Ultrasonics 96 (2019) 24–33

Contents lists available at ScienceDirect

Ultrasonics

journal homepage: www.elsevier.com/locate/ultras

# Robust segmentation of arterial walls in intravascular ultrasound images using Dual Path U-Net[☆]

Ji Yang[1], Mehdi Faraji[1], Anup Basu[*]

*Department of Computing Science, University of Alberta, Canada*

ARTICLE INFO

*Keywords:*
Dual Path UNet
DPU-Net
Real-time augmentor
Intravascular
Ultrasound
IVUS
Segmentation

ABSTRACT

A Fully Convolutional Network (FCN) based deep architecture called Dual Path U-Net (DPU-Net) is proposed for automatic segmentation of the lumen and media-adventitia in **I**ntra**V**ascular **U**ltra**S**ound (IVUS) frames, which is crucial for diagnosis of many cardiovascular diseases and also for facilitating 3D reconstructions of human arteries. One of the most prevalent problems in medical image analysis is the lack of training data. To overcome this limitation, we propose a twofold solution. First, we introduce a deep architecture that is able to learn using a small number of training images and still achieves a high degree of generalization ability. Second, we strengthen the proposed DPU-Net by having a real-time augmentor control the image augmentation process. Our real-time augmentor contains specially-designed operations that simulate three types of IVUS artifacts and integrate them into the training images. We exhaustively assessed our twofold contribution over Balocco's standard publicly available IVUS 20 MHz and 40 MHz B-mode dataset, which contain 109 training image, 326 test images and 19 training images, 59 test images, respectively. Models are trained from scratch with the training images provided and evaluated with two commonly used metrics in the IVUS segmentation literature, namely Jaccard Measure (JM) and Hausdorff Distance (HD). Experimental results show that DPU-Net achieves 0.87 JM, 0.82 mm HD and 0.86 JM, 1.07 mm HD over 40 MHz dataset for segmenting the lumen and the media, respectively. Also, DPU-Net achieves 0.90 JM, 0.25 mm HD and 0.92 JM, 0.30 mm HD over 20 MHz images for segmenting the lumen and the media, respectively. In addition, DPU-Net outperforms existing methods by 8–15% in terms of HD distance. DPU-Net also shows a strong generalization property for predicting images in the test sets that contain a significant amount of major artifacts such as bifurcations, shadows, and side branches that are not common in the training set. Furthermore, DPU-Net runs within 0.03 s to segment each frame with a single modern GPU (Nvidia GTX 1080). The proposed work leverages modern deep learning-based method for segmentation of lumen and the media vessel walls in both 20 MHz and 40 MHz IVUS B-mode images and achieves state-of-the-art results without any manual intervention. The code is available online at https://github.com/Kulbear/IVUS-Ultrasonic.

## 1. Introduction

Cardiovascular diseases, due to their high incidence, high mortality and irreversible after-effect, require accurate/fast detection and treatment. **I**ntra**V**ascular **U**ltra**S**ound (IVUS) is one of the imaging modalities that is commonly used to assist medical staff and helps them diagnose cardiovascular diseases. IVUS provides the medical experts with an inside-out view of the coronary artery. To acquire the IVUS frames, a catheter that carries an ultrasound emitter is inserted into the coronary artery to provide a cross-sectional tomographic view of the artery. Although IVUS frames allow assessing the vessel morphology [1], the coronary images acquired are not easy for the human eyes to interpret. No doubt that automatically and accurately labeling (segment) vessel walls from IVUS frames would be beneficial for many applications. For instance, medical staff can benefit from the automatic labeling of the boundaries of the circular layers to diagnose cardiovascular diseases. Therefore, segmentation of the acquired IVUS images has important clinical implications even though it has always been a challenging task since IVUS images usually contain significant imaging artifacts. In particular, an accurate separation of the interior (lumen) and exterior (media) vessel walls in IVUS images plays a critical role in creating precise 3D reconstructions of the artery and also diagnose cardiovascular diseases where such a quantitative measurement of the coronary artery boundary can affect clinical decisions.

[*] Corresponding author.

*E-mail addresses:* jyang7@ualberta.ca (J. Yang), faraji@ualberta.ca (M. Faraji), basu@ualberta.ca (A. Basu).

[1] Equal Contributions.

https://doi.org/10.1016/j.ultras.2019.03.014
Received 3 December 2018; Received in revised form 19 February 2019; Accepted 16 March 2019
Available online 23 March 2019
0041-624X/ © 2019 Published by Elsevier B.V.

The segmentation of IVUS images is a well-investigated problem from a conventional perspective where numerous ideas and approaches of computer vision and image processing have been used [2–8]. Older methods used various strategies such as shape and intensity priors[6], gradient vector flow in a nonparametric energy function [5], parametric deformable models with probabilistic cost functions [4,2], or even the radio frequency signals [3] to segment lumen and media. In recent years, a significant amount of research has been conducted and supported by successful experimental results. In [7,8], the authors leverage a type of region detector called Extremal Regions of Extremum Level (EREL) [9,10] to cluster the regions of interest, namely, the lumen and media. In order to segment the arterial walls in IVUS frames, Yang et al. [11] and Kim et al. [12], proposed two derivatives of the well-known U-Net [13] that are based on the concept of deep convolutional neural networks. Convolutional Neural Networks (CNNs) play an important role in visual image recognition since 2012 following the success of AlexNet in the ImageNet competition [14]. Semantic segmentation is one of the most active research fields in computer vision and image processing that has been a subject of a large volume of research, such as in[15–19]. A very popular architecture was proposed in [16] that attempted to transfer the knowledge learned from image classification tasks to semantic segmentation by making the network architecture in a fully convolutional fashion. Since then, Fully Convolutional Networks (FCNs) have been widely used to solve the problem of making pixel-level dense predictions.

Several years have passed after the emergence of deep learning [14] and although numerous deep learning based methods have dominated almost every field of study related to computer vision and medical image analysis, not many deep learning based approaches have been used in segmentation of intravascular ultrasound frames. One of the main reasons why only a small number of studies have been conducted in this field is the lack of sufficient training data. Many deep architectures require a very large number of training examples in order to achieve high quality generalization abilities. However, the available IVUS datasets contain a very small number of training images. For example, Balocco's dataset [1] A contains only 19 training images. Having a small number of training images significantly decreases the performance of a deep model especially in IVUS segmentation and since most of the state-of-the-art pre-trained models trained on natural photo images [14,20], transfer learning cannot be a valid option for IVUS segmentation and hence, the model needs to be trained from scratch.

In order to overcome the aforementioned problems, we propose a deep architecture in this paper, that is not only able to better generalize (after training over a small number of images) than the current deep models, but also does not need to be initialized by a pre-trained model. DPU-Net, our proposed fully convolutional network has been built based on the UNet architecture and is an improved extension of the IVUSNet [11] that can automatically delineate the boundary of the lumen and the media vessel walls. We have evaluated the generalization ability of the proposed DPU-Net by comparing it with other two state-of-the-art architectures trained over the same number of frames for the same amount of time, namely SegNet [21] and UNet [13]. Our results show that DPU-Net outperforms both of them in terms of the accuracy of the segmentation denoted by Jaccard Measure (JM), alternatively called Intersection over Union (IoU), and Hausdorff Distance (HD). The fact that DPU-Net can be trained using a small number of images and achieves better generalization, could make it a suitable choice for various medical image analysis problems.

In addition to the proposed DPU-Net, we introduce several augmentation operations that are specially designed to alleviate the effects of three common IVUS artifacts, namely shadow, side vessel and bifurcations. These operations are included in our real-time augmentation framework that is able to generate augmented images for training as quickly as requested.

We evaluated the proposed work on the two test sets of a publicly available IVUS B-mode benchmark dataset [1]. These sets not only contain a small number of training images but also have test sets generated based on different distributions of the artifacts than the training set. The evaluation results reveal that, the proposed DPU-Net enhanced by the real-time augmentor outperforms every existing state-of-the-art approach. Also, due to our IVUS-based augmentation operations, the accuracy results for images contaminated with artifacts are far superior to other automatic methods. Our work is an end-to-end method that requires no human intervention.

The rest of the paper is organized as the follows: Section 3 provides a detailed explanation of our proposed architecture. Section 2.1 states how we build the augmentation pipeline. We also demonstrate multiple experiments that reinforce our contribution in Section 4. Finally, we conclude our study in Section 6.

## 2. Datasets

We exploits a publicly available IVUS dataset [1] to validate DPU-Net. This dataset is designed to be useful in different approaches that might need a single frame or a multi-frame dataset [1]. Therefore, there are two datasets available which were obtained with different two ultrasound frequencies including 20 MHz and 40 MHz. Note that the two datasets are obtained and stored to different resolutions, which are 384-by-384 for the 20 MHz dataset and 512-by-512 for the 40 MHz dataset. The 20 MHz dataset contains two sets (train and test) of IVUS gated frames using a full pullback at the end-diastolic cardiac phase from 10 patients. Dataset frames were manually annotated by four clinical experts. Specifically, two of them repeated the task one week after the first marking [1]. There are 109 and 326 IVUS frames in the training and testing sets, respectively. Also, the test set contains a large number of IVUS artifacts including bifurcation (44 frames), side vessel (93 frames), and shadow (96 frames) artifacts. The remaining 143 frames do not contain any artifacts except for plaque.

The 40 MHz dataset also comes with two predefined sets (train and test) both contain annotated ground truth. The train and test sets consist of 19 and 59 IVUS frames, respectively. Having small number of training images significantly increases the difficulty of training a well-generalized deep model to produce results at a reasonable level. We address the problem of having a limited training examples with DPU-Net along with major data augmentations as mentioned in Section 2.1.

### 2.1. Augmentation

The idea of data augmentation is to make an effort to fill the missing and unseen values by augmenting the observed data, and has been studied in machine learning for several decades [22]. The 'craving for data' however was felt more strongly after the emergence of large deep models, especially in computer vision and image analysis where the great dimensionality of the problem's input domain creates significant variability in the latent space. Therefore, once one decides to employ a deep model, augmenting the training set becomes an inevitable necessity, especially when we do not have a big and diverse training set which is always the case in medical image analysis applications. Most of medical datasets contain a relatively small number of images due to a range of difficulties in acquiring medical images. For example, the Balocco datasets A and B [1] contain only 19 and 109 training images, respectively, which highlights the need for artificially augmenting the training sets. In order to best utilize the available information in the training set, we designed a real-time augmenter class capable of simultaneously transforming and warping the training images and also several new operations specifically designed for IVUS images.

#### 2.1.1. Augmentation operations

Various types (operations) of augmentations have been tried in the literature to create new training images: Elastic distortions are used in visual document analysis [23], RGB color shifting, translation, scale, horizontal shearing, horizontal flipping, and rotation transformations are examples that have been considered in [24–26,14,20,27,17,13]. In medical applications, on the other hand, not all kinds of augmentations might be useful. In fact, the type of medical image augmentation (transformation) strongly depends on the application. Performing redundant augmentation thus increases the training time without fulfilling noticeable accuracy gains [28]. For example translating (shifting) IVUS frames does not sound plausible since the catheter is always located at the center of the images. Thus, the model will never see an IVUS frame with a catheter located at somewhere else rather than the center of the frame. In contrast, rotations and flipping seem reasonable because it is highly probable that the catheter is rotated or moved inside the vessels. Therefore, in this section we explore the effects of various types of the existing transformations on the accuracy of the trained models. We also introduce three new filters that have been exclusively designed for augmenting IVUS frames. We show that applying a combination of the existing transformations and our designed masks on the training frames can grant the model a greater ability to generalize and thus improve the accuracy of the final segmentation.

#### 2.1.2. Proposed augmentation operations

Designing particular strategies to alleviate the effects of ultrasound artifacts has been practiced many times in various conventional studies. In [29], a circular cursor was dragged to cover the catheter artifact. In [30], a preprocessing step was added to the method to clear away the motion artifacts. Many other studies has also made an effort to include the artifact detection steps [6,31–33]. In the proposed study, not only do we use the aforementioned common augmentation operations such as rotation and rescaling, but we have also devised three different types of augmentations to mimic the common IVUS artifacts, namely bifurcation, side vessel, and shadow. These operations are included into our augmentation pipeline to increase the robustness of the DPU-Net architecture in dealing with several types of IVUS artifacts.

In order to mimic bifurcation, side vessel and shadow artifacts, we have designed three different masks with dimensions equal to the size of the training images. Each of these masks were rotated offline 360 times.[2] Fig. 1 illustrates all the masks. The masks designed can then be multiplied by the training images at the augmentation time. One advantage of using masks during augmentation is its low computational cost, since applying the masks only needs the multiplication operation and no warping or convolution is required. Another benefit of using this type of augmentation is that we can apply several operations to one image at the same time. Considering that the masks designed to mimic shadow, side vessel, and bifurcation are called shadow mask, side vessel mask, and bifurcation mask and are denoted by $\mathcal{SH}$, $\mathcal{SV}$, $\mathcal{BF}$, a new augmented image ($\mathcal{I}^*$) can be obtained as follows:

$$\mathcal{I}^* = \mathcal{SH} * \mathcal{SV} * \mathcal{BF} * \mathcal{I} \tag{1}$$

where $\mathcal{I}$ denotes an original training image and $*$ represents an element-wise multiplication. It should be noted that other operations based on each specific mask can be defined as well such as $\mathcal{I}^{\mathcal{SH}} = \mathcal{SH} * \mathcal{I}$ for only applying the shadow mask, $\mathcal{I}^{\mathcal{SV}} = \mathcal{SV} * \mathcal{I}$ for only applying the side vessel mask, $\mathcal{I}^{\mathcal{BF}} = \mathcal{BF} * \mathcal{I}$ for only applying the bifurcation mask, $\mathcal{I}^{\mathcal{SV}*\mathcal{BF}} = \mathcal{SV} * \mathcal{BF} * \mathcal{I}$ for applying the side vessel and bifurcation mask together and so forth. Fig. 2 shows three of these masks randomly selected and the resulting images after applying them to one of the 20 MHz training frames of the Balocco's dataset [1].

#### 2.1.3. Real-time augmenter

From a practical point of view, most types of augmentations are done by applying a specific transformation (such as similarity or affine) on the image. This shows that the image augmentation is inherently a computationally expensive task since a costly warping procedure (that includes interpolation) needs to be performed after applying the transformation on the training images. The cumbersome process of augmenting new images significantly increases the training time. Thus, in order to prevent the training process from being delayed, the new images are augmented in either an *offline* or *online* form. Basically, if the augmented training set consists of a small number of images and enough disk space is available, the training set can be augmented and saved on disk before the training process is started. This way of creating new images is called offline augmentation. However, due to the high disk space requirement, offline augmentation is not practical in many applications that tend to train a large model over large datasets such as ImageNet [34] or Microsoft coco [35]. A simple remedy for this is to let the CPU augment the images while the training process is being performed on GPU which is called online augmentation. This technique has been adopted during training in many of the state-of-the-art deep learning architectures [14,26]. But recent advances in GPU hardware and software has made it possible to feed forward and back-propagate the images incredibly fast and hence even a small computation on CPU will delay the training process. Therefore, in our study, we propose a real-time image augmenter module, a process that runs on the background and controls a number of parallel processes designed to augment the required images. The training process (runs on GPU) can instantly read an image from a buffer that is always full of augmented images. As soon as an image is read from the buffer, the processes that are running on the background in each CPU core, augment another image and place it into an empty location. Fig. 3 illustrates how our proposed real-time augmentation pipeline works.

## 3. Proposed method

Since [14], deep learning (DL) approaches have been shown to achieve superior performance on many vision recognition tasks. In particular, Convolutional Neural Networks (CNN) and its derivatives outperform conventional methods in almost all visual recognition tasks that can be treated as supervised learning tasks [20,18,21,27]. In this section, we first introduce the popular IVUS B-mode dataset we used to train and validate the proposed work. Then, we propose a Dual Path UNet (DPU-Net), which produces a binary prediction mask for either the lumen or media area to delineate the vessel wall, with detailed explanation in terms of its intuition and design.

### 3.1. Dual Path UNet

Dual Path UNet (DPU-Net) is designed incorporating intuitions from human perception and lessons learned from existing popular fully convolutional network (FCN) [16,21,36] and segmentation-purposed refinements for FCNs. We adopt U-Net [13] as the base architecture of our proposed work according to the overall architecture design. As in popular 1-stage architectures, there are two major components for DPU-Net:

1. An encoder network that can downsample and process the input to produce a low-resolution deep feature map.
2. A decoder network that can restore the resolution of the deep feature map output by the encoder network towards the original size.

Due to the limited computational power and memory we had, we first downsample the input images before we feed them to the deep model and upsample the prediction result at the end of the model. This procedure will be explained later in this section. The upsampled output feature map is sent to one more convolutional layer followed by a sigmoid activation to produce the final result.

The encoder network contains 6 encoding blocks whereas the decoder network contains 5 decoding blocks. Starting from the second block in the network, each block receives the feature map from its previous block; and specifically for each decoder block, there is an extra skip connection that can help forward information from the encoder

[2] All of the designed masks are available online.

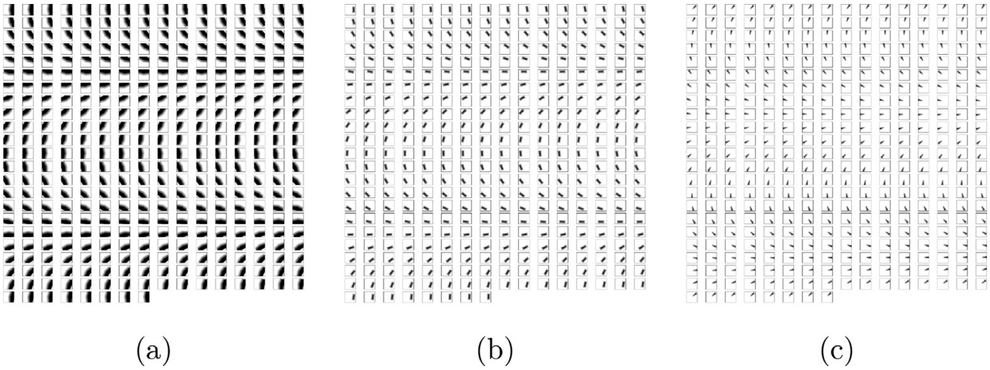

**Fig. 1.** 360 designed masks for emulating the common artifacts of IVUS frames. (a) Designed shadow masks ($\mathcal{SH}$). (b) Designed side vessel masks ($\mathcal{SV}$). (c) Designed bifurcation masks ($\mathcal{BF}$).

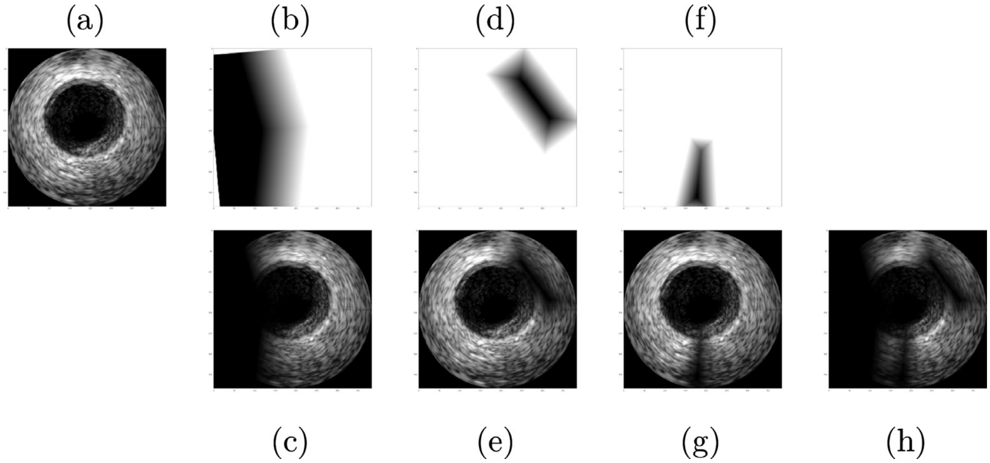

**Fig. 2.** The proposed artifact simulation operations to mimic the three common IVUS artifacts. (a) An original training **I**mage ($\mathcal{I}$). (b) Designed **SH**adow mask ($\mathcal{SH}$). (c) The image obtained after multiplying the shadow mask (b) by the image ($\mathcal{I}^{\mathcal{SH}}$). (d) Designed **S**ide **V**essel mask ($\mathcal{SV}$). (e) The result of multiplying the image by the side vessel mask ($\mathcal{I}^{\mathcal{SV}}$). (f) Designed **B**i**F**urcation mask ($\mathcal{BF}$). (g) The result of multiplying the image by the bifurcation mask ($\mathcal{I}^{\mathcal{BF}}$). (h) The image obtained after applying all of the masks designed to the original image at the same time ($\mathcal{I}^*$).

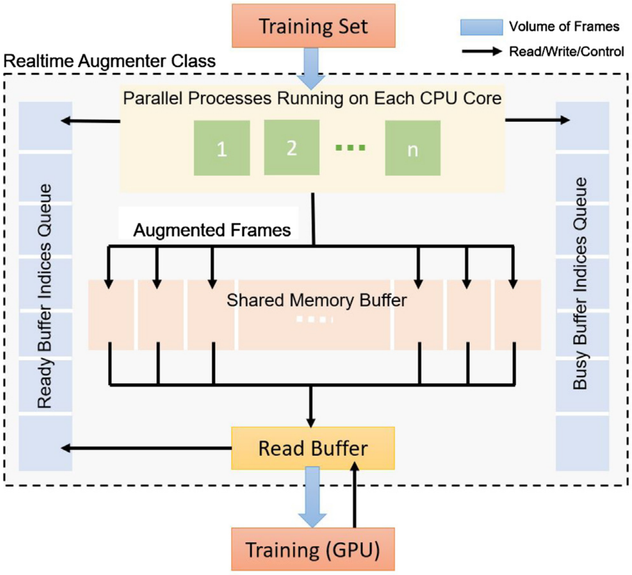

**Fig. 3.** The proposed real-time augmenter.

network. Skip connections from the encoder network to the decoder network give additional information to help restore the feature map to the original size. Particularly, spatial relations among pixels are preserved as the skip connections actually connect corresponding blocks between the encoder and the decoder. Note that these skip connections can also help reinforce the gradient flow in deep models therefore avoiding the common gradient vanishing problem and speed up the training process [37]. The entire architecture is therefore symmetric as shown in Fig. 4. There are minor differences among the blocks in the architecture.

There are two common options for the pooling layer, namely average pooling and max pooling. Max pooling is widely used for downsampling the input feature map obtained from the previous layer. We choose max pooling over average pooling as we consider the IVUS images to be relatively blurry with low resolution. Max pooling forces the network to capture the information in the most activated neuron in a sub-region of the kernel size, but drops other non-significant information.

Except for the first encoding block, each encoding block contains a downsampling branch that downsamples the received feature map, followed by a two-branch convolution path, as shown in Fig. 5(a). To avoid losing information caused by using max pooling and also to reduce the spatial resolution of the input, we build and expand the downsampling branch by using both a 2-by-2 average pooling layer and a 3-by-3 convolutional layer with a stride of 2. Finally, we concatenate the two outputs together at the depth dimension. This aggregation idea is similar to [27,36].

After downsampling, the aggregated feature map output by the downsampling branch is passed to two subsequent branches, namely the refining branch and the main branch. First, we follow the design in [21,13] to include a branch with consecutive convolutional layers

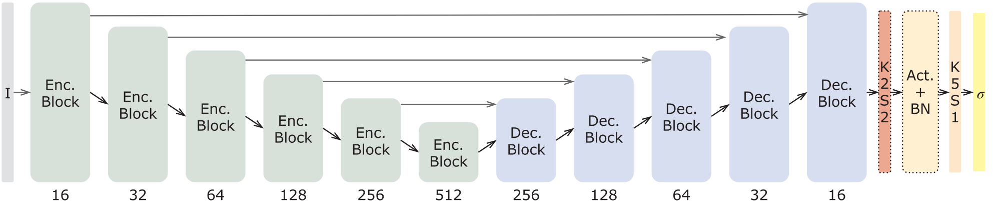

**Fig. 4.** The DPU-Net architecture. Every convolutional layer in the same block has the same output depth as labeled at the bottom of the block. At the end, the layer in red represents a 2-by-2 transposed convolution (deconvolution) with a stride size of 2, where the layer right before the sigmoid output (labeled with yellow) is a 5-by-5 convolution layer. The abbreviations used in the figure mean as follows: "K2S2" means "kernel size 2 and stride size 2", "BN" means "batch normalization". (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

followed by activation and batch normalization, here we call it *main branch*.

A recent trend is to use small kernel size for the feature map refinement [17,19], also the concept of networks in network [27] is widely used in the literature. Therefore, we introduce a *refining branch* that has one convolutional layer with a 3-by-3 kernel size followed by a convolutional layer with a 1-by-1 kernel size to produce similar but refined feature map. A 1-by-1 convolution is able to refine or trim a feature map since it covers only a single pixel without influence from its neighbors. But over all the depth, this idea is similar to the global average pooling [38] with more learning capacity. In addition, as the ability to capture features at different scales are usually desired, we set convolutional layers with a kernel size of 5 in the main branch, compared to the kernel sizes of 3 and 1 in the refining branch. The outputs from the main and refining branches are summed up and passed to the next block and its corresponding decoding block. A critical issue is that deep networks are hard to train due to the gradient vanishing problem. The multi-branch and local networks-in-network architecture not only provide a good local topology but also reinforce the gradient flow to accelerate the training.

Decoding blocks needs a slightly different configuration, as shown in Fig. 5(b). Every decoding block receives the feature map from both its previous block and its corresponding encoding block. Only the feature map received from the previous block is upsampled by a 2-by-2 transposed convolution and then concatenated with the feature map from its corresponding encoding block. Note that this concatenated feature map will only be passed to the main branch, where the refining branch handles the upsampled feature map only.

The activation used in the DPU-Net is the **P**arametric **Re**ctified **L**inear **U**nit (PReLU) [39].

$$\mathrm{PReLU}(x) = \max(0, x) - \alpha \max(0, -x) \tag{2}$$

Compared to the ordinary ReLU activation, PReLU [39] allows a part of the gradients flow through the neuron when it is not activated, whereas ReLU only passes gradients when the neuron is active. As suggested in [39,40], PReLU outperforms ReLU in many benchmarks and also has a more stable performance.

To reduce the computation cost involved during training, we initially downsampled the input image by a factor of 2 (i.e., a 384-by-384 image in the 20 MHz dataset will be resized to a 192-by-192 image). However, the ground truth masks are kept in the original size. The reason why we do not downsample the ground truth masks is to make the model predict a smoother map since if we downsample the binary ground truth masks, errors (especially for pixels around boundary) will be added to the ground truth. Therefore, by not changing the ground truth dimension, we achieve a smoother predictions especially around boundaries. The feature map in all of the available architectures should be in the downsampled size in terms of the width and height dimensions, while we add an extra resize branch that helps restore the feature map to its original size, which is the same as the ground truth mask in width and height.

Finally, the output feature map from the last decoding block is first upsampled by a 2-by-2 transposed convolution layer with a stride size of 2 then refined by a 5-by-5 convolution layer, which is experimentally proved to help improve performance [14]. Thanks to the extra 2-by-2 transposed convolution layer, the obtained final outputs now have the exact same size as the original dataset images. As we want DPU-Net to produce binary masks, the activation used after the last convolutional layer is a sigmoid function. Also, it is worth mentioning that the skip-connections between corresponding encoding and decoding blocks add

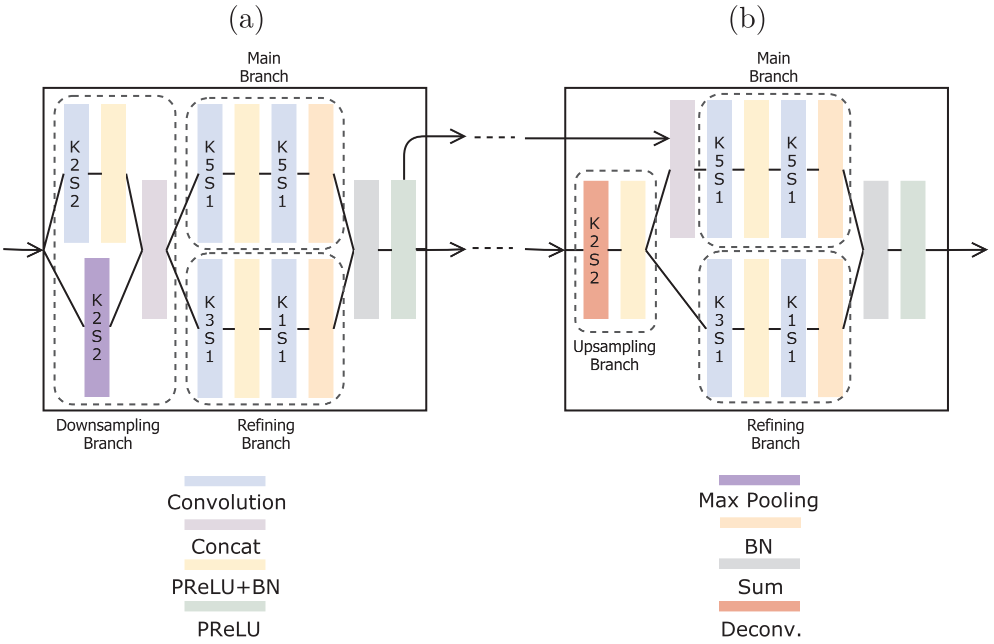

**Fig. 5.** A detailed illustration on the encoding block and the decoding block. Note that the first encoding block does not have the downsampling branch; therefore the *main branch* and *refining branch* will directly accept the raw image as the input. (a) An encoding block with downsampling branch, followed by the main branch and the refining branch. (b) A typical decoding block that accepts feature maps from both the previous block and the skip-connection. The abbreviations used in the figure mean as follows: "K2S2" means "kernel size 2 and stride size 2", "BN" means "batch normalization.".

context information for the decoder network and also provide extra gradient flow to the current architecture.

## 4. Experiments

In this section, we first introduce how we setup and train the DPU-Net. Then, we show a detailed investigation on the effectiveness of different augmentation techniques for training IVUS segmentation models. In addition, we provide a group of comparison experiments that shows our architecture surpassed the performance of some of the best previous work, namely U-Net [13] and SegNet [21]. Finally, we report the accuracy of the segmentation results by DPU-Net and compare it with existing IVUS segmentation literature.

### *4.1. Experiments setup*

The training and evaluation is based on two publicly available IVUS B-mode datasets [1]. Both datasets have been widely used in the IVUS segmentation literature [7,41,29,8]. The two datasets are acquired with different ultrasound frequencies, namely, 20 MHz and 40 MHz. The pattern and texture are similar in these two groups of IVUS images. As no official validation set was provided in [1], we randomly select a small subset from the augmented images as our validation set during the training to do the 5-fold cross validation. As the result of the 5-fold cross validation, we need to train 5 models based on the 5 different training/validation splits. We then use the 5 models to preform inference on their corresponding validations set. After doing this, we should end up with the completed out-of-fold inferenced training set. We evaluated this out-of-fold training set to select the best training configuration of the architecture, namely, the batch size, learning rate, and the depth of each blocks in the architecture. All models are trained end-to-end over only the training set of the two given datasets without involving any other external resources such as extra training images and pretrained model weights.

Metrics used for evaluating segmentation results have been calculated by the provided function in dataset which needs the contour of the predicted segment and the ground truth as input argument. Jaccard Measure (JM) and Hausdorff Distance (HD) are two popular segmentation performance metrics that have been calculated using the aforementioned function. The Jaccard Measure, sometimes called Intersection over Union (IoU), is calculated based on the comparison of the automatic segmentation from the deep model ($y_{pred}$) and the manual segmentation delineated by experts ($y_{true}$).

$$\mathrm{IoU} = \frac{y_{pred} \cap y_{true}}{y_{pred} \cup y_{true}} \tag{3}$$

The Hausdorff Distance between the automatic ($C_{pred}$) and manual ($C_{true}$) curves is the greatest distance of all points belonging to $C_{pred}$ to the closest point [7] in $C_{true}$ and is defined as follows:

$$\mathrm{HD} = \max\{d(C_{pred}, C_{true}), d(C_{true}, C_{pred})\} \tag{4}$$

### *4.2. Training the model*

All the models are trained and evaluated on a computer with a Core i7-8700 K processor, 16 GB of RAM, and a GTX 1080 8 GB graphics card. Training a model from scratch generally takes less than 40 min to complete. To make the training faster and enjoy a relatively large batch size, we downsized every frame of the dataset by a factor of 2 as mentioned earlier.

We implement DPU-Net with TensorFlow [42]. The weights in the model are all initialized using He initialization [39] with the default setting in TensorFlow. Then, we train the model with Adam optimization [43]. The learning rate is set to 0.0001 with no decay scheme. The real-time augmentor is used through training each model for 96 epochs, with a batch size of 6 and 144 iterations in total for each epoch. Note that we need two separate groups of models to predict the lumen area and the media area since the output activation is a sigmoid function:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

#### *4.2.1. Loss function*

We propose a heuristically designed loss function, that helps the model benefit from both Binary Cross Entropy (BCE) and the Soft Dice Loss (SD). We minimize this objective function which is in fact a weighted average of the aforementioned loss functions.

$$L(Y, \widehat{Y}) = \lambda\, \mathrm{SD}(Y, \widehat{Y}) + (1 - \lambda)\mathrm{BCE}(Y, \widehat{Y}) \tag{6}$$

where $\lambda = 0.8$ denotes the weight of each term, the pixel-wise binary cross entropy is

$$\mathrm{BCE}\left(Y, \widehat{Y}\right) = \frac{1}{N}\sum_{i,j} -\left(y_{ij}\log(\widehat{y}_{ij}) + \left(1 - y_{ij}\right)\log\left(1 - \widehat{y}_{ij}\right)\right) \tag{7}$$

and the soft dice loss which is a differentiable form of intersection over union is defined as:

$$\mathrm{SD}\left(Y, \widehat{Y}\right) = 1 - \frac{1}{N}\sum_{i,j} \frac{\widehat{y}_{ij}\cdot y_{ij}}{\widehat{y}_{ij}\cdot\widehat{y}_{ij} + y_{ij}\cdot y_{ij}} \tag{8}$$

where N represents the total number of pixels in each image, $y_{ij}$ is the ground truth value of $i$th row and $j$th column in the image annotation (either 1 or 0) and $\widehat{y}_{ij}$ is the predicted probability of being a foreground pixel (either the lumen or the media).

For training each model, we monitor the average Jaccard Measure without extracting contours. The prediction given by a single model is a probability map of the input image size. We follow the same simple average ensemble practice in [15] to produce the final result. The binarization is performed by using a searched threshold value where the threshold is searched on the out-of-fold version of the training set. Once the prediction maps are generated, ensembled, and binarized, we fill any hole inside the binary region using the 'fillhole' algorithm explained in [44], extract and trace the boundary using the algorithm described in [45], smooth the contour coordinates using 'rloess' [46] method and report it as the final segmentation output.

### *4.3. Results*

In this section, we report the results of our thorough experiments over various augmented sets. Particularly, we present and compare the segmentation output of DPU-Net over 10 augmented sets for both 20 MHz and 40 MHz frames as well as segmentation results of UNet and SegNet trained over the same set. Also, we compare the segmentation output of DPU-Net trained over the best augmented set with existing state-of-the-art IVUS segmentation approaches.

#### *4.3.1. Augmentation results*

In order to have a firm grasp on the effects of various augmentation operations, we have trained our proposed DPU-Net over augmented images generated based on various combinations of the proposed operations. The results are reported and compared together in Table 1 over the original training sets of Balocco's dataset [1] which contains only 19 and 109 images for 40 MHz and 20 MHz IVUS frames (without any augmentation), respectively. The performance of the DPU-Net can be seen in Table 2 where all the evaluations results are obtained over the official test set.

#### *4.3.2. DPU-Net vs SegNet and U-Net*

In the image segmentation literature there are two well-known architectures, namely the SegNet [21] for street scene segmentation and

**Table 1**
Performance of the proposed DPU-Net over the test set of 20 MHz and 40 MHz IVUS frames of Balocco's dataset [1] based on training over various combinations of augmentation operations. The quantitative results are evaluated on the official test set. The evaluation measures are Jaccard Measure (JM), Hausdorff Distance (HD) in mm, and Percentage of Area Difference (PAD).

| | | | | | Dataset A(40 MHz frames) | | | | Dataset B(20 MHz frames) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Lumen | | Media | | Lumen | | Media | |
| Rotation | Side Vessel ($\mathcal{SV}$) | Shadow ($\mathcal{SH}$) | Bifurcation ($\mathcal{BF}$) | Scale | JM | HD | JM | HD | JM | HD | JM | HD |
| ✓ | | | | | 0.856 (0.07) | 0.929 (0.85) | 0.856 (0.08) | 1.175 (0.75) | 0.899 (0.05) | 0.257 (0.2) | 0.887 (0.07) | 0.447 (0.36) |
| ✓ | ✓ | | | | 0.865 (0.06) | 0.850 (0.65) | 0.866 (0.09) | 1.040 (0.71) | 0.898 (0.05) | 0.263 (0.21) | 0.887 (0.08) | 0.430 (0.36) |
| ✓ | | | ✓ | | 0.858 (0.06) | 0.857 (0.61) | 0.862 (0.08) | 1.156 (0.76) | 0.898 (0.05) | 0.263 (0.22) | 0.895 (0.07) | 0.402 (0.34) |
| ✓ | | ✓ | | | 0.865 (0.07) | 0.853 (0.65) | **0.874** (0.08) | 1.001 (0.68) | 0.896 (0.06) | 0.285 (0.25) | 0.892 (0.08) | 0.404 (0.34) |
| ✓ | | | | ✓ | 0.865 (0.06) | 0.840 (0.62) | 0.848 (0.08) | 1.262 (0.73) | 0.899 (0.05) | 0.264 (0.23) | 0.914 (0.06) | 0.332 (0.36) |
| ✓ | ✓ | | ✓ | | 0.869 (0.06) | **0.813** (0.63) | 0.865 (0.09) | 1.070 (0.70) | 0.900 (0.05) | 0.266 (0.23) | 0.888 (0.07) | 0.419 (0.34) |
| ✓ | ✓ | ✓ | | | 0.866 (0.07) | 0.834 (0.63) | 0.873 (0.08) | **0.984** (0.66) | 0.893 (0.06) | 0.292 (0.25) | 0.882 (0.09) | 0.432 (0.36) |
| ✓ | | ✓ | ✓ | | 0.869 (0.06) | 0.829 (0.65) | 0.870 (0.08) | 1.013 (0.65) | 0.896 (0.06) | 0.289 (0.27) | 0.889 (0.08) | 0.401 (0.35) |
| ✓ | ✓ | ✓ | ✓ | | 0.869 (0.06) | 0.838 (0.62) | 0.868 (0.08) | 1.014 (0.64) | 0.894 (0.06) | 0.291 (0.24) | 0.882 (0.08) | 0.445 (0.36) |
| ✓ | ✓ | ✓ | ✓ | ✓ | **0.869** (0.06) | 0.818 (0.61) | 0.863 (0.09) | 1.073 (0.72) | **0.902** (0.05) | **0.252** (0.20) | **0.921** (0.07) | **0.300** (0.35) |

Bold implies best results.

the U-Net [13] for segmentation of neuronal structures in electron microscope stacks. As both architectures can output a prediction map at the original input size (for U-Net, we need to change the valid padding to the same padding for all convolutional layers). To keep the comparison fair, we add the last upsampling transposed convolution layer to the end of both architectures. For illustration purposes, let the original size of the training images be $W \times H$, as we mentioned in Section 3, we downsample the training images to a size of $\frac{W}{2} \times \frac{H}{2}$. We train these two networks and our DPU-Net with the re-scaled low-resolution images and masks at the original size with no augmentation applied. The comparison results are shown in Table 2.

### 4.3.3. *Comparison with existing methods*

In this section, we present experimental results over the test set of 20 MHz and 40 MHz IVUS B-mode datasets [1]. We obtained 5 models according to the 5-fold cross validation for each dataset with the configuration mentioned in Section 4.2 and ensemble the predicted maps by the simple average voting. The quantitative results are shown in Table 3.

## 5. Discussion

In this study, we performed several experiments to evaluate our proposed DPU-Net along with showing the improvement achieved by employing the newly proposed augmentation operations. We drew a comparison between DPU-Net, SegNet and UNet over original IVUS frames and also augmented frames. From the quantitative results we can see that DPU-Net performance is significantly better than the other two existing architectures. We discuss three empirical reasons that help achieve such a result. First, DPU-Net has more convolutional layers but lower depth for each convolutional layer. This is similar to having several layers with a relatively small amount of neurons instead of putting many neurons in a single layer. A multi-layer architecture has stronger ability of learning representation [20,14] as it provides more possible ways of intersecting features and more non-linearity can be introduced by multiple non-linear activations. Since a deeper layer in a neural network usually has a larger receptive field, it can capture features at multiple scales. However, there is always the limitation that a kernel of a fixed size cannot be a universal solution. In the two IVUS datasets, especially the 40 MHz one, the shape and size of the lumen or media region are significantly different among images. In both U-Net and SegNet, there is no particular design to handle or improve the multi-scale segmentation. Our main branch and refining branch handle this problem naturally by having convolutional layers with different kernel sizes. Other improvements of DPU-Net over the original SegNet and U-Net consist of employing two downsampling methods simultaneously, namely the pooling and strided convolution. This modification over these two base architectures can help to ensure that we can leverage the information in multiple ways to increase the diversity of features.

Having shown the superiority of DPU-Net to UNet and SegNet that all were trained over non-augmented IVUS frames, we compared various combinations of our proposed augmentation operations in Table 1 to figure out the best combination of the operations. Although, it was almost obvious that a training set contains all of the operations will achieve the best results, we did the cumbersome processes of training over different combinations of operations to support and prove our hypothesis as it is shown in Table 1 so. Looking at the reported results in Table 2, we see that training UNet and SegNet over these operations significantly improved their resulted segmentation, though DPU-Net still outperforms them. Training over a set created using our proposed augmentation operations can dramatically increases the accuracy of the segmentation. More specifically, it can increase the lumen JM from 0.823 to 0.869 for 40 MHz images and from 0.871 to 0.902 for 20 MHz frames. The accuracy of media segmentation is also increased from 0.775 to 0.863 for 40 MHz images and from 0.855 to 0.921 for 20 MHz

**Table 2**

Comparison of the proposed DPU-Net over the entire 40 MHz and 20 MHz IVUS frames of Dataset A and B with U-Net and SegNet trained over frames with and without augmentation. Symbol * shows that the model was trained over augmented set contained all of operations used in this study. Note that all methods were trained by similar parameters and even the same strategy to smooth the boundary for DPU-Net has been used to smooth the predictions of UNet and SegNet. The evaluation measures are Jaccard Measure (JM) and Hausdorff Distance (HD).

| | Dataset A(40 MHz frames) | | | | Dataset B(20 MHz frames) | | | |
|---|---|---|---|---|---|---|---|---|
| | Lumen | | Media | | Lumen | | Media | |
| | JM | HD | JM | HD | JM | HD | JM | HD |
| DPU-Net | **0.823** (0.08) | **1.070** (0.60) | **0.775** (0.10) | **1.813** (0.90) | **0.871** (0.06) | **0.322** (0.25) | **0.855** (0.07) | 0.647 (0.45) |
| UNet | 0.778 (0.09) | 1.235 (0.82) | 0.746 (0.11) | 1.894 (1.01) | 0.830 (0.06) | 0.356 (0.21) | 0.808 (0.08) | **0.590** (0.43) |
| SegNet | 0.784 (0.09) | 1.239 (0.85) | 0.746 (0.10) | 1.875 (0.99) | 0.824 (0.06) | 0.367 (0.23) | 0.805 (0.11) | 0.590 (0.43) |
| DPU-Net* | **0.869** (0.06) | **0.818** (0.61) | **0.863** (0.09) | **1.073** (0.72) | **0.902** (0.05) | **0.252** (0.20) | **0.921** (0.07) | **0.300** (0.35) |
| UNet* | 0.864 (0.06) | 0.866 (0.61) | 0.836 (0.09) | 1.320 (0.75) | 0.884 (0.05) | 0.289 (0.20) | 0.883 (0.08) | 0.527 (0.4) |
| SegNet* | 0.853 (0.07) | 0.923 (0.61) | 0.833 (0.10) | 1.312 (0.81) | 0.884 (0.05) | 0.289 (0.20) | 0.883 (0.08) | 0.527 (0.46) |

Bold implies best results.

**Table 3**

Performance of the proposed DPU-Net over Balocco's dataset A and B [1]. Measures represent the mean and standard deviation evaluated on the test set frames of the dataset and categorized based on the presence of a specific artifact in each frame. The evaluation measures are Jaccard Measure (JM) and Hausdorff Distance (HD). The values are rounded to two decimal places.

| Artifact | | Dataset A(40 MHz frames) | | | | Dataset B(20 MHz frames) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Lumen | | Media | | Lumen | | Media | |
| | Method | JM | HD | JM | HD | JM | HD | JM | HD |
| All | DPU-Net (Proposed) | **0.87(0.07)** | **0.82(0.61)** | **0.86(0.09)** | **1.07(0.72)** | **0.90(0.05)** | **0.25(0.20)** | **0.92(0.07)** | **0.30(0.35)** |
| | IVUSNet [11] | – | – | – | – | 0.90(0.06) | 0.26(0.25) | 0.86(0.11) | 0.48(0.44) |
| | Faraji et al. [7] | – | – | – | – | 0.87(0.06) | 0.30(0.20) | 0.77(0.17) | 0.67(0.54) |
| | Downe et al. [29] | – | – | – | – | 0.77(0.09) | 0.47(0.22) | 0.74(0.17) | 0.76(0.48) |
| | Exarchos et al. [1] | 0.80(0.14) | 1.32(1.18) | 0.80(0.13) | 1.57(1.03) | 0.81(0.09) | 0.42(0.22) | 0.79(0.11) | 0.60(0.28) |
| | Destrempes et al.[1] (semi) | 0.85(0.12) | 1.16(1.12) | 0.86(0.11) | 1.18(1.02) | 0.88(0.05) | 0.34(0.14) | 0.91(0.04) | 0.31(0.12) |
| None | DPU-Net (Proposed) | **0.88(0.02)** | **0.66(0.38)** | **0.95(0.03)** | **0.53(0.31)** | 10.91(0.04) | 0.25(0.17) | **0.95(0.02)** | **0.17(0.08)** |
| | IVUSNet [11] | – | – | – | – | **0.91(0.03)** | **0.21(0.09)** | 0.92(0.05) | 0.27(0.23) |
| | Faraji et al. [7] | – | – | – | – | 0.88 (0.05) | 0.29 (0.17) | 0.89 (0.07) | 0.31 (0.23) |
| Bifurcation | DPU-Net (Proposed) | 0.83(0.08) | **1.18(0.75)** | **0.81(0.11)** | 1.72(0.91) | 0.85(0.10) | 0.46(0.38) | 0.86(0.10) | 0.60(0.35) |
| | IVUSNet [11] | – | – | – | – | 0.82(0.11) | 0.50(0.58) | 0.78(0.11) | 0.82(0.60) |
| | Faraji et al. [7] | – | – | – | – | 0.79 (0.10) | 0.53 (0.34) | 0.57 (0.13) | 1.22 (0.45) |
| | Downe et al. [29] | – | – | – | – | 0.70(0.11) | 0.64(0.27) | 0.71(0.19) | 0.79(0.53) |
| | Exarchos et al. [1] | 0.79(0.12) | 1.43(1.42) | 0.78(0.13) | 1.75(1.12) | 0.80(0.09) | 0.47(0.23) | 0.78(0.11) | 0.63 (0.25) |
| | Destrempes et al.[1] (semi) | **0.84(0.11)** | 1.37(1.12) | 0.81(0.14) | **1.56(1.36)** | **0.85(0.05)** | **0.42(0.18)** | **0.91(0.03)** | **0.32(0.13)** |
| Side Vessel | DPU-Net (Proposed) | **0.90**(0.01) | **0.68**(0.35) | **0.90**(0.03) | 1.21(0.54) | **0.91(0.04)** | **0.20(0.12)** | 0.91(0.08) | 0.35(0.36) |
| | IVUSNet [11] | – | – | – | – | 0.90(0.04) | 0.23(0.12) | 0.83(0.14) | 0.59(0.49) |
| | Faraji et al. [7] | – | – | – | – | 0.87 (0.05) | 0.24 (0.11) | 0.73 (0.60) | 0.74 (0.18) |
| | Downe et al. [29] | – | – | – | – | 0.70(0.11) | 0.64(0.27) | 0.71(0.19) | 0.79(0.53) |
| | Exarchos et al. [1] | 0.78(0.16) | 1.28(0.95) | 0.78(0.11) | 1.94(1.02) | 0.77(0.09) | 0.53(0.24) | 0.78(0.12) | 0.63 (0.31) |
| | Destrempes et al.[1] (semi) | 0.84(0.13) | 1.07(1.12) | 0.87(0.18) | **1.02(0.58)** | 0.87(0.04) | 0.36(0.15) | **0.91(0.04)** | **0.31(0.12)** |
| Shadow | DPU-Net (Proposed) | **0.87(0.07)** | **0.75(0.61)** | **0.86(0.09)** | **1.13(0.74)** | **0.89(0.05)** | **0.25(0.20)** | 0.88(0.10) | 0.48(0.48) |
| | IVUSNet [11] | – | – | – | – | 0.87(0.06) | 0.27(0.25) | 0.76(0.12) | 0.80(0.45) |
| | Faraji et al. [7] | – | – | – | – | 0.86 (0.07) | 0.29 (0.20) | 0.58 (0.13) | 1.24 (0.39) |
| | Downe et al. [29] | – | – | – | – | 0.70(0.11) | 0.64(0.27) | 0.71(0.19) | 0.79(0.53) |
| | Exarchos et al. [1] | 0.78(0.15) | 1.43(1.18) | 0.80(0.14) | 1.58(1.04) | 0.80(0.10) | 0.46(0.19) | 0.82(0.11) | 0.57 (0.28) |
| | Destrempes et al.[1] (semi) | 0.83(0.13) | 1.25(1.20) | 0.86(0.11) | 1.14(1.05) | 0.87(0.05) | 0.39(0.18) | **0.92(0.03)** | **0.33(0.14)** |

Bold implies best results.

images.

We also compared the segmentation results of DPU-Net with other state-of-the-art IVUS segmentation approaches. The results are reported in Table 3. As we can see, DPU-Net outperforms existing methods by a significant margin. According to the Jaccard Measure, we achieve 4% and 8% improvement for the lumen and the media, respectively. If we look at the Hausdorff distance, DPU-Net obtains 8% and 20% improvement for the lumen and the media, respectively. In addition, DPU-Net performs particularly well on images with no artifact. It considerably increases JM and HD of the segmentation. It is worth noting that DPU-Net significantly improves the result for segmenting the media region, from an JM score 0.79 [1] to 0.92, and HD from 0.60 to 0.30 on the 20 MHz dataset. The reasons behind why DPU-Net does not exceed all the methods in every category of [1] can be addressed from two perspectives. First, the training set is too small to capture all the common artifacts in the real world and even the test set. However, the architecture is still considerably effective as the training set contains only 1 image with side vessel artifact while the test set contains 93 frames with side vessel artifacts. Secondly, the shadow artifacts are generally overlapped with parts of the media area that makes the

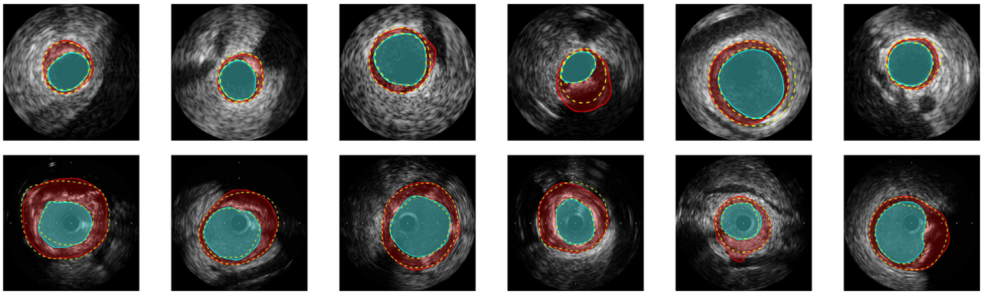

**Fig. 6.** Some of the lumen and media segmentation results for images from dataset B (first row) and dataset A (second row). Segmented lumen and media have been highlighted by cyan and red colors, respectively. The yellow dashed lines illustrate the gold standard that have been delineated by four clinical experts [1]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

segmentation becomes much more challenging since the media regions leak to the background. Some predictions are illustrated in Fig. 6.

## 6. Conclusion

In this paper, we proposed DPU-Net, a fully convolutional deep network, that is able to generalize even if there is a small number of training images for the segmentation of arterial walls in IVUS images. We evaluated the generalization ability of our proposed DPU-Net by comparing it with two existing general-purposed segmentation architectures, namely SegNet and UNet, that were trained over the same number of images for the same amount of time without doing any augmentation. The results indicate a significant improvement for DPU-Net over SegNet and UNet. Specifically, DPU-Net achieved more than 4% and 5% higher accuracy in terms of JM for 40 MHz and 20 MHz datasets, respectively. These emprical outcomes express a higher generalization ability than SegNet and UNet.

The contributions of this paper can be summarized as follows:

- We introduced a domain-specific design for image augmentation that can:
  - Generate various types of augmented images in real-time.
  - Add various combinations of three common IVUS artifacts into the training images.

    We empirically proved that we can produce a significant number of effective augmented images. This can be counted an effective augmentation pipeline and can be generalized for different deep architectures and tasks.
- We proposed DPU-Net that outperforms existing approaches over a publicly available IVUS benchmark dataset [1] which contains IVUS images with a significant number of artifacts. We also compare it with several existing influential architectures in the deep learning literature, namely SegNet [21] and U-Net [13]. This shows that the proposed work has a potential to be used in solving other segmentation problems as well.

To further improve the segmentation performance, we devised our own augmentation framework called real-time augmentor. Our real-time augmentor not only generates augmented images in a way that does not interrupt the training process on GPU(s), but also contains our proposed IVUS artifact-based augmentation operations that include three common IVUS artifacts into the training data to simulate the frames with artifacts. We thoroughly investigated how various augmentation operations affect the final accuracy of the model. Consequently, the experimental results reveal that a DPU-Net model trained using these augmented data outperforms every available state-of-the-art automatic and semi-automatic IVUS segmentation methods by a large margin.

## Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.ultras.2019.03.014.

## References

[1] S. Balocco, C. Gatta, F. Ciompi, A. Wahle, P. Radeva, S. Carlier, G. Unal, E. Sanidas, J. Mauri, X. Carillo, et al., Standardized evaluation methodology and reference database for evaluating IVUS image segmentation, Comput. Med. Imag. Graph. 38 (2) (2014) 70–90.

[2] E.G. Mendizabal-Ruiz, M. Rivera, I.A. Kakadiaris, Segmentation of the luminal border in intravascular ultrasound b-mode images using a probabilistic approach, Med. Image Anal. 17 (6) (2013) 649–670.

[3] G. Mendizabal-Ruiz, I.A. Kakadiaris, A physics-based intravascular ultrasound image reconstruction method for lumen segmentation, Comput. Biol. Med. 75 (2016) 19–29.

[4] A. Taki, Z. Najafi, A. Roodaki, S.K. Setarehdan, R.A. Zoroofi, A. Konig, N. Navab, Automatic segmentation of calcified plaques and vessel borders in IVUS images, Int. J. Comput. Assis. Radiol. Surg. 3 (3–4) (2008) 347–354.

[5] X. Zhu, P. Zhang, J. Shao, Y. Cheng, Y. Zhang, J. Bai, A snake-based method for segmentation of intravascular ultrasound images and its in vivo validation, Ultrasonics 51 (2) (2011) 181–189.

[6] G. Unal, S. Bucher, S. Carlier, G. Slabaugh, T. Fang, K. Tanaka, Shape-driven segmentation of the arterial wall in intravascular ultrasound images, IEEE Trans. Inform. Technol. Biomed. 12 (3) (2008) 335–347.

[7] M. Faraji, I. Cheng, I. Naudin, A. Basu, Segmentation of arterial walls in intravascular ultrasound cross-sectional images using extremal region selection, Ultrasonics 84 (2018) 356–365.

[8] Y. Li, M. Faraji, Erel Selection Using Morphological Relation. Available from: arXiv preprint arXiv:1806.03580.

[9] M. Faraji, J. Shanbehzadeh, K. Nasrollahi, T. Moeslund, Extremal regions detection guided by maxima of gradient magnitude, IEEE Trans. Image Process.

[10] M. Faraji, J. Shanbehzadeh, K. Nasrollahi, T.B. Moeslund, Erel: Extremal regions of extremum levels, 2015 IEEE International Conference on Image Processing (ICIP), IEEE, 2015, pp. 681–685.

[11] J. Yang, L. Tong, M. Faraji, A. Basu, Ivus-net: An Intravascular Ultrasound Segmentation Network. Available from: arXiv preprint arXiv:1806.03583.

[12] S. Kim, Y. Jang, B. Jeon, Y. Hong, H. Shim, H. Chang, Fully automatic segmentation of coronary arteries based on deep neural network in intravascular ultrasound images, Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis, Springer, 2018, pp. 161–168.

[13] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241.

[14] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[15] D. Ciresan, A. Giusti, L.M. Gambardella, J. Schmidhuber, Deep neural networks segment neuronal membranes in electron microscopy images, Advances in Neural Information Processing Systems, 2012, pp. 2843–2851.

[16] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[17] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFS, 2016. Available from: arXiv preprint arXiv:1606.00915.

[18] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, et al., Chexnet: Radiologist-Level Pneumonia Detection on Chest X-rays with Deep Learning. Available from: arXiv preprint arXiv:1711.05225.

[19] C. Peng, X. Zhang, G. Yu, G. Luo, J. Sun, Large Kernel Matters–Improve Semantic Segmentation by Global Convolutional Network. Available from: arXiv preprint arXiv:1703.02719.

[20] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[21] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: a deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495.

[22] M.A. Tanner, W.H. Wong, The calculation of posterior distributions by data augmentation, J. Am. Stat. Assoc. 82 (398) (1987) 528–540.

[23] P.Y. Simard, D. Steinkraus, J.C. Platt, Best practices for convolutional neural networks applied to visual document analysis, Null, IEEE, 2003, p. 958.

[24] D. Cireşan, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification. Available from: arXiv preprint arXiv:1202.2745.

[25] D.C. Cireşan, U. Meier, J. Masci, L.M. Gambardella, J. Schmidhuber, High-performance neural networks for visual object classification. Available from: arXiv preprint arXiv:1102.0183.

[26] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition. Available from: arXiv preprint arXiv:1409.1556.

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[28] M. Paulin, J. Revaud, Z. Harchaoui, F. Perronnin, C. Schmid, Transformation pursuit for image classification, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3646–3653.

[29] R. Downe, A. Wahle, T. Kovarnik, H. Skalicka, J. Lopez, J. Horak, M. Sonka, Segmentation of intravascular ultrasound images using graph search and a novel cost function, in: Proc. 2nd MICCAI Workshop on Computer Vision for Intravascular and Intracardiac Imaging, Citeseer, 2008, pp. 71–79.

[30] M.-H. Cardinal, J. Meunier, G. Soulez, R.L. Maurice, É. Therasse, G. Cloutier, Intravascular ultrasound image segmentation: a three-dimensional fast-marching method based on gray level distributions, IEEE Trans. Med. Imag. 25 (5) (2006) 590–601.

[31] J. Dijkstra, G. Koning, J. Tuinenburg, P. Oemrawsingh, J. Reiber, Automatic border detection in intravascular iltrasound images for quantitative measurements of the vessel, lumen and stent parameters, Computers in Cardiology 2001, IEEE, 2001, pp. 25–28.

[32] M.E. Plissiti, D.I. Fotiadis, L.K. Michalis, G.E. Bozios, An automated method for lumen and media-adventitia border detection in a sequence of IVUS frames, IEEE Trans. Inform. Technol. Biomed. 8 (2) (2004) 131–141.

[33] J.D. Klingensmith, R. Shekhar, D.G. Vince, Evaluation of three-dimensional segmentation algorithms for the identification of luminal and medial-adventitial borders in intravascular ultrasound images, IEEE Trans. Med. Imag. 19 (10) (2000) 996–1011.

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, IEEE, 2009, pp. 248–255.

[35] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: common objects in context, European Conference on Computer Vision, Springer, 2014, pp. 740–755.

[36] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 5987–5995.

[37] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, C. Pal, The importance of skip connections in biomedical image segmentation, Deep Learning and Data Labeling for Medical Applications, Springer, 2016, pp. 179–187.

[38] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2921–2929.

[39] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on imagenet classification, Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1026–1034.

[40] B. Xu, N. Wang, T. Chen, M. Li, Empirical evaluation of rectified activations in convolutional network. Available from: arXiv preprint arXiv:1505.00853.

[41] L. Lo Vercio, J.I. Orlando, M. del Fresno, I. Larrabide, Assessment of image features for vessel wall segmentation in intravascular ultrasound images, Int. J. Comput. Assis. Radiol. Surg. 11 (8) (2016) 1397–1407.

[42] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, Software Available from tensorflow.org, 2015. < https://www.tensorflow.org/ > .

[43] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization. Available from: arXiv preprint arXiv:1412.6980.

[44] P. Soille, Morphological Image Analysis: Principles and Applications, Springer Science & Business Media, 2013.

[45] R.C. Gonzalez, R.E. Woods, S. Eddins, Digital Image Processing using Matlab: Pearson Prentice Hall, Upper Saddle River, New Jersey.

[46] S.J. Orfanidis, Introduction to Signal Processing, Prentice-Hall Inc., 1995.