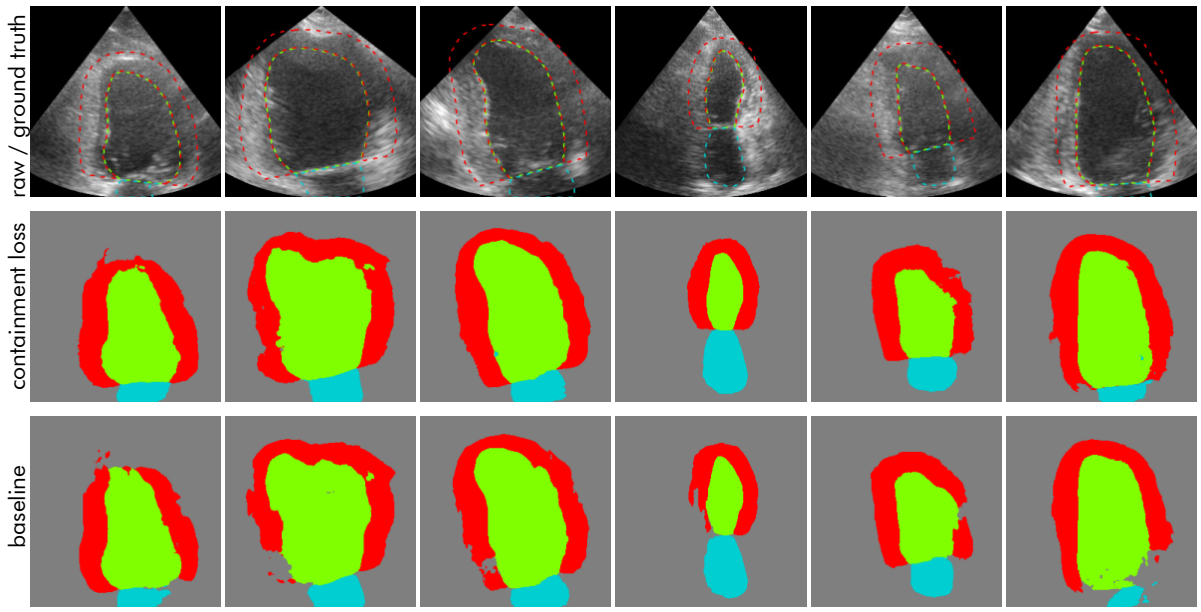

Figure 6.28: The effect of containment loss on cardiac topology. The containment loss helps to predict continuous myocardia (red).

metrics. The contrary happened for cardiac segmentation. Here, the segmentation metrics got worse for small datasets.

RQ 2.4: What types of segmentation errors are reduced by ICA shape priors?
The statements in the answer to **RQ 2.2** are also reflected by the very low error 3 frequencies in Table 6.9 and Table 6.10, which vanish entirely in many cases for larger datasets. Unfortunately, the containment loss tends to increase incorrect patches (error 2) and reduces atrium segmentation performance. When looking at Figure 6.27, it becomes clear that the containment loss tends to generate more rough and frayed boundaries, which also promotes incorrect patches. However, these patches are rather small and do not seem to affect the segmentation metrics very much.

All in all, the containment loss improved topological stability when facing samples rather off the training domain. This effect was larger for U-Net-Res because the baseline DeepLabV3 already tends to generate segmentation masks with correct topology (see 6.1).

6.5 Synthetic Data Generation with GANs

Parts of this section have been published in Bargsten and Schlaefer [22].

We presented the GAN architecture used in our experiments in Section 4.3. Unlike our paper on speckleGAN [22], we now used a multi-scale discriminator instead of a single-scale discriminator. This leads to different results and conclusions, which we will discuss later.

For training the GANs, we used the basic GAN loss functions (Section 3.4). Since we employed a multi-scale discriminator with two stages, the loss functions were evaluated individually for both discriminators and then summed to obtain the total loss function.

For validating and testing the GAN models, we employed the Fréchet Inception Distance (FID) developed by Heusel et al. [102]. It measures the distance between the synthetic and real image data distribution by combining mean values and covariance matrices of network activations obtained by feeding both image sets into an Inception-v3 model [249], which was pre-trained on the ImageNet dataset [57]. Typically, activations of the penultimate network layer are used to calculate the FID score:

$$\mathrm{FID} = \|\mu_1 - \mu_2\|_2^2 + \mathrm{Tr}(C_1 + C_2 - 2(C_1 C_2)^{1/2}). \tag{6.1}$$

Here, μ_1 and μ_2 are the mean vectors, and C_1 and C_2 are the corresponding covariance matrices of the activations. Small FID scores and, thus, small distances between the image data distributions indicate visual similarity of the image sets, as well as diversity of the generated image set, meaning that mode collapse was prevented. It has not been proven so far that low FID scores induce high image quality when applied to medical images. However, previous work indicates correlation between FID score and realistic appearance of generated medical images [169, 263]. Since the FID score is a distance metric, smaller values are better.

For defining a baseline GAN, the speckle layer was replaced with an identity mapping (cyan-colored box in the generator sketch of Figure 4.14). Everything else remained the same. Via preliminary experiments we determined how many epochs were needed to saturate training (with respect to FID score) for all datasets and dataset sizes. We found values between 400 and 1500 epochs for speckleGAN and values between 400 and 6000 epochs for the baseline GAN. During training, conventional data augmentation via image transformations was performed according to Section 6.1.

Validation was performed about 40 times during training, so the interval depended on the number of training epochs. For validating a GAN, 500 synthetic images were generated with conventionally augmented masks from the training set. Afterwards, the FID score between these synthetic images and at least 500 conventionally augmented real training images was calculated. To reach the required amount of masks and images in cases of smaller training datasets, the training datasets were conventionally augmented randomly multiple times. The model checkpoint with the best validation performance was used for testing. To test a GAN, 2000 synthetic images were generated by conditioning the GAN on conventionally augmented training masks. Afterwards, the FID score between these synthetic images and 2000 conventionally augmented images from the test set was calculated. Again, to obtain the required amount of images, the test and training set were augmented randomly multiple times.

As explained previously, the FID score does not completely ensure reliability when used to evaluate realism of medical image sets. In order to further assess the quality of the synthetic images, we calculated two more metrics: The Jensen-Shannon divergence between gray

value distributions of different segmentation classes in ground-truth and synthetic images and the structural similarity (SSIM) index [277] between corresponding ground-truth and synthetic images. The Jensen-Shannon divergence measures the similarity between two probability distributions P and Q and is defined as

$$D_{JS}(P||Q) = \frac{1}{2}\left(D_{KL}(P||M) + D_{KL}(Q||M)\right) \tag{6.2}$$

with $M = \frac{1}{2}(P+Q)$ being the pointwise mean of both distributions P and Q and D_{KL} being the Kullback-Leibler divergence. In the case of image comparison, P and Q are obtained by normalizing the corresponding gray value histograms of both images that are to be compared. For discrete distributions, the Kullback-Leibler divergence is defined as

$$D_{KL}(P||Q) = -\sum_{x} P(x)\,\log\frac{Q(x)}{P(x)} \tag{6.3}$$

We used the Jensen-Shannon divergence to compare the gray value histograms of the real and the synthetic dataset as a whole, not of individual images. Since the Jensen-Shannon divergence is a distance metric, smaller values are better.

The SSIM can be interpreted as the product of luminance similarity, contrast similarity and structure similarity (basically correlation) and takes values between 0 (no similarity) and 1 (identical). The SSIM of two images, x and y, is defined as

$$\mathrm{SSIM}(x,y) = \frac{(2\,\mu_x\,\mu_y + c_1)(2\,\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{6.4}$$

with μ_x and μ_y being the mean values of images x and y, σ_x^2 and σ_y^2 being the variances of images x and y and σ_{xy} being the correlation coefficient of images x and y. The small constants c_1 and c_2 prevent the denominator from becoming 0. To get a single value for testing, the individual SSIM values were simply averaged. Since the SSIM measures similarity, larger values are better.

We used a batch size of 12 and set the learning rates to 10^{-4} for the generator and $4 \cdot 10^{-4}$ for the discriminator.

6.5.1 IVUS Lumen and Vessel Wall Image Generation

Figure 6.29 depicts the results of generating synthetic IVUS lumen and vessel wall image data. We can see that the synthetic speckleGAN dataset achieves far better FID scores, for small datasets in particular. However, the SSIM values obtained by the baseline GAN tend to be slightly higher (up to about 2.3 %) compared to the values by speckleGAN. We cannot see any differences regarding the Jensen-Shannon divergence. Nevertheless, the variances of speckleGAN are smaller.

Exemplary synthetic images for all training dataset sizes are shown in Figure 6.30. Remarkably, the visually assessed image quality does not seem to change with increasing dataset size for both GANs. When we compare the images of both GANs, we can see that the speckle noise in images of the baseline GAN tends to be rather connected and wavy. Especially in images with larger lumen diameter. The speckle noise in images of speckleGAN, on the other hand, is much clearer and distinct.

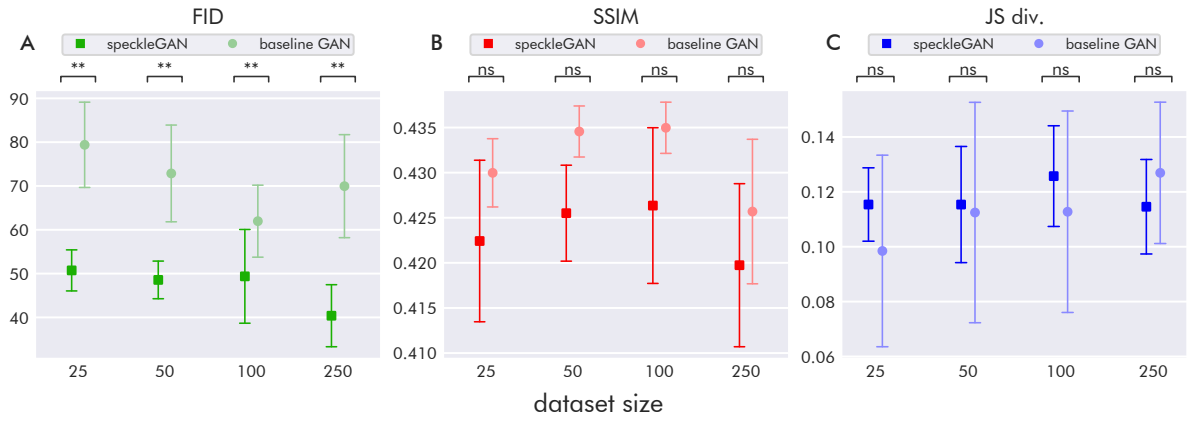

Figure 6.29: Synthetic image generation results regarding the IVUS lumen and vessel wall dataset. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

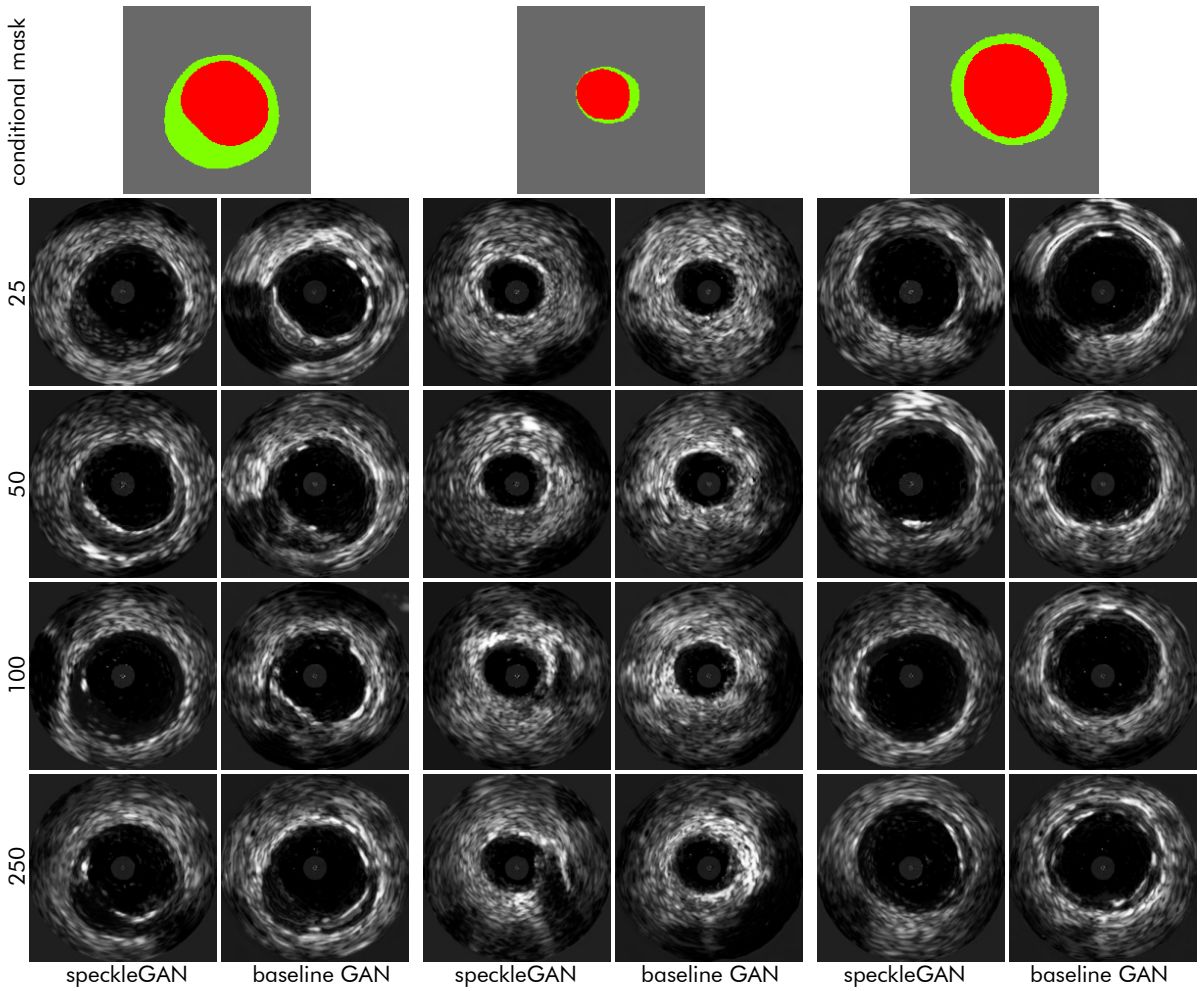

Figure 6.30: Exemplary synthetic images based on the IVUS lumen and vessel wall dataset. The first row depicts the conditional segmentation masks, the other rows denote dataset size. The columns denote GAN architecture.

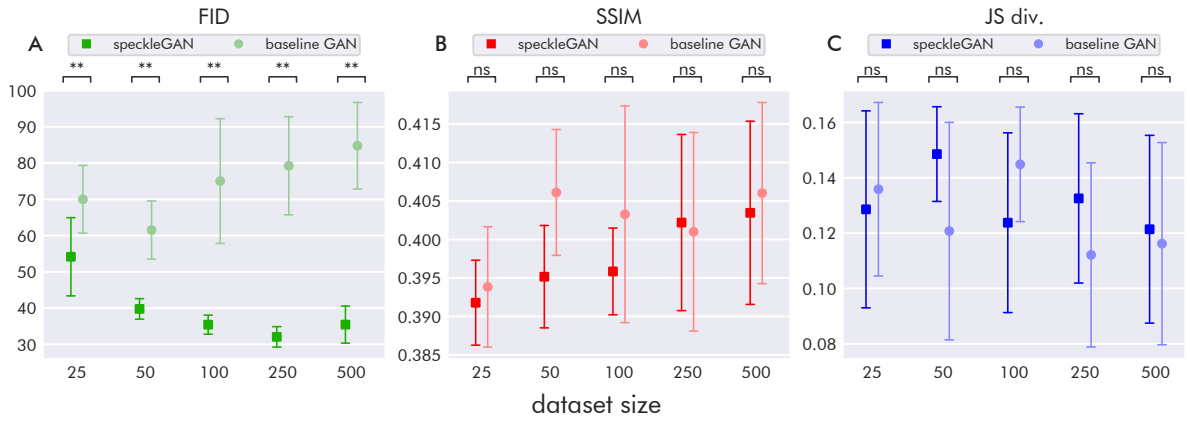

Figure 6.31: Synthetic image generation results regarding the IVUS calcium dataset. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

6.5.2 IVUS Calcium Image Generation

The results of synthetic IVUS calcium data generation are shown in Figure 6.31. As with the previous dataset, speckleGAN greatly outperformed the baseline GAN in terms of FID score. For speckleGAN, the FID score decreases with increasing dataset size, while the FID score achieved by the baseline GAN tends to increase with larger amount of training data. SSIM and Jensen-Shannon divergence are quite constant across dataset sizes do not differ substantially between speckleGAN and the baseline GAN.

Figure 6.32 depicts exemplary synthetic images of both GANs and for varying dataset size. As with the other IVUS dataset, the visually assessed image quality does not seem to change with larger amounts of training data. Again, the speckle noise in images of speckleGAN appears much clearer and distinct. As the lumen and vessel wall shape was not provided as conditional topological data, their geometry tends to fit less for smaller datasets. Also, the usual appearance of shadows behind calcifications is not always considered, for smaller training sets as for larger ones. However, areas with calcium almost always appear bright, as expected.

6.5.3 Cardiac Image Generation

Figure 6.33 depicts the results of cardiac image data generation. Contrary to the results on the IVUS datasets, speckleGAN does not outperform the baseline GAN with respect to the FID score. The baseline GAN even slightly outperforms speckleGAN for medium-sized datasets. The same holds for the SSIM. Interestingly, the FID score variance of the baseline GAN increases for the largest datasets. Both metrics improve with larger datasets and saturate at about 160 training images. Regarding the Jensen-Shannon divergence, the baseline GAN clearly obtains better values for smaller datasets. However, the variances by speckleGAN tend to be much smaller.

Exemplary images of both GANs are shown in Figure 6.34. Opposed to the IVUS datasets, we now can see quite an improvement in visual image qulity with increasing training dataset size. Nevertheless, even for large datasets, we see incorrect wavy shapes of the image sector which usually occurs when the network was trained with elastically transformed images. Compared to the IVUS datasets, the speckle noise generated by speckleGAN and the baseline GAN

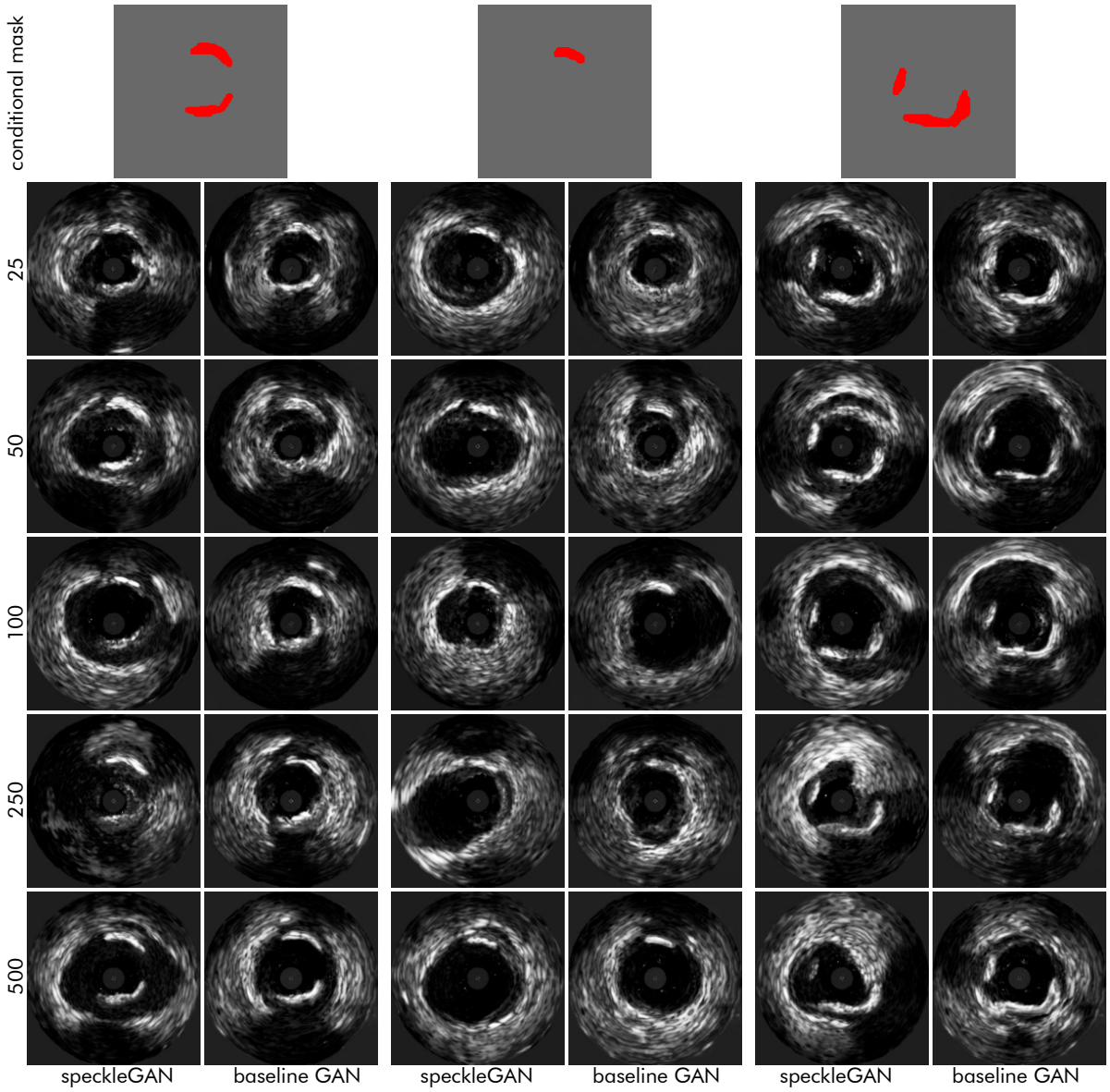

Figure 6.32: Exemplary synthetic images based on the IVUS calcium dataset. The first row depicts the conditional segmentation masks, the other rows denote dataset size. The columns denote GAN architecture.

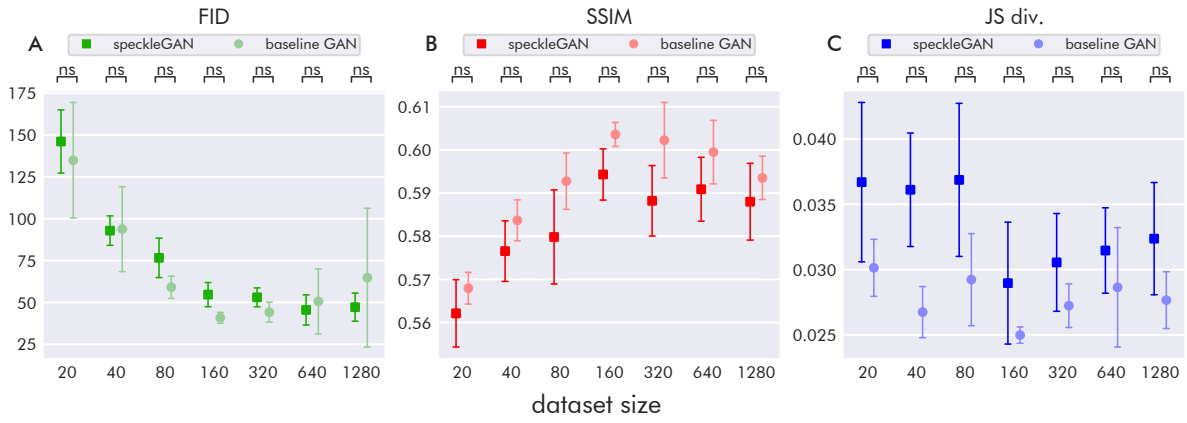

Figure 6.33: Synthetic image generation results regarding the cardiac dataset. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

do not show severe differences in terms of quality and realistic appearance. This is mainly because the speckles in the cardiac dataset are much smaller and less distinct. Moreover, we observe clearly visible borders of the conditional mask input in images by speckleGAN, particularly for smaller datasets. These are also present in images by the baseline GAN, but not very noticeable. For both GANs, these artificial looking shapes tend to fade with larger datasets but are still slightly visible in speckleGAN images.

6.5.4 Neck Muscle Image Generation

The results of synthetic neck muscle image generation are shown in Figure 6.35. The FID scores achieved by baseline GAN are slightly better than the ones by speckleGAN. The same is true for most of the SSIM values. The results in terms of the Jensen-Shannon divergence are rather balanced. The FID scores are larger compared to the other three datasets, indicating poorer variability and quality. While the SSIM for the previous datasets reached values up to about 0.42 (IVUS) and 0.6 (cardiac), here, it reaches values larger than 0.7. The Jensen-Shannon divergence, however, reaches values much larger compared to the IVUS datasets (about 0.12) and the cardiac dataset (0.04 and smaller). This means that for the neck muscle dataset, the luminance similarity, contrast similarity, and structure similarity between synthetic and real images are greater compared to the other datasets, while the gray value histograms fit worse.

Figure 6.36 depicts some exemplary images of both GANs for different amounts of training data. In general, it is hard to assess the images in terms of quality and realistic appearance. Little seems to change with increased dataset size. As with the cardiac dataset, we observe clearly visible borders of the conditional mask input in images of both GANs, which fade for larger dataset sizes. Since speckle noise is not very pronounced in this dataset, we cannot make any statements regarding its quality in the synthetic images.

6.5.5 Summary and Discussion

To summarize, speckleGAN provides the largest benefits for the IVUS datasets. These benefits take shape in substantially smaller FID scores, also for small datasets. The SSIM and the Jensen-Shannon divergence are not really affected. However, it was hard to spot any difference between speckleGAN and the baseline GAN with respect to the visual appearance of the

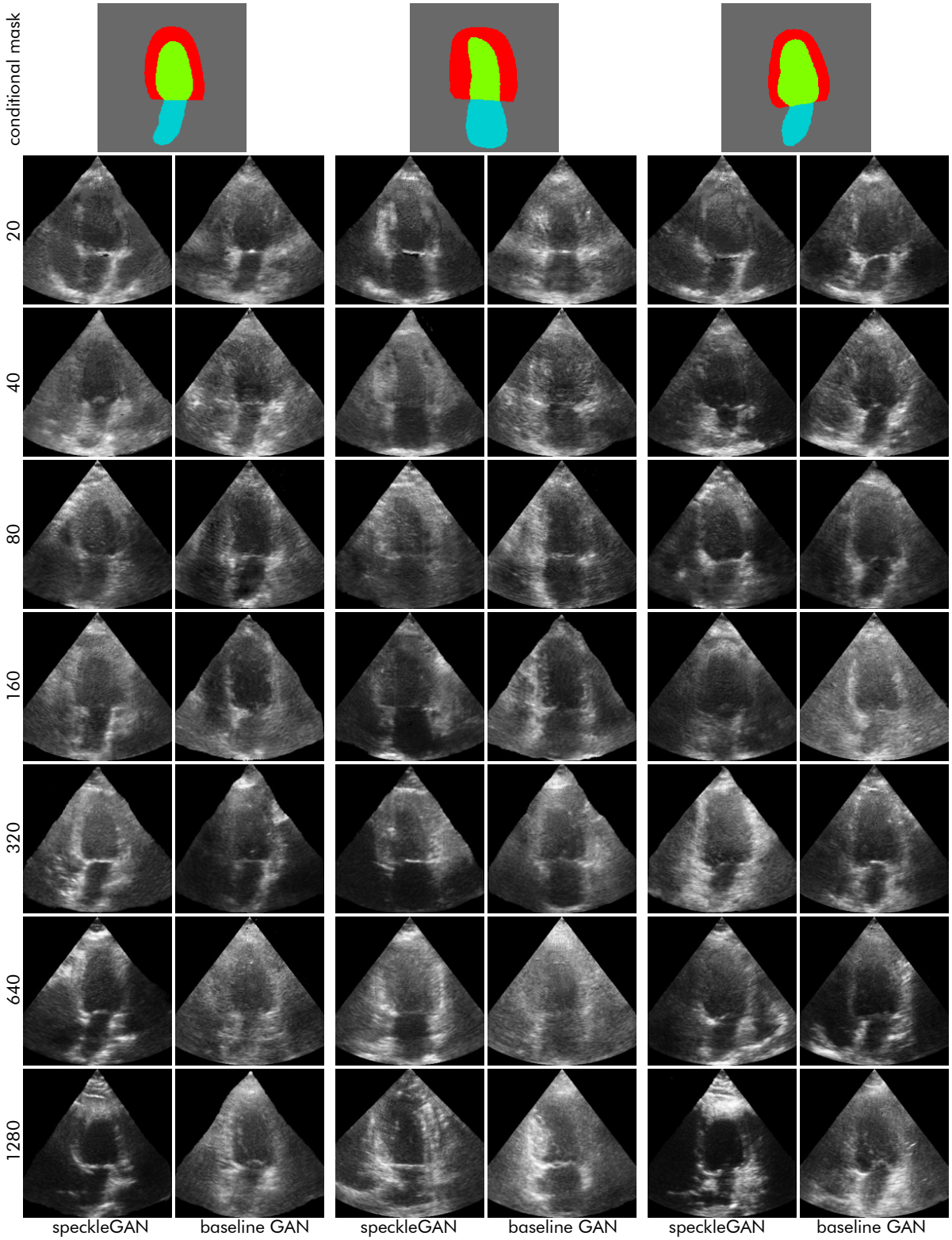

Figure 6.34: Exemplary synthetic images based on the cardiac dataset. The first row depicts the conditional segmentation masks, the other rows denote dataset size. The columns denote GAN architecture.

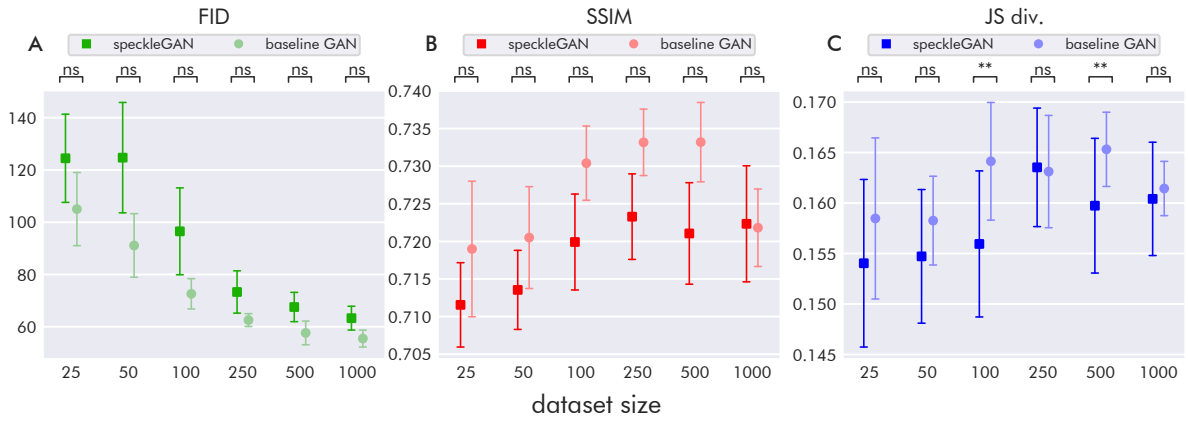

Figure 6.35: Synthetic image generation results regarding the neck muscle dataset. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

generated images. Moreover, the visually assessed image quality did not seem to improve with more training data, although the FID did improve. How can this be explained? We assume that this behavior is due to mode collapse and that FID measures not only image quality but also dataset diversity. In Section 4.3 we explained that mode collapse leads to the GAN generating images from only a few modes of the data distribution. This means that some patterns do not change across the synthetic images leading to a large FID score. And indeed, mode collapse occurs for the baseline GAN in almost all trainings. We can comprehend this with help of Figure 6.37. We can see three exemplary synthetic images from speckleGAN and the baseline GAN for the IVUS lumen and vessel wall, as well as the IVUS calcium dataset. The images are chosen such that the conditional masks do not alter very much. If we look closely, we notice that the speckle noise in baseline GAN images does almost not change, in contrast to the images by speckleGAN. To make this even clearer, we have plotted the mean images of the complete synthetic datasets in the last column. The mean images of the synthetic datasets by speckleGAN look blurry, as expected when taking the mean of more than a thousand images with no correlation in texture. The mean images of the synthetic datasets by the baseline GAN, on the other hand, show a distinct speckle pattern. The same one that can be observed in the corresponding exemplary images. However, the speckle pattern can alter in specific regions like the top right corner of the images of the IVUS lumen and vessel wall dataset, where the distance to the image edges is small. But these sparsely occurring effects do not have a large impact on the mean image. All that shows, that the baseline GAN suffers from mode collapse which manifests in generating virtually just a single speckle pattern for all synthetic images. We call this phenomenon speckle mode collapse.

Figure 6.38 shows corresponding images of the cardiac and neck muscle dataset. In the case of the cardiac images, we again see that the mean image of the synthetic dataset by the baseline GAN exhibits a certain speckle pattern, albeit not as clear and distinct as for the IVUS datasets. If one looks carefully, some of these speckles can also be observed in the exemplary images. Again, the mean image of the synthetic dataset by speckleGAN does not show a certain speckle pattern but only blurry regions as expected. The same basically holds for the neck muscle dataset. However, it is hard to tell if the bright spots in the mean image of the baseline GAN dataset really represent speckles. In any case, they are accumulation points and therefore indicate a (potentially mild) occurrence of mode collapse.

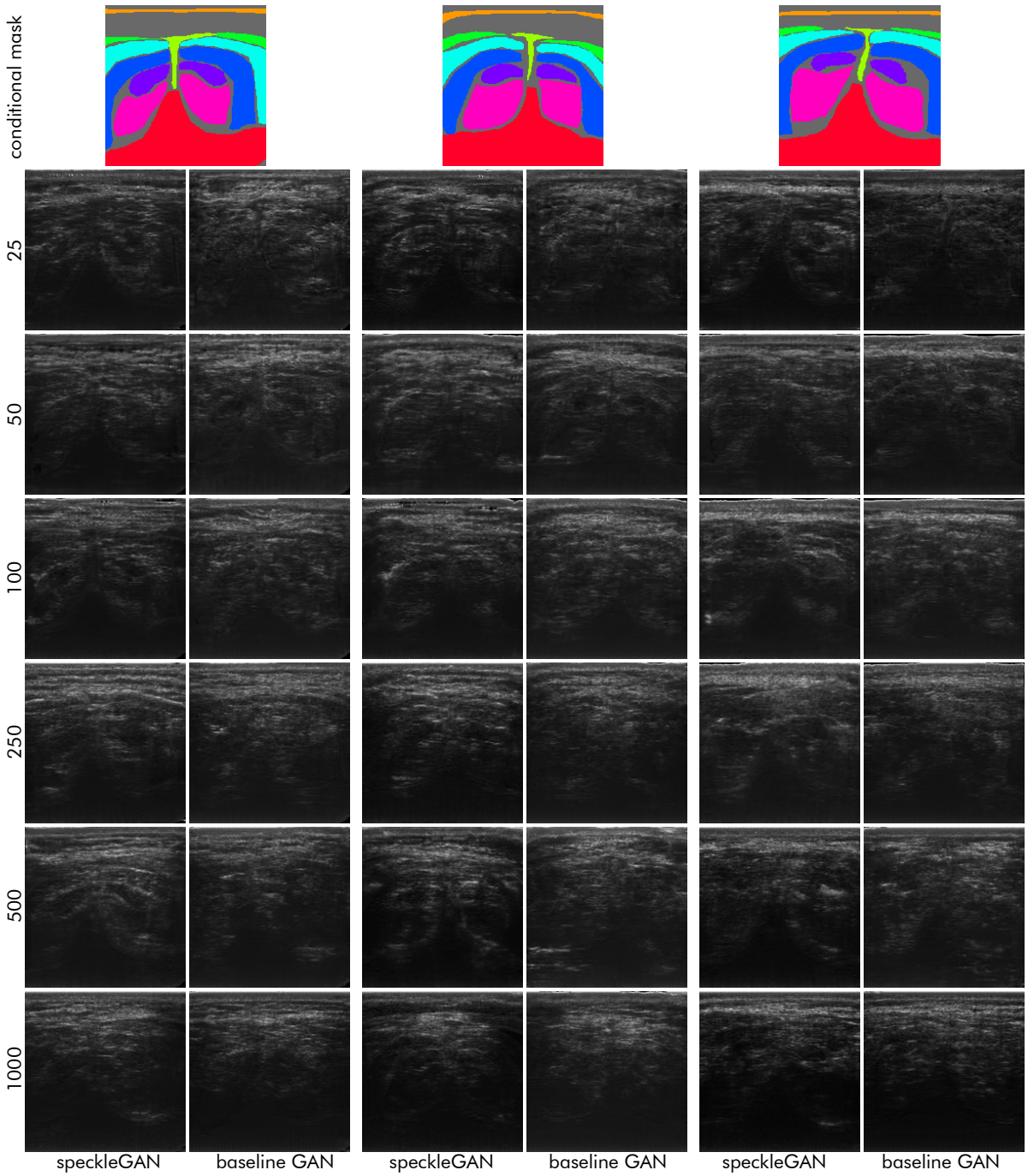

Figure 6.36: Exemplary synthetic images based on the neck muscle dataset. The first row depicts the conditional segmentation masks, the other rows denote dataset size. The columns denote GAN architecture.

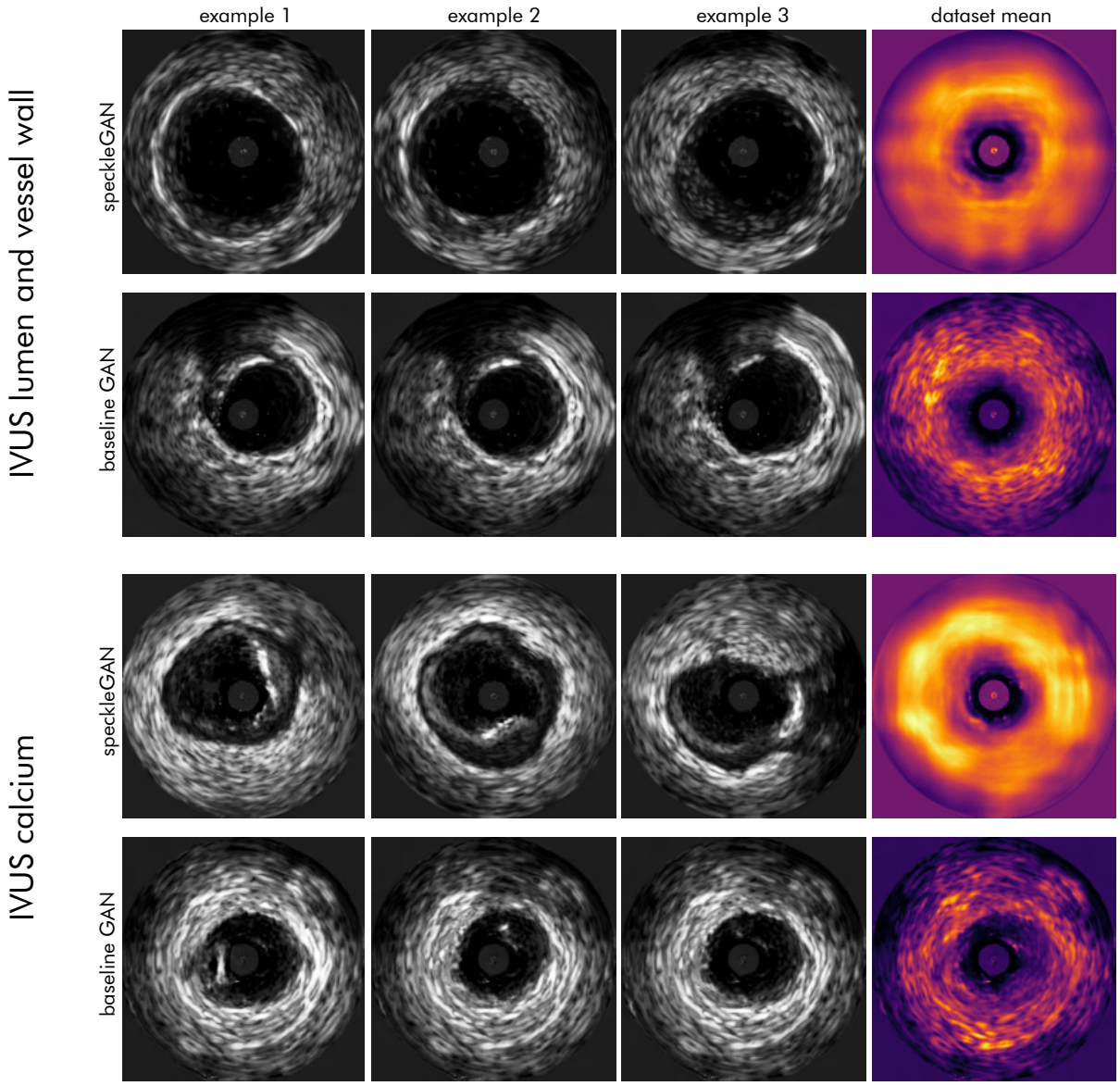

Figure 6.37: Comparison between speckleGAN and the baseline GAN in terms of speckle mode collapse on the IVUS lumen and vessel wall dataset and the IVUS calcium dataset. The last column shows the mean images of the complete synthetic datasets. While the mean images by speckleGAN appear blurry, the mean images by the baseline GAN show a clear speckle pattern. This indicates that the baseline GAN suffers mode collapse and can only generate a single speckle pattern.

All preceding mean images were generated from synthetic datasets by GANs trained on the smallest datasets. Figure 6.39 shows the mean images of synthetic datasets by GANs trained on all datasets and dataset sizes (smallest at the top, largest at the bottom). We can see that the described effects on both the smallest IVUS datasets occur for all dataset sizes. This means that the baseline GAN suffers from mode collapse even when trained with 500 images. However, we should remember that GANs for natural color images are usually trained with far more than 10,000 images. In the case of the cardiac dataset, we see that speckle mode collapse of the baseline GAN decreases with larger dataset sizes but does not vanish completely. It rather seems to increase slightly for 1280 training images. Changes in the neck muscle mean images are not visible when varying the dataset size. Nevertheless, the baseline GAN images show more accumulation points and thus indicate mode collapse.

All that explains why speckleGAN outperforms the baseline GAN on both IVUS datasets in terms of FID score. It does not outperform the baseline GAN in terms of SSIM and Jensen-Shannon divergence since these two do not take mode collapse into account. Hence, speckleGAN has no advantage over the baseline GAN regarding luminance, contrast, and structure similarity, as well as histogram similarity. It will be interesting to see how these properties affect the segmentation performance when using the synthetic images for data augmentation (see Section 6.6).

In the case of the cardiac dataset, the relatively mild mode collapse of the baseline GAN does not seem to have a large effect on the FID score. Moreover, the FID score improves as the SSIM improves. This indicates that improved similarity to the real dataset does also result in a better FID score, which makes sense, since the FID does not only measure dataset variability but also image quality. We can therefore assume that speckleGANs power to add random speckle to the generators feature maps is only of advantage if the speckles are large enough (as in the IVUS datasets) and thus have a large impact on the FID score. The additional capacity that speckleGAN gains from not having to create speckle noise with multiple convolutional layers does not seem to allow it to generate datasets with more variability in terms of image structure and composition. Nevertheless, since the baseline GAN suffers from speckle mode collapse even for the largest IVUS datasets, we suppose that generating variable speckle in GANs is not easily accomplished. This becomes clear if we look at the typical architecture of a GAN generator (see Section 4.3). Image variability is induced by the random seed and the conditional mask. It would make sense to assume that random speckles should be affected by the random seed. The random seed mainly affects the first few layers of the generator. Here, the feature maps are still quite small and determine rather large-scale image contents. However, speckle is a small-scale phenomenon and is therefore created in the last layers of the generator. Here, the influence of the random seed is only very small, which leads to the inability of the speckle GAN to generate variable speckle noise across images. SpeckleGAN does not have this problem since random speckle is generated actively with the speckle layer.

The last topic we want to mention is the increased FID score of the baseline GAN for the larger cardiac datasets. Not only do the mean values increase but also the variances. This behavior is caused by some unstable GAN trainings that led to comparatively poor results. It is not uncommon for a GAN to have a hard start to training due to an unfavorable initialization of its weights. It is easier for the GAN to get back on track if the dataset is small because it does not have to learn a large variability of textures, structures, and compositions, as would be the case for very large datasets. When training with large datasets, it is possible that the GAN

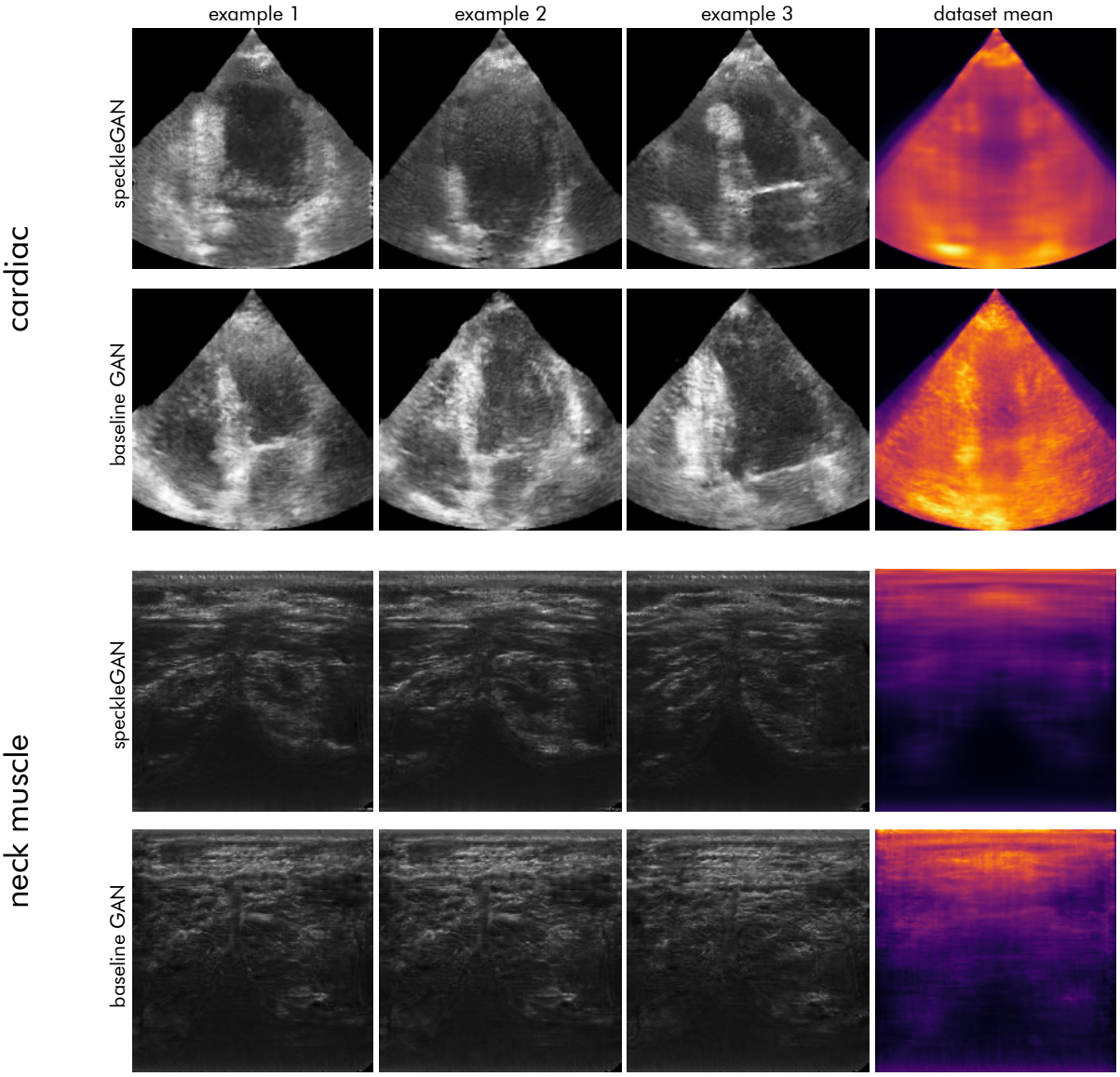

Figure 6.38: Comparison between speckleGAN and the baseline GAN in terms of speckle mode collapse on the cardiac and neck muscle datasets. The last column shows the mean images of the complete synthetic datasets. The mean cardiac image by speckleGAN appears blurry, and the corresponding mean image by the baseline GAN shows a speckle pattern. However, the speckles are not as clear as compared to the IVUS datasets. Hence, mode collapse is less severe on the cardiac dataset. Since the neck muscle images do not exhibit crisp speckle noise, we cannot directly observe speckle mode collapse.

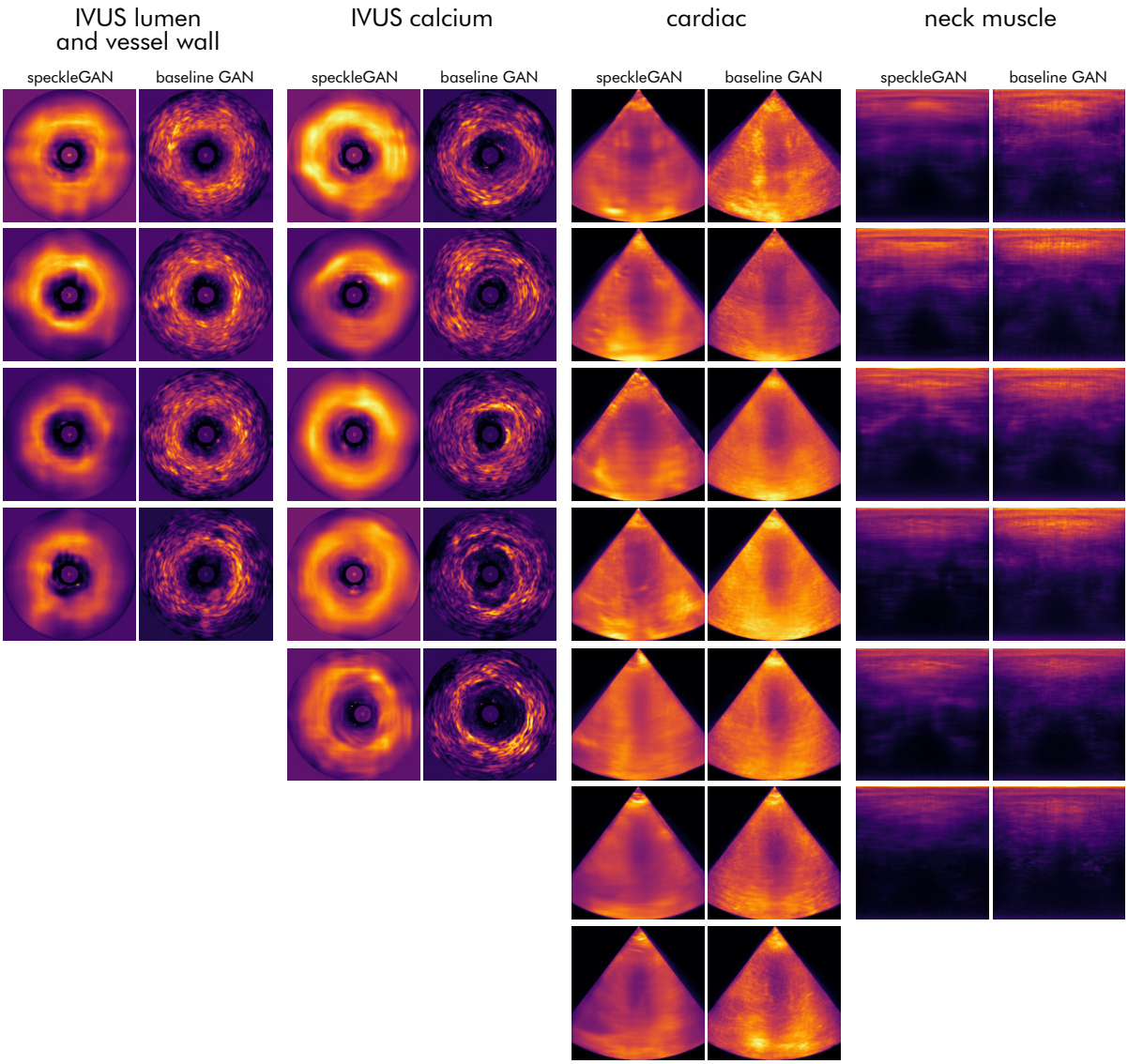

Figure 6.39: Mean images of all synthetic datasets for all training dataset sizes. Rows denote dataset size with larger sizes towards the bottom. The images show that speckleGAN does not suffer speckle mode collapse, while the baseline GAN does, especially on the IVUS datasets.

training recovers, too, but it would take very much time and does usually happen only after saturating for many epochs. SpeckleGAN does not seem to have this problem. An explanation for this could be the realistic speckle noise in the images since the first training iteration. These could lead to large training signals for both the generator and discriminator. The GAN can thus be pushed fast out of unfavorable regions of the loss landscape. We hypothesize that this is why speckleGAN requires much fewer training epochs compared to the baseline GAN. This effect is largest for smaller datasets and vanishes for larger ones. We assume this is the case because, for larger datasets, the richness of different structures and geometries is crucial for the number of epochs required for saturation of training. The effect of added speckle at the beginning of training is then negligible regarding the required number of epochs.

Furthermore, we want to comment on why the baseline GAN in our paper [22] performed significantly worse in terms of image quality than the baseline GAN in this work. The main reason is the multi-scale discriminator that we employed for this work. The discriminator in Bargsten and Schlaefer [22] just contained a single stage which made the training of the baseline GAN quite unstable. The impact of using a multi-scale discriminator on speckleGAN was less strong, as the presence of realistic speckles in the synthetic images at the beginning of training did already substantially stabilize training as described above.

Finally, we want to point out that just because the synthetic images appear visually realistic, this does not imply that they are realistic in terms of important contrast subtleties, correlations between certain features like shadows behind acoustically dense objects, or reasonable gray value gradients. These aspects are important if we want to use synthetic images to augment real datasets and thus improve segmentation performance. Hence, the question arises whether the synthetic datasets provide any benefit when used to augment real datasets, and if so, whether the images by speckleGAN have an advantage over the images by the baseline GAN. These questions will be answered in the next section.

6.6 Synthetic Data Augmentation

In the last section, we investigated the capabilities of GANs to generate synthetic images from small datasets. In this section, we take a look at the segmentation performances when using these synthetic images to augment the real datasets. We compared three settings for both segmentation CNNs (U-Net-Res and DeepLabV3):

1. no data augmentation (original baseline results from Section 6.1)
2. data augmentation with synthetic images by the baseline GAN
3. data augmentation with synthetic images by speckleGAN

For all trainings, conventional augmentation operations like flips, rotations, and elastic transformations were still used according to Chapter 6. The training procedure with synthetic data augmentation was divided into three phases:

1. pre-training with 1000 synthetic images for 20 epochs
2. combined training with real images and the same amount of fake images
3. fine-tuning with real data only

The model checkpoint that performed best on the validation set was used for testing. This could include checkpoints from training phases 2 and 3. All 5 cross-validation folds were augmented with the same synthetic images. Model variations after pre-training therefore originate only from different parameter initializations.

The required amounts of synthetic images were generated by the GANs form Section 6.5. To simulate a realistic scenario, we augmented each dataset using a GAN that was trained on exactly the same dataset. That is especially important regarding different dataset sizes. In practice, augmenting a dataset with 25 images by employing a GAN trained on 500 images is irrelevant. Instead, we would augment a dataset with 25 images by employing a GAN that was trained on the same 25 images. Of the 10 trained speckleGANs and baseline GANs, the models whose performance corresponded to the (upper) median in terms of FID score were used to generate the synthetic images. The required segmentation masks for conditioning the GANs were generated by applying conventional augmentation operations to the training segmentation masks. In addition to the operations listed in Section 6.1, we added elastic transformations also to both IVUS datasets. For all datasets, the elastic transformations for generating the GAN seeds were set to be slightly larger than the ones for conventionally augmenting the cardiac and neck muscle datasets. On top of that, scalings, shearings, rotations, and translations were applied. The resulting images were then used to augment the datasets according to the procedure shown above.

6.6.1 IVUS Lumen and Vessel Wall Segmentation

Figure 6.40 shows the IVUS lumen and vessel wall segmentation metrics for U-Net-Res for all three augmentation settings. The first striking statement we can make is that augmenting with images by speckleGAN did not systematically outperform augmentation with images by the baseline GAN. Synthetic data augmentation was particularly beneficial for vessel wall

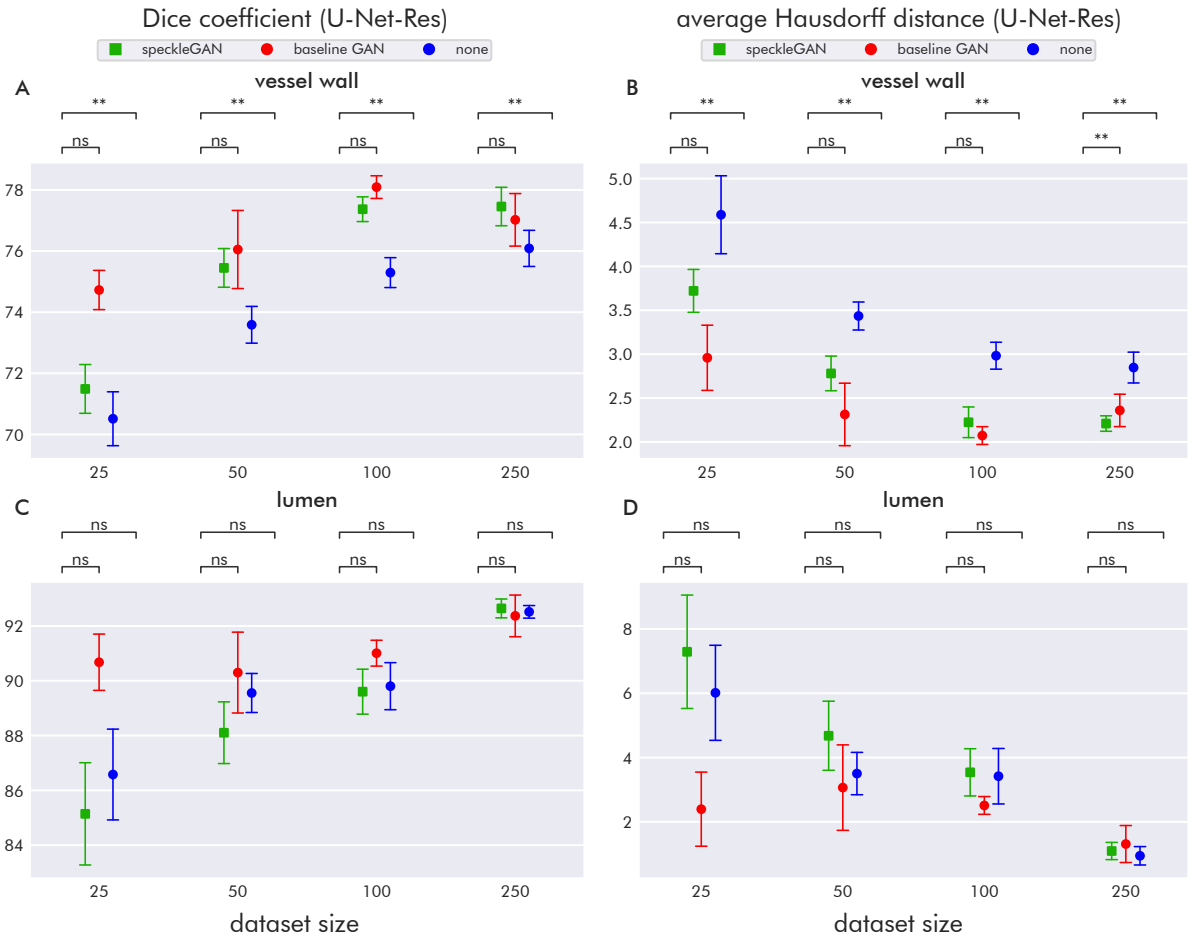

Figure 6.40: IVUS lumen and vessel wall segmentation results of U-Net-Res using synthetic data augmentation. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

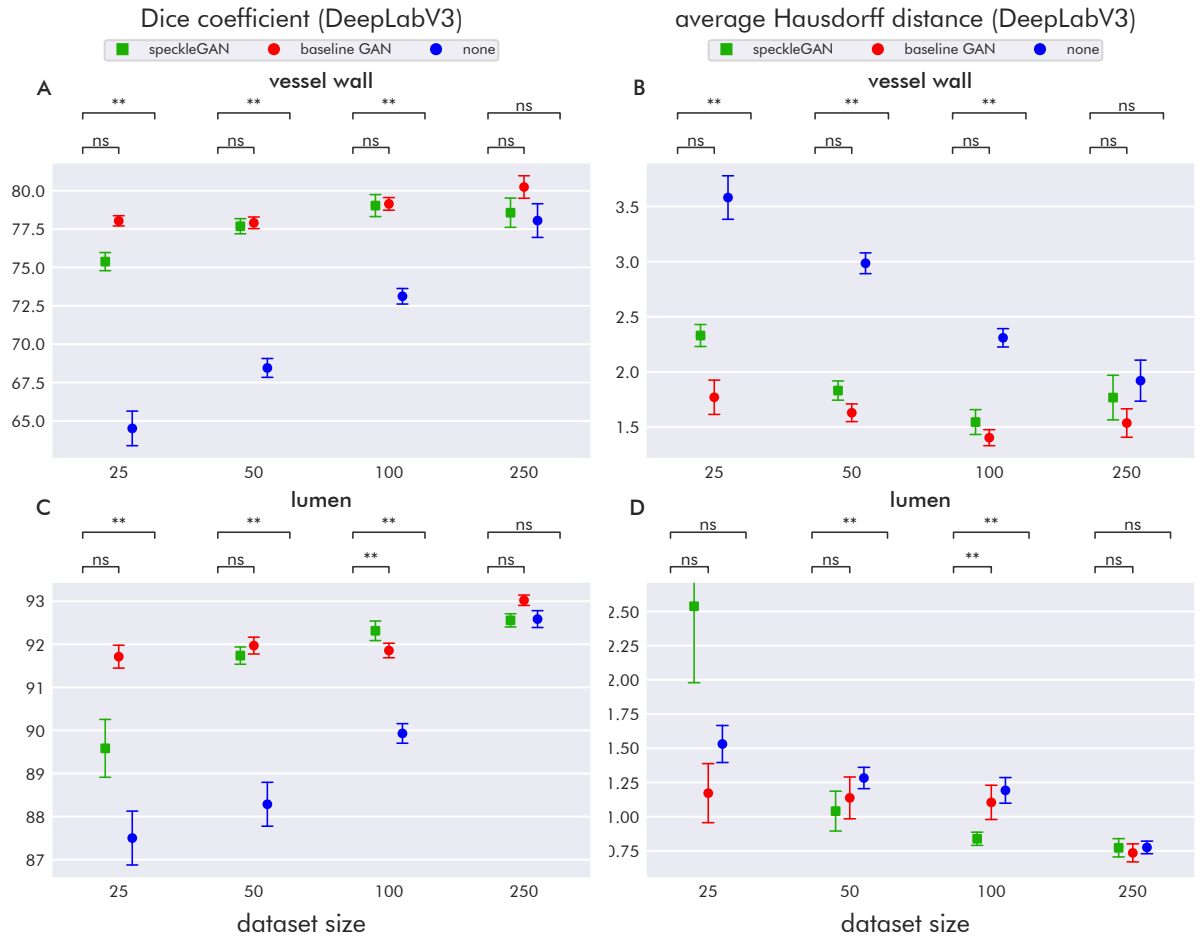

Figure 6.41: IVUS lumen and vessel wall segmentation results of DeepLabV3 using synthetic data augmentation. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

Table 6.11: IVUS lumen and vessel wall segmentation error rates using synthetic data augmentation. For each error, the rates achieved by not applying synthetic data augmentation, by using synthetic images from the baseline GAN, and by using synthetic images from speckleGAN are given, as well as the relative changes of the latter compared to both baselines. The columns are divided by dataset size (25, 50, 100, and 250 images) and CNN architecture (U: U-Net-Res, D: DeepLabV3). Values are given as percentages.

		25		50		100		250	
error	**method**	U	D	U	D	U	D	U	D
error 1	**baseline (no aug.)**	2.7	0.7	2.0	0.7	1.3	0.0	0.7	0.0
	baseline GAN aug.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	speckleGAN aug.	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	rel. change B/SG	**−75.0**	**−100.0**	**−100.0**	**−100.0**	**−100.0**	—	**−100.0**	—
	rel. change BG/SG	—	—	—	—	—	—	—	—
error 2	**baseline (no aug.)**	53.3	14.0	22.7	8.0	22.7	8.7	20.7	8.0
	baseline GAN aug.	34.0	10.0	29.3	12.0	30.0	8.0	24.7	6.0
	speckleGAN aug.	57.3	15.3	34.7	10.0	27.3	8.7	24.7	6.7
	rel. change B/SG	**7.5**	**9.5**	**52.9**	**25.0**	**20.6**	**0.0**	**19.4**	**−16.7**
	rel. change BG/SG	**68.6**	**53.3**	**18.2**	**−16.7**	**−8.9**	**8.3**	**0.0**	**11.1**
error 3	**baseline (no aug.)**	16.0	2.0	12.7	4.7	9.3	4.0	9.3	0.7
	baseline GAN aug.	16.7	1.3	10.7	0.7	8.0	0.7	6.0	1.3
	speckleGAN aug.	11.3	2.7	8.7	1.3	7.3	0.7	6.0	0.7
	rel. change B/SG	**−29.2**	**33.3**	**−31.6**	**−71.4**	**−21.4**	**−83.3**	**−35.7**	**0.0**
	rel. change BG/SG	**−32.0**	**100.0**	**−18.8**	**100.0**	**−8.3**	**0.0**	**0.0**	**−50.0**
error 4	**baseline (no aug.)**	69.3	51.3	54.7	33.3	47.3	31.3	43.3	24.0
	baseline GAN aug.	50.7	35.3	46.0	30.0	48.7	33.3	40.7	30.0
	speckleGAN aug.	68.0	38.7	60.7	28.7	50.7	30.7	42.7	24.7
	rel. change B/SG	**−1.9**	**−24.7**	**11.0**	**−14.0**	**7.0**	**−2.1**	**−1.5**	**2.8**
	rel. change BG/SG	**34.2**	**9.4**	**31.9**	**−4.4**	**4.1**	**−8.0**	**4.9**	**−17.8**
error 5	**baseline (no aug.)**	23.3	25.3	21.3	26.7	20.0	18.7	16.7	10.0
	baseline GAN aug.	37.3	20.0	28.0	19.3	25.3	18.7	21.3	18.7
	speckleGAN aug.	20.7	17.3	22.0	17.3	18.0	18.7	14.0	18.7
	rel. change B/SG	**−11.4**	**−31.6**	**3.1**	**−35.0**	**−10.0**	**0.0**	**−16.0**	**86.7**
	rel. change BG/SG	**−44.6**	**−13.3**	**−21.4**	**−10.3**	**−28.9**	**0.0**	**−34.4**	**0.0**

segmentation. Lumen segmentation was only improved for smaller datasets and only when augmenting with synthetic images by the baseline GAN. When using 250 real training images, synthetic data augmentation did not yield any improvement.

The results by DeepLabV3 are depicted in Figure 6.41. We again see that speckleGAN did not outperform the baseline GAN systematically, especially when using only 25 images for training GANs and segmentation CNNs. However, in almost all cases, augmenting with synthetic images greatly improved segmentation performance.

Table 6.11 shows the error rates for the three augmentation settings (baseline, baseline GAN augmentation, speckleGAN augmentation) as well as the relative improvements of speckleGAN augmentation (SG) compared to the original baseline (B) and baseline GAN augmentation (BG). First and foremost, we see that speckleGAN and baseline GAN augmentation largely reduce error 1 (topological disorder). The error 2 rate (incorrect patches) tends to increase with synthetic data augmentation, whereas the error 5 rate decreases for most dataset sizes. The tendencies of the other error rates look quite inconclusive. We cannot identify distinct correlations between error rates and segmentation metrics.

6.6.2 IVUS Calcium Segmentation

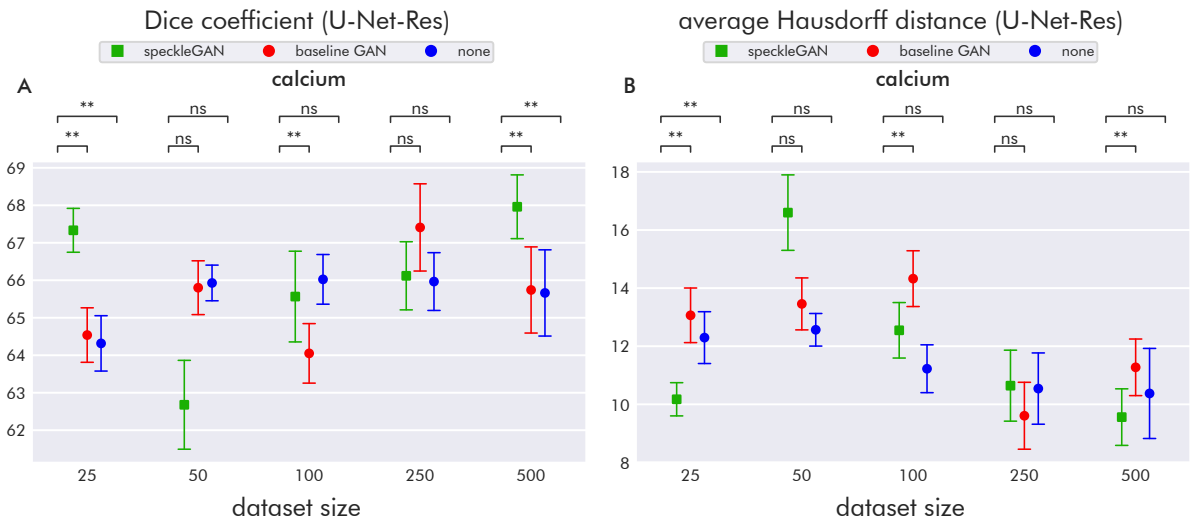

Figure 6.42: IVUS calcium segmentation results of U-Net-Res using synthetic data augmentation. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

The results of IVUS calcium segmentation of U-Net-Res are shown in Figure 6.42. The performance drop of speckleGAN augmentation for 50 training images is quite salient. Significant outperformances by speckleGAN augmentation are only seen for 25 and 500 training images. Apart from that, the results are quite similar and exhibit more or less constant variance.

Figure 6.43 shows the corresponding results of DeepLabV3. In terms of the Dice score, synthetic data augmentation led to improvements for nearly all dataset sizes. However, we observe a performance drop for 100 training images. This behavior is the same as for the original baseline. For 500 training images, the average Hausdorff distance is largely improved by synthetic data augmentation.

Considering the error rates in Table 6.12 does not provide much more insight. While error

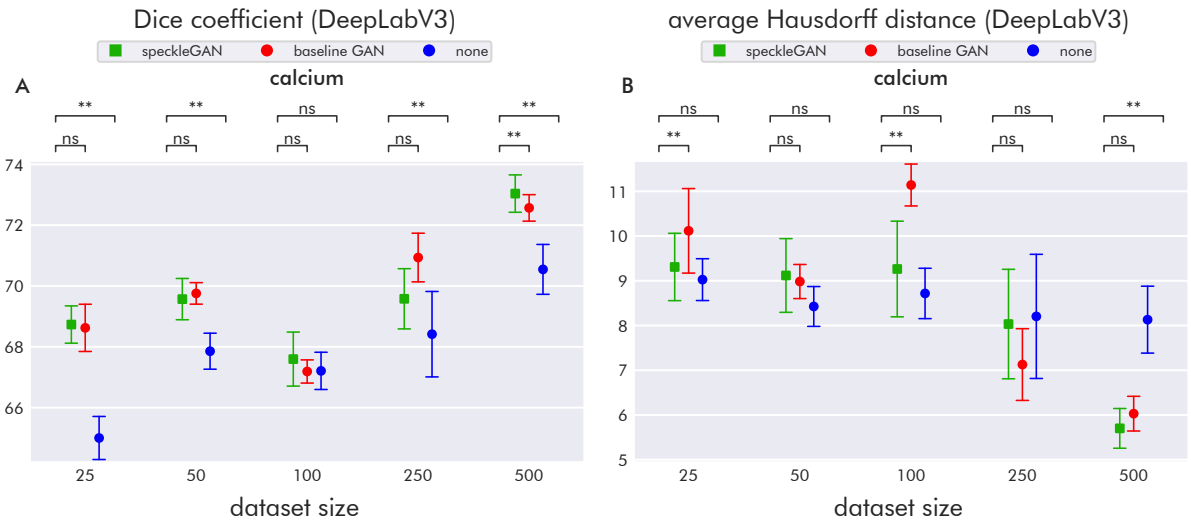

Figure 6.43: IVUS calcium segmentation results of DeepLabV3 using synthetic data augmentation. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

Table 6.12: IVUS calcium segmentation error rates using synthetic data augmentation. For each error, the rates achieved by not applying synthetic data augmentation, by using synthetic images from the baseline GAN, and by using synthetic images from speckleGAN are given, as well as the relative changes of the latter compared to both baselines. The columns are divided by dataset size (25, 50, 100, 250, and 500 images) and CNN architecture (U: U-Net-Res, D: DeepLabV3). Values are given as percentages.

		25		50		100		250		500	
error	**method**	U	D	U	D	U	D	U	D	U	D
	baseline (no aug.)	62.2	56.5	58.0	50.8	49.7	42.0	42.0	39.9	37.3	31.6
	baseline GAN aug.	60.6	49.7	68.4	48.2	49.7	31.6	41.5	21.8	38.9	27.5
error 1	**speckleGAN aug.**	54.4	50.3	41.5	58.5	26.4	39.9	54.9	24.4	27.5	28.5
	rel. change B/SG	**−12.5**	**−11.0**	**−28.6**	**15.3**	**−46.9**	**−4.9**	**30.9**	**−39.0**	**−26.4**	**−9.8**
	rel. change BG/SG	**−10.3**	**1.0**	**−39.4**	**21.5**	**−46.9**	**26.2**	**32.5**	**11.9**	**−29.3**	**3.8**
	baseline (no aug.)	33.7	37.8	36.3	35.8	31.6	32.1	37.8	36.3	36.8	35.8
	baseline GAN aug.	32.1	26.4	29.0	24.9	31.6	25.4	31.1	26.9	32.6	33.7
error 2	**speckleGAN aug.**	26.4	23.3	31.6	28.5	29.5	26.4	34.7	24.4	32.1	26.4
	rel. change B/SG	**−21.5**	**−38.4**	**−12.9**	**−20.3**	**−6.6**	**−17.7**	**−8.2**	**−32.9**	**−12.7**	**−26.1**
	rel. change BG/SG	**−17.7**	**−11.8**	**8.9**	**14.6**	**−6.6**	**4.1**	**11.7**	**−9.6**	**−1.6**	**−21.5**

2 is clearly reduced with speckleGAN augmentation compared to training without synthetic data augmentation, the picture with respect to error 1 is rather inconclusive.

6.6.3 Cardiac Segmentation

Table 6.13: Cardiac segmentation error rates using synthetic data augmentation. For each error, the rates achieved by not applying synthetic data augmentation, by using synthetic images from the baseline GAN, and by using synthetic images from speckleGAN are given, as well as the relative changes of the latter compared to both baselines. The columns are divided by dataset size (20, 40, 80, 160, 320, 640, and 1280 images) and CNN architecture (U: U-Net-Res, D: DeepLabV3). Values are given as percentages.

		20		40		80		160		320		640		1280	
error	method	U	D	U	D	U	D	U	D	U	D	U	D	U	D
error 1	baseline (no aug.)	26.0	0.0	12.6	0.0	7.2	0.0	3.8	0.0	1.2	0.0	0.6	0.0	0.0	0.0
	baseline GAN aug.	0.4	0.0	0.4	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	speckleGAN aug.	0.6	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	rel. change B/SG	−97.7	—	−98.4	—	−100.0	—	−100.0	—	−100.0	—	−100.0	—	—	—
	rel. change BG/SG	50.0	—	−50.0	—	−100.0	—	—	—	—	—	—	—	—	—
error 2	baseline (no aug.)	67.0	18.6	50.6	14.0	39.8	11.6	36.4	6.2	20.2	3.4	14.4	1.6	11.0	1.0
	baseline GAN aug.	37.8	19.4	31.6	14.2	28.0	9.2	21.6	6.0	13.8	3.8	10.2	1.6	6.0	0.8
	speckleGAN aug.	40.2	18.4	33.6	13.0	26.6	10.0	20.0	5.4	14.6	4.2	9.8	1.6	5.0	0.6
	rel. change B/SG	−40.0	−1.1	−33.6	−7.1	−33.2	−13.8	−45.1	−12.9	−27.7	23.5	−31.9	0.0	−54.5	−40.0
	rel. change BG/SG	6.3	−5.2	6.3	−8.5	−5.0	8.7	−7.4	−10.0	5.8	10.5	−3.9	0.0	−16.7	−25.0
error 3	baseline (no aug.)	42.0	18.0	31.0	14.0	20.2	13.0	13.2	6.2	9.0	5.0	6.2	1.8	4.6	2.0
	baseline GAN aug.	18.0	9.8	12.6	7.0	11.2	5.8	9.4	4.4	6.0	3.6	3.8	2.2	2.2	0.6
	speckleGAN aug.	18.6	8.4	15.6	6.4	11.6	6.6	7.6	5.2	4.6	4.0	2.8	1.6	2.6	0.4
	rel. change B/SG	−55.7	−53.3	−49.7	−54.3	−42.6	−49.2	−42.4	−16.1	−48.9	−20.0	−54.8	−11.1	−43.5	−80.0
	rel. change BG/SG	3.3	−14.3	23.8	−8.6	3.6	13.8	−19.1	18.2	−23.3	11.1	−26.3	−27.3	18.2	−33.3
error 4	baseline (no aug.)	22.2	27.0	19.2	18.2	17.4	17.6	10.4	9.8	8.0	7.0	3.0	2.0	0.8	0.8
	baseline GAN aug.	17.2	13.0	15.2	12.4	11.0	8.6	8.2	4.8	6.4	3.0	3.2	1.2	1.0	0.2
	speckleGAN aug.	16.2	14.2	14.0	12.4	12.4	8.2	9.2	6.0	7.8	3.6	2.4	1.4	0.8	0.4
	rel. change B/SG	−27.0	−47.4	−27.1	−31.9	−28.7	−53.4	−11.5	−38.8	−2.5	−48.6	−20.0	−30.0	0.0	−50.0
	rel. change BG/SG	−5.8	9.2	−7.9	0.0	12.7	−4.7	12.2	25.0	21.9	20.0	−25.0	16.7	−20.0	100.0
error 5	baseline (no aug.)	25.0	13.0	21.0	14.0	17.2	8.2	8.8	6.6	4.6	3.6	2.4	1.2	0.6	0.2
	baseline GAN aug.	19.2	9.4	15.0	7.6	13.0	4.2	7.8	3.6	5.2	2.8	2.6	1.2	0.8	0.0
	speckleGAN aug.	20.0	10.2	16.4	8.0	13.8	5.8	8.2	4.4	4.6	2.2	2.2	0.8	0.6	0.0
	rel. change B/SG	−20.0	−21.5	−21.9	−42.9	−19.8	−29.3	−6.8	−33.3	0.0	−38.9	−8.3	−33.3	0.0	−100.0
	rel. change BG/SG	4.2	8.5	9.3	5.3	6.2	38.1	5.1	22.2	−11.5	−21.4	−15.4	−33.3	−25.0	—

Figure 6.44 depicts the cardiac segmentation results of U-Net-Res. In general, synthetic data augmentation improved the segmentation metrics compared to the baseline for small to medium dataset sizes. The improvements are larges for 20 training images. Again, speckleGAN augmentation shows no benefit compared to baseline GAN augmentation. However, speckleGAN augmentation tends to outperform baseline GAN augmentation for larger datasets, particularly for endocardium segmentation. We also observe an instability of atrium segmentation with speckleGAN augmentation and 50 training images. Here, the atrium was not recognized in a few of the 10 trained ensemble models, leading to a Dice score of 0 and an average Hausdorff distance of 128 pixels (half the image size).

The corresponding results of DeepLabV3 are shown in Figure 6.45. Synthetic data augmentation improved the results of almost all metrics and dataset sizes. The improvements are larger than those of U-Net-Res, and the values obtained are better for small and medium dataset sizes. As in the case of segmentation with U-Net-Res, speckleGAN augmentation tends to slightly outperform baseline GAN augmentation for larger amounts of training data.

The error frequencies in Table 6.13 show reductions of almost all errors when using speckleGAN augmentation compared to using no synthetic data augmentation. We cannot see any clear tendency when comparing speckleGAN error rates with baseline GAN error rates. However, improvements with speckleGAN augmentation compared to baseline GAN augmentation tend to aggregate at larger dataset sizes which correlate with the segmentation metrics.

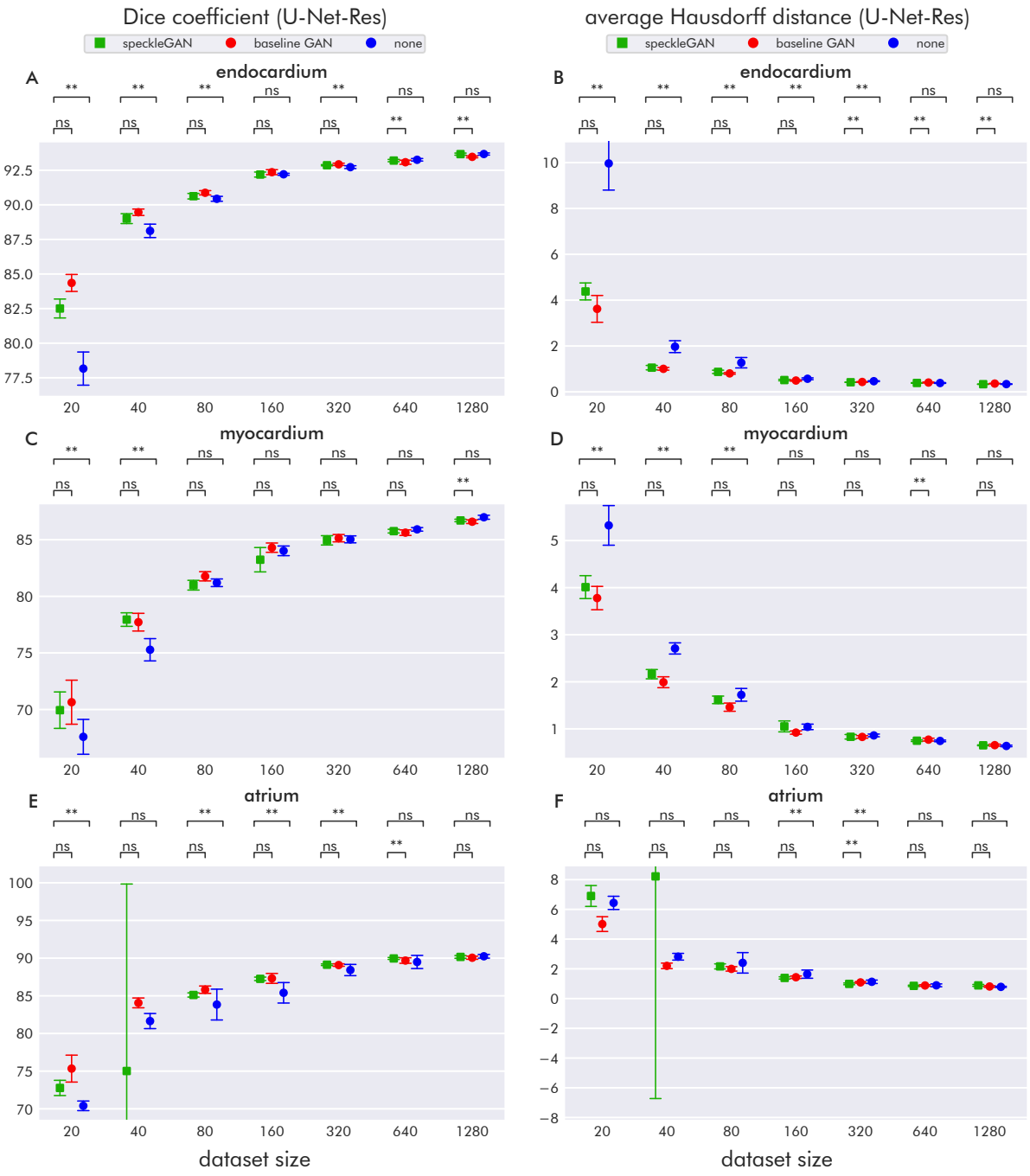

Figure 6.44: Cardiac segmentation results of U-Net-Res using synthetic data augmentation. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

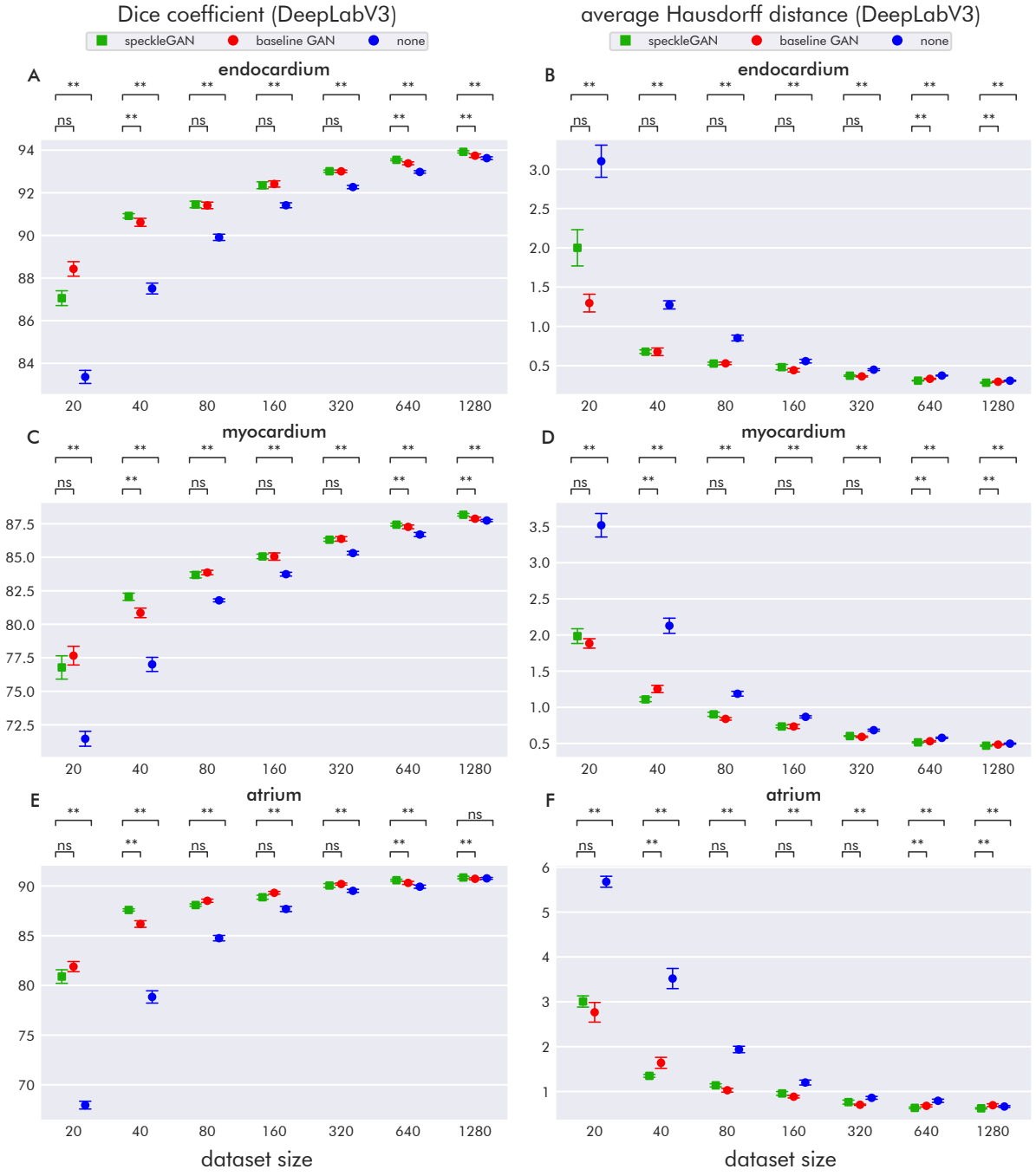

Figure 6.45: Cardiac segmentation results of DeepLabV3 using synthetic data augmentation. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

6.6.4 Neck Muscle Segmentation

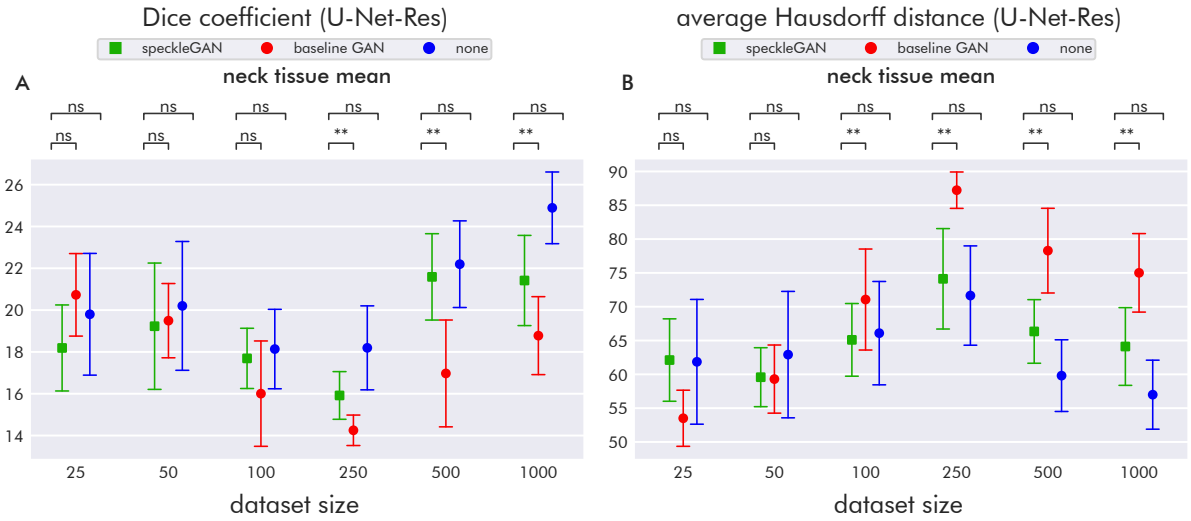

Figure 6.46: Neck muscle segmentation results of U-Net-Res using synthetic data augmentation. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

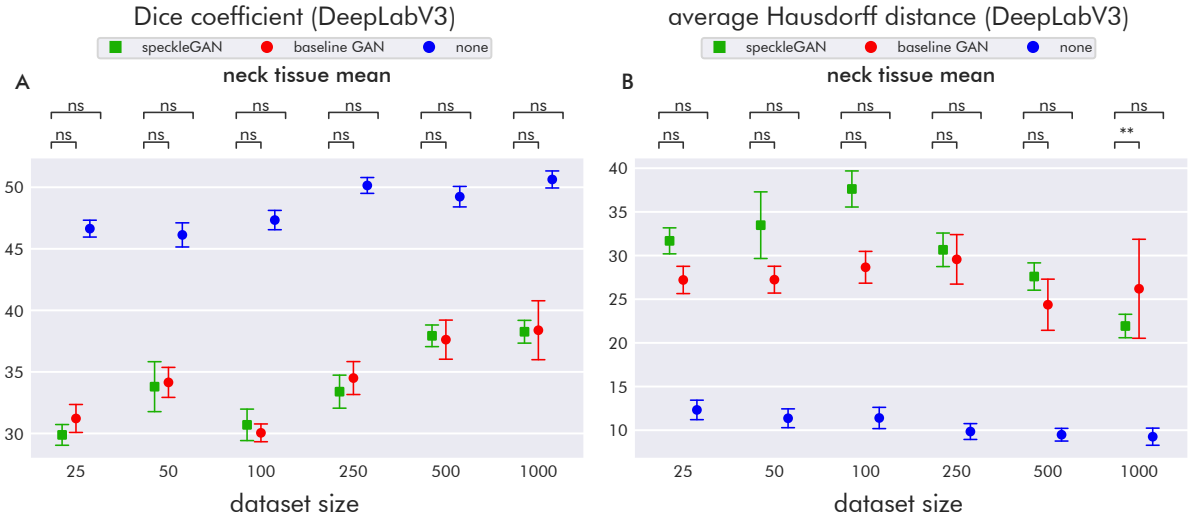

Figure 6.47: Neck muscle segmentation results of DeepLabV3 using synthetic data augmentation. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

Finally, we take a look at the neck muscle segmentation results starting with U-Net-Res in Figure 6.46. SpeckleGAN augmentation outperforms baseline GAN augmentation for datasets larger than 100 images. However, training with synthetic data augmentation did not outperform the baseline in any case. On the contrary, synthetic data augmentation worsens the results on larger datasets.

The results of DeepLabV3 are depicted in Figure 6.47. A striking feature is the drastic deterioration of the results when applying synthetic data augmentation. SpeckleGAN augmentation and baseline GAN augmentation perform more or less equally poorly, except for

the average Hausdorff distance at 25 to 100 training images, where the latter outperforms the former.

6.6.5 Summary and Discussion

In this section, we have seen that data augmentation with synthetic images can largely increase segmentation performance, especially when training on smaller datasets. But also larger datasets can benefit from additional synthetic data in some cases. The IVUS lumen and vessel wall dataset and the cardiac dataset benefited the most, while the IVUS calcium dataset benefited only partially. Surprisingly, the results on the neck muscle dataset actually worsened.

The most striking observation on the IVUS lumen and vessel wall dataset is that synthetic data augmentation with SpeckleGAN images performed worse than augmentation with images by the baseline GAN on the smallest dataset. The differences are larger for U-Net-Res. This behavior is likely caused by an unfavorable bias in the corresponding synthetic dataset leading to a bias of the CNN, too.

The large randomly appearing differences between speckleGAN and baseline GAN metrics on the IVUS calcium dataset are also likely caused by certain biases in the synthetic datasets. These differences only occur for U-Net-Res, not for DeepLabV3. In the latter case, we even observe that the performance drops for 100 training images when employing synthetic data augmentation. This shows that the adverse bias in the dataset was also adopted by the synthetic dataset. We suppose that the IVUS calcium results are, in general, quite inconclusive because the training and test set lack variability (Subsection 5.1.4). Moreover, we see that the metrics improve for dataset sizes of 100 and above, although the corresponding FID scores of the baseline GAN worsen. The FID score and segmentation performance do thus not correlate (negatively) in this case.

The improvements on the heart dataset appear quite systematic. The metrics by DeepLabV3 show that synthetic data augmentation has roughly the same effect as doubling the size of the real dataset. The impact on U-Net-Res is not that large. However, synthetic images seem to provide much valuable information for the segmentation CNNs. As speckle noise is not very pronounced in this dataset, we assume that speckle is not the main source of information. Synthetic images are more likely to exhibit other features that provide meaningful information. More about that later. Now, we want to answer **RQ 2.1** to **RQ 2.4** for synthetic data augmentation.

RQ 2.1: Which CNN architectures benefit from synthetic data augmentation?
Overall, it seems that DeepLabV3 benefits more from synthetic data augmentation than U-Net-Res, also reaching better metric values. This could mean that DeepLabV3 would outperform U-Net-Res in general if enough data were available. This tendency is also reflected by the baseline cardiac segmentation results for the myocardium and atrium on the largest dataset size of 1280 images (which is still quite small for the deep learning field). However, if we remember DeepLabV3's inability to resolve small details below 8 pixels, it becomes clear that this advantage has its limits.

RQ 2.2: How does synthetic data augmentation perform as a function of dataset size?
The improvements are usually largest for the smallest dataset and decrease with increasing

dataset size. However, in the cases of IVUS calcium and cardiac segmentation, synthetic data augmentation can also be beneficial for the largest datasets.

RQ 2.3: Which tissues benefit from synthetic data augmentation?
Most of the investigated tissues benefit largely from synthetic data augmentation. While the improvements in IVUS lumen and vessel wall segmentation and cardiac segmentation are quite systematic, the improvements in IVUS calcium segmentation are unevenly distributed across dataset sizes. One potential reason for this is the small variability of the IVUS calcium dataset, implicating large differences between the training and test data domains. Thus, whether synthetic data augmentation provides valuable additional information depends more on chance.

The instability of U-Net-Res in atrium segmentation with 40 training images is likely due to an unfavorable bias in the synthetic dataset that causes some segmentation ensembles to fail completely in detecting atria. In general, an adverse bias in the synthetic dataset likely caused more than one aberrant behavior. Hence, for future experiments, it would be reasonable to repeat trainings with different synthetic datasets for every dataset size.

DeepLabV3's performance on the neck muscle dataset drastically decreased. But also, U-Net-Res cannot outperform the baseline by means of synthetic data. It is possible that the edges of the conditional masks, which are still clearly visible in the synthetic images, provide features that are not present in the real images, thus confusing the CNNs. This means that the other image features present in the synthetic images do not provide any additional meaningful information for segmentation. For large datasets, the synthetic datasets reach relatively small FID scores below 60 and SSIM values larger than 0.72, but still, the segmentation results worsen. This indicates that most of the features in the neck muscle images are not useful for segmentation at all.

RQ 2.4: What types of segmentation errors are reduced by synthetic data augmentation?
Synthetic data augmentation tends to improve all segmentation errors more or less uniformly. However, there are exceptions like error 2 of the IVUS lumen and vessel wall dataset, which is largely increased. But based on our findings, we cannot assume any reasons for this behavior.

Generally, we cannot observe a clear tendency regarding which GAN generates more favorable synthetic images for data augmentation. Notably, a better FID score does not imply better segmentation performance. The same holds for SSIM and Jensen-Shannon divergence. In the last section, we assumed that the improved FID of speckleGAN for the IVUS datasets was mainly due to not suffering from speckle mode collapse in contrast to the baseline GAN. What follows is that speckle variability across the synthetic images does not have a positive impact on segmentation performance. Also, we cannot guarantee that the speckle noise added by the speckle layer is similar to real speckle, although the size and distribution can be learned during training. However, the subsequent layers have the potential to adapt the speckle texture such that the generator can fool the discriminator. But this is by no means measurable.

Moreover, we could hardly detect any correlation between segmentation and GAN performances. This means that in addition to the features that affect FID, SSIM, and Jensen-Shannon divergence, there are other features in the synthetic images that have an impact on segmentation performance. We assume that these are the kinds of features we described earlier, like

contrast subtleties, correlations between certain features like shadows behind acoustically dense objects, or reasonable gray value gradients. All these cannot be measured directly by the three metrics mentioned above.

Recalling Figure 4.11 about different configurations of the real and synthetic data distribution, we can assume that especially the cardiac dataset facilitates beneficial configurations. Here, the synthetic data distributions seem to include many images that do not lie in the main modes of the data distribution but are similar to images from the test set. But why is that so? The test set comprises several hundred images from more than a hundred patients. Therefore, synthetic images are likely to lie near some modes of the test data distribution. This is not the case for the IVUS datasets, which comprise images from only a few patients (see Subsection 5.1.4) and thus exhibit less variability in the training and test set. The underlying data distributions therefore have fewer modes making overlap between synthetic and real data distributions less likely. This effect is more pronounced for the IVUS calcium dataset, leading to more variability regarding the improvement through synthetic data augmentation. The same holds for the neck muscle dataset (see Subsection 5.3.3). Additionally, due to the visible boundaries through the conditional masks in the synthetic neck muscle images, we can assume that the synthetic data distribution has almost no overlap with the real data distribution. Hence, the synthetic images do not provide additional information for neck muscle segmentation. They rather lead to adverse network configurations after pre-training, making it harder for the network to achieve advantageous configurations during fine-tuning. The results can thus even get worse compared to the baseline. Ultimately, more research is needed to gain further insight into this topic.

6.7 Combinations of Methods

Finally, we want to show what improvements are possible compared to the original baseline when all the methods presented so far are combined. For each dataset, we combined all applicable methods. The combinations for the individual datasets were as follows:

- **IVUS lumen and vessel wall**
  - Combining wavelet scattering and CNNs
  - ICA shape prior
  - Topological constraints
  - Synthetic data augmentation with speckleGAN
- **IVUS calcium**
  - Combining wavelet scattering and CNNs
  - Synthetic data augmentation with speckleGAN
- **Cardiac**
  - Combining wavelet scattering and CNNs
  - ICA shape prior
  - Topological constraints
  - Synthetic data augmentation with speckleGAN
- **Neck muscle**
  - Combining wavelet scattering and CNNs
  - ICA shape prior

In the case of the neck muscle dataset, we refrained from adding synthetic data augmentation since it drastically worsened the results by DeepLabV3. For cases that employed SEST and ICA shape prior, we inserted the SEST blocks only into the main branch of the CNNs, not into the second parallel branch that processes the ICs. The other parameters stayed the same. Note that we compare the results with the original baseline. Therefore, improvements due to a larger number of parameters and due to the methods add up.

6.7.1 IVUS Lumen and Vessel Wall Segmentation

Figure 6.48 depicts the IVUS lumen and vessel wall segmentation results when employing a combination of all previously presented methods. In most cases, the combination of methods largely outperforms that baseline. While U-Net-Res seems to have some trouble on the smallest dataset, DeepLabV3 benefits quite largely. An exception is the average Hausdorff distance of lumen. Here, the baseline achieved better values for 25 training images. The improvements of the combination of methods vanish for 250 training images, except for vessel wall segmentation of U-Net-Res. However, the baseline results were comparatively poor in these cases.

The error frequencies are shown in Table 6.14. With a few exceptions in error 5 and error 2, the combination of methods achieved large improvements, mostly more than 50 %. Exceptions are errors 4 and 5. Despite moderate improvements in most cases, the error rates still remain

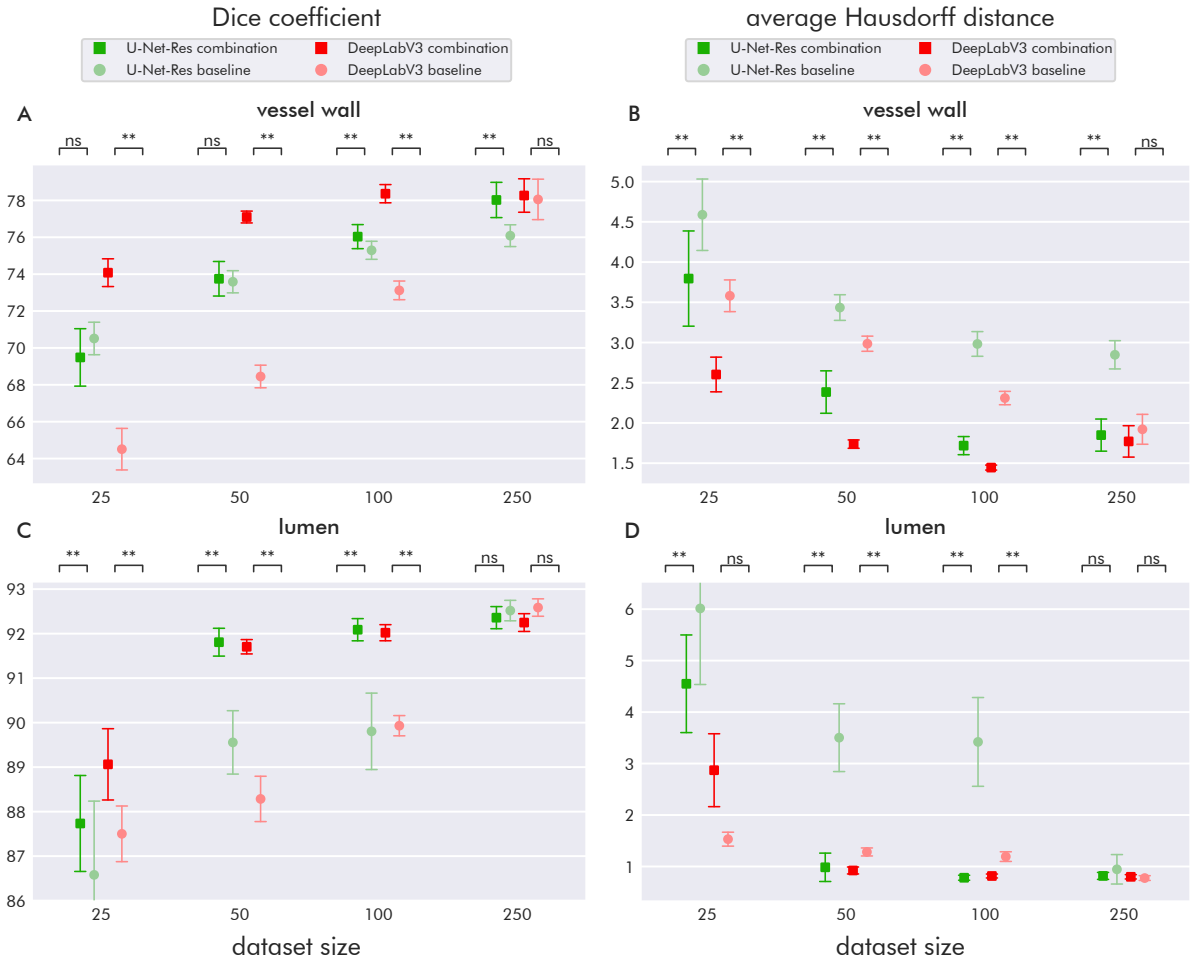

Figure 6.48: IVUS lumen and vessel wall segmentation results using a combination of methods. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

Table 6.14: IVUS lumen and vessel wall segmentation error rates using a combination of methods. For each error, the rates achieved by the baseline and the combination of methods are given, as well as the relative change of the latter compared to the former. The columns are divided by dataset size (25, 50, 100, and 250 images) and CNN architecture (U: U-Net-Res, D: DeepLabV3). Values are given as percentages.

		25		50		100		250	
error	**method**	U	D	U	D	U	D	U	D
error 1	**baseline**	2.7	0.7	2.0	0.7	1.3	0.0	0.7	0.0
	comb	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	rel. change	**−25.0**	**−100.0**	**−100.0**	**−100.0**	**−100.0**	—	**−100.0**	—
error 2	**baseline**	53.3	14.0	22.7	8.0	22.7	8.7	20.7	8.0
	comb	19.3	15.3	5.3	3.3	9.3	2.7	2.0	4.0
	rel. change	**−63.7**	**9.5**	**−76.5**	**−58.3**	**−58.8**	**−69.2**	**−90.3**	**−50.0**
error 3	**baseline**	16.0	2.0	12.7	4.7	9.3	4.0	9.3	0.7
	comb	2.0	0.0	2.0	2.0	1.3	0.0	0.0	0.0
	rel. change	**−87.5**	**−100.0**	**−84.2**	**−57.1**	**−85.7**	**−100.0**	**−100.0**	**−100.0**
error 4	**baseline**	69.3	51.3	54.7	33.3	47.3	31.3	43.3	24.0
	comb	62.0	46.0	38.7	26.7	36.7	24.7	28.0	22.0
	rel. change	**−10.6**	**−10.4**	**−29.3**	**−20.0**	**−22.5**	**−21.3**	**−35.4**	**−8.3**
error 5	**baseline**	23.3	25.3	21.3	26.7	20.0	18.7	16.7	10.0
	comb	16.7	15.3	20.7	19.3	21.3	14.7	17.3	11.3
	rel. change	**−28.6**	**−39.5**	**−3.1**	**−27.5**	**6.7**	**−21.4**	**4.0**	**13.3**

quite high for 250 training examples with more than 20 % (error 4) and more than 10 % (error 5). We even see small increases for error rate 5.

6.7.2 IVUS Calcium Segmentation

Table 6.15: IVUS calcium segmentation error rates using a combination of methods. For each error, the rates achieved by the baseline and the combination of methods are given, as well as the relative change of the latter compared to the former. The columns are divided by dataset size (25, 50, 100, 250, and 500 images) and CNN architecture (U: U-Net-Res, D: DeepLabV3). Values are given as percentages.

		25		50		100		250		500	
error	**method**	U	D	U	D	U	D	U	D	U	D
error 1	**baseline**	62.2	56.5	58.0	50.8	49.7	42.0	42.0	39.9	37.3	31.6
	comb	51.3	46.6	52.3	36.3	42.0	37.3	26.9	34.7	21.2	23.3
	rel. change	**−17.5**	**−17.4**	**−9.8**	**−28.6**	**−15.6**	**−11.1**	**−35.8**	**−13.0**	**−43.1**	**−26.2**
error 2	**baseline**	33.7	37.8	36.3	35.8	31.6	32.1	37.8	36.3	36.8	35.8
	comb	25.4	21.8	23.3	22.8	25.9	30.6	27.5	25.4	28.5	26.9
	rel. change	**−24.6**	**−42.5**	**−35.7**	**−36.2**	**−18.0**	**−4.8**	**−27.4**	**−30.0**	**−22.5**	**−24.6**

The IVUS calcium segmentation results when employing a combination of methods are shown in Figure 6.49. The methods used are SEST and synthetic data augmentation with speckleGAN. Regarding the Dice coefficient, the combination of methods outperforms the baseline in almost all cases. An exception is the dip of U-Net-Res for 50 training images. The same dip can be observed for the average Hausdorff distance. Here, we see almost only improvements for U-Net-Res. However, U-Net-Res does not outperform DeepLabV3. The combination

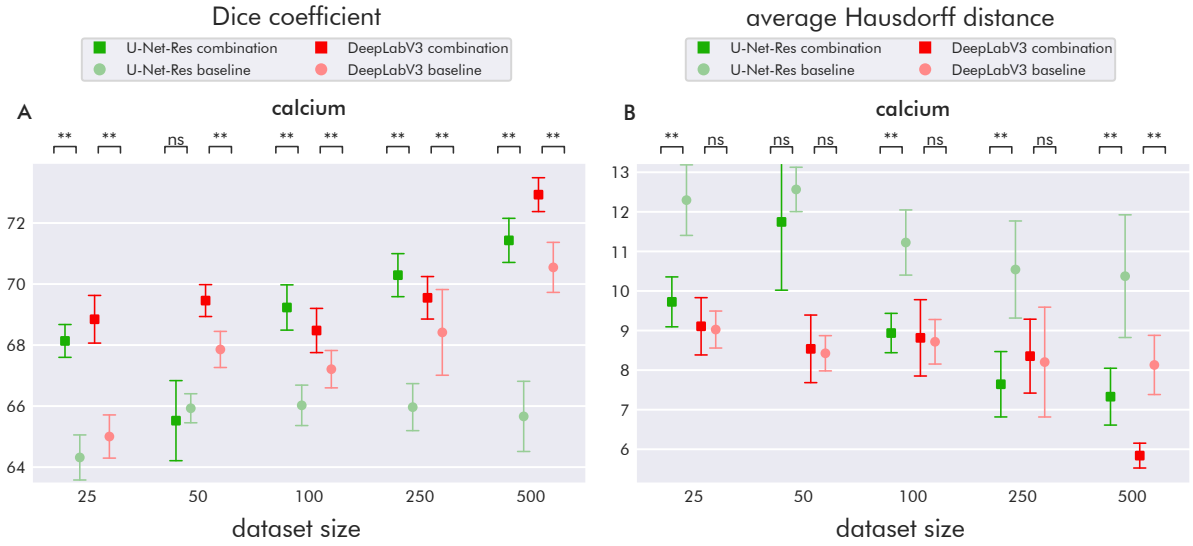

Figure 6.49: IVUS calcium segmentation results using a combination of methods. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

of methods does only outperform the baseline for 500 training images. The best results are obtained by DeepLabV3 for 500 training examples. Here, the improvement is still quite large, with 25 % in terms of the average Hausdorff distance and about 3 % in terms of the Dice coefficient.

Table 6.15 depicts the corresponding error frequencies. We only see improvements between 4.8 % and 43.1 %. Error 1 tends to improve a little more on larger datasets, and vice versa for error 2. However, even for 500 training images, the error 1 rates of more than 20 % and the error 2 rates of more than 25 % remain.

6.7.3 Cardiac Segmentation

Table 6.16: Cardiac segmentation error rates using a combination of methods. For each error, the rates achieved by the baseline and the combination of methods are given, as well as the relative change of the latter compared to the former. The columns are divided by dataset size (20, 40, 80, 160, 320, 640, and 1280 images) and CNN architecture (U: U-Net-Res, D: DeepLabV3). Values are given as percentages.

		20		40		80		160		320		640		1280	
error	method	U	D	U	D	U	D	U	D	U	D	U	D	U	D
error 1	baseline	26.0	0.0	12.6	0.0	7.2	0.0	3.8	0.0	1.2	0.0	0.6	0.0	0.0	0.0
	comb	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	rel. change	**−100.0**	—	**−100.0**	—	**−100.0**	—	**−100.0**	—	**−100.0**	—	**−100.0**	—	—	—
error 2	baseline	67.0	18.6	50.6	14.0	39.8	11.6	36.4	6.2	20.2	3.4	14.4	1.6	11.0	1.0
	comb	42.0	19.8	30.2	13.0	19.2	9.2	14.0	5.0	10.2	4.2	4.6	2.0	2.0	1.2
	rel. change	**−37.3**	**6.5**	**−40.3**	**−7.1**	**−51.8**	**−20.7**	**−61.5**	**−19.4**	**−49.5**	**23.5**	**−68.1**	**25.0**	**−81.8**	**20.0**
error 3	baseline	42.0	18.0	31.0	14.0	20.2	13.0	13.2	6.2	9.0	5.0	6.2	1.8	4.6	2.0
	comb	8.0	5.0	6.6	3.8	5.0	3.4	3.6	1.6	1.4	2.2	0.0	0.0	0.0	0.0
	rel. change	**−81.0**	**−72.2**	**−78.7**	**−72.9**	**−75.2**	**−73.8**	**−72.7**	**−74.2**	**−84.4**	**−56.0**	**−100.0**	**−100.0**	**−100.0**	**−100.0**
error 4	baseline	22.2	27.0	19.2	18.2	17.4	17.6	10.4	9.8	8.0	7.0	3.0	2.0	0.8	0.8
	comb	10.0	10.0	8.4	5.8	7.8	6.8	5.2	3.8	3.8	4.2	0.2	1.2	0.6	0.0
	rel. change	**−55.0**	**−63.0**	**−56.2**	**−68.1**	**−55.2**	**−61.4**	**−50.0**	**−61.2**	**−52.5**	**−40.0**	**−93.3**	**−40.0**	**−25.0**	**−100.0**
error 5	baseline	25.0	13.0	21.0	14.0	17.2	8.2	8.8	6.6	4.6	3.6	2.4	1.2	0.6	0.2
	comb	19.0	7.0	14.2	6.4	13.2	5.0	3.8	4.4	4.2	2.6	0.6	1.4	0.2	0.0
	rel. change	**−24.0**	**−46.2**	**−32.4**	**−54.3**	**−23.3**	**−39.0**	**−56.8**	**−33.3**	**−8.7**	**−27.8**	**−75.0**	**16.7**	**−66.7**	**−100.0**

Figure 6.50 depicts the results of cardiac segmentation when employing a combination of

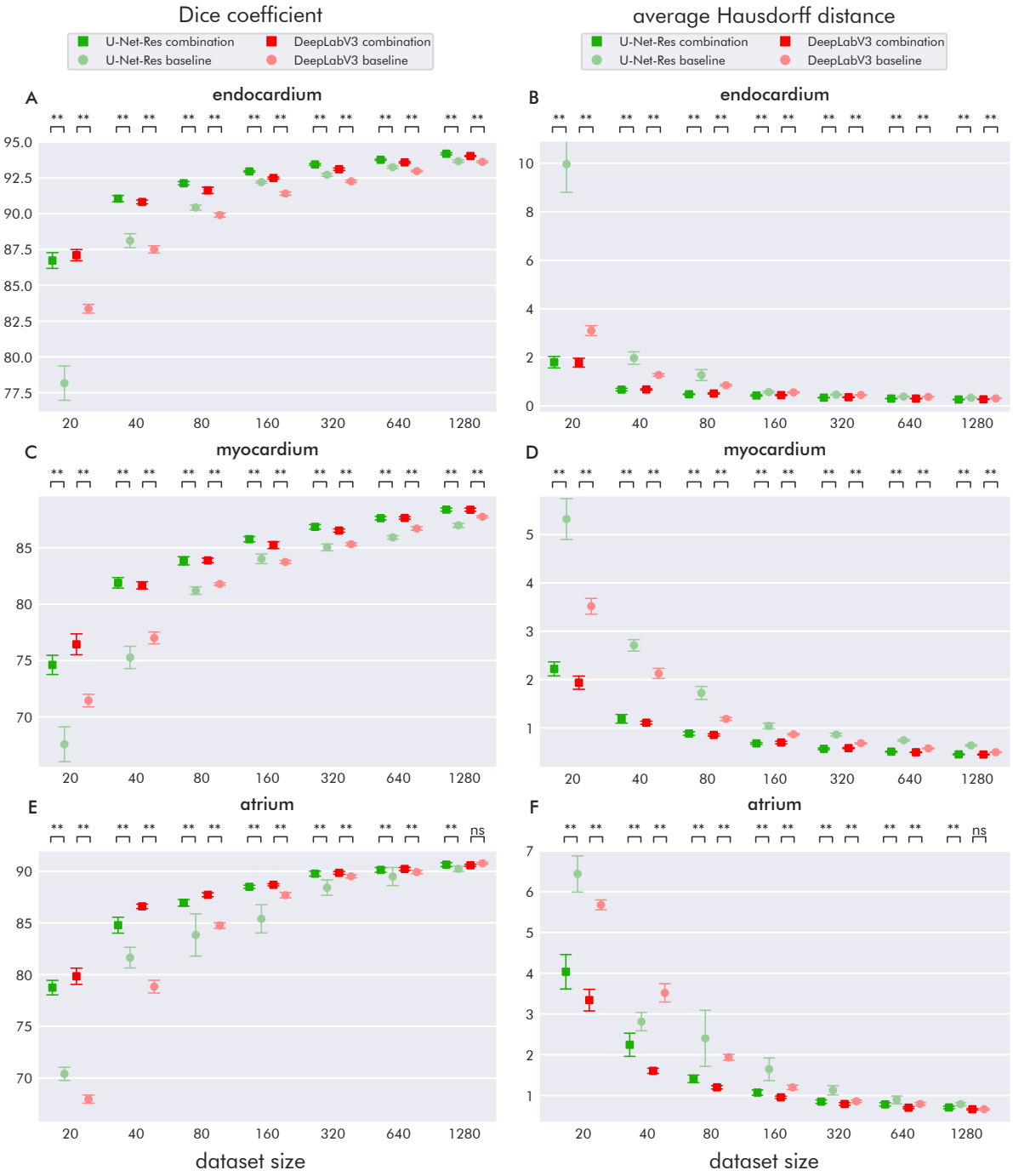

Figure 6.50: Cardiac segmentation results using a combination of methods. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

all presented methods. The combination led to improvements in almost all cases. The only exception is atrium segmentation by DeepLabV3 with 1280 training images. For both CNNs, the combination of methods tends to have the same effect as doubling the dataset size. In some cases, even more. For smaller datasets, the improvements of U-Net-Res on endocardium and myocardium segmentation tend to be larger since the baseline results are a little poorer. U-Net-Res is thus able to reach quite the same performance as DeepLabV3. For larger datasets, U-Net-Res can even outperform DeepLabV3. Atrium segmentation is slightly dominated by DeepLabV3.

If we take a look at the error frequencies in Table 6.16, we see significant improvements in almost all cases. Only DeepLabV3 seems to suffer sometimes with respect to error 2 (incorrect patches), especially for larger datasets. However, the values for the baseline are already quite small and only worsen by up to 0.8 % (absolute). The combination of methods led U-Net-Res to completely erase error 1 (topological disorder). Also, error 3 (discontinuous myocardium) was reduced massively, completely vanishing for 640 and 1280 training images. In the case of 1280 training images, DeepLabV3 got rid of almost all errors. Just error 2 remained with a frequency of 1.2 %. But also U-Net-Res reached very small error rates with only 2 % remaining for error 2, 0.6 % remaining for error 4 (myocardium placed around atrium), and 0.2 % for error 5 (endocardium and atrium partially confused).

6.7.4 Neck Muscle Segmentation

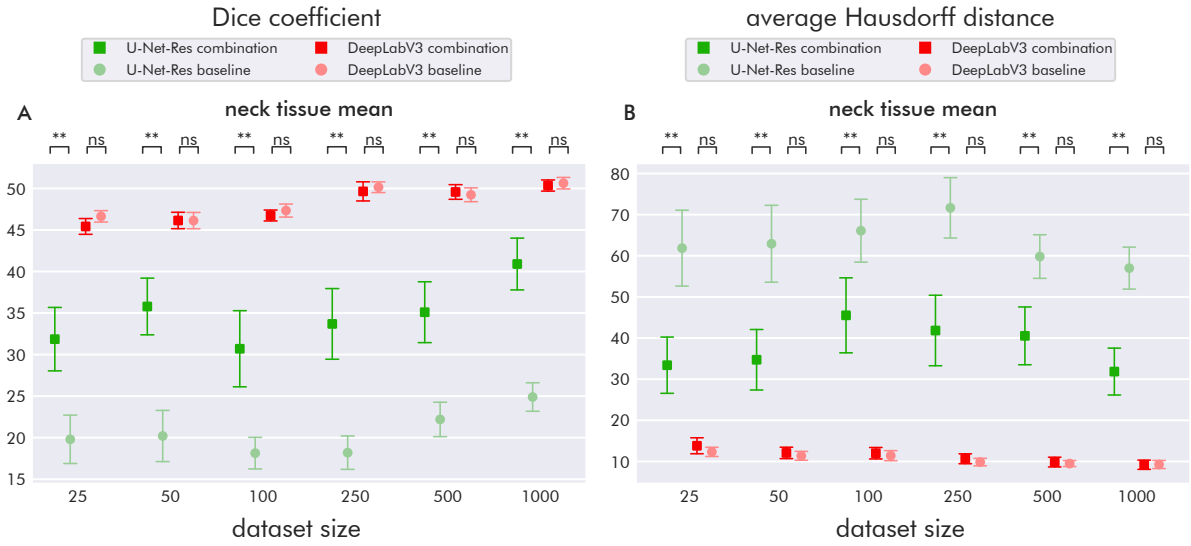

Figure 6.51: Neck muscle segmentation results using a combination of methods. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

Figure 6.51 shows the results of neck muscle segmentation when employing a combination of SEST and ICA shape prior. We observe no impact on DeepLabV3. However, the performance of U-Net-Res was improved by up to 40 %. Still, U-Net-Res was not able to reach the performance of DeepLabV3. All in all, the performance of both CNNs on the neck muscle dataset is still far away from being useful.

6.7.5 Summary and Discussion

While the improvements on the cardiac dataset are almost entirely statistically significant, this is not the case for the IVUS datasets. Here, we see that U-Net-Res does not outperform the baseline with respect to vessel wall Dice score for smaller datasets. Instead, we see improvements for larger dataset sizes, where values similar to DeepLabV3 are reached. In the case of DeepLabV3 and lumen segmentation, the improvements vanish for the largest dataset size. This indicates that our proposed methods cannot improve IVUS lumen and vessel wall segmentation results for datasets of 250 images and larger. However, the error rates 1 to 4 were still reduced for 250 training images. Partially even quite largely (errors 2 and 3). It is therefore likely that our methods could be beneficial with regard to error rates even if the IVUS lumen and vessel wall dataset is further extended.

The same basically holds for the IVUS calcium dataset. Again, combining our presented methods improved both error rates for 500 training images. Moreover, the segmentation metrics for the largest dataset were also improved. Therefore, we can assume that combining our methods could also benefit even larger IVUS calcium datasets. The performance dip of U-Net-Res at 50 training images corresponds to the dip in synthetic data augmentation with the same settings and is likely induced by an adverse bias in the synthetic dataset. The same holds for DeepLabV3's large improvement in average Hausdorff distance for 500 training images.

While the improvements on the cardiac dataset are the largest for the smallest training datasets, they are still present in the cases of 1280 training images. Especially for U-Net-Res. Only atrium segmentation by DeepLabV3 does not improve significantly for 1280 training images. While in the original baseline DeepLabV3 usually performed better than U-Net-Res, it is now often the other way around. U-Net-Res even outperforms DeepLabV3 on the largest training datasets, with the atrium Hausdorff distance being the only exception. We can therefore conclude that the combination of the presented methods has a greater positive impact on U-Net-Res than on DeepLabV3 in the case of cardiac segmentation.

Regarding neck muscle segmentation, we see that combining SEST and IC shape priors with U-Net-Res has only little benefit compared to IC shape priors alone. We see no effect on DeepLabV3, which shows that DeepLabV3 still generates average segmentation masks that do not adapt to the individual images. Therefore, the combination of both methods cannot extract additional information from the dataset.

At first glance, one could get the impression that the improvements in the combination of methods originate solely from synthetic data augmentation, especially since the improvements by performing synthetic data augmentation were similarly large. However, this is only true for some cases as Figure 6.52, Figure 6.53 and Figure 6.54 show. U-Net-Res, in particular, benefits from the other methods, too. DeepLabV3, on the other hand, did often reach maximum performances when solely employing synthetic data augmentation. Exceptions are endocardium and myocardium segmentation, as well as some other scattered instances. However, the error rates mostly improved anyway. Examples are error 2 (incorrect patches) and error 4 (myocardium placed around atrium) of the IVUS lumen and vessel wall dataset and error 1 (false positives) of the IVUS calcium dataset. In the case of cardiac segmentation, the error rates mostly decrease for synthetic data augmentation and the combination of methods. However, the improvements for the latter are substantially larger. Since synthetic data augmentation improves the error rates relatively uniformly, it is impossible to determine whether some of the other methods are redundant to synthetic data augmentation. However, the impact of the

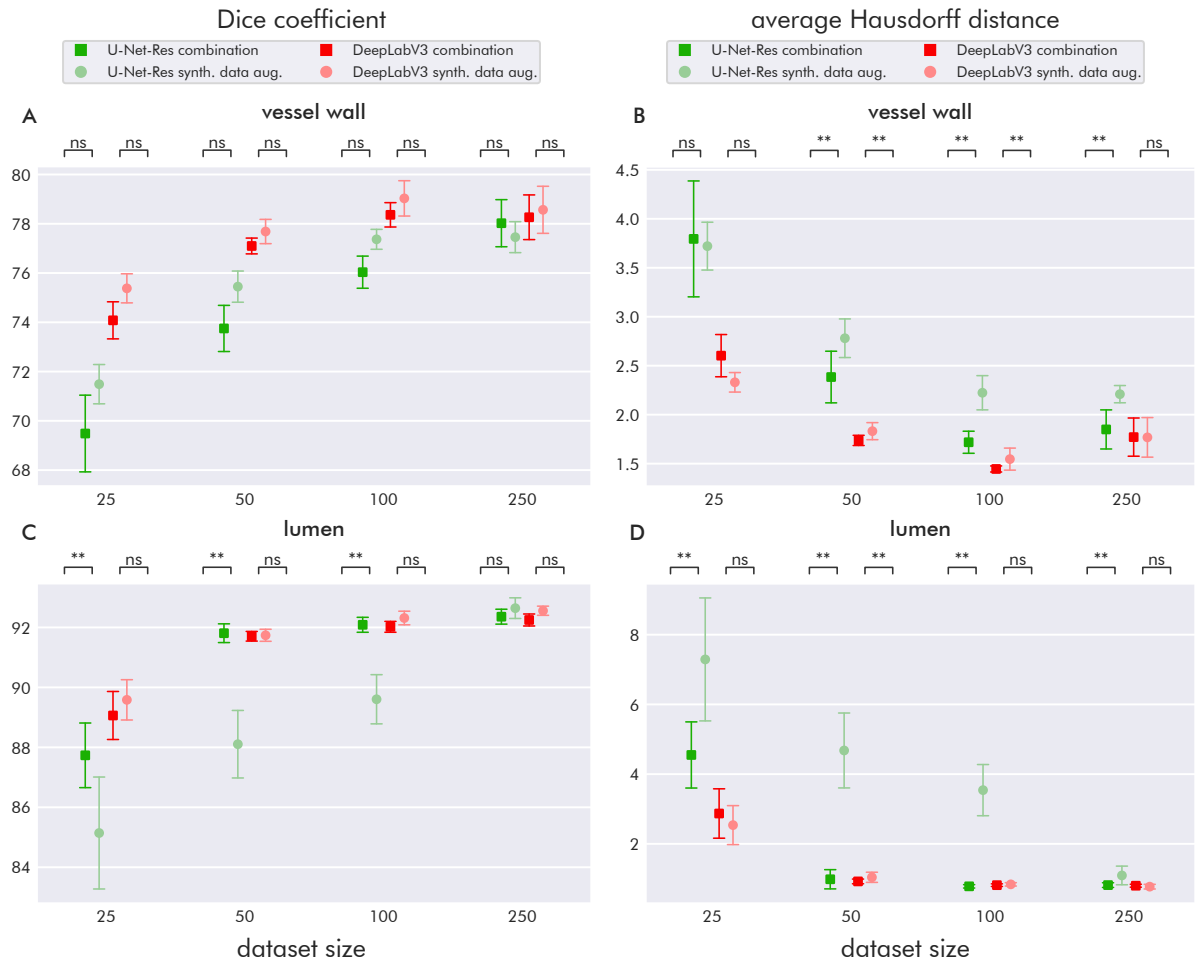

Figure 6.52: Comparison between the combination of methods and synthetic data augmentation regarding IVUS lumen and vessel wall segmentation. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

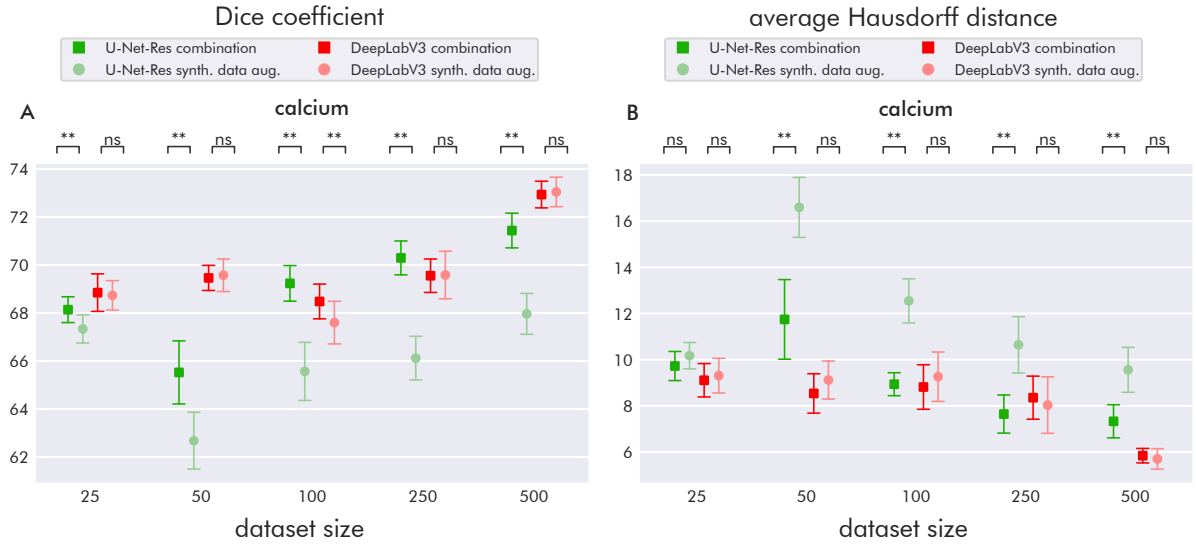

Figure 6.53: Comparison between the combination of methods and synthetic data augmentation regarding IVUS calcium segmentation. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

other methods will likely decrease if more synthetic data of sufficient quality is available for training. This requires that the synthetic data is similar to the real data in the sense that the synthetic data contains the information that the other methods are intended to take advantage of. This includes correct image textures for SEST, correct shapes for ICA shape priors, and correct topology for the containment loss. To test this, we would have to perform experiments with all possible combinations of the presented methods, which is not feasible within the scope of this work. We now want to give short answers to **RQ 2** for the combination of methods.

RQ 2.1: Which CNN architectures benefit from combining methods?
Both U-Net-Res and DeepLabV3 benefited strongly from combining the presented methods, both in terms of segmentation metrics and error rates. However, Figure 6.52, Figure 6.53 and Figure 6.54 show that DeepLabV3 gains its improvements mostly from synthetic data augmentation, while U-Net-Res does also benefit from the other methods.

RQ 2.2: How does combining methods perform as a function of dataset size?
Combining methods improves performance mostly for all dataset sizes. However, no performance improvement is seen for IVUS lumen and vessel wall segmentation with 250 images.

RQ 2.3: Which tissues benefit from combining methods?
Almost all tissues benefit from combining the methods. Exceptions are the vessel wall Dice score by U-Net-Res and neck muscle segmentation by DeepLabV3 in general.

RQ 2.4: What types of segmentation errors are reduced by combining methods?
Combining the presented methods reduced almost all error rates, even up to the largest dataset sizes. Exceptions are error 5 (background marked in lumen or vessel wall) of the IVUS lumen and vessel wall dataset, which slightly worsened for 250 training images, and error 2 (incorrect

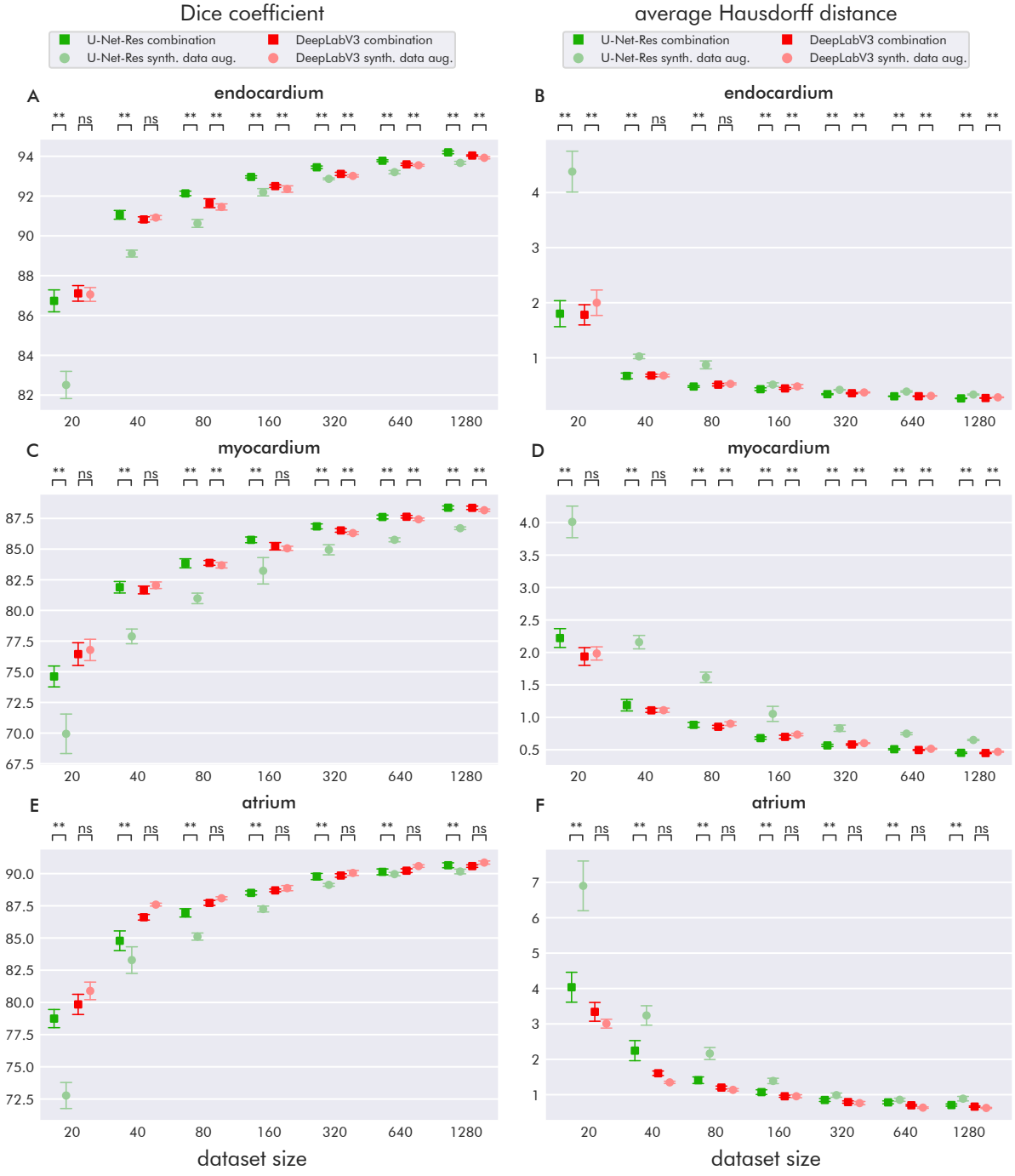

Figure 6.54: Comparison between the combination of methods and synthetic data augmentation regarding cardiac segmentation. Note the different scaling of the vertical axes. The error bars indicate a standard deviation.

patches) of the cardiac dataset, which further exacerbated for DeepLabV3 on the largest dataset sizes.

To summarize, combining the presented methods turned out to be very efficient. While the performance of DeepLabV3 was mainly improved only by synthetic data augmentation, U-Net-Res could reach performances beyond synthetic data augmentation by employing all methods at once. Both networks benefited strongly from combining all methods regarding error rates, which could be greatly reduced in most cases, even for the largest datasets.

7 Discussion

Our results show that each method has the potential to improve ultrasound image segmentation results for at least some settings. However, none of the methods led to improvements for all tissue classes, dataset sizes, and CNN architectures. Combining multiple methods largely increased the chance for improvements and significantly reduced all error rates. However, which methods to combine for a maximum outcome is likely highly dependent on the underlying dataset as well as the CNN architecture and has to be found through rigorous experiments. Although no statements valid for all investigated cases can be derived from the results, we will nevertheless identify and discuss some patterns that can be found among the results.

While research question **RQ 1** was answered in Chapter 4, we now want to discuss the results with respect to research question **RQ 2**. **RQ 2.1** to **RQ 2.4** have already been considered for every method individually in the corresponding discussion sections. Now we would like to answer the research questions, as far as possible, on a more general level. We will leave out the neck muscle dataset for now and discuss it in more detail later.

RQ 2.1: Which CNN architectures benefit from which methods?

Regarding segmentation metrics, the results indicate quite well that both U-Net-Res and DeepLabV3 benefit greatly from synthetic data augmentation. The other methods have a positive effect on U-Net-Res in particular, while DeepLabV3 does not benefit much from them. However, in terms of error rate reduction, both CNNs benefit from all methods.

Likely, this behavior occurs because DeepLabV3's baseline predictions are already quite robust in terms of correct topology compared to U-Net-Res. We discussed the reasons for this, e.g., DeepLabV3's inability to generate details smaller than 8 pixels, in Subsection 6.1.5. The methods other than synthetic data augmentation mainly aim at improving this robustness. SEST tries to draw information from image textures and thus highlights important regions via an attention map. The ICA shape priors aim at restricting possible tissue shapes via linearly combining ICs of the segmentation masks and using these for an attention mechanism. The containment loss aims at penalizing predictions that do not fulfill a containment condition of specific tissues, e.g., lumen surrounded by vessel wall or endocardium surrounded by myocardium and atrium. Hence, all these methods do not affect the predictions by DeepLabV3 very much. Since U-Net-Res does not exhibit this robustness in its baseline version, it benefits much more from these methods, even outperforming DeepLabV3 in some cases when combining all methods.

While the increased topological robustness of DeepLabV3 compared to U-Net-Res does, in many cases, lead to lower error rates, it does not automatically implicate better segmentation metrics. As we have seen in the case of neck muscle segmentation (compare Subsection 6.1.5), DeepLabV3 tends to generate average segmentation maps with small variability across images if the amount of texture information in the images is quite limited. We therefore assume that DeepLabV3 tends to generate segmentation masks that are topologically correct but somewhat

less accurate when the training dataset is rather small. This behavior could explain why the segmentation performance by DeepLabV3 is sometimes equal to or even worse than the performance of U-Net-Res, although the error rates are lower.

RQ 2.2: How do the presented methods perform as a function of dataset size?
In Chapter 1, we hypothesized that the smaller the training datasets, the greater the performance gains that would result from the proposed methods. The results show that this behavior occurs only in some cases, especially on the cardiac dataset. Here, synthetic data augmentation, combinations, and partially SEST show decreasing improvement with increasing dataset sizes. Another pattern is composed of small improvements or even declines for smaller datasets, larger improvements for medium-sized datasets, and, again, small improvements or declines for larger datasets. Examples are U-Net-Res SEST for IVUS lumen and vessel wall segmentation, DeepLabV3 ICA for endocardium and myocardium segmentation, and DeepLabV3 ICA for neck muscle segmentation. This pattern indicates that for smaller datasets, the additional information that was supposed to be drawn from the images by the affected methods did not contribute to the generalizability of the CNN. This effect is quite prominent in the IVUS datasets. Likely, the comparably low variability in these datasets contributes to this effect resulting in larger domain shifts between the training and test sets. This effect is particularly pronounced for the small dataset sizes. Moreover, the low variability in the IVUS datasets promotes greater variance in performance across different dataset sizes, as adding favorable or unfavorable training data with respect to the test set tends to be a matter of chance. In the case of the cardiac data set, this problem does not arise because the variability is much greater, and thus, the domain shift between training and test data is smaller.

The containment loss is an exception that led to equal or slightly worse results for small datasets and then improved the results for larger datasets. We found that this behavior originates in the method's ability to fix topological errors in test images that are far off the training domain, mainly due to poor quality.

RQ 2.3: Which tissues benefit from the presented methods?
We want to recapitulate briefly which kinds of tissues can benefit from the individual methods. Since SEST and, thus, wavelet scattering extracts mainly texture information from the images, it is suitable for tissues that show some kind of distinct texture, i.e., the IVUS lumen and vessel wall dataset and the cardiac dataset. IVUS calcium did not benefit, as the calcifications are relatively bright and do not exhibit a distinct texture. However, squeeze and excitation, in general, turned out to be a suitable tool for segmenting smaller structures like calcium. The ICA shape priors worked best for tissues with limited shape variability across the dataset, i.e., the vessel lumen. It turned out that ICA shape priors act rather restricting on shapes with more variability, like vessel walls or the cardiac tissues. The main reason is likely the small amount of ICs for small datasets making the shape prior rather rigid. The containment loss improved the results of IVUS lumen and vessel wall, as well as cardiac segmentation for larger datasets. Not only regarding the contained tissues but also the containing tissues, except for the atrium. Synthetic data augmentation was beneficial in virtually all tissues since our GAN architectures were able to generate meaningful images even for the smallest training datasets. Lastly, combining all methods improved the results on all datasets and tissues.

The methods' individual capabilities are strongly confounded with other factors. The most

important contributing factor is the dataset variability and, thus, the corresponding domain shift between training and test data, as we have explained above in the answer to **RQ 2.2**. Furthermore, the employed CNN architecture plays a major role in the way we discussed in the answer to **RQ 2.1**.

RQ 2.4: What types of segmentation errors are reduced by the presented methods?
Across all methods, the largest improvements in error rates were achieved in terms of discontinuous tubular structures like the vessel wall or myocardium (error 3). Furthermore, topological disorder (error 1 in the IVUS lumen and vessel wall as well as the cardiac dataset) by U-Net-Res was usually improved quite strongly. Basically, all methods had the tendency to increase topological robustness, even for DeepLabV3.

Synthetic data augmentation and the combination of methods, in particular, reduced basically all error rates, also for the largest dataset sizes. This indicates that these methods will likely improve error rates for even larger datasets.

Comments on the neck muscle dataset We have seen that the neck muscle dataset differs severely from the other datasets, not only in terms of visual appearance but also in terms of the results. While we could not improve the performance of DeepLabV3, synthetic data augmentation even worsened the results drastically, likely due to the artifacts produced by the conditional segmentation masks (compare Subsection 6.6.5). U-Net-Res benefited from the ICA shape priors and marginally from SEST, still reaching only half the performance of DeepLabV3. Furthermore, the performances were rather constant across various dataset sizes.

We found that DeepLabV3 could only outperform U-Net-Res because it learned an average segmentation mask with a mostly correct topology that was quite rigid across images and did not adapt to different tissue shapes. The baseline version of U-Net-Res only produced topological disorder. However, especially ICA shape priors helped U-Net-Res to approach DeepLabV3's behavior of generating average segmentation masks with improved topological correctness.

Loram et al. [162] showed that a cascade of multiple methods, including neural networks, can outperform our end-to-end CNN approaches. However, they used more training data and additionally employed a second dataset. We therefore believe that the neck muscle dataset, as we used it in this work, does not contain any further potential to improve the segmentation performance beyond the extent of our methods.

The above shows that the CNNs are not able to draw sufficient amounts of information from the individual tissue classes, especially from the deep ones. These are, in many cases, barely visible even to the eye. Additionally, it is debatable whether the ground truth is sufficiently accurate since registration errors between ultrasound images and corresponding MRI images are unknown [53, 54]. Furthermore, annotators used general shape patterns from anatomical atlases to complete invisible tissue boundaries [53, 54], which likely led to large shape biases in the dataset.

In summary, the neck muscle dataset, as it was presented in this work, does not seem to provide enough information to facilitate segmentation with an end-to-end CNN approach.

Relevance of the results in terms of clinical applicability
Whether a particular segmentation algorithm performs sufficiently well depends strongly on

the underlying application. If the segmentations are to be used for calculating certain quantities, e.g., object diameter or cross-sectional area, one would have to define a maximum error or variance that should not be exceeded on a given test set. For qualitative applications, the requirements do not necessarily have to be that strict. If one is mainly interested in the shape of an object for supporting a diagnosis, e.g., for breast mass or prostate assessment, the requirements could be less strict, as precise geometric quantities are not always needed. The same is true for segmentation as a pre-processing step for further analyses, e.g., for limiting texture classification to the object of interest only, not the whole image. Another application that does not require large spatial accuracy is localization. Here, the exact detection of boundaries is not necessary since only the presence of an object should be emphasized through a bounding box.

Every application faces its individual trade-off between the amount of available data and the required precision of the predictions. The results show that our methods have the potential to greatly improve the segmentation performance both in terms of segmentation metrics and error rates. Therefore, these methods can help to achieve the required precision with a given amount of data. We think that applications that especially use segmentation masks in a qualitative manner can benefit from strongly reduced topological error rates and, thus, from our presented methods.

We now want to comment on each of our investigated datasets concerning clinical applicability. In our opinion, only cardiac segmentation reaches performances that could be practically useful. As we saw in Subsection 6.4.3 for the largest datasets, the errors leading to deteriorated segmentation metrics are concentrated in a small number of images with poor quality or with an appearance far from the training domain. For images with high or average quality, the CNNs were able to generate segmentation masks similar to the manual annotations. Therefore, automatic calculation of ejection fraction or global longitudinal strain (compare Section 5.2) would reach a precision similar to values calculated with manual segmentations. Of course, physicians should have the opportunity to manually correct some parts of the automatic segmentation.

In the case of IVUS lumen and vessel wall segmentation, we do not see this opportunity when restricted to the dataset used in this work. However, the performance reached for lumen segmentation is already quite impressive. The segmentation errors are more distributed across images, especially in terms of the vessel wall. The predictions would therefore need more manual correction compared to the cardiac dataset in order to reach a precision that allows for robust estimation of lumen diameter or vessel wall thickness.

The results on the IVUS calcium dataset are very far from clinically relevant. Since the calcifications are often rather small compared to the whole image, small deviations from the ground truth lead to large deteriorations of the segmentation metrics. The CNNs are still rather prone to confusing other bright areas of the vessel with calcium. Less bright calcifications are still often not recognized as such. Moreover, we did not consider ambiguities between stent struts and calcium in this work, as stents do not appear in the IVUS calcium dataset.

To improve the results on the datasets in this work until they are suitable for clinical applications, we either need more data or further research (or maybe both). Ideas about further improvements to our methods are discussed below.

Recommendations for employing the presented methods

For this paragraph, we assume that a certain amount of data is already available, i.e., annotated and pre-processed, and thus ready for CNN training. First and foremost, one has to ensure

that the data is of sufficient quality. This includes minimal intra-observer variability, i.e., the labels are consistent across images, as well as a reasonably small domain shift between the training data and the data that is seen during deployment. Second, every cunning method can usually be outperformed by just extending the dataset. Therefore, one should determine whether acquiring additional data is possible with reasonable effort. If these possibilities are exhausted, it is time to take care of the deep learning methods.

Based on our findings, U-Net-Res (or generally an encoder-decoder architecture) should be used if fine-scaled details are important, while topological robustness does not play a major role. DeepLabV3, on the other hand, is the means of choice if the focus lies on topological robustness without the need to resolve small details. Our results show that synthetic data augmentation did virtually always improve the results greatly and is therefore a reasonable approach. If variable speckle noise across the synthetic images is probably important for the underlying application, one should employ speckleGAN. Otherwise, an ordinary GAN could likely also do the job. However, if severe mode collapse occurs, one should try using speckleGAN. To recognize possible adverse biases in the synthetic datasets, we highly recommend generating multiple synthetic datasets from multiple GANs and testing their capabilities to improve segmentation performance individually.

If U-Net-Res was chosen initially, it is reasonable to add the other methods we presented in this work and test whether the performance improves. Testing all possible combinations of methods would reveal the distinct optimum. However, if the given resources do not allow for that strategy, one should add methods greedily, starting with the most promising method. Employing methods other than synthetic data augmentation is particularly important if the error rates associated with the underlying dataset are still high despite augmenting the dataset with synthetic images. However, which methods lead to improved performance highly depends on the underlying dataset (see the answer to **RQ 2.3** above). If DeepLabV3 was chosen initially, it is likely that the other methods do not contribute much to improvements in the segmentation metrics. Nevertheless, they still have the potential to improve error rates.

If the overall results are promising and indicate possible clinical deployment, it is reasonable to put more effort into collecting more high-quality data and thus pushing performance to its limits. In the case of a clinical application, post-processing the segmentation masks is extremely important but not the scope of this work. See Gonzalez and Woods [82] and Jähne [119] for further information.

Directions of further research on the presented methods

The presented methods provide room for further experiments and development. In this paragraph, we want to give some exemplary directions for further research on the individual methods.

Besides the biorthogonal dual-tree wavelet transformation that we employed, many other wavelets exist that could be suitable for SEST. Likely, different wavelets are favorable for different datasets. It would be interesting to find whether certain systematic relationships exist between wavelets and dataset properties. Furthermore, the segmentation performance could depend on the positions SEST is inserted into the CNN, e.g., adding SEST blocks only to the top level, the encoding path, or the decoding path of U-Net-Res, probably affects the results. These questions could be answered with further experiments.

Regarding the ICA shape priors, the results show that the benefits of this approach decline

if the shape variability increases. Further experiments could systematically investigate which level of shape variability still allows this approach to work. The problem of shape variability could potentially be tackled by employing conventional data augmentation for the segmentation masks before calculating the ICs. Another possibility to leverage the ICA shape priors could include feeding intermediate results of the secondary branch into different levels of the decoding path of U-Net-Res. In this case, concatenation or summing could be more beneficial than the attention mechanism we used in this work.

The containment loss can only be considered if the tissue topology allows it. New approaches that generalize this idea to arbitrary topological relationships would be quite valuable, i.e., a loss function that penalizes predictions in which the myocardium is adjacent to the atrium. In that way, a larger set of conditions could be defined and converted into corresponding loss functions that push the CNN toward generating segmentation masks that match these conditions.

Synthetic data augmentation turned out to be a powerful method, regardless of with or without speckleGAN. We saw that the virtually infinite variability of speckle noise that speckleGAN provides in contrast to the baseline GAN did not affect segmentation performance. Further experiments could therefore investigate whether the speckle variability provided by speckleGAN positively influences image classification rather than segmentation.

In general, all presented methods can readily be transferred to 3D images. Moreover, SEST, ICA shape priors, and the containment loss can also be applied to other imaging modalities like CT or MRI. The same holds for synthetic data augmentation if the speckle layer is removed or replaced with a layer that adds characteristic noise of the underlying imaging modality. Importantly, such a layer has to be differentiable in order to facilitate the CNN to learn the noise-affecting parameters.

Besides enhancing the presented methods, another possibility to improve segmentation performance on small datasets by means of these methods is by integrating them into a larger image processing pipeline. Let's take the IVUS calcium dataset as an example. For such datasets with the objects of interest being rather small, localization instead of segmentation would probably be much easier. Applying a segmentation algorithm only to the resulting bounding boxes could increase the segmentation performance and reduce false positives. Furthermore, providing images with bounding boxes instead of segmentation masks is easier, facilitating the creation of larger datasets. Another approach to improve calcium segmentation is by first segmenting the vessel wall and then restricting calcium segmentation to this area, as calcifications only occur in the vessel wall.

Finally, applying an appropriate post-processing pipeline can greatly impact the resulting segmentation performance. Since discussing the large variety of post-processing methods is not feasible in the context of this work, we refer the reader to Gonzalez and Woods [82] and Jähne [119] for further details.

8 Conclusion

The goal of this work was to take a step towards solving the problem of data scarcity in the medical domain, which still limits the possibilities of automated image processing with convolutional neural networks (CNNs), especially regarding image segmentation. We have contributed to this field by developing and investigating methods extending vanilla CNNs to improve segmentation performance on small ultrasound datasets. Our results are relevant since they show that each method has the potential to improve ultrasound image segmentation results for different cases.

The proposed methods try to tackle the problem of inefficient filters that are learned by CNNs when trained with only small amounts of data. Inefficient filters are unable to extract meaningful features from the images, leading to poor generalizability and robustness with respect to unseen data. Our methods include squeeze and excitation with scattering transformation (SEST), incorporating shape priors via independent component analysis (ICA) into CNNs, containment loss for penalizing incorrect topologies in predicted segmentation masks, and synthetic data augmentation with speckleGAN, a generative adversarial network (GAN) that comprises a learnable speckle layer which adds random speckle to feature maps.

We tested each method with 2 CNN architectures, U-Net-Res and DeepLabV3, on 4 different datasets and varying amounts of training images. The datasets included two intravascular ultrasound (IVUS) datasets for lumen and vessel wall segmentation, as well as calcium segmentation, a cardiac segmentation dataset, and a neck muscle segmentation dataset. We found that none of the methods improves performance in all cases. Instead, suitable combinations of methods for a certain dataset have to be found through rigorous experiments. Therefore, we understand our methods as a toolbox from which individual tools can be taken, depending on their applicability to a particular problem.

Regarding segmentation metrics, U-Net-Res benefited from all methods, while DeepLabV3 benefited mainly from synthetic data augmentation. Most methods tended to decrease improvements with increasing dataset size, while the containment loss showed the opposite behavior. Whether a certain tissue class benefited from the methods was highly dependent on the method itself and the visual properties of the tissue. All methods helped to reduce topological segmentation errors like discontinuous tubular structures (vessel wall, myocardium), in some cases even for the largest dataset sizes.

Synthetic data augmentation turned out to be the most powerful approach, also for small datasets with less than 50 images. The multi-scale discriminator architecture was an important factor in generating visually appealing synthetic images from very small datasets. Incorporating the speckle layer into the generator was beneficial for generating images with larger speckles, as in IVUS images. U-Net-Res benefited, in particular, from combining synthetic data augmentation with the other methods both in terms of segmentation metrics and error rates.

Besides enhancing our presented methods (see Chapter 7), future research should generally focus more on methods designed for small datasets since this is an inherent problem in the

medical field. While future datasets of quite prevalent pathologies may be created with massive efforts, i.e., laborious work by dedicated experts over a long time, datasets of non-prevalent pathologies will remain small or grow very slowly. Hence, suitable methods to leverage such datasets are needed. Our work is a valuable step in this direction.

Bibliography

[1] Abou, R., Bijl, P. van der, Bax, J. J., and Delgado, V. "Global longitudinal strain: clinical use and prognostic implications in contemporary practice". *Heart* 106.18 (2020), pp. 1438–1444.

[2] Adlam, D., Tweet, M. S., Gulati, R., Kotecha, D., Rao, P., Moss, A. J., and Hayes, S. N. "Spontaneous Coronary Artery Dissection: Pitfalls of Angiographic Diagnosis and an Approach to Ambiguous Cases". *JACC: Cardiovascular Interventions* 14.16 (2021), pp. 1743–1756.

[3] Ahn, C. Y., Jung, Y. M., Kwon, O. I., and Seo, J. K. "Fast segmentation of ultrasound images using robust Rayleigh distribution decomposition". *Pattern Recognition* 45.9 (2012), pp. 3490–3500.

[4] Akagi, R. and Kusama, S. "Comparison Between Neck and Shoulder Stiffness Determined by Shear Wave Ultrasound Elastography and a Muscle Hardness Meter". *Ultrasound in Medicine & Biology* 41.8 (2015), pp. 2266–2271.

[5] Akbari, H. and Fei, B. "3D ultrasound image segmentation using wavelet support vector machines". *Medical Physics* 39.6Part1 (2012), pp. 2972–2984.

[6] Albanese, A., Abbruzzese, G., Dressler, D., Duzynski, W., Khatkova, S., Marti, M. J., Mir, P., Montecucco, C., Moro, E., Pinter, M., Relja, M., Roze, E., Skogseid, I. M., Timerbaeva, S., and Tzoulis, C. "Practical guidance for CD management involving treatment of botulinum toxin: a consensus statement". *Journal of Neurology* 262.10 (2015), pp. 2201–2213.

[7] Ali, Y., Beheshti, S., and Janabi-Sharifi, F. "Echocardiogram segmentation using active shape model and mean squared eigenvalue error". *Biomedical Signal Processing and Control* 69 (2021), p. 102807.

[8] Ali, Y., Janabi-Sharifi, F., and Beheshti, S. "Echocardiographic image segmentation using deep Res-U network". *Biomedical Signal Processing and Control* 64 (2021), p. 102248.

[9] Alsinan, A. Z., Rule, C., Vives, M., Patel, V. M., and Hacihaliloglu, I. "GAN-Based Realistic Bone Ultrasound Image and Label Synthesis for Improved Segmentation". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*. 2020, pp. 795–804.

[10] Araki, T., Banchhor, S. K., Londhe, N. D., Ikeda, N., Radeva, P., Shukla, D., Saba, L., Balestrieri, A., Nicolaides, A., Shafique, S., Laird, J. R., and Suri, J. S. "Reliable and Accurate Calcium Volume Measurement in Coronary Artery Using Intravascular Ultrasound Videos". *Journal of Medical Systems* 40.3 (2016), p. 51.

[11] Araki, T., Ikeda, N., Dey, N., Acharjee, S., Molinari, F., Saba, L., Godia, E. C., Nicolaides, A., and Suri, J. S. "Shape-based approach for coronary calcium lesion volume measurement on intravascular ultrasound imaging and its association with carotid intima-media thickness". *Journal of Ultrasound in Medicine* 34.3 (2015), pp. 469–482.

[12] Athanasiou, L. S., Karvelis, P. S., Tsakanikas, V. D., Naka, K. K., Michalis, L. K., Bourantas, C. V., and Fotiadis, D. I. "A novel semiautomated atherosclerotic plaque characterization method using grayscale intravascular ultrasound images: Comparison with virtual histology". *IEEE Transactions on Information Technology in Biomedicine* 16.3 (2012), pp. 391–400.

[13] Badawy, S. M., Zidan, H. E., Mohamed, A. E.-N. A., Hefnawy, A. A., GadAllah, M. T., and El-Banby, G. M. "A Wavelet - Fuzzy Combination Based Approach for Efficient Cancer Characterization in Breast Ultrasound Images". In: *2021 International Conference on Electronic Engineering (ICEEM)*. 2021, pp. 1–8.

[14] Bahdanau, D., Cho, K., and Bengio, Y. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *3rd International Conference on Learning Representations (ICLR)*. 2015.

[15] Balocco, S., Gatta, C., Ciompi, F., Wahle, A., Radeva, P., Carlier, S., Unal, G., Sanidas, E., Mauri, J., Carillo, X., Kovarnik, T., Wang, C.-W., Chen, H.-C., Exarchos, T. P., Fotiadis, D. I., Destrempes, F., Cloutier, G., Pujol, O., Alberti, M., Mendizabal-Ruiz, E. G., Rivera, M., Aksoy, T., Downe, R. W., and Kakadiaris, I. A. "Standardized evaluation methodology and reference database for evaluating IVUS image segmentation". *Computerized Medical Imaging and Graphics* 38.2 (2014), pp. 70–90.

[16] Bamber, J. C. and Dickinson, R. J. "Ultrasonic B-scanning: a computer simulation". *Physics in Medicine & Biology* 25.3 (1980), pp. 463–479.

[17] Barbosa, D., Dietenbeck, T., Schaerer, J., D'hooge, J., Friboulet, D., and Bernard, O. "B-Spline Explicit Active Surfaces: An Efficient Framework for Real-Time 3-D Region-Based Segmentation". *IEEE Transactions on Image Processing* 21.1 (2012), pp. 241–251.

[18] Bargsten, L., Raschka, S., and Schlaefer, A. "Capsule networks for segmentation of small intravascular ultrasound image datasets". *International Journal of Computer Assisted Radiology and Surgery* 16.8 (2021), pp. 1243–1254.

[19] Bargsten, L., Riedl, K. A., Wissel, T., Brunner, F. J., Schaefers, K., Grass, M., Blankenberg, S., Seiffert, M., and Schlaefer, A. "Deep learning for calcium segmentation in intravascular ultrasound images". *Current Directions in Biomedical Engineering* 7.1 (2021), pp. 96–100.

[20] Bargsten, L., Riedl, K. A., Wissel, T., Brunner, F. J., Schaefers, K., Grass, M., Blankenberg, S., Seiffert, M., and Schlaefer, A. "Attention via Scattering Transforms for Segmentation of Small Intravascular Ultrasound Data Sets". In: *Proceedings of the Fourth Conference on Medical Imaging with Deep Learning*. Vol. 143. Proceedings of Machine Learning Research. PMLR, 2021, pp. 34–47.

[21] Bargsten, L., Riedl, K. A., Wissel, T., Brunner, F. J., Schaefers, K., Sprenger, J., Grass, M., Seiffert, M., Blankenberg, S., and Schlaefer, A. "Tailored methods for segmentation of intravascular ultrasound images via convolutional neural networks". In: *Medical Imaging 2021: Ultrasonic Imaging and Tomography*. Vol. 11602. 2021, pp. 1–7.

[22] Bargsten, L. and Schlaefer, A. "SpeckleGAN: a generative adversarial network with an adaptive speckle layer to augment limited training data for ultrasound image processing". *International Journal of Computer Assisted Radiology and Surgery* 15.9 (2020), pp. 1427–1436.

[23] Barreiros, A., Chiorean, L., Braden, B., and Dietrich, C. "Ultrasound in Rare Diffuse Liver Disease". *Zeitschrift für Gastroenterologie* 52 (Nov. 2014), pp. 1247–1256.

[24] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. "Automatic Differentiation in Machine Learning: A Survey". *Journal of Machine Learning Research* 18.1 (2017), pp. 5595–5637.

[25] BenTaieb, A. and Hamarneh, G. "Topology Aware Fully Convolutional Networks for Histology Gland Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*. 2016, pp. 460–468.

[26] Bernard, O., Bosch, J. G., Heyde, B., Alessandrini, M., Barbosa, D., Camarasu-Pop, S., Cervenansky, F., Valette, S., Mirea, O., Bernier, M., Jodoin, P.-M., Domingos, J. S., Stebbing, R. V., Keraudren, K., Oktay, O., Caballero, J., Shi, W., Rueckert, D., Milletari, F., Ahmadi, S.-A., Smistad, E., Lindseth, F., Stralen, M. van, Wang, C., Smedby, Ö., Donal, E., Monaghan, M., Papachristidis, A., Geleijnse, M. L., Galli, E., and D'hooge, J. "Standardized Evaluation System for Left Ventricular Segmentation Algorithms in 3D Echocardiography". *IEEE Transactions on Medical Imaging* 35.4 (2016), pp. 967–977.

[27] Binder, T., Süssner, M., Moertl, D., Strohmer, T., Baumgartner, H., Maurer, G., and Porenta, G. "Artificial neural networks and spatial temporal contour linking for automated endocardial contour detection on echocardiograms: a novel approach to determine left ventricular contractile function". *Ultrasound in Medicine & Biology* 25.7 (1999), pp. 1069–1076.

[28] Bjorkkvist, J. E., Peterson, G., and Peolsson, A. "Ultrasound Investigation of Dorsal Neck Muscle Deformation During a Neck Rotation Exercise". *Journal of Manipulative and Physiological Therapeutics* 43.9 (2020), pp. 864–873.

[29] Boutillon, A., Borotikar, B., Burdin, V., and Conze, P.-H. "Combining Shape Priors with Conditional Adversarial Networks for Improved Scapula Segmentation in Mr Images". In: *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. 2020, pp. 1164–1167.

[30] Bowles, C., Chen, L., Guerrero, R., Bentley, P., Gunn, R., Hammers, A., Dickie, D. A., Hernández, M. V., Wardlaw, J., and Rueckert, D. "GAN Augmentation: Augmenting Training Data using Generative Adversarial Networks". *ArXiv* 1810.10863 (2018).

[31] Breiman, L. "Random Forests". *Machine Learning* 45.1 (2001), pp. 5–32.

[32] Bruna, J. and Mallat, S. "Invariant Scattering Convolution Networks". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1872–1886.

[33] Buoso, S., Joyce, T., and Kozerke, S. "Personalising left-ventricular biophysical models of the heart using parametric physics-informed neural networks". *Medical Image Analysis* 71 (2021), p. 102066.

[34] Burckhardt, C. "Speckle in ultrasound B-mode scans". *IEEE Transactions on Sonics and Ultrasonics* 25.1 (1978), pp. 1–6.

[35] Chai, Y., Liu, H., and Xu, J. "Glaucoma diagnosis based on both hidden features and domain knowledge through deep learning models". *Knowledge-Based Systems* 161 (2018), pp. 147–156.

[36] Chen, C., Qin, C., Qiu, H., Tarroni, G., Duan, J., Bai, W., and Rueckert, D. "Deep Learning for Cardiac Image Segmentation: A Review". *Frontiers in Cardiovascular Medicine* 7 (2020), p. 25.

[37] Chen, C., Wang, Y., Niu, J., Liu, X., Li, Q., and Gong, X. "Domain Knowledge Powered Deep Learning for Breast Cancer Diagnosis Based on Contrast-Enhanced Ultrasound Videos". *IEEE Transactions on Medical Imaging* 40.9 (2021), pp. 2439–2451.

[38] Chen, D. R., Chang, R. F., Kuo, W. J., Chen, M. C., and Huang, Y. L. "Diagnosis of breast tumors with sonographic texture analysis using wavelet transform and neural networks". *Ultrasound in Medicine & Biology* 28.10 (2002), pp. 1301–1310.

[39] Chen, F., Ma, R., Liu, J., Zhu, M., and Liao, H. "Lumen and media-adventitia border detection in IVUS images using texture enhanced deformable model". *Computerized Medical Imaging and Graphics* 66.July 2017 (2018), pp. 1–13.

[40] Chen, J., You, H., and Li, K. "A review of thyroid gland segmentation and thyroid nodule segmentation methods for medical ultrasound images". *Computer Methods and Programs in Biomedicine* 185 (2020), p. 105329.

[41] Chen, L., Bentley, P., Mori, K., Misawa, K., Fujiwara, M., and Rueckert, D. "Self-supervised learning for medical image analysis using image context restoration". *Medical Image Analysis* 58 (2019), p. 101539.

[42] Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2018), pp. 834–848.

[43] Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. "Rethinking Atrous Convolution for Semantic Image Segmentation". *ArXiv* 1706.05587 (2017).

[44] Chen, Y., Li, D., Zhang, X., Jin, J., and Shen, Y. "Computer aided diagnosis of thyroid nodules based on the devised small-datasets multi-view ensemble learning". *Medical Image Analysis* 67 (2021), p. 101819.

[45] Cheng, H. D., Jiang, X. H., Sun, Y., and Wang, J. "Color image segmentation: Advances and prospects". *Pattern Recognition* 34.12 (2001), pp. 2259–2281.

[46] China, D., Illanes, A., Poudel, P., Friebe, M., Mitra, P., and Sheet, D. "Anatomical Structure Segmentation in Ultrasound Volumes Using Cross Frame Belief Propagating Iterative Random Walks". *IEEE Journal of Biomedical and Health Informatics* 23.3 (2019), pp. 1110–1118.

[47] Chitradevi, A. and Sadasivam, V. "Various Approaches for Medical Image Segmentation: A Survey". *Current Medical Imaging* 12.2 (2016), pp. 77–94.

[48] Contarino, M. F., Smit, M., Dool, J. van den, Volkmann, J., and Tijssen, M. A. "Unmet needs in the management of cervical dystonia". *Frontiers in Neurology* 7 (2016), p. 165.

[49] Cortes, C. and Vapnik, V. N. "Support-Vector Networks". *Machine Learning* 20.3 (1995), pp. 273–297.

[50] Cotter, F. *The Learnable ScatterNet*. `https://github.com/fbcotter/scatnet_learn`. Accessed: 2020-10-29. 2019.

[51] Cotter, F. and Kingsbury, N. "A Learnable Scatternet: Locally Invariant Convolutional Layers". In: *IEEE International Conference on Image Processing (ICIP)*. 2019, pp. 350–354.

[52] Cronin, N. J., Finni, T., and Seynnes, O. "Using deep learning to generate synthetic B-mode musculoskeletal ultrasound images". *Computer Methods and Programs in Biomedicine* 196 (2020), p. 105583.

[53] Cunningham, R. J., Harding, P. J., and Loram, I. D. "Real-Time Ultrasound Segmentation, Analysis and Visualisation of Deep Cervical Muscle Structure". *IEEE Transactions on Medical Imaging* 36.2 (2017), pp. 653–665.

[54] Cunningham, R. J., Sánchez, M. B., and Loram, I. D. "Ultrasound segmentation of cervical muscle during head motion: A dataset and a benchmark using deconvolutional neural networks". *engrXiv* (Feb. 2019).

[55] Darmoch, F., Alraies, M. C., Al-Khadra, Y., Pacha, H. M., Pinto, D. S., and Osborn, E. A. "Intravascular Ultrasound Imaging-Guided Versus Coronary Angiography-Guided Percutaneous Coronary Intervention: A Systematic Review and Meta-Analysis". *Journal of the American Heart Association* 9.5 (2020), e013678.

[56] Dash, T., Chitlangia, S., Ahuja, A., and Srinivasan, A. "A Review of Some Techniques for Inclusion of Domain-Knowledge into Deep Neural Networks". *ArXiv* 2107.10295 (2021).

[57] Deng, J., Dong, W., Socher, R., Li, L., Kai Li, and Li Fei-Fei. "ImageNet: A large-scale hierarchical image database". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 248–255.

[58] Dice, L. R. "Measures of the Amount of Ecologic Association Between Species". *Ecology* 26.3 (1945), pp. 297–302.

[59] Dilna, K. T. and Jude Hemanth, D. "Fibroid Segmentation in Ultrasound Uterus Images Using Wavelet Filter and Active Contour Model". In: *Data Analytics and Management*. 2021, pp. 509–517.

[60] Dokur, Z. and Ölmez, T. "Segmentation of Ultrasound Images by Using a Hybrid Neural Network". *Pattern Recognition Letters* 23.14 (2002), pp. 1825–1836.

[61] Dong, L., Jiang, W., Lu, W., Jiang, J., Zhao, Y., Song, X., Leng, X., Zhao, H., Wang, J., Li, C., and Xiang, J. "Automatic segmentation of coronary lumen and external elastic membrane in intravascular ultrasound images using 8-layer U-Net". *BioMedical Engineering Online* 20.1 (2021), p. 16.

[62] Donnez, M., Carton, F.-X., Lann, F. L., Schlichting, E. de, and Chabanas, M. "Realistic synthesis of brain tumor resection ultrasound images with a generative adversarial network". *Medical Imaging 2021: Image-Guided Procedures, Robotic Interventions, and Modeling* 11598 (2021), pp. 1–6.

[63] Dössel, O. *Bildgebende Verfahren in der Medizin*. Springer Vieweg, 2016.

[64] Dubuisson, M.-P. and Jain, A. "A modified Hausdorff distance for object matching". In: *Proceedings of 12th International Conference on Pattern Recognition*. Vol. 1. 1994, pp. 566–568.

[65] Duck, F. and Martin, K. "A History of Medical Ultrasound Physics - Part I". *Medical Physics International* Special Issue.History of Medical Physics 5 (2021).

[66] Dumoulin, V. and Visin, F. "A guide to convolution arithmetic for deep learning". 1603.07285 (2016).

[67] El Jurdi, R., Petitjean, C., Honeine, P., Cheplygina, V., and Abdallah, F. "High-level prior-based loss functions for medical image segmentation: A survey". *Computer Vision and Image Understanding* 210 (2021), p. 103248.

[68] Escobar, M., Castillo, A., Romero, A., and Arbeláez, P. "UltraGAN: Ultrasound Enhancement Through Adversarial Generation". In: *Simulation and Synthesis in Medical Imaging*. 2020, pp. 120–130.

[69] Feder, J. M., Paredes, E. S. de, Hogge, J. P., and Wilken, J. J. "Unusual Breast Lesions: Radiologic-Pathologic Correlation". *RadioGraphics* 19 (1999), S11–S26.

[70] Finlayson, S. G., Lee, H., Kohane, I. S., and Oakden-Rayner, L. "Towards generative adversarial networks as a new paradigm for radiology education". *ArXiv* 1812.01547 (2018).

[71] Foster, B., Bagci, U., Mansoor, A., Xu, Z., and Mollura, D. J. "A review on segmentation of positron emission tomography images". *Computers in Biology and Medicine* 50 (2014), pp. 76–96.

[72] Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. "GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification". *Neurocomputing* 321 (2018), pp. 321–331.

[73] Fujieda, S., Takayama, K., and Hachisuka, T. "Wavelet Convolutional Neural Networks for Texture Classification". *ArXiv* 1707.07394 (2017).

[74] Fujioka, T., Kubota, K., Mori, M., Katsuta, L., Kikuchi, Y., Kimura, K., Kimura, M., Adachi, M., Oda, G., Nakagawa, T., Kitazume, Y., and Tateishi, U. "Virtual Interpolation Images of Tumor Development and Growth on Breast Ultrasound Image Synthesis With Deep Convolutional Generative Adversarial Networks". *Journal of Ultrasound in Medicine* 40.1 (2021), pp. 61–69.

[75] Fujioka, T., Mori, M., Kubota, K., Kikuchi, Y., Katsuta, L., Adachi, M., Oda, G., Nakagawa, T., Kitazume, Y., and Tateishi, U. "Breast ultrasound image synthesis using deep convolutional generative adversarial networks". *Diagnostics* 9.4 (2019), pp. 1–9.

[76] Fukushima, K. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". *Biological Cybernetics* 36.4 (1980), pp. 193–202.

[77] Gadermayr, M., Li, K., Müller, M., Truhn, D., Krämer, N., Merhof, D., and Gess, B. "Domain-specific data augmentation for segmenting MR images of fatty infiltrated human thighs with neural networks". *Journal of Magnetic Resonance Imaging* 49.6 (2019), pp. 1676–1683.

[78] Ganaye, P.-A., Sdika, M., Triggs, B., and Benoit-Cattin, H. "Removing segmentation inconsistencies with semi-supervised non-adjacency constraint". *Medical Image Analysis* 58 (2019), p. 101551.

[79] Gao, Y., Huang, R., Yang, Y., Zhang, J., Shao, K., Tao, C., Chen, Y., Metaxas, D. N., Li, H., and Chen, M. "FocusNetv2: Imbalanced large and small organ segmentation with adversarial shape constraint for head and neck CT images". *Medical Image Analysis* 67 (2021), p. 101831.

[80] Gao, Z., Chung, J., Abdelrazek, M., Leung, S., Hau, W. K., Xian, Z., Zhang, H., and Li, S. "Privileged Modality Distillation for Vessel Border Detection in Intracoronary Imaging". *IEEE Transactions on Medical Imaging* 39.5 (2020), pp. 1524–1534.

[81] Giannoglou, V. G., Stavrakoudis, D. G., Theocharis, J. B., and Petridis, V. "Genetic fuzzy rule based classification systems for coronary plaque characterization based on intravascular ultrasound images". *Engineering Applications of Artificial Intelligence* 38 (2015), pp. 203–220.

[82] Gonzalez, R. C. and Woods, R. E. *Digital Image Processing*. 4th ed. Pearson, 2018.

[83] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

[84] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems*. 2014, pp. 2672–2680.

[85] Goodman, J. W. *Introduction to Fourier Optics*. McGraw-Hill, 1968.

[86] Goodman, J. W. *Speckle Phenomena in Optics: Theory and Applications*. Roberts and Company Publishers, 2007.

[87] Gordillo, N., Montseny, E., and Sobrevilla, P. "State of the art survey on MRI brain tumor segmentation". *Magnetic Resonance Imaging* 31.8 (2013), pp. 1426–1438.

[88] Goudarzi, S., Asif, A., and Rivaz, H. "Fast Multi-Focus Ultrasound Image Recovery Using Generative Adversarial Networks". *IEEE Transactions on Computational Imaging* 6 (2020), pp. 1272–1284.

[89] Guo, F., Ng, M., Goubran, M., Petersen, S. E., Piechnik, S. K., Neubauer, S., and Wright, G. "Improving cardiac MRI convolutional neural network segmentation on small training datasets and dataset shift: A continuous kernel cut approach". *Medical Image Analysis* 61 (2020), p. 101636.

[90] Guo, L., Lei, B., Chen, W., Du, J., Frangi, A. F., Qin, J., Zhao, C., Shi, P., Xia, B., and Wang, T. "Dual attention enhancement feature fusion network for segmentation and quantitative analysis of paediatric echocardiography". *Medical Image Analysis* 71 (2021), p. 102042.

[91] Hagerty, J. R., Stanley, R. J., Almubarak, H. A., Lama, N., Kasmi, R., Guo, P., Drugge, R. J., Rabinovitz, H. S., Oliviero, M., and Stoecker, W. V. "Deep Learning and Handcrafted Method Fusion: Higher Diagnostic Accuracy for Melanoma Dermoscopy Images". *IEEE Journal of Biomedical and Health Informatics* 23.4 (2019), pp. 1385–1391.

[92] Hammouche, A., Cloutier, G., Tardif, J. C., Hammouche, K., and Meunier, J. "Automatic IVUS lumen segmentation using a 3D adaptive helix model". *Computers in Biology and Medicine* 107 (2019), pp. 58–72.

[93] Han, C., Murao, K., Satoh, S., and Nakayama, H. "Learning More with Less: GAN-based Medical Image Augmentation". *ArXiv* 1904.00838 (2019).

[94] Han, C., Rundo, L., Araki, R., Furukawa, Y., Mauri, G., Nakayama, H., and Hayashi, H. *Infinite Brain MR Images: PGGAN-Based Data Augmentation for Tumor Detection*. Springer Singapore, 2020, pp. 291–303.

[95] Han, L., Huang, Y., Dou, H., Wang, S., Ahamad, S., Luo, H., Liu, Q., Fan, J., and Zhang, J. "Semi-supervised segmentation of lesion from breast ultrasound images with attentional generative adversarial network". *Computer Methods and Programs in Biomedicine* 189 (2020), p. 105275.

[96] Handee, W. and Ritenour, E. *Medical Imaging Physics*. Wiley-Liss, 2002.

[97] Hangiandreou, N. J. "AAPM/RSNA Physics Tutorial for Residents: Topics in US". *RadioGraphics* 23.4 (2003), pp. 1019–1033.

[98] Hansson, M., Brandt, S. S., Lindström, J., Gudmundsson, P., Jujić, A., Malmgren, A., and Cheng, Y. "Segmentation of B-mode cardiac ultrasound data by Bayesian Probability Maps". *Medical Image Analysis* 18.7 (2014), pp. 1184–1199.

[99] Harikumar, R. and Vinoth kumar, B. "Performance analysis of neural networks for classification of medical images with wavelets as a feature extractor". *International Journal of Imaging Systems and Technology* 25.1 (2015), pp. 33–40.

[100] Hausdorff, F. *Grundzüge der Mengenlehre*. Leipzig: Veit & Comp., 1914.

[101] He, K., Zhang, X., Ren, S., and Sun, J. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[102] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". In: *Conference on Neural Information Processing Systems*. 2017, pp. 6626–6637.

[103] Hinton, G. E., Osindero, S., and Teh, Y.-W. "A Fast Learning Algorithm for Deep Belief Nets". *Neural Computation* 18.7 (2006), pp. 1527–1554.

[104] Hochreiter, S. and Schmidhuber, J. "Long Short-Term Memory". *Neural Computation* 9.8 (1997), pp. 1735–1780.

[105] Hsu, W. Y. "Automatic atrium contour tracking in ultrasound imaging". *Integrated Computer-Aided Engineering* 23.4 (2016), pp. 401–411.

[106] Hu, J., Shen, L., and Sun, G. "Squeeze-and-Excitation Networks". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7132–7141.

[107] Hu, Y., Gibson, E., Lee, L.-L., Xie, W., Barratt, D. C., Vercauteren, T., and Noble, J. A. "Freehand Ultrasound Image Simulation with Spatially-Conditioned Generative Adversarial Networks". In: *Molecular Imaging, Reconstruction and Analysis of Moving Body Organs, and Stroke Imaging and Treatment*. Springer International Publishing, 2017, pp. 105–115.

[108] Huang, L., Qin, J., Zhou, Y., Zhu, F., Liu, L., and Shao, L. "Normalization Techniques in Training DNNs: Methodology, Analysis and Application". *ArXiv* 2009.12836 (2020).

[109] Huang, X., Zhu, H., and Wang, J. "Adoption of Snake Variable Model-Based Method in Segmentation and Quantitative Calculation of Cardiac Ultrasound Medical Images". *Journal of Healthcare Engineering* 2021 (2021).

[110] Huang, Y., Yan, W., Xia, M., Guo, Y., Zhou, G., and Wang, Y. "Segmentation of Media and Lumen in Intravascular Ultrasound Image Using Guided Multiscale Normalized Cut". *Journal of Medical Imaging and Health Informatics* 9.7 (2019), pp. 1498–1504.

[111] Humeau-Heurtier, A. "Texture Feature Extraction Methods: A Survey". *IEEE Access* 7 (2019), pp. 8975–9000.

[112] Hwang, Y. N., Lee, J. H., Kim, G. Y., Shin, E. S., and Kim, S. M. "Characterization of coronary plaque regions in intravascular ultrasound images using a hybrid ensemble classifier". *Computer Methods and Programs in Biomedicine* 153 (2018), pp. 83–92.

[113] Iglesias, J. E. and Sabuncu, M. R. "Multi-atlas segmentation of biomedical images: A survey". *Medical Image Analysis* 24.1 (2015), pp. 205–219.

[114] Ioffe, S. and Szegedy, C. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". *ArXiv* 1502.03167 (2015).

[115] Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., Marklund, H., Haghgoo, B., Ball, R., Shpanskaya, K., Seekins, J., Mong, D. A., Halabi, S. S., Sandberg, J. K., Jones, R., Larson, D. B., Langlotz, C. P., Patel, B. N., Lungren, M. P., and Ng, A. Y. "CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison". *ArXiv* 1901.07031 (2019).

[116] Isola, P., Efros, A. A., Ai, B., and Berkeley, U. C. "Image-to-Image Translation with Conditional Adversarial Networks". *ArXiv* 1611.07004 (2016).

[117] Jacob, G., Noble, J., Behrenbruch, C., Kelion, A., and Banning, A. "A shape-space-based approach to tracking myocardial borders and quantifying regional left-ventricular function applied in echocardiography". *IEEE Transactions on Medical Imaging* 21.3 (2002), pp. 226–238.

[118] Jafari, M. H., Girgis, H., Van Woudenberg, N., Liao, Z., Rohling, R., Gin, K., Abolmaesumi, P., and Tsang, T. "Automatic biplane left ventricular ejection fraction estimation with mobile point-of-care ultrasound using multi-task learning and adversarial training". *International Journal of Computer Assisted Radiology and Surgery* 14.6 (2019), pp. 1027–1037.

[119] Jähne, B. *Digital Image Processing*. 6th ed. Springer, 2005.

[120] Jang, J. H., Kim, D. H., Yang, D. H., Woo, S. I., Kwan, J., Park, K. S., and Shin, S. H. "Spontaneous coronary artery dissection by intravascular ultrasound in a patient with myocardial infarction". *Korean Journal of Internal Medicine* 29.1 (2014), pp. 106–110.

[121] Jin, D., Xu, Z., Tang, Y., Harrison, A. P., and Mollura, D. J. "CT-realistic lung nodule simulation from 3D conditional generative adversarial networks for robust lung segmentation". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI)*. 2018, pp. 732–740.

[122] Jing, L. and Tian, Y. "Self-Supervised Visual Feature Learning with Deep Neural Networks: A Survey". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.11 (2021), pp. 4037–4058.

[123] Joel, T. and Sivakumar, R. "An extensive review on Despeckling of medical ultrasound images using various transformation techniques". *Applied Acoustics* 138 (2018), pp. 18–27.

[124] Johnson, T. W., Räber, L., Di Mario, C., Bourantas, C., Jia, H., Mattesini, A., Gonzalo, N., De La Torre Hernandez, J. M., Prati, F., Koskinas, K., Joner, M., Radu, M. D., Erlinge, D., Regar, E., Kunadian, V., Maehara, A., Byrne, R. A., Capodanno, D., Akasaka, T., Wijns, W., Mintz, G. S., and Guagliumi, G. "Clinical use of intracoronary imaging. Part 2: acute coronary syndromes, ambiguous coronary angiography findings, and guiding interventional decision-making: an expert consensus document of the European Association of Percutaneous Cardiovascular Interventions". *European Heart Journal* 40.31 (2019), pp. 2566–2584.

[125] Jost, W. H., Hefter, H., Stenner, A., and Reichel, G. "Rating scales for cervical dystonia: A critical evaluation of tools for outcome assessment of botulinum toxin therapy". *Journal of Neural Transmission* 120.3 (2013), pp. 487–496.

[126] Karimi, D., Warfield, S. K., and Gholipour, A. "Transfer learning in medical image segmentation: New insights from analysis of the dynamics of model parameters and learned representations". *Artificial Intelligence in Medicine* 116 (2021), p. 102078.

[127] Karpatne, A., Atluri, G., Faghmous, J. H., Steinbach, M., Banerjee, A., Ganguly, A., Shekhar, S., Samatova, N., and Kumar, V. "Theory-guided data science: A new paradigm for scientific discovery from data". *IEEE Transactions on Knowledge and Data Engineering* 29.10 (2017), pp. 2318–2331.

[128] Karras, T., Aila, T., Laine, S., and Lehtinen, J. "Progressive Growing of GANs for Improved Quality, Stability, and Variation". *ArXiv* 1710.10196 (2017).

[129] Katouzian, A., Angelini, E. D., Carlier, S. G., Suri, J. S., Navab, N., and Laine, A. F. "A State-of-the-Art Review on Segmentation Algorithms in Intravascular Ultrasound (IVUS) Images". *IEEE Transactions on Information Technology in Biomedicine* 16.5 (2012), pp. 823–834.

[130] Kaymak, B., Kara, M., Gürçay, E., and Özçakar, L. "Sonographic Guide for Botulinum Toxin Injections of the Neck Muscles in Cervical Dystonia". *Physical Medicine and Rehabilitation Clinics of North America* 29.1 (2018), pp. 105–123.

[131] Kazeminia, S., Baur, C., Kuijper, A., van Ginneken, B., Navab, N., Albarqouni, S., and Mukhopadhyay, A. "GANs for medical image analysis". *Artificial Intelligence in Medicine* 109 (2020), p. 101938.

[132] Kermani, A. and Ayatollahi, A. "A new nonparametric statistical approach to detect lumen and Media-Adventitia borders in intravascular ultrasound frames". *Computers in Biology and Medicine* 104 (2019), pp. 10–28.

[133] Khatami, A., Nazari, A., Beheshti, A., Nguyen, T. T., Nahavandi, S., and Zieba, J. "Convolutional Neural Network for Medical Image Classification using Wavelet Features". In: *2020 International Joint Conference on Neural Networks (IJCNN)*. 2020, pp. 1–8.

[134] Kim, G. Y., Lee, J. H., Hwang, Y. N., and Kim, S. M. "A novel intensity-based multi-level classification approach for coronary plaque characterization in intravascular ultrasound images". *BioMedical Engineering Online* 17.2 (2018), p. 151.

[135] Kim, M., Kang, T. W., Jang, K. M., Kim, Y. K., Kim, S. H., Ha, S. Y., Sinn, D. H., and Gu, S. "Tumefactive Gallbladder Sludge at US: Prevalence and Clinical Importance". *Radiology* 283.2 (2017), pp. 570–579.

[136] Kim, S., Jang, Y., Jeon, B., Hong, Y., Shim, H., and Chang, H. "Fully Automatic Segmentation of Coronary Arteries Based on Deep Neural Network in Intravascular Ultrasound Images". In: *Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*. 2018, pp. 161–168.

[137] Kim, T., Hedayat, M., Vaitkus, V. V., Belohlavek, M., Krishnamurthy, V., and Borazjani, I. "Automatic segmentation of the left ventricle in echocardiographic images using convolutional neural networks". *Quantitative Imaging in Medicine and Surgery* 11.5 (2021), pp. 1763–1781.

[138] Kingma, D. P. and Ba, J. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations (ICLR)*. 2015.

[139] Kingma, D. P. and Welling, M. "Auto-encoding variational bayes". *ArXiv* 1312.6114 (2014).

[140] Kingsbury, N. "Image Processing with Complex Wavelets". *Philosophical Transactions of the Royal Society A* 357 (1999), pp. 2543–2560.

[141] Kossoff, G., Garrett, W., Carpenter, D., Jellins, J., and Dadd, M. "Principles and classification of soft tissues by grey scale echography". *Ultrasound in Medicine & Biology* 2.2 (1976), pp. 89–105.

[142] Kulkarni, P. and Madathil, D. "Echocardiography image segmentation using semi-automatic numerical optimisation method based on wavelet decomposition thresholding". *International Journal of Imaging Systems and Technology* (2021), pp. 1–10.

[143] Kumar, G. and Bhatia, P. K. "A Detailed Review of Feature Extraction in Image Processing Systems". In: *Fourth International Conference on Advanced Computing & Communication Technologies*. 2014, pp. 5–12.

[144] Leclerc, S., Smistad, E., Østvik, A., Cervenansky, F., Espinosa, F., Espeland, T., Rye Berg, E. A., Belhamissi, M., Israilov, S., Grenier, T., Lartizien, C., Jodoin, P.-M., Lovstakken, L., and Bernard, O. "LU-Net: A Multistage Attention Network to Improve the Robustness of Segmentation of Left Ventricular Structures in 2-D Echocardiography". *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 67.12 (2020), pp. 2519–2530.

[145] Leclerc, S., Smistad, E., Pedrosa, J., Østvik, A., Cervenansky, F., Espinosa, F., Espeland, T., Berg, E. A. R., Jodoin, P.-M., Grenier, T., Lartizien, C., D'hooge, J., Lovstakken, L., and Bernard, O. "Deep Learning for Segmentation Using an Open Large-Scale Dataset in 2D Echocardiography". *IEEE Transactions on Medical Imaging* 38.9 (2019), pp. 2198–2210.

[146] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. "Backpropagation Applied to Handwritten Zip Code Recognition". *Neural Computation* 1.4 (1989), pp. 541–551.

[147] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. "Gradient-based learning applied to document recognition". *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[148] Lee, J. H., Kim, G. Y., Hwang, Y. N., and Kim, S. M. "Automatic detection of dense calcium and acoustic shadow in intravascular ultrasound images by dual-threshold-based segmentation approach". *Sensors and Materials* 30.8 (2018), pp. 1841–1852.

[149] Lee, J., Hwang, Y. N., Kim, G. Y., Kwon, J. Y., and Kim, S. M. "Automated classification of dense calcium tissues in gray-scale intravascular ultrasound images using a deep belief network". *BMC Medical Imaging* 19.1 (2019), p. 103.

[150] Lei, Y., Fu, Y., Roper, J., Higgins, K., Bradley, J. D., Curran, W. J., Liu, T., and Yang, X. "Echocardiographic image multi-structure segmentation using Cardiac-SegNet". *Medical Physics* 48.5 (2021), pp. 2426–2437.

[151] Lenchik, L., Heacock, L., Weaver, A. A., Boutin, R. D., Cook, T. S., Itri, J., Filippi, C. G., Gullapalli, R. P., Lee, J., Zagurovskaya, M., Retson, T., Godwin, K., Nicholson, J., and Narayana, P. A. "Automated Segmentation of Tissues Using CT and MRI: A Systematic Review". *Academic Radiology* 26.12 (2019), pp. 1695–1706.

[152] Li, Y.-C., Shen, T.-Y., Chen, C.-C., Chang, W.-T., Lee, P.-Y., and Huang, C.-C. J. "Automatic Detection of Atherosclerotic Plaque and Calcification From Intravascular Ultrasound Images by Using Deep Convolutional Neural Networks". *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 68.5 (2021), pp. 1762–1772.

[153] Li, L., Xu, M., Liu, H., Li, Y., Wang, X., Jiang, L., Wang, Z., Fan, X., and Wang, N. "A Large-Scale Database and a CNN Model for Attention-Based Glaucoma Detection". *IEEE Transactions on Medical Imaging* 39.2 (2020), pp. 413–424.

[154] Liang, J. and Chen, J. "Data Augmentation of Thyroid Ultrasound Images Using Generative Adversarial Network". In: *IEEE International Ultrasonics Symposium (IUS)*. 2021, pp. 1–4.

[155] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. "Focal Loss for Dense Object Detection". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.2 (2020), pp. 318–327.

[156] *Microsoft COCO: Common Objects in Context*. 2014, pp. 740–755.

[157] Liu, F., Wang, K., Liu, D., Yang, X., and Tian, J. "Deep pyramid local attention neural network for cardiac structure segmentation in two-dimensional echocardiography". *Medical Image Analysis* 67 (2021), p. 101873.

[158] Liu, P., Zhang, H., Lian, W., and Zuo, W. "Multi-Level Wavelet Convolutional Neural Networks". *IEEE Access* 7 (2019), pp. 74973–74985.

[159] Liu, S., Neleman, T., Hartman, E. M., Ligthart, J. M., Witberg, K. T., van der Steen, A. F., Wentzel, J. J., Daemen, J., and van Soest, G. "Automated Quantitative Assessment of Coronary Calcification Using Intravascular Ultrasound". *Ultrasound in Medicine & Biology* 46.10 (2020), pp. 2801–2809.

[160] Liu, X., Fan, Y., Li, S., Chen, M., Li, M., Hau, W. K., Zhang, H., Xu, L., and Lee, A. P.-W. "Deep learning-based automated left ventricular ejection fraction assessment using 2-D echocardiography". *American Journal of Physiology-Heart and Circulatory Physiology* 321.2 (2021), H390–H399.

[161] Lo Vercio, L., del Fresno, M., and Larrabide, I. "Lumen-intima and media-adventitia segmentation in IVUS images using supervised classifications of arterial layers and morphological structures". *Computer Methods and Programs in Biomedicine* 177 (2019), pp. 113–121.

[162] Loram, I., Siddique, A., Sánchez, M. B., Harding, P., Silverdale, M., Kobylecki, C., and Cunningham, R. "Objective Analysis of Neck Muscle Boundaries for Cervical Dystonia Using Ultrasound Imaging and Deep Learning". *IEEE Journal of Biomedical and Health Informatics* 24.4 (2020), pp. 1016–1027.

[163] Lu, H., Wang, H., Zhang, Q., Won, D., and Yoon, S. W. "A Dual-Tree Complex Wavelet Transform Based Convolutional Neural Network for Human Thyroid Medical Image Segmentation". In: *2018 IEEE International Conference on Healthcare Informatics (ICHI)*. 2018, pp. 191–198.

[164] Malaiapan, Y., Leung, M., and White, A. J. "The role of intravascular ultrasound in percutaneous coronary intervention of complex coronary lesions". *Cardiovascular Diagnosis and Therapy* 10.5 (2020), pp. 1371–1388.

[165] Mallat, S. "Group Invariant Scattering". *Communications on Pure and Applied Mathematics* 65 (Oct. 2012), pp. 1331–1398.

[166] Matsoukas, C., Haslum, J. F., Sorkhei, M., Söderberg, M., and Smith, K. "What Makes Transfer Learning Work For Medical Images: Feature Reuse & Other Factors". *ArXiv* 2203.01825 (2022).

[167] Meiburger, K. M., Acharya, U. R., and Molinari, F. "Automated localization and segmentation techniques for B-mode ultrasound images: A review". *Computers in Biology and Medicine* 92 (2018), pp. 210–235.

[168] Mharib, A. M., Ramli, A. R., Mashohor, S., and Mahmood, R. B. "Survey on liver CT image segmentation methods". *Artificial Intelligence Review* 37.2 (2012), pp. 83–95.

[169] Middel, L., Palm, C., and Erdt, M. "Synthesis of Medical Images Using GANs". In: *First International Workshop, UNSURE 2019, and 8th International Workshop, CLIP 2019, Held in Conjunction with MICCAI 2019*. 2019, pp. 125–134.

[170] Miki, K., Fujii, K., Nakata, T., Shibuya, M., Fukunaga, M., Kawai, K., Kawasaki, D., Masutani, M., Ohyanagi, M., and Masuyama, T. "The utility of intravascular ultrasound for the diagnosis and management of spontaneous coronary artery dissection in a middle-aged woman with acute inferior myocardial infarction". *Journal of Cardiology Cases* 6.3 (2012), e78–e80.

[171] Mikic, I., Krucinski, S., and Thomas, J. "Segmentation and tracking in echocardiographic sequences: active contours guided by optical flow estimates". *IEEE Transactions on Medical Imaging* 17.2 (1998), pp. 274–284.

[172] Milletari, F., Navab, N., and Ahmadi, S. "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation". In: *2016 Fourth International Conference on 3D Vision (3DV)*. 2016, pp. 565–571.

[173] Mirikharaji, Z. and Hamarneh, G. "Star Shape Prior in Fully Convolutional Networks for Skin Lesion Segmentation". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. 2018, pp. 737–745.

[174] Mirza, M. and Osindero, S. "Conditional Generative Adversarial Nets". *ArXiv* 1411.1784 (2014).

[175] Mishra, D., Chaudhury, S., Sarkar, M., and Soin, A. S. "Ultrasound Image Enhancement Using Structure Oriented Adversarial Network". *IEEE Signal Processing Letters* 25.9 (2018), pp. 1349–1353.

[176] Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. "Spectral Normalization for Generative Adversarial Networks". In: *6th International Conference on Learning Representations (ICLR 2018)*. 2018.

[177] Mohagheghi, S. and Foruzan, A. H. "Incorporating prior shape knowledge via data-driven loss model to improve 3D liver segmentation in deep CNNs". *International Journal of Computer Assisted Radiology and Surgery* 15.2 (2020), pp. 249–257.

[178] Mohan, A. T., Lubbers, N., Livescu, D., and Chertkov, M. "Embedding Hard Physical Constraints in Neural Network Coarse-Graining of 3D Turbulence". *ArXiv* 2002.00021 (2020).

[179] Mok, T. C. W. and Chung, A. C. S. "Learning Data Augmentation for Brain Tumor Segmentation with Coarse-to-Fine Generative Adversarial Networks". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI)*. 2018, pp. 70–80.

[180] Montero, A., Bonet-Carne, E., and Burgos-Artizzu, X. P. "Generative adversarial networks to improve fetal brain fine-grained plane classification". *Sensors* 21.23 (2021), pp. 1–14.

[181] Nair, A. A., Tran, T. D., Reiter, A., and Bell, M. A. L. "A Generative Adversarial Neural Network for Beamforming Ultrasound Images : Invited Presentation". In: *2019 53rd Annual Conference on Information Sciences and Systems (CISS)*. 2019, pp. 1–6.

[182] Nandamuri, S., China, D., Mitra, P., and Sheet, D. "SUMNet: Fully Convolutional Model For Fast Segmentation of Anatomical Structures in Ultrasound Volumes". In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. 2019, pp. 1729–1732.

[183] Negi, A., Raj, A. N. J., Nersisson, R., Zhuang, Z., and Murugappan, M. "RDA-UNET-WGAN: An Accurate Breast Ultrasound Lesion Segmentation Using Wasserstein Generative Adversarial Networks". *Arabian Journal for Science and Engineering* 45.8 (2020), pp. 6399–6410.

[184] Nguyen, X. B., Lee, G. S., Kim, S. H., and Yang, H. J. "Self-supervised learning based on spatial awareness for medical image analysis". *IEEE Access* 8 (2020), pp. 162973–162981.

[185] Nishiguchi, T., Tanaka, A., Ozaki, Y., Taruya, A., Fukuda, S., Taguchi, H., Iwaguro, T., Ueno, S., Okumoto, Y., and Akasaka, T. "Prevalence of spontaneous coronary artery dissection in patients with acute coronary syndrome". *European Heart Journal* 5.3 (2016), pp. 263–270.

[186] Noble, J. A. "Ultrasound image segmentation and tissue characterization". *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine* 224.2 (2010), pp. 307–316.

[187] Noble, J. A. and Boukerroui, D. "Ultrasound image segmentation: A survey". *IEEE Transactions on Medical Imaging* 25.8 (2006), pp. 987–1010.

[188] Noguchi, S., Nishio, M., Yakami, M., Nakagomi, K., and Togashi, K. "Bone segmentation on whole-body CT using convolutional neural network with novel data augmentation techniques". *Computers in Biology and Medicine* 121 (2020), p. 103767.

[189] Ohri, K. and Kumar, M. "Review on self-supervised image recognition using deep neural networks". *Knowledge-Based Systems* 224 (2021), p. 107090.

[190] Oktay, O., Ferrante, E., Kamnitsas, K., Heinrich, M., Bai, W., Caballero, J., Cook, S. A., Marvao, A. de, Dawes, T., O'Regan, D. P., Kainz, B., Glocker, B., and Rueckert, D. "Anatomically Constrained Neural Networks (ACNNs): Application to Cardiac Image Enhancement and Segmentation". *IEEE Transactions on Medical Imaging* 37.2 (2018), pp. 384–395.

[191] Olender, M. L., Athanasiou, L. S., Michalis, L. K., Fotiadis, D. I., and Edelman, E. R. "A Domain Enriched Deep Learning Approach to Classify Atherosclerosis Using Intravascular Ultrasound Imaging". *IEEE Journal of Selected Topics in Signal Processing* 14.6 (2020), pp. 1210–1220.

[192] Onishi, Y., Teramoto, A., Tsujimoto, M., Tsukamoto, T., Saito, K., Toyama, H., Imaizumi, K., and Fujita, H. "Automated Pulmonary Nodule Classification in Computed Tomography Images Using a Deep Convolutional Neural Network Trained by Generative Adversarial Networks". *BioMed Research International* 2019 (2019), p. 6051939.

[193] Osuala, R., Kushibar, K., Garrucho, L., Linardos, A., Szafranowska, Z., Klein, S., Glocker, B., Diaz, O., and Lekadir, K. "A Review of Generative Adversarial Networks in Cancer Imaging: New Applications, New Solutions". *ArXiv* 2107.09543 (2021).

[194] Otsu, N. "A Threshold Selection Method from Gray-Level Histograms". *IEEE Transactions on Systems, Man, and Cybernetics* 9.1 (1979), pp. 62–66.

[195] Ouahabi, A. "A review of wavelet denoising in medical imaging". In: *2013 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA)*. 2013, pp. 19–26.

[196] Oyallon, E., Belilovsky, E., and Zagoruyko, S. "Scaling the Scattering Transform: Deep Hybrid Networks". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 5619–5628.

[197] Painchaud, N., Skandarani, Y., Judge, T., Bernard, O., Lalande, A., and Jodoin, P.-M. "Cardiac Segmentation With Strong Anatomical Guarantees". *IEEE Transactions on Medical Imaging* 39.11 (2020), pp. 3703–3713.

[198] Pang, T., Wong, J. H. D., Ng, W. L., and Chan, C. S. "Semi-supervised GAN-based Radiomics Model for Data Augmentation in Breast Ultrasound Mass Classification". *Computer Methods and Programs in Biomedicine* 203 (2021), p. 106018.

[199] Park, T., Liu, M.-Y., Wang, T.-C., and Zhu, J.-Y. "Semantic Image Synthesis with Spatially-Adaptive Normalization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

[200] Pavlov, I., Prado, E., Navab, N., and Zahnd, G. "Towards in-vivo ultrasound-histology: Plane-waves and generative adversarial networks for pixel-wise speed of sound reconstruction". In: *2019 IEEE International Ultrasonics Symposium (IUS)*. 2019, pp. 1913–1916.

[201] Pedrosa, J., Barbosa, D., Heyde, B., Schnell, F., Rösner, A., Claus, P., and D'hooge, J. "Left Ventricular Myocardial Segmentation in 3-D Ultrasound Recordings: Effect of Different Endocardial and Epicardial Coupling Strategies". *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 64.3 (2017), pp. 525–536.

[202] Peng, B., Huang, X., Wang, S., and Jiang, J. "A Real-Time Medical Ultrasound Simulator Based on a Generative Adversarial Network Model". In: *2019 IEEE International Conference on Image Processing (ICIP)*. 2019, pp. 4629–4633.

[203] Petrou, M. and Petrou, C. *Image Processing: The Fundamentals*. 2nd ed. John Wiley & Sons, 2010.

[204] Pillen, S. and Alfen, N. van. "Skeletal muscle ultrasound". *Neurological Research* 33.10 (2011), pp. 1016–1024.

[205] Pillen, S., Boon, A., and Van Alfen, N. "Chapter 42 - Muscle ultrasound". In: *Neuroimaging Part II*. Ed. by Masdeu, J. C. and González, R. G. Vol. 136. Handbook of Clinical Neurology. Elsevier, 2016, pp. 843–853.

[206] Pirri, C., Fede, C., Fan, C., Guidolin, D., Macchi, V., De Caro, R., and Stecco, C. "Ultrasound Imaging of Head/Neck Muscles and Their Fasciae: An Observational Study". *Frontiers in Rehabilitation Sciences* 2 (2021).

[207] Pizurica, A., Wink, A. M., Vansteenkiste, E., Philips, W., and Roerdink, B. T. "A Review of Wavelet Denoising in MRI and Ultrasound Brain Imaging". *Current Medical Imaging Reviews* 2.2 (2006), pp. 247–260.

[208] Plissiti, M. E., Fotiadis, D. I., Michalis, L. K., and Bozios, G. E. "An automated method for lumen and media-adventitia border detection in a sequence of IVUS frames". *IEEE Transactions on Information Technology in Biomedicine* 8.2 (2004), pp. 131–141.

[209] Prabusankarlal, K. M., Thirumoorthy, P., and Manavalan, R. "Segmentation of Breast Lesions in Ultrasound Images through Multiresolution Analysis Using Undecimated Discrete Wavelet Transform". *Ultrasonic Imaging* 38.6 (2016), pp. 384–402.

[210] Räber, L., Mintz, G. S., Koskinas, K. C., Johnson, T. W., Holm, N. R., Onuma, Y., Radu, M. D., Joner, M., Yu, B., Jia, H., Meneveau, N., De La Torre Hernandez, J. M., Escaned, J., Hill, J., Prati, F., Colombo, A., Di Mario, C., Regar, E., Capodanno, D., Wijns, W., Byrne, R. A., Guagliumi, G., Alfonso, F., Bhindi, R., Ali, Z., and Carter, R. "Clinical use of intracoronary imaging. Part 1: Guidance and optimization of coronary interventions. An expert consensus document of the European Association of Percutaneous Cardiovascular Interventions". *European Heart Journal* 39.35 (2018), pp. 3281–3300.

[211] Raghu, M., Zhang, C., Kleinberg, J., and Bengio, S. "Transfusion: Understanding transfer learning for medical imaging". In: *33rd Conference on Neural Information Processing Systems*. 2019.

[212] Raissi, M., Perdikaris, P., and Karniadakis, G. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". *Journal of Computational Physics* 378 (2019), pp. 686–707.

[213] Rajpurkar, P., Irvin, J., Bagul, A., Ding, D., Duan, T., Mehta, H., Yang, B., Zhu, K., Laird, D., Ball, R. L., Langlotz, C., Shpanskaya, K., Lungren, M. P., and Ng, A. Y. *MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs*. 2018.

[214] Rani Krithiga, R. and Lakshmi, C. "A novel automated classification technique for diagnosing liver disorders using wavelet and texture features on liver ultrasound images". *Multimedia Tools and Applications* 79.5 (2020), pp. 3761–3773.

[215] Rankin, G., Stokes, M., and Newham, D. "Size and shape of the posterior neck muscles measured by ultrasound imaging: normal values in males and females of different ages". *Manual Therapy* 10.2 (2005), pp. 108–115.

[216] Reddy, C., Gopinath, K., and Lombaert, H. "Brain Tumor Segmentation using Topological Loss in Convolutional Networks". In: *Medical Imaging with Deep Learning: MIDL 2019 – Extended Abstract Track*. 2019.

[217] Ronneberger, O., Fischer, P., and Brox, T. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2015, pp. 234–241.

[218] Rosenblatt, F. "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological Review* 65.6 (1958), pp. 386–408.

[219] Rosenblatt, F. "PRINCIPLES OF NEURODYNAMICS. PERCEPTRONS AND THE THEORY OF BRAIN MECHANISMS". *American Journal of Psychology* 76 (1963).

[220] Rotemberg, V., Kurtansky, N., Betz-Stablein, B., Caffery, L., Chousakos, E., Codella, N., Combalia, M., Dusza, S., Guitera, P., Gutman, D., Halpern, A., Helba, B., Kittler, H., Kose, K., Langer, S., Lioprys, K., Malvehy, J., Musthaq, S., Nanda, J., Reiter, O., Shih, G., Stratigos, A., Tschandl, P., Weber, J., and Soyer, H. P. "A patient-centric dataset of images and metadata for identifying melanomas using clinical context". *Scientific Data* 8.1 (2021), p. 34.

[221] Roy, A. G., Navab, N., and Wachinger, C. "Recalibrating Fully Convolutional Networks With Spatial and Channel "Squeeze and Excitation" Blocks". *IEEE Transactions on Medical Imaging* 38.2 (2019), pp. 540–549.

[222] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. "Learning Internal Representations by Error Propagation". In: *Parallel Distributed Processing – Explorations in the Microstructure of Cognition*. MIT Press, 1986. Chap. 8, pp. 318–362.

[223] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. "ImageNet Large Scale Visual Recognition Challenge". *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252.

[224] Saba, T., Sameh Mohamed, A., El-Affendi, M., Amin, J., and Sharif, M. "Brain tumor detection using fusion of hand crafted and deep learning features". *Cognitive Systems Research* 59 (2020), pp. 221–230.

[225] Sandfort, V., Yan, K., Pickhardt, P. J., and Summers, R. M. "Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks". *Scientific Reports* 9.1 (2019), pp. 1–9.

[226] Santos Filho, E., Saijo, Y., Tanaka, A., and Yoshizawa, M. "Detection and Quantification of Calcifications in Intravascular Ultrasound Images by Automatic Thresholding". *Ultrasound in Medicine & Biology* 34.1 (2008), pp. 160–165.

[227] Saunders, S. L., Leng, E., Spilseth, B., Wasserman, N., Metzger, G. J., and Bolan, P. J. "Training Convolutional Networks for Prostate Segmentation with Limited Data". *IEEE Access* 9 (2021), pp. 109214–109223.

[228] Shan, C., Tan, T., Han, J., and Huang, D. "Ultrasound tissue classification: a review". *Artificial Intelligence Review* 54.4 (2021), pp. 3055–3088.

[229] Sheng, C., Xin, Y., Liping, Y., and Kun, S. "Segmentation in echocardiographic sequences using shape-based snake model combined with generalized Hough transformation". *International Journal of Cardiovascular Imaging* 22.1 (2006), pp. 33–45.

[230] Shi, G., Wang, J., Qiang, Y., Yang, X., Zhao, J., Hao, R., Yang, W., Du, Q., and Kazihise, N. G.-F. "Knowledge-guided synthetic medical image adversarial augmentation for ultrasonography thyroid nodule classification". *Computer Methods and Programs in Biomedicine* 196 (2020), p. 105611.

[231] Shiji, T. P., Remya, S., Lakshmanan, R., Pratab, T., and Thomas, V. "Evolutionary intelligence for breast lesion detection in ultrasound images: A wavelet modulus maxima and SVM based approach". *Journal of Intelligent and Fuzzy Systems* 38.5 (2020), pp. 6279–6290.

[232] Shirly, S. and Ramesh, K. "Review on 2D and 3D MRI Image Segmentation Techniques". *Current Medical Imaging* 15.2 (2019), pp. 150–160.

[233] Shorten, C. and Khoshgoftaar, T. M. "A survey on Image Data Augmentation for Deep Learning". *Journal of Big Data* 6.1 (2019).

[234] Shrimali, V., Anand, R. S., and Kumar, V. "Current Trends in Segmentation of Medical Ultrasound B-mode Images: A Review". *IETE Technical Review* 26.1 (2009), pp. 8–17.

[235] Simpson, A. L., Antonelli, M., Bakas, S., Bilello, M., Farahani, K., Ginneken, B. van, Kopp-Schneider, A., Landman, B. A., Litjens, G., Menze, B., Ronneberger, O., Summers, R. M., Bilic, P., Christ, P. F., Do, R. K. G., Gollub, M., Golia-Pernicka, J., Heckers, S. H., Jarnagin, W. R., McHugo, M. K., Napel, S., Vorontsov, E., Maier-Hein, L., and Cardoso, M. J. "A large annotated medical image dataset for the development and evaluation of segmentation algorithms". *ArXiv* 1902.09063 (2019).

[236] Singh, A. and Kingsbury, N. "Efficient Convolutional Network Learning Using Parametric Log Based Dual-Tree Wavelet ScatterNet". In: *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. 2017, pp. 1140–1147.

[237] Singh, A. and Kingsbury, N. "Dual-Tree wavelet scattering network with parametric log transformation for object classification". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017, pp. 2622–2626.

[238] Singh, N. K. and Raza, K. "Medical Image Generation Using Generative Adversarial Networks: A Review". In: *Health Informatics: A Computational Perspective in Healthcare*. Ed. by Patgiri, R., Biswas, A., and Roy, P. Singapore: Springer Singapore, 2021, pp. 77–96.

[239] Sinha, P., Wu, Y., Psaromiligkos, I., and Zilic, Z. "Lumen & Media Segmentation of IVUS Images via Ellipse Fitting Using a Wavelet-Decomposed Subband CNN". In: *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*. 2020, pp. 1–6.

[240] Sommer, F. G., Joynt, L. F., Carroll, B. A., and Macovski, A. "Ultrasonic characterization of abdominal tissues via digital analysis of backscattered waveforms". *Radiology* 141.3 (1981), pp. 811–817.

[241] Sonka, M., Zhang, X., Siebes, M., Dejong, S., McKay, C., and Collins, S. "Automated segmentation of coronary wall and plaque from intravascular ultrasound image sequences". In: *Computers in Cardiology 1994*. 1994, pp. 281–284.

[242] Sørensen, T. "A method of establishing group of equal amplitude in plant sociobiology based on similarity of species content and its application to analyses of the vegetation on Danish commons". *Biologiske skrifter / Kongelige Danske Videnskabernes Selskab* 5 (4 1948), pp. 1–34.

[243] Sorin, V., Barash, Y., Konen, E., and Klang, E. "Creating Artificial Images for Radiology Applications Using Generative Adversarial Networks (GANs) – A Systematic Review". *Academic Radiology* 27.8 (2020), pp. 1175–1185.

[244] Sudarshan, V. K., Mookiah, M. R. K., Acharya, U. R., Chandran, V., Molinari, F., Fujita, H., and Ng, K. H. "Application of wavelet techniques for cancer diagnosis using ultrasound images: A Review". *Computers in Biology and Medicine* 69 (2016), pp. 97–111.

[245] Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., and Jorge Cardoso, M. "Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations". In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. 2017, pp. 240–248.

[246] Swarnalatha, A. and Manikandan, M. "Intravascular Ultrasound Image Classification Using Wavelet Energy Features and Random Forest Classifier". In: *Intelligent Computing in Engineering*. 2020, pp. 803–810.

[247] Szabo, T. L. *Diagnostic Ultrasound Imaging: Inside Out*. 2nd ed. Elsevier, 2014.

[248] Szarski, M. and Chauhan, S. "Improved real-time segmentation of Intravascular Ultrasound images using coordinate-aware fully convolutional networks". *Computerized Medical Imaging and Graphics* 91 (2021), p. 101955.

[249] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. "Rethinking the Inception Architecture for Computer Vision". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2818–2826.

[250] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. "Going deeper with convolutions". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1–9.

[251] Taha, A. A. and Hanbury, A. "Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool". *BMC medical imaging* 15 (2015), p. 29.

[252] Tan, J., Huo, Y., Liang, Z., and Li, L. "Expert knowledge-infused deep learning for automatic lung nodule detection". *Journal of X-Ray Science and Technology* 27.1 (2019), pp. 17–35.

[253] Tang, M., Marin, D., Ben Ayed, I., and Boykov, Y. "Kernel Cuts: Kernel and Spectral Clustering Meet Regularization". *International Journal of Computer Vision* 127.5 (2019), pp. 477–511.

[254] Tang, Y.-B., Oh, S., Tang, Y.-X., Xiao, J., and Summers, R. M. "CT-realistic data augmentation using generative adversarial network for robust lymph node segmentation". In: *Medical Imaging 2019: Computer-Aided Diagnosis*. Vol. 10950. 2019, pp. 976–981.

[255] Tappeiner, E., Pröll, S., Fritscher, K., Welk, M., and Schubert, R. "Training of head and neck segmentation networks with shape prior on small datasets". *International Journal of Computer Assisted Radiology and Surgery* 15.9 (2020), pp. 1417–1425.

[256] Tessler, F. N., Middleton, W. D., Grant, E. G., Hoang, J. K., Berland, L. L., Teefey, S. A., Cronan, J. J., Beland, M. D., Desser, T. S., Frates, M. C., Hammers, L. W., Hamper, U. M., Langer, J. E., Reading, C. C., Scoutt, L. M., and Stavros, A. T. "ACR Thyroid Imaging, Reporting and Data System (TI-RADS): White Paper of the ACR TI-RADS Committee". *Journal of the American College of Radiology* 14.5 (2017), pp. 587–595.

[257] Tian, D. P. "A review on image feature extraction and representation techniques". *International Journal of Multimedia and Ubiquitous Engineering* 8.4 (2013), pp. 385–395.

[258] Tian, D.-Z. and Ha, M.-H. "Applications of wavelet transform in medical image processing". In: *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*. Vol. 3. 2004, pp. 1816–1821.

[259] Tom, F. and Sheet, D. "Simulating patho-realistic ultrasound images using deep generative networks with adversarial learning". In: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. 2018, pp. 1174–1177.

[260] Tuysuzoglu, A., Tan, J., Eissa, K., Kiraly, A. P., Diallo, M., and Kamen, A. "Deep Adversarial Context-Aware Landmark Detection for Ultrasound Imaging". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. 2018, pp. 151–158.

[261] Ulyanov, D., Vedaldi, A., and Lempitsky, V. S. "Improved Texture Networks: Maximizing Quality and Diversity in Feed-Forward Stylization and Texture Synthesis". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 4105–4113.

[262] Unser, M. and Aldroubi, A. "A review of wavelets in biomedical applications". *Proceedings of the IEEE* 84.4 (1996), pp. 626–638.

[263] Uzunova, H., Ehrhardt, J., Jacob, F., Frydrychowicz, A., and Handels, H. "Multi-scale GANs for Memory-efficient Generation of High Resolution Medical Images". In: *Medical Image Computing and Computer Assisted Intervention (MICCAI)*. 2019, pp. 112–120.

[264] Vakalopoulou, M., Chassagnon, G., Bus, N., Marini, R., Zacharaki, E. I., Revel, M.-P., and Paragios, N. "AtlasNet: Multi-atlas Non-linear Deep Networks for Medical Image Segmentation". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. 2018, pp. 658–666.

[265] Vakanski, A., Xian, M., and Freer, P. E. "Attention-Enriched Deep Learning Model for Breast Tumor Segmentation in Ultrasound Images". *Ultrasound in Medicine & Biology* 46.10 (2020), pp. 2819–2833.

[266] Vantaram, S. R. and Saber, E. "Survey of contemporary trends in color image segmentation". *Journal of Electronic Imaging* 21.4 (2012), p. 040901.

[267] Veni, G., Moradi, M., Bulu, H., Narayan, G., and Syeda-Mahmood, T. "Echocardiography segmentation based on a shape-guided deformable model driven by a fully convolutional network prior". In: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. 2018, pp. 898–902.

[268] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. J. "Phoneme recognition using time-delay neural networks". *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37.3 (1989), pp. 328–339.

[269] Wang, J., Zhu, H., Wang, S. H., and Zhang, Y. D. "A Review of Deep Learning on Medical Image Analysis". *Mobile Networks and Applications* 26.1 (2021), pp. 351–380.

[270] Wang, L., Chitiboi, T., Meine, H., Günther, M., and Hahn, H. K. "Principles and methods for automatic and semi-automatic tissue segmentation in MRI data". *Magnetic Resonance Materials in Physics, Biology and Medicine* 29.2 (2016), pp. 95–110.

[271] Wang, R., Cao, S., Ma, K., Zheng, Y., and Meng, D. "Pairwise learning for medical image segmentation". *Medical Image Analysis* 67 (2021), p. 101876.

[272] Wang, S., Han, K., and Jin, J. "Review of image low-level feature extraction methods for content-based image retrieval". *Sensor Review* 39.6 (2019), pp. 783–809.

[273] Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. M. "ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases". *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).

[274] Wang, X., Peng, Y., Lu, L., Lu, Z., and Summers, R. M. "TieNet: Text-Image Embedding Network for Common Thorax Disease Classification and Reporting in Chest X-Rays". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 9049–9058.

[275] Wang, Y., Gao, X., Wang, Y., and Sun, J. "Adventitia segmentation in intravascular ultrasound images based on improved Snake algorithm". *Optik* 241 (2021), p. 167175.

[276] Wang, Z., She, Q., and Ward, T. E. "Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy". *ACM Computing Surveys* 54.2 (2021), p. 37.

[277] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. "Image quality assessment: from error visibility to structural similarity". *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612.

[278] Wijntjes, J. and Alfen, N. van. "Muscle ultrasound: Present state and future opportunities". *Muscle & Nerve* 63.4 (2021), pp. 455–466.

[279] Wissel, T., Riedl, K. A., Schaefers, K., Nickisch, H., Brunner, F. J., Schnellbächer, N. D., Blankenberg, S., Seiffert, M., and Grass, M. "Cascaded learning in intravascular ultrasound: coronary stent delineation in manual pullbacks". *Journal of Medical Imaging* 9.2 (2022), p. 025001.

[280] Xia, M., Yan, W., Huang, Y., Guo, Y., Zhou, G., and Wang, Y. "IVUS image segmentation using superpixel-wise fuzzy clustering and level set evolution". *Applied Sciences (Switzerland)* 9.22 (2019).

[281] Xia, M., Yan, W., Huang, Y., Guo, Y., Zhou, G., and Wang, Y. "Extracting Membrane Borders in IVUS Images Using a Multi-Scale Feature Aggregated U-Net". *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine & Biology Society, EMBS* (2020), pp. 1650–1653.

[282] Xiao, G., Brady, M., Noble, J., and Zhang, Y. "Segmentation of ultrasound B-mode images with intensity inhomogeneity correction". *IEEE Transactions on Medical Imaging* 21.1 (2002), pp. 48–57.

[283] Xie, X., Niu, J., Liu, X., Chen, Z., Tang, S., and Yu, S. "A survey on incorporating domain knowledge into deep learning for medical image analysis". *Medical Image Analysis* 69 (2021), p. 101985.

[284] Xing, J., Li, Z., Wang, B., Qi, Y., Yu, B., Zanjani, F. G., Zheng, A., Duits, R., and Tan, T. "Lesion Segmentation in Ultrasound Using Semi-pixel-wise Cycle Generative Adversarial Nets". *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (2020), pp. 1–10.

[285] Xu, H.-X., Xie, X.-Y., Lu, M.-D., Liu, G.-J., Xu, Z.-F., Liang, J.-Y., and Chen, L.-D. "Unusual Benign Focal Liver Lesions". *Journal of Ultrasound in Medicine* 27.2 (2008), pp. 243–254.

[286] Yang, J., Faraji, M., and Basu, A. "Robust segmentation of arterial walls in intravascular ultrasound images using Dual Path U-Net". *Ultrasonics* 96 (2019), pp. 24–33.

[287] Yang, W., Dong, Y., Du, Q., Qiang, Y., Wu, K., Zhao, J., Yang, X., and Zia, M. B. "Integrate domain knowledge in training multi-task cascade deep learning model for benign–malignant thyroid nodule classification on ultrasound images". *Engineering Applications of Artificial Intelligence* 98 (2021), p. 104064.

[288] Yasmin, M., Sharif, M., Mohsin, S., and Azam, F. "Pathological Brain Image Segmentation and Classification: A Survey". *Current Medical Imaging* 10.3 (2014), pp. 163–177.

[289] Yazdani, S., Yusof, R., Karimian, A., Pashna, M., and Hematian, A. "Image Segmentation Methods and Applications in MRI Brain Images". *IETE Technical Review* 32.6 (2015), pp. 413–427.

[290] Yi, X., Walia, E., and Babyn, P. "Generative adversarial network in medical imaging: A review". *Medical Image Analysis* 58 (2019), p. 101552.

[291] Yue, Q., Luo, X., Ye, Q., Xu, L., and Zhuang, X. "Cardiac Segmentation from LGE MRI Using Deep Neural Network Incorporating Shape and Spatial Priors". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*. 2019, pp. 559–567.

[292] Zafer, İ., Kurnaz, M. M. N., Dokur, Z., Ölmez, T., İş, Z., and İşcan, Z. "Ultrasound Image Segmentation by Using Wavelet Transform and Self- Organizing Neural Network". *Neural Information Processing-Letters and Reviews* 10.8-9 (2006), pp. 183–191.

[293] Zaidi, H. and El Naqa, I. "PET-guided delineation of radiation therapy treatment volumes: A survey of image segmentation techniques". *European Journal of Nuclear Medicine and Molecular Imaging* 37.11 (2010), pp. 2165–2187.

[294] Zaman, A., Park, S. H., Bang, H., Park, C. woo, Park, I., and Joung, S. "Generative approach for data augmentation for deep learning-based bone surface segmentation from ultrasound images". *International Journal of Computer Assisted Radiology and Surgery* 15.6 (2020), pp. 931–941.

[295] Zayed, N., Badwi, A., Elsayad, A., Elsherif, M., and Youssef, A.-B. "Wavelet segmentation for fetal ultrasound images". In: *Proceedings of the 44th IEEE 2001 Midwest Symposium on Circuits and Systems (MWSCAS)*. Vol. 1. 2001, pp. 501–504.

[296] Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. "mixup: beyond empirical risk minimization". In: *International Conference on Learning Representations (ICLR)*. 2018.

[297] Zhang, Q., Wang, Y., Wang, W., Ma, J., Qian, J., and Ge, J. "Automatic Segmentation of Calcifications in Intravascular Ultrasound Images Using Snakes and the Contourlet Transform". *Ultrasound in Medicine & Biology* 36.1 (2010), pp. 111–129.

[298] Zhang, R., Lu, W., Wei, X., Zhu, J., Jiang, H., Liu, Z., Gao, J., Li, X., Yu, J., Yu, M., and Yu, R. "A Progressive Generative Adversarial Method for Structurally Inadequate Medical Image Data Augmentation". *IEEE Journal of Biomedical and Health Informatics* Early Access (2021).

[299] Zhang, Z., Chen, P., Sapkota, M., and Yang, L. "TandemNet: Distilling Knowledge from Medical Images Using Diagnostic Reports as Optional Semantic References". In: *Medical Image Computing and Computer Assisted Intervention - MICCAI 2017*. 2017, pp. 320–328.

[300] Zhao, C., Xia, B., Chen, W., Guo, L., Du, J., Wang, T., and Lei, B. "Multi-scale wavelet network algorithm for pediatric echocardiographic segmentation via hierarchical feature guided fusion". *Applied Soft Computing* 107 (2021), p. 107386.

[301] Zheng, H., Lin, L., Hu, H., Zhang, Q., Chen, Q., Iwamoto, Y., Han, X., Chen, Y.-W., Tong, R., and Wu, J. "Semi-supervised Segmentation of Liver Using Adversarial Learning with Deep Atlas Prior". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*. 2019, pp. 148–156.

[302] Zheng, S. and Bing-Ru, L. "Fast retrieval of calcification from sequential intravascular ultrasound gray-scale images". *Bio-Medical Materials and Engineering* 27.2-3 (2016), pp. 183–195.

[303] Zheng, Y., Chen, Z., Wang, J., Jiang, S., and Liu, Y. "Active contour models-based segmentation of left ventricle in ultrasound images for different axes views". *Journal of Mechanics in Medicine and Biology* 21.3 (2021), p. 2150031.

[304] Zheng, Z., Yan, H., Setzer, F. C., Shi, K. J., Mupparapu, M., and Li, J. "Anatomically Constrained Deep Learning for Automating Dental CBCT Segmentation and Lesion Detection". *IEEE Transactions on Automation Science and Engineering* 18.2 (2021), pp. 603–614.

[305] Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. "Scene parsing through ADE20K dataset". *Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 5122–5130.

[306] Zhou, Y. T. and Chellappa, R. "Computation of optical flow using a neural network". In: *IEEE 1988 International Conference on Neural Networks*. Vol. 2. 1988, pp. 71–78.

[307] Zhou, Z., Wang, Y., Guo, Y., Jiang, X., and Qi, Y. "Ultrafast Plane Wave Imaging With Line-Scan-Quality Using an Ultrasound-Transfer Generative Adversarial Network". *IEEE Journal of Biomedical and Health Informatics* 24.4 (2020), pp. 943–956.

[308] Zhou, Z., Wang, Y., Guo, Y., Qi, Y., and Yu, J. "Image Quality Improvement of Hand-Held Ultrasound Devices With a Two-Stage Generative Adversarial Network". *IEEE Transactions on Biomedical Engineering* 67.1 (2020), pp. 298–311.

[309] Zhu, J. Y., Park, T., Isola, P., and Efros, A. A. "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks". In: *IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2242–2251.

[310] Zhu, Y., Papademetris, X., Sinusas, A. J., and Duncan, J. S. "A coupled deformable model for tracking myocardial borders from real-time echocardiography using an incompressibility constraint". *Medical Image Analysis* 14.3 (2010), pp. 429–448.

[311] Ziemer, P. G. P., Bulant, C. A., Orlando, J. I., Maso Talou, G. D., Álvarez, L. A. M., Guedes Bezerra, C., Lemos, P. A., García-García, H. M., and Blanco, P. J. "Automated lumen segmentation using multi-frame convolutional neural networks in intravascular ultrasound datasets". *European Heart Journal - Digital Health* 1.1 (2020), pp. 75–82.

[312] Zotti, C., Luo, Z., Lalande, A., and Jodoin, P.-M. "Convolutional Neural Network With Shape Prior Applied to Cardiac MRI Segmentation". *IEEE Journal of Biomedical and Health Informatics* 23.3 (2019), pp. 1119–1128.

List of Figures

List of Tables

List of Abbreviations

cGAN	conditional generative adversarial network
CNN	convolutional neural network
CT	computed tomography
DTCWT	dual-tree complex wavelet transformation
DWT	discrete wavelet transformation
FID	Fréchet Inception distance
fMRI	functional magnetic resonance imaging
GAN	generative adversarial network
IC	independent component
ICA	independent component analysis
IVUS	intravascular ultrasound
LReLU	leaky rectified linear unit
MRI	magnetic resonance imaging
PET	positron emission tomography
ReLU	rectified linear unit
RF	random forest
RQ	research question
SCAD	spontaneous coronary artery dissection
SEST	squeeze and excitation with scattering transformation
SPADE	spatially-adaptiove normalization
SSIM	structural similarity
SVM	support-vector machine

Mathematical Notation

$*$	convolution operator
$\star$	cross-correlation operator
α	absorption coefficient
β_1	hyperparameter for the Adam optimizer
β_2	hyperparameter for the Adam optimizer
$\boldsymbol{\beta}$	bias tensor for feature map normalization
$\boldsymbol{b}$	bias vector of a neural network layer
$\mathrm{BN}(\cdot)$	batch normalization
c	phase velocity of a wave
D_{JS}	Jensen-Shannon divergence
D_{KL}	Kullback-Leibler divergence
$d_H^{ave}(\cdot,\cdot)$	average Hausdorff distance
$DC(\cdot,\cdot)$	Dice coefficient
$D(\cdot)$	discriminator function of a GAN
$d_H(\cdot,\cdot)$	Hausdorff distance
ε	small constant value, usually $\approx 10^{-8}$
η	learning rate
$FID(\cdot,\cdot)$	Fréchet Inception distance
$\mathcal{F}^{-1}(\cdot)$	inverse Fourier transformation
$\mathcal{F}(\cdot)$	Fourier transformation
f	frequency
$f^*(\cdot)$	function that defines the relationship between input $\boldsymbol{x}$ and the correct output y
$f(\cdot;\theta)$	mathematical function embodied by a neural network with learnable parameters θ
$g(\cdot)$	non-linear activation function
$\boldsymbol{\gamma}$	scaling tensor for feature map normalization

$G(\cdot)$	generator function of a GAN
$\mathcal{H}(\cdot)$	smooth Heaviside function
$\boldsymbol{h}^{\ell}$	output of hidden layer ℓ
$\mathcal{H}[\cdot]$	Hilbert transformation
I_{sp}	image with added speckle
I	intensity of a wave
$\mathrm{IN}(\cdot)$	instance normalization
j	imaginary unit
$\Lambda_{J,K}^{m}$	the set of all possible paths $p = (\lambda_1, ..., \lambda_m)$ with length m
$\Lambda_{J,K}$	the set of all possible $lambda = (j, k)$ with scaling index $j \in \{1, ..., J\}$ and rotation index $k \in \{1, ..., K\}$
L_c^{cardiac}	cardiac containment loss
L_c^{IVUS}	IVUS containment loss
$L_{DC}(\cdot, \cdot)$	Dice loss
$\mathcal{L}_D(\cdot)$	discriminiator loss function of a GAN
$L_{gDC}(\cdot, \cdot)$	generalized Dice loss
$\mathcal{L}_G(\cdot)$	generator loss function of a GAN
$\nabla_\theta \mathcal{L}$	gradients of the loss function $\mathcal{L}$ with respect to the network parameters θ
λ	a tuple (j, k) of scaling index j and rotation index k
λ	wavelength
$\mathcal{L}(\cdot)$	loss function
$\mathrm{LReLU}(\cdot)$	leaky rectified linear unit activation function
μ	mean value
P	probability distribution
p	a path of λs, i.e., a tuple $p = (\lambda_1, \lambda_2, ..., \lambda_m$
ϕ_j	scaling function with scaling index j
Q	probability distribution
r	atrous convolution rate
ρ	volumetric mass density
$\mathrm{rect}_d(\cdot, \cdot)$	rectangular window function with size d

R	reflection coefficient
$\mathrm{ReLU}(\cdot)$	rectified linear unit activation function
R_θ	rotational matrix with angle θ
S	ground truth segmentation mask
$\hat{S}$	predicted segmentation mask
$S[p]x$	scattering coefficients of x along path p
σ	standard deviation
$\sigma(\cdot)$	sigmoid activation function
$\mathrm{sinc}_d(\cdot,\cdot)$	sinc function with scaling d
$\mathrm{softmax}(\cdot)$	softmax function
$\mathrm{SPADE}(\cdot)$	spatially-adaptive normalization
$SSIM(\cdot,\cdot)$	structural similarity index
θ	learnable parameters (or weights) of a neural network
$\tanh(\cdot)$	hyperbolic tangent activation function
T	transmission coefficient
$U(\cdot,\cdot)$	field amplitude of a wave after diffraction
$u(t)$	time signal
$U[p]x$	auxiliary wavelet operator that is defined as $U[p]x = U[\lambda_m] \ldots U[\lambda_2]\, U[\lambda_1]\, x$
$U[\lambda]x$	auxiliary wavelet operator that is defined as $\lvert x * \psi_\lambda \rvert$
u	two-dimensional vector
$v(t)$	envelope of a time signal
$\tilde{W}U[\Lambda^m_{J,K}]x$	all scattering coefficients of x
$\tilde{W}$	wavelet modulus operator
W	wavelet transformation
w_c	class weights for the generalized Dice loss
$\boldsymbol{W}$	weight matrix of a neural network layer
$\boldsymbol{x}$	input to a neural network
$\boldsymbol{y}$	correct output of a network input
$\hat{\boldsymbol{y}}$	output of a neural network, i.e., a prediction
$\psi_{j,k}$	2D wavelet filters with scaling index j and rotation index k

$\boldsymbol{z}$	random seed of a GAN
Z	acoustic impedance
z	penetration depth of a wave