Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2020) 28: 2261 – 2277
© TÜBİTAK
doi:10.3906/elk-1912-97

PERI-Net: a parameter efficient residual inception network for medical image segmentation

Fatmatülzehra USLU[1,*], Cher BASS[2†], Anil A. BHARATH[2]
[1]Department of Electrical and Electronics Engineering, Bursa Technical University, Bursa, Turkey
[2]Department of Bioengineering, Imperial College London, London, UK

Received: 16.12.2019 • **Accepted/Published Online:** 11.04.2020 • **Final Version:** 29.07.2020

Abstract: Recent developments in deep networks allow us to train networks with more parameters by yielding better performance given sufficient amount of data. However, we are still restricted with the availability of labelled data in medical image segmentation, where the problem is exacerbated with high intra- and intervariability of anatomical structures. In order to bypass this problem without compromising network performance, this study introduces a PERI-Net, which promises to achieve higher performance while being with smaller parameter count such as on the order of 0.8 million than its counterparts. The network benefits from rich features generated by our versions of inception modules, better communication between encoding and decoding paths and an effective way of segmentation mask generation. We evaluate the performance of our architecture on the segmentation of retinal vasculature in fundus image datasets of DRIVE, CHASE_DB1 and IOSTAR and the segmentation of axons in a 2-photon microscopy image dataset. According to the results of our experiments, PERI-Net achieves state of the art performance on sensitivity and G-mean metrics with a significant margin for the 3 datasets, by outperforming our training of a U-net sharing the same properties and training strategies as PERI-Net.

Key words: Inception modules, U-net, axon, residual connections, vessels

1. Introduction

The segmentation of elongated anatomical structures such as vasculature, neuronal axons structures and lymphatic conduits is an initial step to characterise the geometrical properties of branching or connected structures, providing insight into normal biological processes and disease progression. The variety of biomedical imaging techniques (microscopy, ultrasound, optical coherence tomography) and contrast mechanisms (interferometry, reflectance, fluorescence) present significant challenges to approaches based on deep machine learning. This variability is compounded by vastly different imaging geometries, and biological variability. For example, the retinal vasculature manifests a tree like structure with widths of branches spanning a range of 1:20 [1]. On the other hand, axons have relatively regular widths but highly varying axon densities, a more tortuous geometry and appearance, due to limitations of imaging optics, the presence of microanatomical features (such as synaptic boutons) and very complex background appearance.

Recently, deep networks have been shown to outperform many state of the art methods on medical image analysis [2, 3]. In addition to their high performance, their ability to automatically learn features is a big advantage for medical image analysis, where intra- and intervariability of anatomical structures are high.

[*] Correspondence: fatmatulzehra.uslu@tbtu.edu.tr
[†] She is currently with Department of Biomedical Engineering in King's College London.

However, deep networks are usually of millions of parameters and require large amount of labelled data for training. This requirement may limit their performance in areas where obtaining labelled data is exhausting such as medical image segmentation, where pixel-level annotation is needed.

A commonly applied solution to this problem is data augmentation, where new images from the existence one can be produced with small perturbations to the colour or shape of objects in images or with flipping or rotating images themselves. Liskowski and Krawiec increased the size of original training dataset by a factor of 11 with this method [4]. Despite of it increasing variety in training data to some extend, this method typically increases the size of training data and it, sometimes, may produce images, which are meaningful for algorithms but not for real applications. Another solution can be to design smaller networks with high capacity such as inception modules [5]. Inception modules constitute important parts of ”GoogleNet” [5], which outperformed other state of the art methods on image classification and detection tasks in imagenet large-scale visual recognition challenge 2014 (ILSVRC14). Later, inception modules were used for a few image segmentation methods [6, 7]. The most limiting property of inception modules is that they have large parameter numbers though they increase diversity of features learned in the modules.

Recent development in deep learning architectures such as residual [8] and dense connections [9] have been shown to improve the performance of deep networks by improving gradient flow through networks. Lately, Dolz et al. extended inception modules by adding residual connections [10]. In addition to provide reuse of previously learned features in the same resolution [9], recently, dense connections have been used to provide links between features at different depths of a network in either encoding or decoding path or through skip connections and also between different data modalities [10–12].

In this study, we introduce PERI-Net, which stands for parameter efficient residual inception network. PERI-Net is an encoder-decoder network designed to improve the segmentation of vague structures such as vessels with the central light reflection in medical images. The capacity of the network is increased without a rise in parameter count with the use of rich feature sets generated by the proposed parameter efficient inception modules and its variants: residual inception modules and densely connected inception modules. In order to enhance segmentation performance of the network on structures with confusing appearance, we also introduce a way to generate segmentation masks, where 3 loss layers at different depths of the decoding path are designed to approximate the same reference segmentation mask for an input image. Eventually, we combine the output masks with max projection to obtain a single output mask.

We evaluate the performance of our method on retinal vasculature segmentation in well known fundus image datasets, namely, DRIVE, CHASE_DB1 and IOSTAR and on axon segmentation in a 2-photon microscopy image dataset [13]. According to the results of our experiments, the proposed architecture outperforms state of the art methods, including those using deep learning, based on performance metrics of sensitivity and G-mean. We also compared the performance of our network with that of U-net, which shares similar parameter count and training strategies with our architecture. We observed a consistent improvement in balanced accuracy, sensitivity and G-mean scores with our architecture over the performance of U-net for the 4 datasets (see Tables 1 and 2).

Table 1. The performance of the proposed method on various datasets, where images with the minimum and maximum performance are selected based on G-mean considering the the proposed network.

Dataset			Balanced accuracy	Sensitivity	Specificity	G-Mean
DRIVE	Min	**PERI-Net**	0.90	0.82	0.98	0.89
		U-net	0.87	0.75	0.99	0.86
	Max	**PERI-Net**	0.96	0.95	0.97	0.96
		U-net	0.95	0.93	0.97	0.95
CHASE_ DB1	Min	**PERI-Net**	0.90	0.82	0.97	0.89
		U-net	0.88	0.79	0.97	0.88
	Max	**PERI-Net**	0.94	0.91	0.97	0.94
		U-net	0.94	0.91	0.98	0.94
IOSTAR	Min	**PERI-Net**	0.90	0.81	0.98	0.89
		U-net	0.88	0.77	0.99	0.87
	Max	**PERI-Net**	0.94	0.91	0.97	0.94
		U-net	0.94	0.91	0.98	0.94
Axon data	Min	**PERI-Net**	0.83	0.68	0.98	0.82
		U-net	0.82	0.66	0.98	0.80
	Max	**PERI-Net**	0.94	0.91	0.98	0.94
		U-net	0.92	0.86	0.99	0.92

2. Background and related work

2.1. Inception modules

Szegedy et al. proposed a type of ”network in network” [14] architecture called inception modules, which consist of various sizes of filters in each module [5]. A basic version of the module typically contains 1×1, 3×3, 5×5 filters and, maybe, a pooling layer. Activation maps generated by the filters are augmented at the output of the module, as demonstrated in Figure 1a.

The superiority of an inception module to conventional convolutional units is that an input image is analysed through various filters providing information at a range of scales, in contrast to traditional convolutional network structures requiring the image to be passed through multiple layers of the network to produce a similar variety of features. However, using multiple filters in the same module has a big disadvantage, which is the increased number of parameters. In order to reduce the number of parameters of a module, Szegedy et al. used 1×1 filters to reduce the number of channels, prior to 3×3 and 5×5 filters. They also replaced large filters with separable filters; for example, a 5×5 filter is replaced with two 3×3 filters [15]. Moreover, they suggested the use of inception modules only in the deep layers of a network, where numbers of filters are traditionally larger than those in its earlier layers. Figure 1b shows a recent version of the inception module with the aforementioned modifications.

2.2. Dense connections

Huang et al. introduced densely connected networks for object recognition problems [9]. Their network was based on reusing features generated at previous layers in the next ones, with successive concatenation of feature maps of the same size, as demonstrated in Figure 2a. Dense connections have been used in different settings in image analysis such as the reuse of features between layers [11] or between image modalities [10].

Table 2. The vessel segmentation performance of our method and previous methods on DRIVE, CHASE_ DB1 and the axon datasets.

Dataset	Year	Method	Balanced accuracy	Sensitivity	Specificity	G-mean	AUC	Parameter count (million)
DRIVE	2019	**PERI-Net**	**0.93 ± 0.02**	**0.88 ± 0.04**	0.97 ± 0.01	**0.93 ± 0.02**	0.98 ± 0.00	**0.8**
		PERI-Net *	0.91 ± 0.02	0.87 ± 0.04	0.95 ± 0.01	0.91 ± 0.02	0.97 ± 0.00	1.
		U-net **	0.90 ± 0.02	0.84 ± 0.05	0.97 ± 0.01	0.90 ± 0.02	0.98 ± 0.01	1.
	2018	U-net [29] ***		0.75	0.98	0.86		
	2017	Orlando et al. [31]		0.79	0.97	0.88		
	2016	Liskowski and Krawiec [4]		0.75	0.98	0.86		48
	2016	Oliveira et al. [32]		0.87	0.96	0.91		
	2015	Li et al.[33]		0.76	0.98	0.86		
	2015	Wang et al. [34]		0.82	0.97	0.89		
	2014	Cheng et al. [35]		0.73	0.98	0.84		
	2013	Fraz et al. [36]		0.73	0.98	0.84		
CHASE_ DB1	2019	**PERI-Net**	**0.92 ± 0.02**	**0.87 ± 0.03**	0.97 ± 0.00	**0.92 ± 0.02**	0.99 ± 0.00	**0.8**
		PERI-Net *	0.90 ± 0.02	0.85 ± 0.04	0.95 ± 0.01	0.90 ± 0.02	0.97 ± 0.01	1.
		U-net **	0.91 ± 0.02	0.85 ± 0.04	0.97 ± 0.00	0.91 ± 0.02	0.99 ± 0.00	1.
	2018	Unet [29] ***		0.83	0.97	0.90		
	2017	Orlando et al. [31]		0.73	0.97	0.84		
	2015	Li et al.[33]		0.71	**0.98**	0.84		
	2012	Fraz et al.[21]		0.72	0.97	0.84		
IOSTAR	2019	**PERI-Net**	**0.92 ± 0.02**	**0.86 ± 0.04**	0.98 ± 0.01	**0.91 ± 0.02**	0.98 ± 0.01	**0.8**
		PERI-Net *	0.91 ± 0.02	0.84 ± 0.04	0.97 ± 0.01	0.90 ± 0.01	0.98 ± 0.01	1.
		U-net **	0.91 ± 0.02	0.83 ± 0.04	0.98 ± 0.01	**0.91 ± 0.02**	0.98 ± 0.00	1.
	2019	Uslu [30]		0.81	0.98	0.89		1.
	2016	Zhang et al. [22]		0.75	0.97	0.85		
	2017	Na et al. [37]		0.76	0.98	0.86		
Axon data	2019	**PERI-Net**	**0.90 ± 0.03**	**0.83 ± 0.06**	0.97 ± 0.02	**0.90 ± 0.04**	0.98 ± 0.00	**0.8**
		PERI-Net *	0.93 ± 0.03	0.89 ± 0.04	0.96 ± 0.02	0.93 ± 0.02	0.98 ± 0.01	1.
		U-net **	0.87 ± 0.03	0.76 ± 0.06	**0.98 ± 0.01**	0.86 ± 0.04	0.98 ± 0.00	1.

[*] This architecture is a modified version of PERI-Net where the RIMs and DCRIMs were replaced with naive inception modules while the rest of the architecture was maintained. See text for details.

[**] The results of our training of U-net.

[***] The results were reported by Alom et al. in [29], not in the original paper of U-net [17].

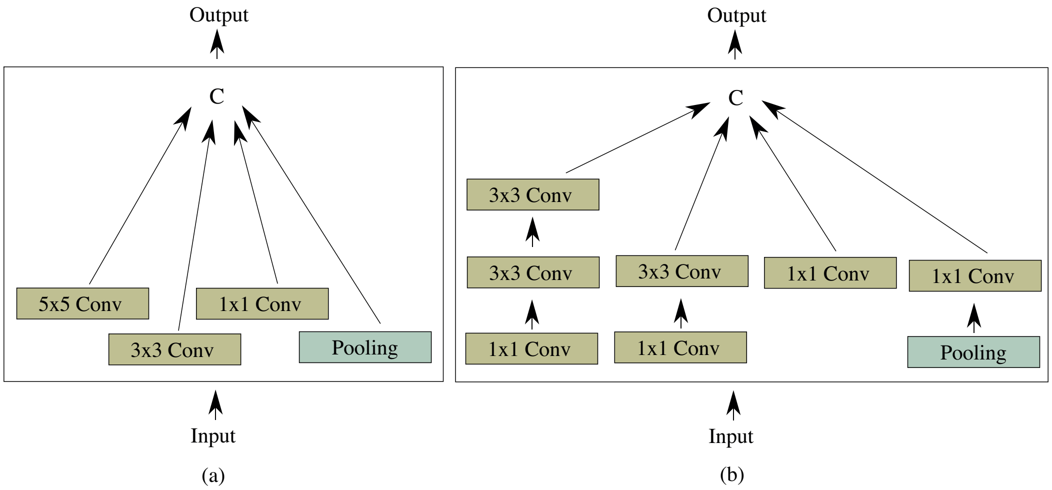


Figure 1. (a) The naïve inception module [5]. (b) The inception module with 1×1 filters prior to 3×3 filters and separable filters, where 5×5 filter is represented with two 3×3 filters [16] (Best viewed in color.)

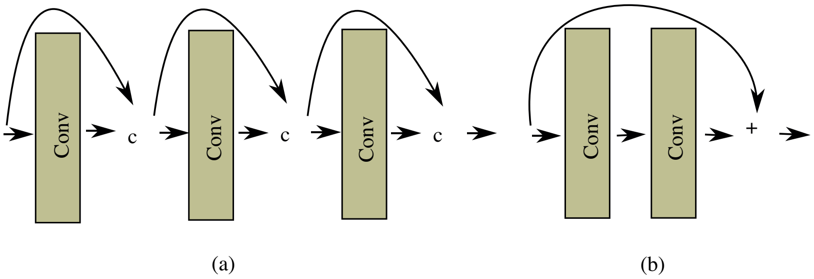


Figure 2. (a) A dense block, where '**c**' shows concatenation of features from previous layers with that from the current layer prior to entering the next layer. (b) A residual block, where '+' shows the addition of features from a previous layer with that from the current layer prior to entering the next layer (Best viewed in color.)

2.3. Residual connections

He et al. introduced residual connections to avoid gradient vanishing in the training of deep networks [8]. With residual connections, a feature set input to a stack of convolutional layers is added to the features generated by the stack, as demonstrated in Figure 2b. The output of a residual block can be formulated with equation (1).

$$F_o = F_i + G(F_i), \tag{1}$$

where F_i is input features, $G(\cdot)$ is a function performed by a stack of convolutional layers. F_o refers to the output features generated by the stack. $F_r = G(F_i)$, features generated by the stack, are called "residual features".

3. Method

The appearance of structures in many medical images has large intra- and intervariability in terms of size, contrast and pathologies. However, there are insufficient amounts of labelled data to exemplify this variation. In order to deal with the shortage of labelled data, this paper presents parameter efficient inception modules and its variants and proposes a network integrating the modules with an effective way of object mask generation.

3.1. Parameter efficient inception modules

The original inception modules have 1 branch for each size of filter, and 1 branch for the pooling layer (see Figure 1). The type of connections of these filters, which is parallel, may lead a network to learn redundant features, because of the repetitive use of the same size of filters for each branch. For example, the architecture in Figure 1(b) has four 1×1 filters to reduce the dimension of input features, which may be replaced with a single 1×1 filter. Similarly, the repetitive use of 3×3 filters for both 3×3 and 5×5 filters may be avoided with careful design.

We redesign the original inception modules by only keeping its 1 branch representing 5×5 filters with 2 consecutive 3×3 filters and adding max pooling if necessary, but sending features generated with each convolution as output to the next layer. This design, demonstrated in Figure 3, saves us 3 other branches in the original inception module. In order to keep the design simpler, we use the same number of filters for each filter type, which also gives equal chance to the detection of various sizes of structures. We also propose 2 variants of the module: residual inception modules and densely connected residual inception modules, which are illustrated in Figure 4. As indicated by the results of our experiments, which will be shown later, the modules play an important role in the performance of PERI-Net which outperform U-net, with the slightly larger parameter count than that of our network, with a remarkable margin on sensitivity and G-mean metrics.

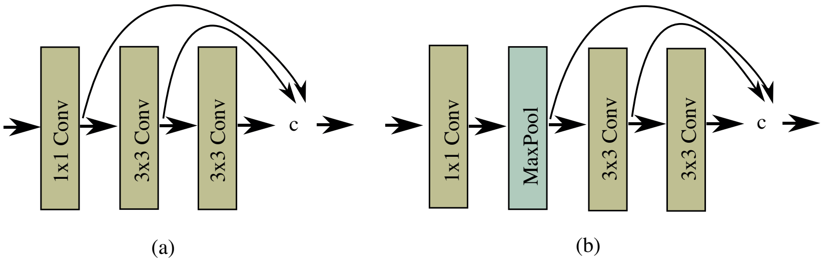


Figure 3. Parameter efficient inception modules without pooling (a) and with it (b). (Best viewed in color.)

3.2. Residual inception modules (RIM)

This variant of the parameter efficient inception modules contains residual connections. As far as we are aware, residual features have not been used to richen feature sets learned by a network but only to improve gradient flow [8]. Residual features can be written with equation (2).

$$F_r = F_o - F_i \tag{2}$$

A RIM combines residual features as follows:

$$F_{RIM} = F_r^1 \cup F_r^2 \cup \cdots \cup F_r^k \cup F_o^k, \tag{3}$$

where $\cup$ shows the concatenation of related features. k denotes the number of convolutional layers in a RIM. In this study, we set $k = 3$.

3.3. Densely connected residual inception modules (DCRIM)

In contrast to RIM, merely integrating residual features, a DCRIM combines different types of features: namely residual and input features, by extending the use of dense connections [9]. In a DCRIM, the input to n^{th}

convolutional layer can be formulated as $F_i^n = F_i^{n-1} \cup F_r^{n-1}$. A DCRIM incorporates the output features of each convolutional layers with the residual features of the last convolutional layer, as follows:

$$F_{DCRIM} = F_o^1 \cup F_o^2 \cup \cdots \cup F_o^k \cup F_r^k. \tag{4}$$

Similar to a RIM, we set $k = 3$.

With the concatenation of residual features from the last unit (unit 3) in the output of the module, DCRIM finds another path in the module to access early input features in addition to having different types of features.

3.4. PERI-Net

PERI-Net is an encoder-decoder network with skip connections, which uses the guidance of global information to better learn local information, in particular, in the segmentation of structures with vague appearance (see Figure 4). This network contains 2 DCRIMs, 5 RIMs and 3 output layers, denoted m_1, m_2, m_3. The output layers all aim to recover full resolution of segmentation masks by using different levels of abstraction. Because dense connections lead to slower training and increased use of memory, we use DCRIMs only in the early layers of encoding path and in the late layers of decoding path; therefore, we aim to improve localisation of detected structures by effectively reusing features learned in a module through dense connections and by evaluating the usefulness of the features while directly using them in the generation of segmentation maps with the increased level of abstraction in the output layers.

In contrast to U-net [17], where skip connections one-to-one link layers in encoding path to those in decoding path, the proposed network has extra skip connections, which may be also called dense connections due to the reuse of features between layers: one transfers the output of the first DCRIM in the encoding path to the input of the third output layer and the other one passes the output of the second DCRIM in the encoding path both to the input of the second output layer and to the input of last DCRIM in the decoding path. Because the three output layers have the same opportunity to reach features from the same early layers in the encoding path, they can approximate the same reference segmentation mask for an input image after appropriately increasing the resolution of input features. This also has the benefit of using the same loss function with the same ground truth image, without adapting the resolution of ground truth image depending on the resolution provided by an output layer.

Similar to U-net, we doubled the number of features when downsampling and halved it when upsampling. We realise downsampling with a 3×3 convolution with a stride of 2, which produced better performance than max pooling in preliminary experiments, and upsampling with bilinear interpolation; we did not use deconvolution because it increases parameter number unnecessarily. Through skip connections, we copy features from the encoding path to the decoding path, and we upsample feature resolution if it is necessary.

Throughout the proposed network, we use batch normalisation before each convolutional layer, followed by a ReLU activation function apart from the loss layers, which are of sigmoid activation function. We avoid the use of dropout in our network by considering the work of Li et al., where they showed that using both dropout and batch normalisation during training may lead a network to perform worse during test time [18].

We optimised the parameters of the proposed network with soft dice loss [19] with L_2 weight decay, with equation (5).

$$\mathcal{L} = \alpha L_D(M, m_1) + \beta L_D(M, m_2) + \beta L_D(M, m_3) + \lambda \left|W\right|_2^2 \tag{5}$$

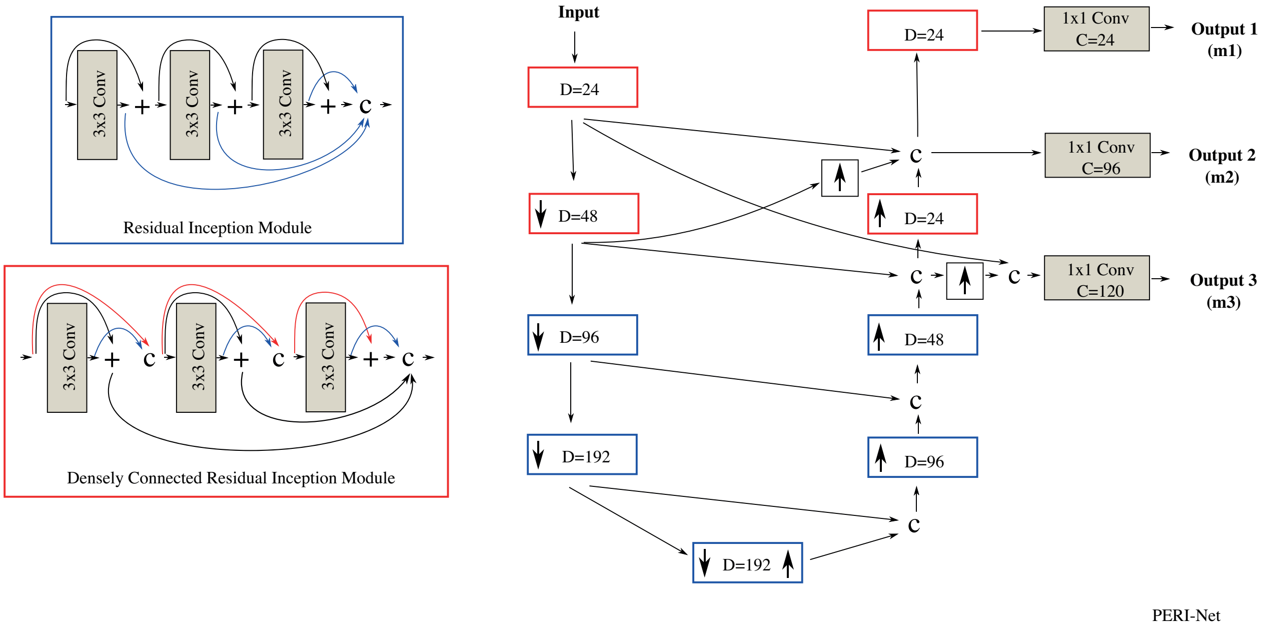


Figure 4. Overview of the proposed architecture, PERI-Net, and its novel components, residual inception module (RIM) and densely connected residual inception module (DCRIM). PERI-Net, in the right side, contains RIMs with blue frames and DCRIMs with red frames. Down and up arrows inside the RIMs and DCRIMs respectively symbolise downscaling and upscaling by a factor of 2, which are respectively realised with 3×3 filters with a stride of 2 and, bilinear interpolation. D and C in light brown boxes show the depth of a layer and input channel number consecutively. $+$ and C respectively denote addition and concatenation. The subfigures in the left side illustrate the architectures of a RIM and a DCRIM in more details.(Best viewed in color.)

where $L_D(\cdot)$ denote soft dice loss between a reference segmentation mask M and an output mask m generated by the proposed architecture. $\alpha = 0.6$ and $\beta = 0.2$ are weightings for corresponding output losses and $\lambda = 0.001$ is a weight decay rate for the weights W of the network.

4. Material and experimental setup

4.1. Material

We evaluated the segmentation performance of the proposed architecture on 3 fundus image datasets with different characteristics –the DRIVE [1] [20], the CHASE_DB1 [2] [21] and IOSTAR [3] [22], which are all available online – and a 2-photon microscopy image dataset. Table 3 tabulates some characteristics of the datasets.

Table 3. The properties of the datasets used in following experiments.

Name	Camera	FOV	Resolution	No. Of Images	Country
DRIVE	Canon CR5	45^o	768×584	40	Netherlands
CHASE_DB1	Nidek NM- 200-D	30^o	1280×960	28	The UK
Axon	2-photon imaging	—	512×512	152	The UK

[1] DRIVE dataset [online]. Website https://www.isi.uu.nl/Research/Databases/DRIVE/ [accessed 16 April 2020].
[2] CHASE_DB1 dataset [online]. Website https://blogs.kingston.ac.uk/retinal/chasedb1/ [accessed 16 April 2020].
[3] IOSTAR dataset [online]. Website https://www.idiap.ch/software/bob/docs/bob/bob.db.iostar/stable/ [accessed 16 April 2020].

DRIVE: This dataset consists of low resolution fundus images mainly collected from healthy people. In order to standardise the performance evaluation of supervised methods, the dataset was divided into 2 sets of the equal size. 3 images in training set and 4 images in test set show signs of diabetic retinopathy. The training set has manually traced vessel segmentation maps, while the test set has 2 sets of labelled vessel maps.

CHASE_DB1: The distinguishing characteristics of this dataset from DRIVE are that data have been captured from (i) both eyes of (ii) school children (iii) from a range of ethnicities. The main difficulties with this dataset are (i) the presence of the central light reflex, (ii) noneven background illumination and (iii) poor contrast of blood vessels. The dataset contains 2 sets of manually traced vessel maps, each labelled with a different expert. Because the dataset is not readily accompanied with FOV masks, we generated them by using the technique in [23].

IOSTAR: The dataset consists of 30 scanning laser ophthalmoscopy (SLO) images, whose sizes are 1024×1024 pixels. The images were taken by an EasyScan camera with a FOV of 45^o. The dataset is shared with its FOV masks and binary vessel masks.

Axon data: The dataset contains 152 gray scale $2D$, 2-photon microscopy images, whose sizes are 512×512 pixels. 20 images were used for performance evaluation of methods and the rest is for training the network. Manual labelling of the images was realised one of the authors. Gray scale images were obtained by using maximum projection over z direction of the image stack. The image dataset is significantly different to fundus images, since it is produced by maximum intensity projections through multiple slice (confocal) stacks. There are bright protrusions (synapses) on or very close to the axons, and there is varied background structure and imaging noise. Also, overlapping and dense axon regions are very common. The segmentation maps only includes the most visible axons due to the imprecise nature of manual labelling.

Despite availability of more than 1 set of reference segmentation mask for the fundus image datasets, the first set is accepted ground truth images with a general consensus [1]. We conformed to the division of training and test sets for the DRIVE dataset. The first 20 images of CHASE_DB1 were used for training and the rest of it was used for the performance evaluation of methods. Similarly, the first 20 images of the IOSTAR dataset were used for training and its last 10 images were allocated for performance evaluation.

4.2. Experimental setup

We randomly sampled $30,000$ patches from each image in a training set with a size of 96×96 pixels for the DRIVE, CHASE_DB1, IOSTAR and the axon datasets. In initial experiments, we observed that using larger image patches, among 48×48, 64×64 and 96×96 pixels, improved segmentation performance. This may be explained with that a larger image patch may provide more information regarding the connectivity of the structure to its neighbouring ones, which may eventually improve segmentation performance.

80% of the sampled patches was used for training and the rest was allocated in a validation set. Then, we trained the network with the 3 channels of RGB images for fundus image datasets and with gray scale images for the axon dataset, after applying channel-wise normalisation, where we only subtracted the mean of pixel values in each color channel calculated over image patches sampled from the training set. The same set of mean channel intensities was also subtracted from validation and test sets. In addition, we applied color jittering to image patches sampled from the training set, on the fly, to make the training less sensitive to various imaging conditions and different camera characteristics. Although this strategy increases the variety in contrast, the degree of apparent blurriness and saturation of image patches in the training set, it does not increase the training time.

We initialize network parameters with the method of He et al. [24] then optimize them with the Adam optimization algorithm [25] with default values, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Gradient updates during training were made after calculating errors over minibatches with a size of 32 patches for all datasets.

In order to accelerate training, we used warm-up training [26], where we gradually increased learning rate from 0 to 0.0008 over 4 epochs. Then, the initial learning rate of 0.0008 was exponentially reduced as $\alpha \cdot \gamma^n$, where α is the learning rate, γ is a learning rate decay parameter and n is epoch number. γ was set to 0.9 for all datasets. The network was trained for 100 epochs for the all datasets. After investigating training and validation loss curves, we used models trained for 60 epochs for the DRIVE dataset, 70 epochs for the CHASE_DB1 dataset, 80 epochs for the IOSTAR dataset and 60 epochs for the axon data.

The parameter count of the proposed network was 868,857 and a single size was used for the 3 datasets. We obtained initial segmentation masks for entire images after combining the outputs of the network with a stride of 30 pixels. Because the network generates three output maps for each input image, we use max projection to obtain a single output map, where we take the maximum of the output maps at each pixel location.

4.3. Evaluation criteria

Binary segmentation images can be generated by thresholding the probability maps produced by a network, at a certain level. We calculated a threshold for each probability map by using Otsu's method [27], which adaptively finds a threshold based on the minimisation of intraclass variance of thresholded foreground and background pixels.

In the binary images, we used the standard approach of denoting the proportion of correctly classified vessel pixels as true positives (TPs), missed vessel pixels as false negatives (FNs), erroneously labeled background pixels as false positives (FPs), and correctly labeled nonvessel pixels as true negatives (TNs). *Sensitivity (Sens)*, *specificity (Spec)*, *balanced accuracy (B. acc)* and *geometric mean (G_mean)* [28] are then calculated as follows:

$$Sens = \frac{|TP|}{|TP| + |FN|} \tag{6}$$

$$Spec = \frac{|TN|}{|TN| + |FP|} \tag{7}$$

$$B.acc = \frac{1}{2}(Sens + Spec) \tag{8}$$

$$G\_mean = \sqrt{Sens \cdot Spec} \tag{9}$$

where $|P|$ and $|\hat{P}|$ respectively refer to the number of positives in a reference segmentation mask and that of positives in its corresponding predicted one.

We computed the performance metrics per image and then present their average scores across the test set. Because the ratio of the number of pixels belonging to structures of interest to that of the other pixels in medical images is very small [2], we calculate balanced accuracy score instead of accuracy. Similarly, G-mean score is provided to better evaluate the performance of the imbalanced dataset. We also produce receiver operating characteristic (ROC) curve and the area under the ROC curve (AUC) for each image. For consistency with other work, we report the performance metrics by considering only pixels inside FOV masks for fundus images.

5. Results

5.1. Segmentation performance

We evaluated the performance of our network on DRIVE, CHASE_DB1 and IOSTAR and the microscopy axon data.

The evaluation of segmentation performance among output maps generated at various depths of the proposed network: Table 4 tabulates the performance of our network regarding output maps generated at different layers of the proposed network and the output map generated after applying max projection to these output maps. Although we provide the same local information for the 3 output maps by copying features generated at layers 1 and 2 to the output layers (layer 7, layer 8 and layer 9, which respectively generate output 1, output 2 and output 3), we observe a small but consistent increase in sensitivity across output maps from the first one to the third one while specificity score is maintained. This may be explained with better identification of structures with challenging appearance such as vessels with the central light reflection when more global information is combined with local information. We also observe that, according to the table, the best performance scores belong to the maximum projected output maps for all datasets. This result is expected because maximum projection seems to integrate the best of all output maps.

Table 4. Performance evaluation of segmentation maps generated at different depth of the proposed architecture

Dataset		Balanced accuracy	Sensitivity	Specificity	G-Mean
DRIVE	Output 1	0.91 ± 0.02	0.84 ± 0.05	0.98 ± 0.01	0.91 ± 0.02
	Output 2	0.91 ± 0.02	0.85 ± 0.04	0.98 ± 0.01	0.91 ± 0.02
	Output 3	0.92 ± 0.02	0.86 ± 0.04	0.98 ± 0.01	0.91 ± 0.02
	Max Projection	**0.93 ± 0.02**	**0.88 ± 0.04**	0.97 ± 0.01	**0.93 ± 0.02**
CHASE_ DB1	Output 1	0.91 ± 0.02	0.84 ± 0.04	0.98 ± 0.01	0.91 ± 0.02
	Output 2	0.91 ± 0.02	0.85 ± 0.04	0.98 ± 0.01	0.91 ± 0.02
	Output 3	**0.92 ± 0.02**	0.86 ± 0.03	0.97 ± 0.01	0.91 ± 0.02
	Max Projection	**0.92 ± 0.02**	**0.87 ± 0.03**	0.97 ± 0.00	**0.92 ± 0.02**
IOSTAR	Output 1	0.90 ± 0.02	0.82 ± 0.04	0.98 ± 0.01	0.89 ± 0.02
	Output 2	0.90 ± 0.02	0.82 ± 0.04	0.98 ± 0.01	0.90 ± 0.02
	Output 3	0.91 ± 0.02	0.84 ± 0.04	0.98 ± 0.01	**0.91 ± 0.02**
	Max Projection	**0.92 ± 0.02**	**0.86 ± 0.04**	0.98 ± 0.01	**0.91 ± 0.02**
Axon Data	Output 1	0.88 ± 0.03	0.78 ± 0.06	0.98 ± 0.01	0.87 ± 0.04
	Output 2	0.88 ± 0.03	0.79 ± 0.07	0.98 ± 0.01	0.88 ± 0.04
	Output 3	**0.90 ± 0.03**	0.82 ± 0.07	0.97 ± 0.02	0.89 ± 0.04
	Max Projection	**0.90 ± 0.03**	**0.83 ± 0.06**	0.97 ± 0.02	**0.90 ± 0.04**

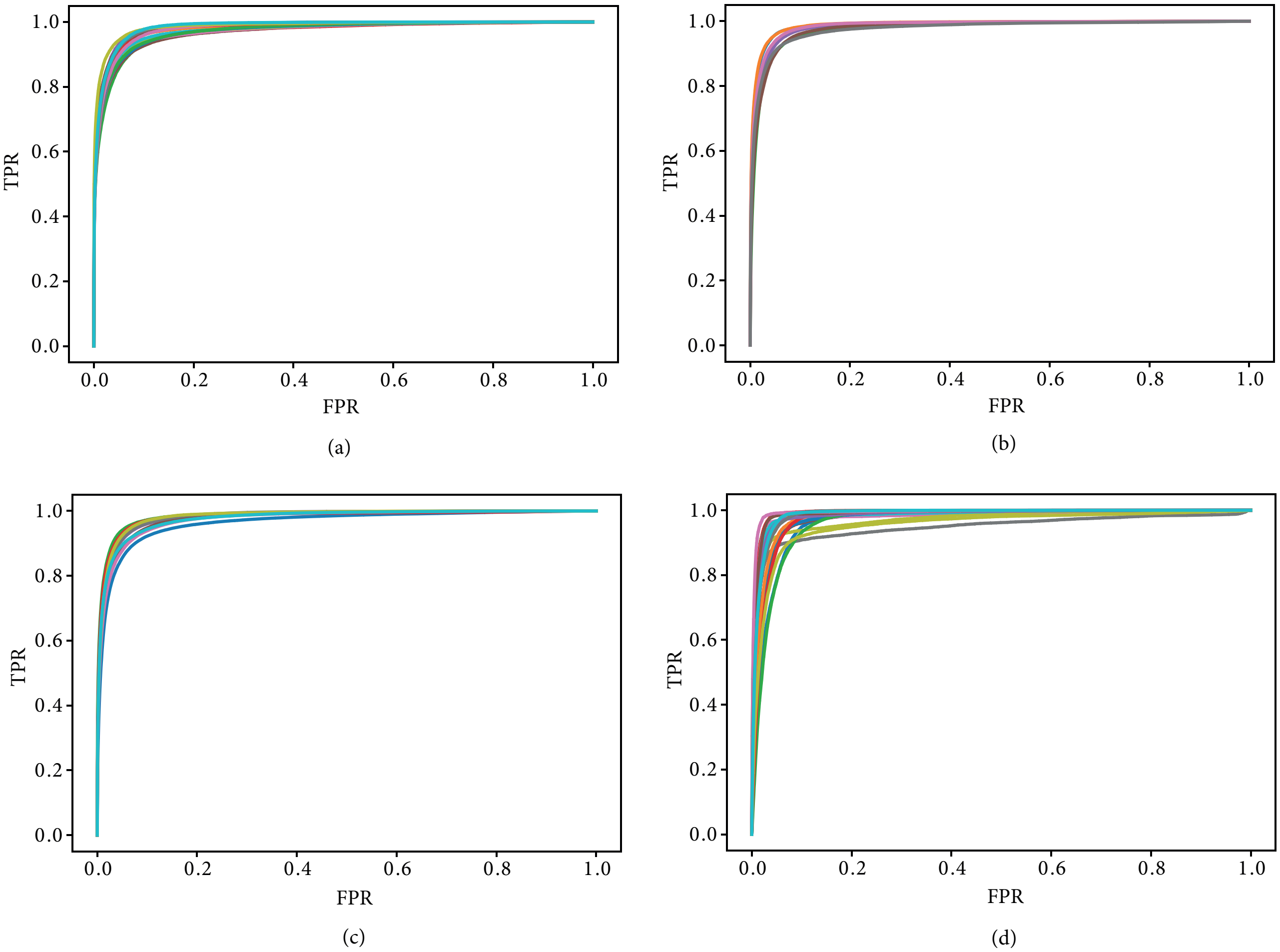
Comparison of segmentation performance of the proposed network with that of previous methods: Table 2 compares the performance of our method with that of previous methods regarding performance metrics of balanced accuracy, sensitivity, specificity, G-mean and AUC. In order to compare the effect of architectural difference on segmentation performance, we trained a U-net with similar parameter count (which is $1,020,283$) to our network and the same downsampling-upsampling techniques, with the same loss function and training strategies explained in Section 4.2. We also asses the effect of the proposed RIMs and DCRIMs on the segmentation performance of PERI-Net by replacing them with naive inception modules [5], as shown in Figure 1a. In the latter modules, we used 3 types of kernels; 3×3, 5×5, 7×7 and the latter 2 were realised with separable kernels.

According to Table 2, the proposed network outperforms previous methods with significant margins on sensitivity and G-mean for the 3 datasets, including the network of Liskowski and Krawiec, which is of approximately 48 million parameters, in contrast to our network with slightly over 0.8 million parameters. Although U-net has a slightly larger parameter count (approximately $200,000$ parameters) than our network, the superiority of the performance of our method to that of U-net on balanced accuracy, sensitivity and G-mean is striking. With the training strategies detailed in the paper, we also outperfom U-net trained with Alom et al. [29]. The performance of the modified version of PERI-Net follows that of PERI-Net for fundus image dataset; however, it produces better performance for the axon dataset. Moreover, we obtained better performance than a similar method of Uslu [30], where inception modules were stated to be the main inspiration for the proposed network. In contrast to our network, Uslu's network was of 3 paths with various kernel sizes, such as 1×1, 3×3 and 5×5. These results show the effectiveness of our architecture in the segmentation of fundus and axon images.

With respect to AUC, PERI-Net generates similar performance to U-net for the all datasets, which shows similar confidence of the 2 methods at vessel segmentation. However, it should be noted that the ratio of the number of vessel pixels to that of background pixels in fundus images is very small [1]. Figure 5 demonstrates ROCs for the 4 datasets.

Table 1 shows the best and worst segmentation performance obtained by our network and U-net, based on G-mean. It should be emphasised that the same order is valid for sensitivity score too, which shows that the performance improvement provided by our network is due to better recognition of structures of interest. According to the table, our architecture leads to a significant improvement on the detection of structures for the worst and the best segmented images for the 3 datasets, when compared with the performance of U-net on the same images. Figures 6–9 respectively demonstrate segmentation masks generated by our method for DRIVE, CHASE_DB1, IOSTAR and the axon dataset. DRIVE dataset contains some images carrying pathologies of diabetic retinopathy. As demonstrated in Figure 6, vessel segmentation is well performed despite the presence of white spots in the image.

Figure 7 shows an oversaturated image and a very dark one from CHASE_DB1, the latter one with prominent central light reflection and inhomogeneous image background. These characteristics of the image may significantly degrade the segmentation performance. However, the proposed method manages to correctly segment the majority of vessel trees of both images, which strongly relates to the use of maximum projection of the 3 output masks. In contrast to DRIVE and CHASE_DB1, IOSTAR dataset contains SLO images, where hue variation is larger than fundus camera images. According to Figure 8, the proposed method also perform well on this dataset, which only misses few small vessels. When compared with fundus images, contrast range in the axon dataset is larger due to the presence of very bright synapses and faint axons. As illustrated in Figure 9, the proposed method correctly identifies many axons and also detects some of those missed by the expert during manual tracing.



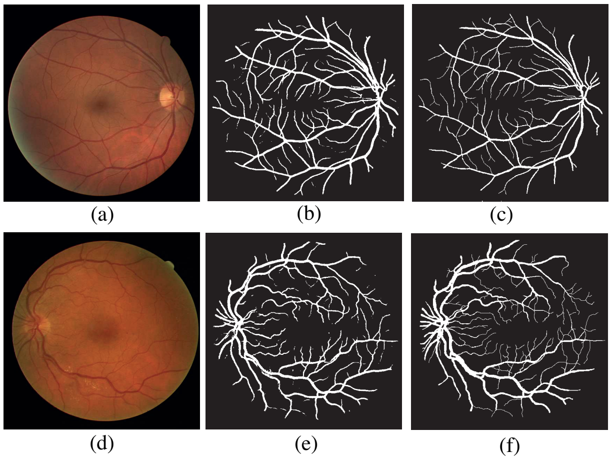
(a)

(b)

(c)

(d)

Figure 5. ROCs for each image in (a) DRIVE, (b) CHASE_ DB1, (c) IOSTAR, and (d) axon datasets.



(a) (b) (c)

(d) (e) (f)

Figure 6. Performance evaluation in DRIVE: The top row relates to the fundus image (19_*test*) with the maximum G-mean score in Table 1 and the bottom row is associated with the fundus image (3_*test*) with the minimum G-mean. Columns from left to right show the fundus images (a) and (d) and vessel maps generated by the proposed method (b) and (e) and ground truth vessel maps (c) and (f). (Best viewed in color).

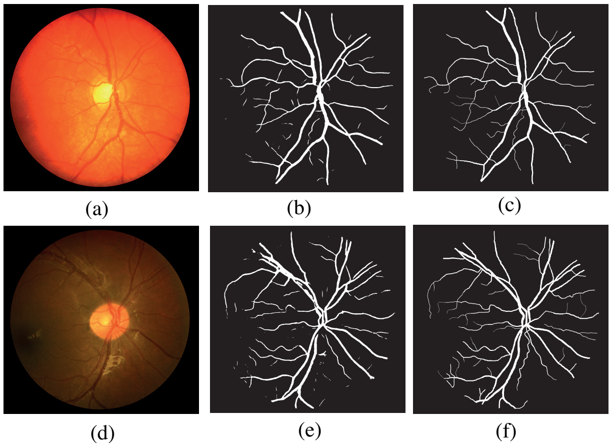


Figure 7. Performance evaluation on CHASE_ DB1: The top row relates to the fundus image ($Image\_11R$) with the maximum G-mean in Table 1 and the bottom row is associated with the fundus image ($Image\_13R$) with the minimum G-mean. Columns from left to right show the fundus images (a) and (d), vessel maps generated by the proposed method (b) and (e) and ground truth vessel maps (c) and (f). (Best viewed in color).

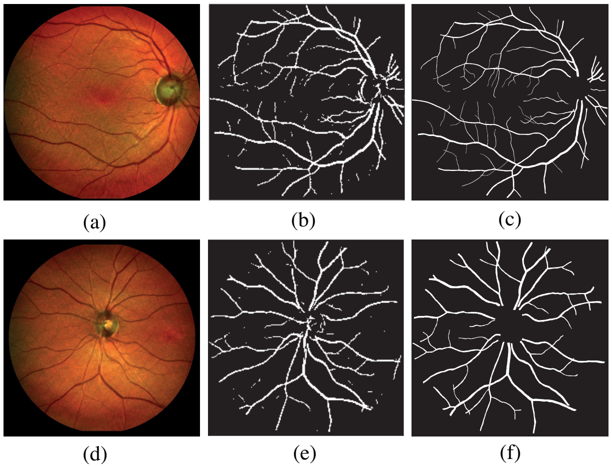


Figure 8. Performance evaluation on IOSTAR: The top row relates to the fundus image ($STAR\ 45\_ODC$) with the maximum G-mean in Table 1 and the bottom row is associated with the fundus image ($STAR\ 48\_OSN$) with the minimum G-mean. Columns from left to right show the fundus images (a) and (d), vessel maps generated by the proposed method (b) and (e) and ground truth vessel maps (c) and (f). (Best viewed in color).

6. Conclusions

This paper presents PERI-Net, a deep architecture for the segmentation of structures with vague appearance in biomedical images. With this architecture, we seek to address the scarcity of labelled data in medical image analysis by effectively processing input image with parameter efficient inception modules and its variants; RIM and DCRIM. We also introduce a way to compute a final segmentation map, where local and global information is combined in a balanced way, which is particularly found useful to resolve ambiguity on the detection of vessels with the central light reflection.

According to the results of our experiments, the proposed architecture outperform U-net, on balanced accuracy, sensitivity and G-mean for the 4 datasets (3 fundus and 1 axon image dataset). In experiments where we trained U-net with a similar parameter count and the same training strategies, we observed a general improvement with the proposed network on both the best and worst performing images over the performance of U-net. This was particularly true for the detection of vessels with the central light reflection.

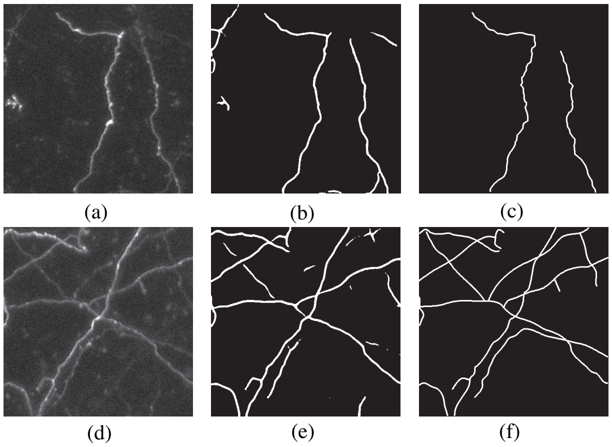


Figure 9. Performance evaluation on the axon data: The top row relates to the axon image ($image\_002$) with the maximum G-mean in Table 1 and the bottom row is associated with the axon image ($image\_018$) with the minimum G-mean. Columns from left to right show the axon images (a) and (d), axon maps generated by the proposed method (b) and (e) and, and ground truth axon maps (c) and (f).

Future work will concentrate on the segmentation of multiscale structures in $3D$ medical images, where a network with inherently small size may be a better choice for such problems demanding much larger parameters than its $2D$ counterparts.

Acknowledgment

Fatmatulzehra Uslu made analysis and drafted the paper. Cher Bass provided data for the analysis and revised the paper. Anil A. Bharath edited the paper.

References

[1] Fraz MM, Remagnino P, Hoppe A, Uyyanonvara B, Rudnicka AR et al. Blood vessel segmentation methodologies in retinal images-a survey. Computer Methods and Programs in Biomedicine 2012; 108 (1): 407-433.

[2] Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F et al. A survey on deep learning in medical image analysis. Medical Image Analysis 2017; 42: 60-88.

[3] Zaimi A, Wabartha M, Herman V, Antonsanti PL, Perone CS et al. AxonDeepSeg: automatic axon and myelin segmentation from microscopy data using convolutional neural networks. Scientific reports 2018; 8 (1): 1-11.

[4] Liskowski P, Krawiec K. Segmenting retinal blood vessels with deep neural networks. IEEE Transactions on Medical Imaging 2016; 35 (11): 2369-2380.

[5] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D et al. Going deeper with convolutions. In: The IEEE Conference on Computer Vision and Pattern Recognition; Boston, Massachusetts, USA; 2015. pp. 1-9.

[6] Chen L, Bentley P, Mori K, Misawa K, Fujiwara M et al. DRINet for medical image segmentation. IEEE Transactions on Medical Imaging 2018; 37 (11): 2453-2462.

[7] Chudzik P, Majumdar S, Caliva F, Al-Diri B, Hunter A. Exudate segmentation using fully convolutional neural networks and inception modules. In: International Society for Optics and Photonics Conference; Houston, Texas, United States; 2018. pp. 1057430.

[8] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition; Las Vegas, NV, USA; 2016. pp. 770-778.

[9] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: The IEEE Conference on Computer Vision and Pattern Recognition; Honolulu, Hawaii, USA; 2017. pp. 4700-4708.

[10] Dolz J, Desrosiers C, Ayed IB. IVD-Net: Intervertebral disc localization and segmentation in MRI with a multi-modal UNet. In: International Workshop and Challenge on Computational Methods and Clinical Applications for Spine Imaging; Granada, Spain; 2018. pp. 130-143.

[11] Bilinski P, Prisacariu V. Dense decoder shortcut connections for single-pass semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition; Salt Lake City, Utah, USA; 2018. pp. 6596-6605.

[12] Zhang J, Jin Y, Xu J, Xu X, Zhang Y. MDU-Net: Multi-scale Densely Connected U-Net for biomedical image segmentation. arXiv preprint 2018; arXiv:1812.00352.

[13] Bass C, Dai T, Billot B, Arulkumaran K, Creswell A et al. Image synthesis with a convolutional capsule generative adversarial network. In:Medical Imaging with Deep Learning; London, UK; 2019. pp. 39-62.

[14] Lin M, Chen Q, Yan S. Network in network. arXiv preprint 2013; arXiv:1312.4400.

[15] Szegedy C, Ioffe S, Vanhoucke V, Alemi A A. Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-first AAAI Conference on Artificial Intelligence; San Francisco, California, USA; 2017. pp. 4278-4284

[16] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: The IEEE Conference on Computer Vision and Pattern Recognition; Las Vegas, NV, USA; 2016. pp. 2818-2826.

[17] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-assisted Intervention; Munich, Germany; 2015. pp. 234-241.

[18] Li X, Chen S, Hu X, Yang J. Understanding the disharmony between dropout and batch normalization by variance shift. In: The IEEE Conference on Computer Vision and Pattern Recognition; Long Beach California, USA; 2019. pp. 2682-2690.

[19] Milletari F, Navab N, Ahmadi SA. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: The Fourth International Conference on 3D Vision; Stanford, CA, USA; 2016. pp. 565-571.

[20] Staal J, Abrámoff MD, Niemeijer M, Viergever MA, Van Ginneken B. Ridge-based vessel segmentation in color images of the retina. IEEE Transactions on Medical Imaging 2004; 23 (4): 501-509.

[21] Fraz MM, Remagnino P, Hoppe A, Uyyanonvara B, Rudnicka AR et al. An ensemble classification-based approach applied to retinal blood vessel segmentation. IEEE Transactions on Biomedical Engineering 2012; 59 (9): 2538-2548.

[22] Zhang J, Dashtbozorg B, Bekkers E, Pluim J P, Duits R et al. Robust retinal vessel segmentation via locally adaptive derivative frames in orientation scores. IEEE Transactions on Medical Imaging 2016; 35 (12): 2631-2644.

[23] Gadriye D, Khandale G, Nawkhare R. System for diagnosis of diabetic retinopathy using neural network. International Journal of Technical Research and Applications 2014; 4 (2): 76-80.

[24] He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: The IEEE International Conference on Computer Vision; Santiago, Chile; 2015. pp. 1026-1034.

[25] Kingma D P, Ba,J. Adam: a method for stochastic optimization. arXiv preprint 2014; arXiv:1412.6980.

[26] Goyal P, Dollár P, Girshick R, Noordhuis P, Wesolowski L et al. Accurate, large minibatch sgd: Training imagenet in 1 hour. arXiv preprint 2017; arXiv:1706.02677.

[27] Otsu N. A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics 1979; 9 (1): 62-66.

[28] Akosa, J. Predictive accuracy: a misleading performance measure for highly imbalanced data. In: The SAS Global Forum; Orlando, FL, USA; 2017; pp. 2-5.

[29] Alom MZ, Yakopcic C, Hasan M, Taha TM, Asari VK. Recurrent residual U-Net for medical image segmentation. Journal of Medical Imaging 2019; 6 (1): 014006.

[30] Uslu F. An inception inspired deep network to analyse fundus images. arXiv preprint 2019; arXiv:1911.08715.

[31] Orlando JI, Prokofyeva E, Blaschko MB. A discriminatively trained fully connected conditional random field model for blood vessel segmentation in fundus images. IEEE Transactions on Biomedical Engineering 2016; 64 (1): 16-27.

[32] Oliveira WS, Teixeira JV, Ren TI, Cavalcanti GD, Sijbers J. Unsupervised retinal vessel segmentation using combined filters. PloS One 2016; 11 (2).

[33] Li Q, Feng B, Xie L, Liang P, Zhang H, Wang T. A cross-modality learning approach for vessel segmentation in retinal images. IEEE Transactions on Medical Imaging 2015; 35 (1): 109-118.

[34] Wang S, Yin Y, Cao G, Wei B, Zheng Y et al. Hierarchical retinal blood vessel segmentation based on feature and ensemble learning. Neurocomputing 2015; 149: 708-717.

[35] Cheng E, Du L, Wu Y, Zhu Y J, Megalooikonomou V et al. Discriminative vessel segmentation in retinal images by fusing context-aware hybrid features. Machine Vision and Applications 2014; 25 (7): 1779-1792.

[36] Fraz MM, Basit A, Barman SA. Application of morphological bit planes in retinal blood vessel extraction. Journal of Digital Imaging 2013; 26 (2): 274-286.

[37] Na T, Zhao Y, Zhao Y, Liu Y. Superpixel-based line operator for retinal blood vessel segmentation. In: Annual Conference on Medical Image Understanding and Analysis; Quebec City, QC, Canada; 2017. pp. 15-26.