International Journal of Computer Assisted Radiology and Surgery (2021) 16:1243–1254
https://doi.org/10.1007/s11548-021-02417-x

ORIGINAL ARTICLE

Capsule networks for segmentation of small intravascular ultrasound image datasets

Lennart Bargsten[1] · **Silas Raschka**[1] · **Alexander Schlaefer**[1]

Received: 11 January 2021 / Accepted: 21 May 2021 / Published online: 14 June 2021

Abstract

Purpose Intravascular ultrasound (IVUS) imaging is crucial for planning and performing percutaneous coronary interventions. Automatic segmentation of lumen and vessel wall in IVUS images can thus help streamlining the clinical workflow. State-of-the-art results in image segmentation are achieved with data-driven methods like convolutional neural networks (CNNs). These need large amounts of training data to perform sufficiently well but medical image datasets are often rather small. A possibility to overcome this problem is exploiting alternative network architectures like capsule networks.

Methods We systematically investigated different capsule network architecture variants and optimized the performance on IVUS image segmentation. We then compared our capsule network with corresponding CNNs under varying amounts of training images and network parameters.

Results Contrary to previous works, our capsule network performs best when doubling the number of capsule types after each downsampling stage, analogous to typical increase rates of feature maps in CNNs. Maximum improvements compared to the baseline CNNs are 20.6% in terms of the Dice coefficient and 87.2% in terms of the average Hausdorff distance.

Conclusion Capsule networks are promising candidates when it comes to segmentation of small IVUS image datasets. We therefore assume that this also holds for ultrasound images in general. A reasonable next step would be the investigation of capsule networks for few- or even single-shot learning tasks.

Keywords Deep learning · Capsule networks · Intravascular ultrasound · Small datasets · Image segmentation

Introduction

Intravascular ultrasound (IVUS) is a commonly used imaging modality worldwide. Via IVUS experienced, physicians can assess vessel morphologies and thereby estimate important shape parameters like lumen diameter, vessel wall thickness or plaque burden. This effectively improves treatment planning and thus the success of percutaneous coronary interventions [21].

In order to derive vessel shape parameters from IVUS, physicians have to manually delineate the respective structures in multiple images. This procedure is rather time-consuming, and the results depend strongly on the physicians' experience. Automatic segmentation of lumen and vessel wall can streamline the derivation of meaningful vessel parameters and therefore improve the efficiency of respective clinical workflows.

Automatic segmentation of lumen and vessel wall via non-data-driven methods has been studied before [1,13,14,23,27]. Many of these approaches rely on active contour models, level sets, gradient-based techniques or thresholding. For example, in [27], the authors propose a fuzzy clustering approach with superpixels for reducing the influence of speckle noise, followed by a level set evolution algorithm with a new edge indicator. Reviews regarding IVUS segmentation approaches can be found in [1,13]. Data-driven methods include support vector machines, random forests or convolutional neural networks (CNNs). The authors of [4], e.g., combine an ensemble support vector machine pixel-wise classifier with a deformable model to extract lumen and media-adventitia borders. Approaches using CNNs mainly rely on encoder–decoder architectures like U-Net [20] and report state-of-the-art results for segmentation of lumen and vessel wall [7,15,17,19,28,29,31]. However, CNNs depend heavily on the size of the underlying dataset as well as the

✉ Lennart Bargsten
lennart.bargsten@tuhh.de

[1] Hamburg University of Technology, Institute of Medical Technology and Intelligent Systems, Hamburg, Germany

quality of the corresponding annotations. To ensure high quality, annotations have to be created by trained experts in a time-consuming process which generally leads to rather small datasets in the medical domain. Therefore, it is essential to develop methods which also perform well and robustly on small datasets.

Possible directions to achieve this are incorporating domain knowledge into the CNN [2] or exploiting new sophisticated network architectures. Such a rather novel network architecture is the capsule network [9,22]. Capsules are neurons grouped into tensors, like vectors or matrices, which correspond to entities and their respective properties (e.g., pose, texture, deformation, etc.) present in the image. These capsules form the basic network elements instead of single neurons as in the case of CNNs. An iterative routing algorithm couples child capsules to parent capsules which thus form a part-whole relationship. The overall network can therefore be interpreted as some kind of parse tree.

Recent experimental studies showed that capsule networks can outperform CNNs when dealing with small natural image datasets [11,12,30]. We study whether this also holds for small ultrasound image datasets. We consider the task of segmenting lumen and vessel wall in IVUS images. So far, capsule networks have been applied to X-ray as well as computed tomography image segmentation. Ultrasound images differ a lot from the former modalities regarding texture and noise structure (speckle). Therefore, we assume that the capsule network architecture has to be tuned in order to achieve sufficient segmentation performance on ultrasound images. Our contribution is twofold. First, we present an optimized capsule network for IVUS image segmentation. Second, we provide a detailed analysis of capsule networks and a state-of-the-art CNN with respect to the amount of training data available.

Material and methods

Dataset

For this study, we used a publicly available IVUS segmentation dataset consisting of 435 annotated IVUS frames with a size of 384×384 pixels obtained from ten different patients [1]. The images were acquired in a gated fashion with a 20 MHz phased array transducer and annotated by clinical experts by delineating the lumen border and the external elastic membrane as the transition between media and adventitia. The contours were transformed into pixel masks comprising three classes: lumen, vessel wall (as the union of intima and media) as well as background (adventitia and surrounding tissue). See Fig. 3 for exemplary images with corresponding segmentation contours (yellow dashed lines).

In addition, we used another IVUS dataset also provided by [1]. This dataset comprises 77 images from 22 patients with a size of 512×512 pixels. The images were acquired with a rotational transducer and a frequency of 40 MHz. Analogous to the other dataset, the annotations delineate lumen border and external elastic membrane. However, these are much less visible compared to the 20 MHz dataset and thus generally harder to detect (see Fig. 4).

Capsule networks

Capsules have been developed in order to integrate parse tree-like child–parent relationships into neural networks. Capsules are groups of multiple neurons and can have different forms like vectors [22] or matrices [10]. The general idea is that an active capsule represents a specific entity present in the image, whereas the activities of the corresponding neurons encode its properties like pose, texture or deformation. Capsules in subsequent layers are coupled via an iterative routing process which ensures a part-whole tree structure throughout the network. This means that capsules $\mathbf{u}_i$ in layer L (child capsules) with a strong coupling to specific capsules $\mathbf{v}_j$ in layer $L+1$ (parent capsules) can be interpreted as parts of entities represented by the respective parent capsules. To perform the routing procedure, child capsules are transformed into the parent capsules' feature space via transformation matrices $\mathbf{W}_{ij}$ which are learned via backpropagation.

Since each image entity is associated with a capsule, the activation of a capsule is independent of the entity's pose. Therefore, capsule layers are—at least heuristically—equivariant [10]. Not only in the case of translations, as CNNs, but also for more complex transformations like rotations or reflections. This could be a reason why capsule networks can outperform CNNs when trained with small datasets as shown in [11,12,30].

Considering the case of capsules with vector outputs, the transformation of child capsule output vectors $\mathbf{u}_i$ into the parent capsules' feature space can be written as

$$\hat{\mathbf{u}}_{j|i} = \mathbf{W}_{ij}\,\mathbf{u}_i.$$

The transformed child capsule output vectors $\hat{\mathbf{u}}_{j|i}$ are linearly combined with weights c_{ij}, which are derived from the dynamic routing process in every forward pass (see [22] for details):

$$\mathbf{s}_j = \sum_i c_{ij}\,\hat{\mathbf{u}}_{j|i}.$$

Finally, the parent capsule outputs $\mathbf{v}_j$ are computed via the *squash* activation function:

$$\mathbf{v}_j = \text{squash}(\mathbf{s}_j) = \frac{\|\mathbf{s}_j\|^2}{1+\|\mathbf{s}_j\|^2}\frac{\mathbf{s}_j}{\|\mathbf{s}_j\|}.$$

By learning a reverse-encoding of object properties, capsule networks provide improved generalizability to unseen transformations and viewpoint changes while requiring less training data than CNNs when performing pose prediction [10]. Furthermore, the preservation of spatial part-whole relationships can better represent constraints regarding anatomical information which could be quite beneficial for semantic segmentation tasks [22].

The first attempt of using capsule networks for image segmentation was SegCaps [16]. SegCaps introduced locally constrained dynamic routing, which restricts the set of child capsules routed to a specific parent capsule to a relatively small window of size 5×5, analogous to the convolutional kernel size in CNNs. We refer to this type of layer as *convolutional capsule layer*. Furthermore, SegCaps makes use of shared transformation matrices for capsules inside these specific windows. The basic architecture follows a U-Net-like structure incorporating downsampling and upsampling via strided routing windows and skip connections between the encoding and decoding path. The numbers of capsule types—as an analogue to feature maps in CNNs—after each level of the encoding path are {1,4,8,8}. We refer to this expression as the *shape* of the network, because the decoding path usually exhibits the same structure but vice versa.

In contrast to SegCaps, Matwo-CapsNet [3] consists of capsules represented as matrices as proposed by Hinton et al. [10]. Matwo-CapsNet extends the idea of a 4×4 capsule pose matrix by introducing an additional 5×5 appearance matrix and a dual routing algorithm combining the information from both matrices. The term pose matrix should not indicate that this matrix has specific properties which hold for pose matrices in robotics and navigation. Like SegCaps, Matwo-CapsNet exhibits a U-Net-like architecture with convolutional capsule layers and a shape of {5,5,6,7}, whereas the decoding path has six capsule types instead of five.

The forward propagation in Matwo-CapsNet works basically the same as when using vector capsules. Pose matrices $\mathbf{P}_i$ and appearance matrices $\mathbf{A}_i$ of layer L are transformed via transformation matrices $\mathbf{W}_{ij}^P$ and $\mathbf{W}_{ij}^A$:

$$\hat{\mathbf{P}}_{j|i} = \mathbf{P}_i\,\mathbf{W}_{ij}^P \qquad \hat{\mathbf{A}}_{j|i} = (\mathbf{A}_i + b_{ij})\,\mathbf{W}_{ij}^A,$$

where b_{ij} denotes learnable biases. The transformed matrices $\hat{\mathbf{P}}_{j|i}$ and $\hat{\mathbf{A}}_{j|i}$ are linearly combined with weights c_{ij} which are the same for both types of matrices.

$$\tilde{\mathbf{P}}_j = \sum_i c_{ij}\,\hat{\mathbf{P}}_{j|i} \qquad \tilde{\mathbf{A}}_j = \sum_i c_{ij}\,\hat{\mathbf{A}}_{j|i}$$

The weights are derived from the dual routing procedure (see [3] for details). The output matrices of layer $L+1$ are then calculated by applying the nonlinear activation functions *Psquash* and *squash*.

$$\mathbf{P}_j = \text{Psquash}(\tilde{\mathbf{P}}_j) = \frac{\tilde{\mathbf{P}}_j}{\max(\text{abs}(\tilde{\mathbf{P}}_j))},$$

$$\mathbf{A}_j = \text{squash}(\tilde{\mathbf{A}}_j) = \frac{\|\tilde{\mathbf{A}}_j\|^2}{1+\|\tilde{\mathbf{A}}_j\|^2}\frac{\tilde{\mathbf{A}}_j}{\|\tilde{\mathbf{A}}_j\|}.$$

Capsule networks offer the possibility of incorporating a regularization by performing a reconstruction of the input image from the network's last capsule layer. In the case of classification, this can be accomplished by feeding the active capsule from the classification layer into a decoder network [22]. In the case of binary segmentation, SegCaps masks out all capsules of the last network layer which do not belong to the target class and feeds the remaining capsules into a decoder consisting of three 1×1 convolutional layers. Matwo-CapsNet waives the idea of a regularization via reconstruction.

Optimization of the capsule network architecture

Preliminary experiments with the SegCaps architecture [16] revealed severe weaknesses. As also observed in [3], SegCaps was not able to produce reasonable results when used for multi-class segmentation. We thus forewent investigating this architecture any further and completely focused on Matwo-CapsNet.

So far, the performance of Matwo-CapsNet has only been demonstrated for chest X-ray as well as computed tomography images. These modalities are very different from ultrasound in terms of texture and noise structure. Ultrasound images are typically governed by speckle noise which tends to make borders between different tissues rather unclear and harder to detect. Furthermore, parts of the images are often obscured by shadow artifacts leading to a local reduction of information. We can thus assume that Matwo-CapsNet's hyperparameters have to be tuned in order to optimize the network structure toward IVUS image segmentation. This procedure was performed on the 20 MHz dataset.

As already mentioned in the previous section the following structural parameters play an important role in Matwo-CapsNet and have been investigated regarding their impact on the IVUS segmentation results:

- Treatment of the pose matrix
- Routing type and number of routing iterations
- Performing a reconstruction regularization
- Window size of locally constrained routing
- Pose matrix shape
- Appearance matrix shape
- Number of capsule types throughout the network

Comparison between capsule network and U-Net Res

We compared our tuned capsule network with a state-of-the-art encoder–decoder CNN similar to the U-Net [20] but built with residual blocks [8] analogous to [18]. We call it U-Net Res throughout this work. Both the baseline CNN and the capsule network had an equal number of parameters. We chose a U-Net-like baseline CNN due to two reasons. First, previous work reports state-of-the-art results using encoder-decoder CNNs [7,15,17,19,28,29,31]. Second, the capsule network also features an encoder–decoder structure which makes both networks more comparable. We furthermore studied how both networks behave when the number of parameters is reduced. Small networks with less parameters are of great importance when it comes to running these on embedded systems or mobile devices [6], because here the amount of available memory is usually rather limited.

We used the 20 MHz dataset and training set sizes of 250, 150 and only 50 training images and investigated which of the networks were able to cope better with smaller datasets. Networks which generally perform better on such small datasets are advantageous for medical image datasets, particularly for few-shot learning tasks [26]. In addition, we evaluated our approach on the 40 MHz dataset in order to investigate whether the capsule architecture optimized for the 20 MHz dataset could readily be used for slightly different data.

Training and evaluation

Preliminary experiments showed that Matwo-CapsNet performed best with the spread loss, which was introduced specifically for capsule networks [22]. The U-Net Res on the other hand performed best with the generalized Dice loss, a state-of-the-art loss function for medical image segmentation [24]. We therefore used the spread loss for all capsule networks and the generalized Dice loss for all U-Net Res.

We carried out fivefold cross-validation (CV) for all experiments in order to get meaningful statistics. We investigated three different training set sizes of the 20 MHz dataset:

1. 250 training images: every CV-fold comprised 50 images of a single patient, resulting in five different patients in the training set. The remaining 185 images of the dataset, again from five different patients, were used for testing.
2. 150 training images: same as (1) but with only 30 images per patient in the CV-folds. Same test set as (1).
3. 50 training images: only data of a single patient divided into CV-folds of ten images. This setting makes it difficult for networks to generalize to the unseen test data because the validation sets highly correlate with the training sets. Same test set as (1) and (2).

A detailed overview of the CV schemes is depicted in Fig. 1. All images were resized to 256×256 pixels and augmented by random rotations and flips on-the-fly during training. As evaluation metrics we chose the Dice coefficient as a measure of overlap and the average Hausdorff distance [5] as a measure of edge alignment between the predicted and ground-truth segmentation masks. The average Hausdorff distance between two sets A and B is defined as

$$d_H^{ave} = \max \left\{ \underset{a \in A}{\text{mean}} \min_{b \in B} d(a,b), \quad \underset{b \in B}{\text{mean}} \min_{a \in A} d(a,b) \right\}$$

with the Euclidean distance $d(\cdot,\cdot)$. Due to the mean operations, d_H^{ave} is less sensitive to outliers [5,25] which makes comparing segmentation pixel masks more meaningful than using the ordinary Hausdorff distance. The average Hausdorff distance is therefore quite similar to the average symmetric surface distance which computes the mean instead of the max of both directed distances. For completeness and comparability to previous work, we do also report the ordinary Hausdorff distance.

All networks were trained with the Adam optimizer. Via preliminary grid-searching, we found a learning rate of $\ell = 1e-3$ to be optimal for the Matwo-CapsNet, whereas it was $\ell = 2e-4$ for the U-Net Res. We trained every network for 200 epochs and validated after every epoch with the validation set by computing Dice coefficients. After training, the model which performed best on the validation set was chosen to be evaluated with the test set.

Additionally, we evaluated our approach on the 40 MHz dataset. Due to its small size of 77 images, we only evaluated a single training set size. We performed fivefold cross-validation with ten images per fold and 27 images in the test set. All other settings were the same as above.

Results and discussion

Optimization of the capsule network architecture

Grid-searching all possible architecture hyperparameters was not feasible regarding temporal and computing resources. We thus used a partially greedy approach starting with a set of

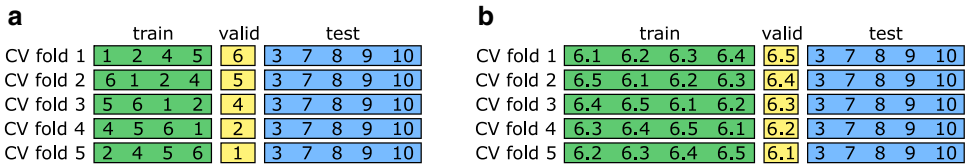

Fig. 1 Overview of the used CV schemes and the distribution of patients among the individual sets. **a** CV scheme for scenarios one (250 training images) and two (150 training images). Scenario two only uses 60% of images from every patient. **b** CV scheme for scenario three (50 training images). All images in the training and validation sets originate solely from patient six

Table 1 Segmentation performances as a function of different treatments of the pose (or pose transformation) matrix

Pose matrix treatment	Dice coefficient		Ave. Hausdorff distance [px]	
	Vessel wall	Lumen	Vessel wall	Lumen
Normalized transf. w/coord. add.	66.27 ± 2.77	90.50 ± 1.34	2.73 ± 0.08	1.27 ± 0.86
Normalized w/ coord. add.	**79.11 ± 1.15**	**93.52 ± 1.05**	**1.37 ± 0.24**	**0.66 ± 0.70**
Normalized w/o coord. add.	69.98 ± 2.69	90.15 ± 0.88	3.11 ± 0.85	1.38 ± 0.50
No modifications	68.3 ± 2.07	89.31 ± 0.74	3.06 ± 0.59	1.64 ± 0.55

Bold values indicate best results

Table 2 Segmentation performances as a function of different routing algorithms performed with three routing iterations

Routing algorithm	Dice coefficient		Ave. Hausdorff distance [px]	
	Vessel wall	Lumen	Vessel wall	Lumen
Dynamic routing	73.14 ± 1.23	90.89 ± 0.78	2.13 ± 0.43	0.96 ± 0.44
Dual routing	**79.11 ± 1.15**	**93.52 ± 1.05**	**1.37 ± 0.24**	**0.66 ± 0.70**

Bold values indicate best results

Table 3 Segmentation performances as a function of different numbers of routing iterations performed with dual routing

# Iterations	Dice coefficient		Ave. Hausdorff distance [px]	
	Vessel wall	Lumen	Vessel wall	Lumen
1	55.06 ± 1.60	86.23 ± 0.79	6.13 ± 0.84	2.89 ± 0.25
2	76.36 ± 0.77	91.71 ± 1.07	1.73 ± 0.17	0.70 ± 0.32
3	**79.11 ± 1.15**	**93.52 ± 1.05**	**1.37 ± 0.24**	**0.66 ± 0.70**
4	74.48 ± 2.61	91.49 ± 1.27	1.95 ± 0.14	0.89 ± 0.59
5	67.77 ± 3.78	90.18 ± 1.57	2.29 ± 0.44	1.10 ± 0.44

Bold values indicate best results

parameters used in the original Matwo-CapsNet paper [3]. However, we changed the numbers of capsule types in the encoding path (network shape) from {5,5,6,7} to {3,5,7,9} and used two convolutional capsule layers per level. The order of the numbers of capsule types in the decoding path is vice versa. The initial shape of the pose matrix was 4×4, whereas the appearance matrix had a shape of 5×5. If improvements were found, these were integrated into the network. Exceptions are mentioned in the text. For the sake of clarity, we used only the average Hausdorff distance measured in pixels as the basis for evaluation in this section, in addition to the Dice coefficient.

First, we investigated how different treatments of the pose matrix affected the segmentation performance. Originally, Hinton et al. [10] did not normalize the pose matrix but proposed to add scaled coordinates to the last matrix column relative to the center of the capsule's receptive field. Bonheur et al. [3] introduced the idea of normalizing every column of the pose transformation matrix such that these have unit length. We compared this method with three other ones: normalizing the pose matrix with subsequent addition of scaled coordinates, normalizing the pose matrix without adding scaled coordinates and no manipulation at all. The corresponding results are given in Table 1. We can see that the approach of normalizing the pose matrix with subsequent scaled coordinate addition led to the best segmentation performance by far.

We then investigated how the results were affected by using either dual routing or dynamic routing as well as the number of routing iterations. Tables 2 and 3 show that using

Table 4 Segmentation performances as a function of different approaches to adding a reconstruction regularization. The underlying network shape was {3,5,7,9}

Reconstruction	Dice coefficient		Ave. Hausdorff distance [px]	
	Vessel wall	Lumen	Vessel wall	Lumen
Without	**79.11 ± 1.15**	**93.52 ± 1.05**	**1.37 ± 0.24**	**0.66 ± 0.70**
From all classes	77.62 ± 1.39	92.23 ± 1.39	1.60 ± 0.28	0.71 ± 0.37
From pos. classes	76.37 ± 2.10	92.52 ± 1.05	1.65 ± 0.15	0.82 ± 0.67
From lowest level	76.60 ± 1.56	92.49 ± 0.78	1.60 ± 0.26	0.77 ± 0.60

Bold values indicate best results

Table 5 Segmentation performances as a function of different pose matrix sizes obtained with a network of shape {3,5,7,9} and an appearance matrix with shape 5×5

Pose matrix shape	Dice coefficient		Ave. Hausdorff distance [px]	
	Vessel wall	Lumen	Vessel wall	Lumen
2×2	37.82 ± 29.02	58.19 ± 24.43	24.12 ± 26.14	26.67 ± 31.33
3×3	75.71 ± 1.84	91.33 ± 2.07	1.73 ± 0.23	0.79 ± 0.60
4×4	**79.11 ± 1.15**	**93.52 ± 1.05**	**1.37 ± 0.24**	**0.66 ± 0.70**
5×5	77.16 ± 1.44	92.76 ± 0.65	1.71 ± 0.24	0.76 ± 0.52

Bold values indicate best results

Table 6 Segmentation performances as a function of different appearance matrix sizes obtained with a network of shape {3,5,7,9} and a pose matrix with shape 4×4

Appearance matrix shape	Dice coefficient		Ave. Hausdorff distance [px]	
	Vessel wall	Lumen	Vessel wall	Lumen
2×2	75.60 ± 1.87	91.22 ± 1.45	1.67 ± 0.06	0.81 ± 0.44
3×3	76.94 ± 1.24	92.45 ± 1.14	1.75 ± 0.40	0.82 ± 0.60
4×4	76.65 ± 0.52	92.64 ± 0.69	1.59 ± 0.18	0.64 ± 0.43
5×5	**79.11 ± 1.15**	**93.52 ± 1.05**	**1.37 ± 0.24**	0.66 ± 0.70
6×6	76.07 ± 1.62	92.18 ± 1.25	1.78 ± 0.19	**0.62 ± 0.14**

Bold values indicate best results

Table 7 Segmentation performances as a function of network depth and the number of capsule types per level

Network shape	Dice coefficient		Ave. Hausdorff distance [px]	
	Vessel wall	Lumen	Vessel wall	Lumen
{3,4,5,6}	75.63 ± 2.20	91.94 ± 0.29	1.80 ± 0.19	0.66 ± 0.12
{3,4,5,6,7}	**76.39 ± 1.23**	**92.94 ± 0.42**	**1.56 ± 0.26**	**0.43 ± 0.06**
{3,5,7,9}	**79.11 ± 1.15**	**93.52 ± 1.05**	**1.37 ± 0.24**	0.66 ± 0.70
{3,5,7,9,11}	78.97 ± 0.67	93.33 ± 0.60	1.30 ± 0.20	**0.46 ± 0.13**
{3,6,12}	74.38 ± 2.10	90.54 ± 1.05	1.73 ± 0.19	1.14 ± 0.46
{3,6,12,24}	79.98 ± 0.73	92.99 ± 0.82	1.32 ± 0.25	0.51 ± 0.26
{3,6,12,24,48}	**81.16 ± 1.88**	**94.59 ± 0.38**	**1.02 ± 0.30**	**037 ± 0.70**

Bold values indicate best results

dual routing with three routing iterations performed best. This means that treating appearance and pose features separately is also beneficial for IVUS segmentation. Increasing the number of routing iterations to values higher than three leads to a decrease in segmentation performance, a tendency also shown in [10] for classification. Due to the larger number of routing iterations, the capacity of the network increases, which eventually leads to overfitting.

The resulting segmentation performance when using three different approaches for reconstruction as a regularization method is shown in Table 4. First, the reconstruction was performed from the capsules belonging to all classes of the last layer. Second, only the capsules from the positive classes of the last layer were used. And third, the capsules of the lowest network level were used. We found no performance improvement through adding a reconstruction regularization. Additionally, incorporating a reconstruction heavily increased training time and VRAM load. We therefore refrained from using a reconstruction just like Bonheur et al. [3].

We then investigated different sizes of the pose and appearance matrix. Tables 5 and 6 show the corresponding results.

Table 8 Segmentation performances as a function of different window sizes for locally constrained routing obtained with a network of shape {3,6,12,24}

Window size	Dice coefficient		ave. Hausdorff distance [px]	
	Vessel wall	Lumen	Vessel wall	Lumen
3×3	77.25 ± 1.03	92.42 ± 1.34	1.63 ± 0.13	0.80 ± 0.58
5×5	79.78 ± 2.03	$\mathbf{94.40 \pm 0.40}$	1.27 ± 0.41	$\mathbf{0.38 \pm 0.26}$
7×7	$\mathbf{80.46 \pm 1.48}$	93.40 ± 0.95	$\mathbf{1.24 \pm 0.22}$	0.44 ± 0.16

Bold values indicate best results

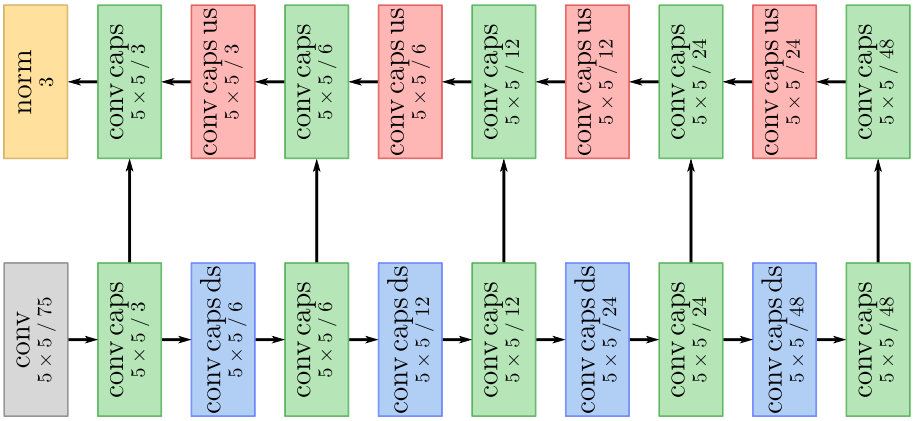

Fig. 2 Sketch of the optimized capsule network architecture. The ordinary convolutional layer is colored gray. Convolutional capsule layers (with downsampling/upsampling) are colored green (blue/red). The digits indicate window size as well as the number of capsule types (feature maps) after convolutional capsule layers (convolutional layers). The last layer computes the Frobenius norm of pose matrix and appearance matrix and multiplies both resulting values for each capsule (i.e., pixel) and all three segmentation classes

Table 9 Segmentation performances of capsule networks and baseline U-Net Res for different sizes of the 20 MHz dataset measured by Dice coefficient

# Images	# Params	Network	Dice coefficient	
			Vessel wall	Lumen
50	32 k	CapsNet	$\mathbf{71.05 \pm 2.25}$	$\mathbf{90.63 \pm 1.06}$
		U-Net Res	59.42 ± 5.98	78.60 ± 9.20
	102 k	CapsNet	$\mathbf{72.00 \pm 2.71}$	$\mathbf{91.73 \pm 0.79}$
		U-Net Res	61.31 ± 4.77	76.07 ± 5.23
	420 k	CapsNet	$\mathbf{73.99 \pm 1.55}$	$\mathbf{91.58 \pm 0.78}$
		U-Net Res	68.71 ± 2.09	87.04 ± 2.45
150	32 k	CapsNet	$\mathbf{77.17 \pm 0.92}$	$\mathbf{93.00 \pm 0.40}$
		U-Net Res	73.17 ± 1.55	90.28 ± 1.33
	102 k	CapsNet	$\mathbf{78.74 \pm 1.02}$	$\mathbf{93.22 \pm 1.01}$
		U-Net Res	75.12 ± 1.83	90.58 ± 0.95
	420 k	CapsNet	$\mathbf{79.07 \pm 1.40}$	$\mathbf{93.84 \pm 0.33}$
		U-Net Res	78.92 ± 1.48	93.38 ± 0.83
250	32 k	CapsNet	$\mathbf{79.11 \pm 1.15}$	$\mathbf{93.52 \pm 1.05}$
		U-Net Res	75.79 ± 2.11	92.07 ± 2.86
	102 k	CapsNet	$\mathbf{79.98 \pm 0.73}$	$\mathbf{92.99 \pm 0.82}$
		U-Net Res	76.67 ± 2.26	92.24 ± 1.21
	420 k	CapsNet	$\mathbf{81.16 \pm 1.88}$	$\mathbf{94.59 \pm 0.38}$
		U-Net Res	80.15 ± 1.35	94.21 ± 0.76

Bold values indicate best results

Table 10 Segmentation performances of capsule networks and baseline U-Net Res for different sizes of the 20 MHz dataset measured by ordinary and average Hausdorff distance

# Images	# Params	Network	Hausdorff distance [mm]		Ave. Hausdorff dist. [mm]	
			Vessel wall	Lumen	Vessel wall	Lumen
50	32 k	CapsNet	**.521 ± .059**	**.355 ± .040**	**.060 ± .009**	**.022 ± .003**
		U-Net Res	.825 ± .116	.629 ± .123	.158 ± .059	.097 ± .050
	102 k	CapsNet	**.570 ± .042**	**.353 ± .056**	**.062 ± .010**	**.016 ± .003**
		U-Net Res	.953 ± .164	.633 ± .090	.203 ± .060	.126 ± .045
	420 k	CapsNet	**.419 ± .040**	**.297 ± .020**	**.046 ± .005**	**.017 ± .002**
		U-Net Res	.627 ± .084	.378 ± .055	.086 ± .019	.053 ± .020
150	32 k	CapsNet	**.393 ± .033**	**.252 ± .026**	**.036 ± .003**	**.013 ± .002**
		U-Net Res	.592 ± .082	.427 ± .087	.066 ± .007	.032 ± .010
	102 k	CapsNet	**.406 ± .049**	**.261 ± .015**	**.036 ± .008**	**.014 ± .003**
		U-Net Res	.518 ± .076	.365 ± .079	.055 ± .013	.028 ± .007
	420 k	CapsNet	**.352 ± .035**	**.243 ± .038**	**.035 ± .004**	**.013 ± .005**
		U-Net Res	.416 ± .139	.265 ± .067	.038 ± .010	.015 ± .011
250	32 k	CapsNet	**.470 ± .067**	**.273 ± .091**	**.036 ± .006**	**.017 ± .016**
		U - Net Res	.625 ± .174	.507 ± .233	.060 ± .019	.049 ± .064
	102 k	CapsNet	**.394 ± .072**	**.234 ± .038**	**.033 ± .010**	**.010 ± .006**
		U-Net Res	.643 ± .175	.342 ± .095	.065 ± .030	.028 ± .019
	420 k	CapsNet	**.313 ± .052**	.207 ± .048	**.027 ± .007**	.010 ± .006
		U-Net Res	.353 ± .063	**.196 ± .033**	.031 ± .005	**.008 ± .002**

Bold values indicate best results

Using a pose matrix with shape 4×4 and an appearance matrix with shape 5×5 led to the best results. Interestingly, the performance drops when choosing the larger matrix sizes.

Regarding the underlying encoder–decoder architecture, we investigated how different network depths (and thus different numbers of downsamplings) affect the segmentation performance. Furthermore, we compared different alternatives for increasing the number of capsule types in the encoding path by either adding a fixed number of capsule types or doubling these in each level. The approaches in [3,16] are non-doubling (likely due to limitations of computational resources) but Table 7 shows that doubling is rather beneficial when performed along with increasing the depth to five levels.

The window size for locally constrained routing is an important hyperparameter because it drastically affects the number of weights and the size of the capsules' receptive fields. Table 8 depicts the segmentation performances with different window sizes. Due to limitations with respect to computational resources, we were not able to apply window sizes of 7×7 to networks with shape {3,6,12,24,48}. We therefore used a network with shape {3,6,12,24} for this comparison. We do not see clear improvements when switching from 5×5 to 7×7 windows. We therefore stuck to a window size of 5×5 for further experiments which is the same as in [3,16].

The structural parameters of Matwo-CapsNet which led to the best segmentation performance are as follows:

- Normalizing pose matrix and adding scaled coordinates
- Dual routing with three iterations
- No reconstruction
- Routing window size: 5×5
- Pose matrix shape: 4×4
- Appearance matrix shape: 5×5
- Network shape: {3, 6, 12, 24, 48}

The resulting architecture differs from the original Matwo-CapsNet architecture proposed in [3]. The major differences are the treatment of the pose matrix (normalizing the pose matrix instead of the pose transformation matrix), the increased network depth of five levels and the doubling of capsule types at each level leading to 48 capsule types at the lowest level. Making the network deeper while only adding a fixed amount of capsule types per level increased the performance substantially less or even led to performance drops. Figure 2 depicts a sketch of the optimized capsule network architecture.

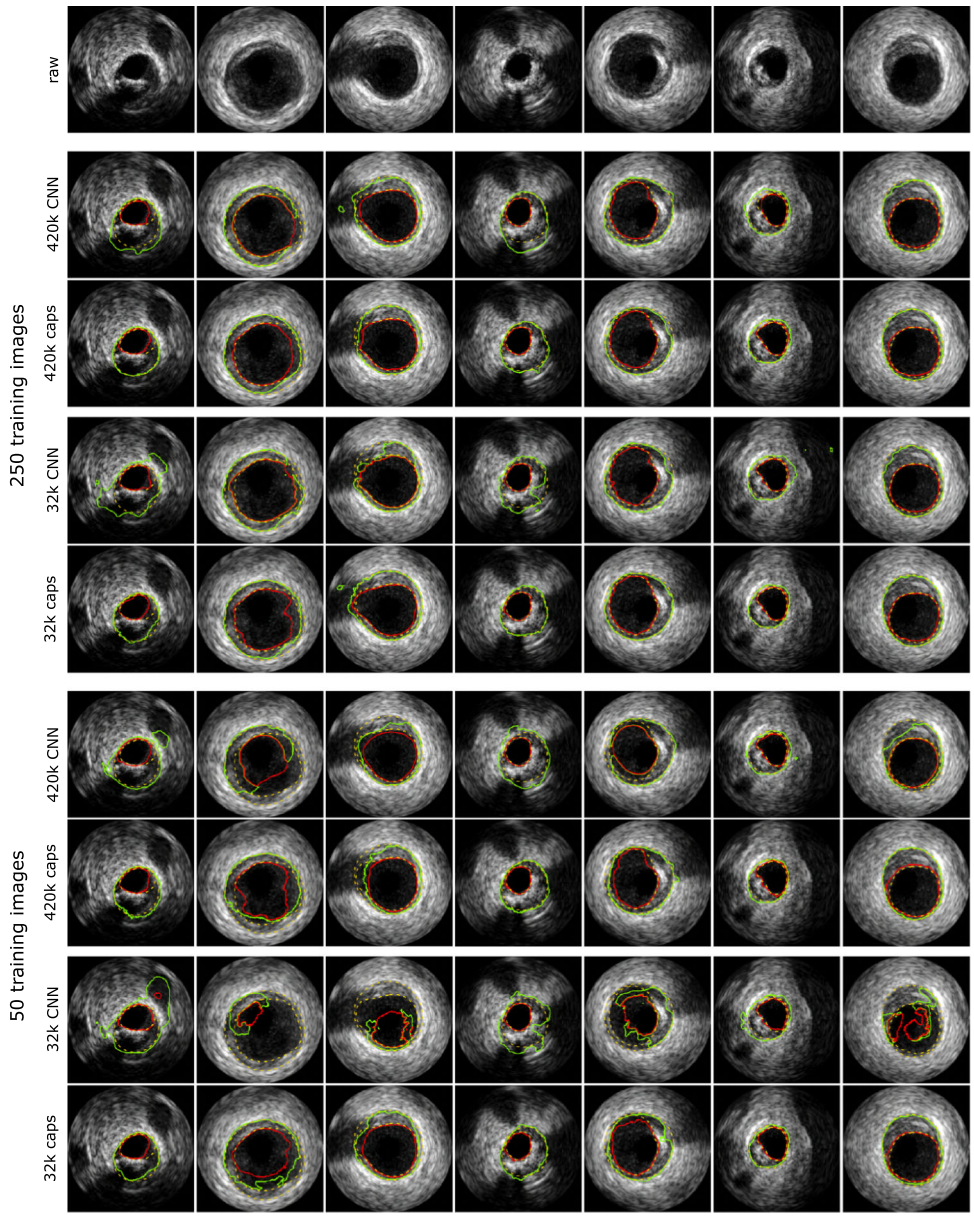

Fig. 3 Comparison of exemplary segmentation results between capsule networks and CNNs for the 20 MHz dataset. Shown are predictions of large networks with 420 k parameters and small networks with 32 k parameters. Ground truth annotations of lumen border and external elastic membrane are depicted with yellow dashed lines. The predicted contours with red and green solid lines, respectively

Comparison between capsule network and U-Net Res

The resulting segmentation performances on the 20 MHz dataset are given in Tables 9 and 10. One can clearly see the tendency of the capsule network to outperform the U-Net Res when the training sets get smaller as well as when the network sizes decrease. We can thus deduce that developing part-whole relationships in capsule networks is beneficial for the segmentation of ultrasound images when dealing with data scarcity or small networks.

For vessel wall segmentation with 250 training images, the relative improvement regarding the Dice coefficient is 1.3% in the case of networks with 420 k parameters and increases to

Table 11 Segmentation performances of capsule networks and baseline U-Net Res on the 40 MHz dataset measured by Dice coefficient

# Params	Network	Dice coefficient	
		Vessel wall	Lumen
32 k	CapsNet	**66.57 ± 2.17**	**88.85 ± 0.84**
	U-Net Res	58.89 ± 0.44	86.90 ± 0.70
102 k	CapsNet	**67.99 ± 1.88**	**88.50 ± 0.53**
	U-Net Res	61.64 ± 1.60	85.42 ± 0.68
420 k	CapsNet	**73.09 ± 1.54**	**90.84 ± 0.68**
	U-Net Res	70.26 ± 2.00	90.57 ± 0.36

Bold values indicate best results

Table 12 Segmentation performances of capsule networks and baseline U-Net Res on the 40 MHz dataset measured by ordinary and average Hausdorff distance

# Params	Network	Hausdorff distance [mm]		Ave. Hausdorff dist. [mm]	
		Vessel wall	Lumen	Vessel wall	Lumen
32 k	CapsNet	**1.115 ± .073**	**.632 ± .047**	**.121 ± .009**	**.034 ± .003**
	U-Net Res	1.599 ± .177	.995 ± .047	.198 ± .019	.065 ± .008
102 k	CapsNet	**.874 ± .097**	**.514 ± .096**	**.087 ± .008**	**.025 ± .005**
	U-Net Res	1.286 ± .106	1.652 ± .110	.162 ± .018	.078 ± .026
420 k	CapsNet	**.857 ± .063**	**.463 ± .048**	**.085 ± .006**	**.022 ± .006**
	U-Net Res	.996 ± .139	.522 ± .065	.097 ± .017	.028 ± .006

Bold values indicate best results

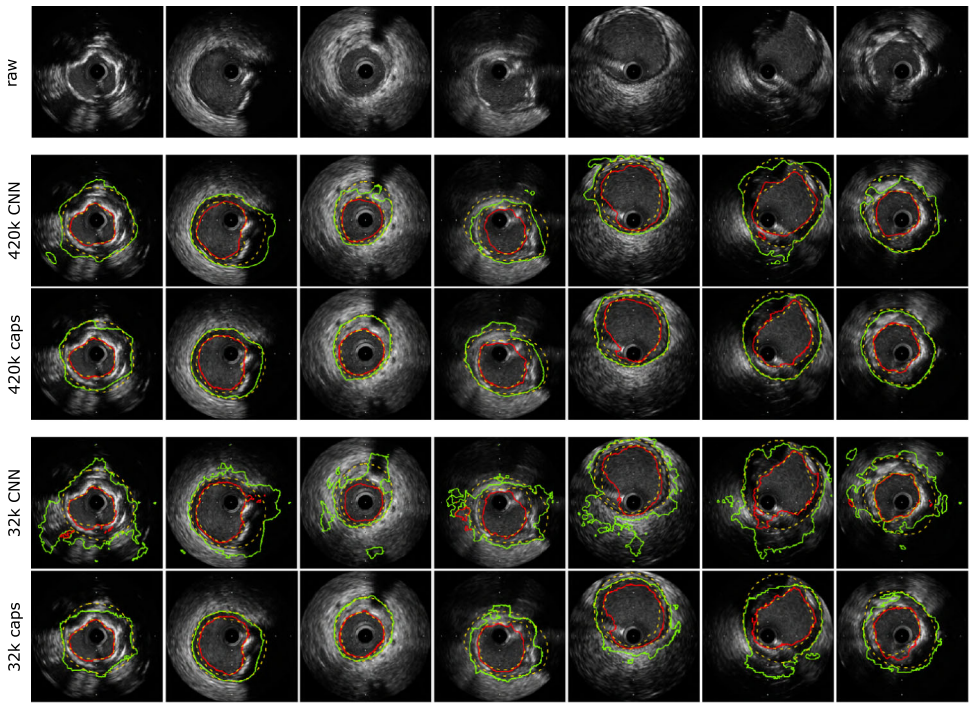

Fig. 4 Comparison of exemplary segmentation results between capsule networks and CNNs for the 40 MHz dataset. Shown are predictions of large networks with 420 k parameters and small networks with 32 k parameters. Ground truth annotations of lumen border and external elastic membrane are depicted with yellow dashed lines. The predicted contours with red and green solid lines, respectively

4.5% for networks with 32 k parameters. The corresponding improvements of the average Hausdorff distance are 12.8% and 46.1%. When using 50 training images, the improvements of the Dice score are 4.6% for networks with 420 k parameters and 19.6% for networks with 32 k parameters. The corresponding average Hausdorff distances improve by 26.5% and 61.9%.

Furthermore, we see that the performance drops of the capsule networks, when decreasing the number of parameters, are substantially smaller compared to the baseline CNNs. In the case of the vessel wall, the Dice scores drop about − 2.4% vs. − 5.8% for networks trained with 250 images and − 1.2% vs. − 15.6% for networks trained with 50 images.

Figure 3 shows exemplary segmentation results for the cases of 250 and 50 training images. It can be seen that the capsule networks are able to complete the vessel wall shape in shadowed regions quite well (see, e.g., Fig. 3 columns 1, 3 and 4), whereas the CNNs fail to do so. Additionally, the predictions of the capsule networks always exhibit a closed vessel wall shape which completely surrounds the lumen. This is not always the case for the CNN predictions (see Fig. 3 columns 2, 3 and 5). Hence, we can assume that the capsule network learned some kind of shape representation of vessel walls and is able to interpolate missing grayvalue gradient information.

In addition, we provide segmentation results for the 40 MHz dataset in Tables 11 and 12. The picture is generally the same as for the 20 MHz dataset. Exemplary segmentation results are depicted in Fig. 4. It can be seen that the capsule network is capable of inferring vessel borders in shadowed regions, as was the case for 20 MHz images. Furthermore, the decrease in performance when reducing the number of network parameters is substantially smaller compared to the baseline CNN. All in all, this shows that the capsule network

architecture optimized for the 20 MHz dataset can be readily used for the 40 MHz dataset.

The major drawback of the capsule network is the long training time compared to the U-Net Res. The largest capsule network needed approximately 16 h training time for five-fold cross-validation, whereas training the corresponding U-Net Res only took roughly 45 min. Also the required amount of graphics memory differed largely. The largest U-Net Res model needed about 3.5 GB of VRAM, whereas the largest capsule network occupied about 20 GB. All experiments were performed on an NVIDIA Titan RTX GPU with 24 GB of VRAM. The main reason for this large difference is the iterative routing process. This also affects the inference time which was more than 30 times longer than the corresponding CNN inference time (e.g., 100 ms vs. 3 ms for networks with 420 k parameters).

Nevertheless, in the case of IVUS, image segmentation capsule networks turned out to be quite performant on small datasets, even with a rather small network size of 32 k parameters. This makes capsule networks promising candidates for few-shot learning tasks like patient adaptation or detection of diseases with small prevalence as well as for applications on mobile devices.

Conclusion

We systematically optimized a capsule network architecture for segmentation of intravascular ultrasound (IVUS) images. The approach of doubling the number of capsule types at each downsampling level analogous to typical CNN architectures turned out to be quite beneficial. We showed that our capsule network performs particularly well on a small dataset compared to a corresponding U-Net Res. We thus assume that capsule networks are promising candidates for ultrasound image segmentation in general when dealing with data scarcity. This could make capsule networks suitable for few- or even single-shot learning tasks as well as applications for mobile devices. Further research should focus on tackling such tasks with capsule networks.

Funding Open Access funding enabled and organized by Projekt DEAL. This work was partially funded by the European Regional Development Fund (ERDF) and the Free and Hanseatic City of Hamburg in the Hamburgische Investitions- und Förderbank (IFB)-Program PROFI Transfer Plus under grant MALEKA.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Not applicable

References

1. Balocco S, Gatta C, Ciompi F, Wahle A, Radeva P, Carlier S, Unal G, Sanidas E, Mauri J, Carillo X, Kovarnik T, Wang CW, Chen HC, Exarchos TP, Fotiadis DI, Destrempes F, Cloutier G, Pujol O, Alberti M, Mendizabal-Ruiz EG, Rivera M, Aksoy T, Downe RW, Kakadiaris IA (2014) Standardized evaluation methodology and reference database for evaluating IVUS image segmentation. Comput Med Imaging Graph 38(2):70–90
2. Bargsten L, Schlaefer A (2020) SpeckleGAN: a generative adversarial network with an adaptive speckle layer to augment limited training data for ultrasound image processing. Int J Comput Assist Radiol Surg 15(9):1427–1436
3. Bonheur S, Štern D, Payer C, Pienn M, Olschewski H, Urschler M (2019) Matwo-capsnet: a multi-label semantic segmentation capsules network. Med Image Comput Comput Assist Interv 2019:664–672
4. Chen F, Ma R, Liu J, Zhu M, Liao H (2018) Lumen and media-adventitia border detection in ivus images using texture enhanced deformable model. Comput Med Imaging Graph 66(July 2017):1–13
5. Dubuisson MP, Jain A (1994) A modified hausdorff distance for object matching. In: Proceedings of the 12th international conference on pattern recognition, pp 566–568
6. Dzhioeva O (2019) Mobile ultrasound systems as a modern tool for the doctor. Med Univ 2(4):134–138
7. Gao Z, Chung J, Abdelrazek M, Leung S, Hau WK, Xian Z, Zhang H, Li S (2020) Privileged modality distillation for vessel border detection in intracoronary imaging. IEEE Trans Med Imaging 39(5):1524–1534
8. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 770–778
9. Hinton GE, Krizhevsky A, Wang SD (2011) Transforming auto-encoders. Artif Neural Netw Mach Learn 2011:44–51
10. Hinton GE, Sabour S, Frosst N (2018) Matrix capsules with EM routing. In: International conference on learning representations
11. Jayasundara V, Jayasekara S, Jayasekara H, Rajasegaran J, Seneviratne S, Rodrigo R (2019) Textcaps: handwritten character recognition with very small datasets. In: 2019 IEEE winter conference on applications of computer vision (WACV), pp 254–262
12. Jiménez-Sánchez A, Albarqouni S, Mateus D (2018) Capsule networks against medical imaging data challenges. In: Intravascular imaging and computer assisted stenting and large-scale annotation of biomedical data and expert label synthesis, pp 150–160
13. Katouzian A, Angelini ED, Carlier SG, Suri JS, Navab N, Laine AF (2012) A state-of-the-art review on segmentation algorithms in intravascular ultrasound (ivus) images. IEEE Trans Inf Technol Biomed 16(5):823–834

14. Kermani A, Ayatollahi A (2019) A new nonparametric statistical approach to detect lumen and media-adventitia borders in intravascular ultrasound frames. Comput Biol Med 104:10–28
15. Kim S, Jang Y, Jeon B, Hong Y, Shim H, Chang H (2018) 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, pp 161–168
16. LaLonde R, Bagci U (2018) Capsules for object segmentation. ArXiv arXiv:1804.04241
17. Li YC, Shen TY, Chen CC, Chang WT, Lee PY, Huang CC (2021) Automatic detection of atherosclerotic plaque and calcification from intravascular ultrasound images by using deep convolutional neural networks. IEEE Trans Ultrason Ferroelectr Freq Control
18. Milletari F, Navab N, Ahmadi S (2016) V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth international conference on 3D vision (3DV), pp 565–571
19. Nandamuri S, China D, Mitra P, Sheet D (2019) Sumnet: fully convolutional model for fast segmentation of anatomical structures in ultrasound volumes. In: 2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019), pp 1729–1732
20. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: Medical image computing and computer-assisted intervention (MICCAI), pp 234–241
21. Räber L, Mintz GS, Koskinas KC, Johnson TW, Holm NR, Onuma Y, Radu MD, Joner M, Yu B, Jia H, Meneveau N, de la Torre Hernandez JM, Escaned J, Hill J, Prati F, Colombo A, di Mario C, Regar E, Capodanno D, Wijns W, Byrne RA, Guagliumi G, Group ESD (2018) 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. Eur Heart J 39(35):3281–3300
22. Sabour S, Frosst N, Hinton GE (2017) Dynamic routing between capsules. In: Proceedings of the 31st international conference on neural information processing systems, pp 3859–3869
23. Sonka M, Zhang X, Siebes M, Dejong S, McKay CR, Collins SM (1994) Automated segmentation of coronary wall and plaque from intravascular ultrasound image sequences. Comput Cardiol 1994:281–284
24. Sudre CH, Li W, Vercauteren T, Ourselin S, Jorge Cardoso M (2017) 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, pp 240–248
25. Taha AA, Hanbury A (2015) Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. BMC Med Imaging 15:1–28
26. Wang Y, Yao Q, Kwok JT, Ni LM (2020) Generalizing from a few examples: a survey on few-shot learning. ACM Comput Surv 53(3):1–34
27. Xia M, Yan W, Huang Y, Guo Y, Zhou G, Wang Y (2019) Ivus image segmentation using superpixel-wise fuzzy clustering and level set evolution. Appl Sci 9(22):4967
28. Xia M, Yan W, Huang Y, Guo Y, Zhou G, Wang Y (2020) Extracting membrane borders in ivus images using a multi-scale feature aggregated u-net. In: Proceedings of the annual international conference of the IEEE engineering in medicine and biology society, EMBS, pp 1650–1653
29. Yang J, Faraji M, Basu A (2019) Robust segmentation of arterial walls in intravascular ultrasound images using Dual Path U-Net. Ultrasonics 96:24–33
30. Zhang X, Luo P, Hu X, Wang J, Zhou J (2018) Research on classification performance of small-scale dataset based on capsule network. In: Proceedings of the 4th international conference on robotics and artificial intelligence, pp 24–28
31. Ziemer PGP, Bulant CA, Orlando JI, Maso Talou GD, Álvarez LAM, Guedes Bezerra C, Lemos PA, García-García HM, Blanco PJ (2020) Automated lumen segmentation using multi-frame convolutional neural networks in intravascular ultrasound datasets. Eur Heart J Digit Health 1(1):75–82