DENSE-INCEPTION U-NET FOR MEDICAL IMAGE SEGMENTATION

Ziang Zhang[a], Chengdong Wu[a, *], Sonya Coleman[b], Dermot Kerr[b]

[a] *Faculty of Robot Science and Engineering, Northeastern University, 110004, China*

[b] *School of Computing, Engineering and Intelligent Systems, Ulster University, BT52 1SA, United Kingdom*

** Corresponding author.*

E-mail addresses: zza13838978934@gmail.com (Ziang Zhang), wuchengdong@mail.neu.edu.cn (Chengdong Wu), sa.coleman@ulster.ac.uk (Sonya Coleman), d.kerr@ulster.ac.uk (Dermot Kerr)

Abstract

Background and objective: Convolutional neural networks (CNNs) play an important role in the field of medical image segmentation. Among many kinds of CNNs, the U-net architecture is one of the most famous fully convolutional network architectures for medical semantic segmentation tasks. Recent work shows that the U-net network can be substantially deeper thus resulting in improved performance on segmentation tasks. Though adding more layers directly into network is a popular way to make a network deeper, it may lead to gradient vanishing or redundant computation during training.
Methods: A novel CNN architecture is proposed that integrates the Inception-Res module and densely connecting convolutional module into the U-net architecture. The proposed network model consists of the following parts: firstly, the Inception-Res block is designed to increase the width of the network by replacing the standard convolutional layers; secondly, the Dense-Inception block is designed to extract features and make the network more deep without additional parameters; thirdly, the down-sampling block is adopted to reduce the size of feature maps to accelerate learning and the up-sampling block is used to resize the feature maps.
Results: The proposed model is tested on images of blood vessel segmentations from retina images, the lung segmentation of CT Data from the benchmark Kaggle datasets and the MRI scan brain tumor segmentation datasets from MICCAI BraTS 2017. The experimental results show that the proposed method can provide better performance on these two tasks compared with the state-of-the-art algorithms. The results reach an average Dice score of 0.9857 in the lung segmentation. For the blood vessel segmentation, the results reach an average Dice score of 0.9582. For the brain tumor segmentation, the results reach an average Dice score of 0.9867.
Conclusions: The experiments highlighted that combining the inception module with dense connections in the U-Net architecture is a promising approach for semantic medical image segmentation.

Keywords Deep Learning; Medical Image Segmentation; U-net; GoogLeNet; DenseNet

1. Introduction

Medical image segmentation has played an important role in the field of medical image analysis and attracted much attention from researchers in image processing [1]. Compared with the classical segmentation methods [2], algorithms based on Deep Learning have provided state-of-art performance and have become very popular [3]. During recent years, with the development of hardware and GPUs, several deep learning models have been proposed, such as AlexNet [4], VGG [5], DeepLab [6], GoogLeNet [7], Residual network [8] and DenseNet [9]. These network models have achieved good performance in the field of computer vision.

With respect to semantic medical image segmentation, the Fully Convolutional Networks (FCN) [10] provide superior performance when compared with other deep learning models. Among the various FCN architectures, the U-net [11] model is one of the most popular fully convolutional network models and it is widely used in the medical image processing field. The U-net model is a pixel-to-pixel, end-to-end fully convolutional network with skip layers between analysis path and synthesis path [11]. The U-Net needs less training resources and can reserve more important feature information. However, the standard U-Net architecture contains only a few layers and therefore it is not currently deep enough to gain

improved performance over other existing networks. To solve the problem, adding more layers directly to the network can enlarge the parameter space and make the network deeper, which may lead to gradient vanishing and redundant computation during training [12]. Gradient vanishing means if the network contains too many hidden layers, the learning rate will decrease with forward propagation, which may decrease the overall network learning. Therefore, the novel contribution of this paper is to make the U-Net deeper and wider without redundant computation and gradient vanishing to improve medical image segmentation.

In order to solve the aforementioned problems, two steps are considered: firstly, reduce redundant computation when making the network wider; secondly, avoid gradient vanishing while making the network deeper. With respect to reducing redundant computation, existing research indicates that sparse matrices can be clustered into dense sub-matrices to improve calculation and performance [13]. Network models like AlexNet or VGG demonstrate good experimental performance but they are computationally expensive. Compared with the models above, the GoogLeNet architecture proposed the concept of an "Inception module" to build a sparse, high performance computing network architecture [7]. The main advantage of this module is improving the utilization rate of computing resources by increasing the depth and width of network with the inception module while keeping the computational budget constant [14]. In the second step, recent research shows that the neural network does not have to be a progressive architecture and remains converging even though some layers are dropped randomly [15]. Inspired by this idea, a deep network model called DenseNet was proposed as a kind of convolutional neural network with dense connections. The DenseNet architecture has various advantages: parameter simplicity, vanishing-gradient reducing and feature reuse [16]. These characteristics make the networks a very good fit for semantic segmentation as they naturally induce skip connections and multi-scale supervision.

Inspired by the works introduced above, in order to develop a deep learning network appropriate for medical image segmentation tasks, we propose an architecture that combines the inception module with the densely connected convolutions based on the U-Net architecture named Dense-Inception U-net (DIU-Net). The details are as followed:

(1) Based on the architecture of U-Net, the analysis path and synthesis path are used in a network with skip connections to transmit feature maps directly from the down-sampling process to the up-sampling process.
(2) Based on GoogLeNet, in order to make the network wider without gradient vanishing, every convolutional layer is replaced by an Inception-Res module with different sizes of convolutional kernels, and residual connection is used in each module.
(3) In the middle of network, dense connecting is used to make the network deeper, and in these dense blocks standard convolutional layers are also replaced by the Inception-Res modules.
(4) All the convolutional blocks except the bottleneck layers are followed by a batch normalization layer [17] to avoid gradient vanishing.

The proposed architecture will be evaluated with respect to accuracy and practicality, on three clinical segmentation problems, namely lung segmentation in CT Data from the Kaggle datasets [18] , the blood vessel segmentations from retina images [19] and the MRI brain tumor segmentation from Multimodal Brain Tumor Segmentation Challenge 2017 [20].

In this paper, a novel module is proposed by combining the inception module with dense connection and

construct the network architecture based on the U-Net with the module. The rest of the paper is organized as follows: in Section 2, we detail the novel DIU-Net architecture and in Section 3 the performance evaluation is presented and compared with the standard U-Net, residual U-Net, FCN-8s [10] and SegNet [21]. Finally, Section 4 concludes the work.

2. Methods

Inspired by the U-Net model, GoogLeNet's Inception-Res module and the Dense-Net model, a novel fully connected network is proposed which integrates the inception module and the dense connections into the U-Net architecture. Fig. 1 demonstrates the architecture of the proposed Dense-Inception U-Net, denoted as DIU-Net. This network model contains the analysis path and the synthesis path, and these two paths mainly consist of four kinds of blocks, namely the Inception-Res block, the Dense-Inception block, the down-sample block and the up-sample block. The analysis path consists of Three Inception-Res blocks, one Dense-Inception block and four down-sample blocks. The synthesis path consists of three Inception-Res blocks, one Dense-Inception block and four up-sample blocks. In the middle of the network, a single Dense-Inception block is deployed and this block contains much more inception layers than the others.

Fig. 1 Overall architecture of the DIU-Net model

2.1 Inception-Res Block

A modified residual inception module is proposed to be used in both the analysis path and the synthesis path. The main purpose is to aggregate feature maps from different branches of kernels of different sizes, which can make the network wider and capable of learning more features [21]. Moreover, the residual connections make the learning easier since a residual inception block learns a function with reference to the input feature maps, instead of learning an unreferenced function [22]. Fig. 2 shows the proposed modified Inception-Res block. Different from the original Inception-Res architecture, each convolutional layer is followed by a batch normalization (BN) layer except for bottleneck layers. Batch normalization layer can avoid gradient vanishing while retaining convolutional layers.

Fig. 2 Overall architecture of the Inception-Res block

Here let's assume x_l is the output of the l^{th} layer. The $g_{n\times n}(\cdot)$ is a $n \times n$ kernel convolutional layer and $g_b(\cdot)$ denotes the BN layer. The function of concatenation is replaced by the sign of $\circ$ and the 1×1 size convolution kernel represents the bottleneck layer [23]. Thus, the output of the module from the analysis path is as follows:

$$x_{l+1} = g_{1\times1}(g_{1\times1}(x_l) \circ g_b(g_{3\times3}(g_{1\times1}(x_l))) \circ g_b(g_{3\times3}(g_b(g_{3\times3}(g_{1\times1}(x_l))))) + x_l \quad (1)$$

2.2 Dense-Inception Block

In the Dense-Inception block, the inception module proposed above is inserted into the dense connection block. By setting the padding as "Same" mode in the convolutional layer, the outputs of the inception

module will remain the same size of feature map as the inputs. Thus, the inception module can be regarded as a much wider convolutional layer since it is an aggregate of different kernel sizes with pooling layers [22]. The main purpose is to fit the inception module into the dense connection architecture and therefore make the network simultaneously deeper and wider without gradient vanishing or redundant computation. Fig. 3 illustrates an overview of the Dense-Inception block and Fig. 4 shows the detail of each Inception-Res block in the Dense-Inception block.

Fig. 3 Overall architecture of a 4-layer Dense-Inception block

Fig. 4 Overall architecture of each Inception-Res block in Dense-Inception block

The inception module with residual connection in the dense connection block is different from the standard residual inception module as the batch normalization layer is also used after each convolutional layer. The dense connection's main purpose is to make the network deeper by concatenating former convolution outputs but narrower with the small hyper-parameter of growth rate. Therefore, in order to fit the residual inception module in the dense connection block, the dense connection module is modified to improve the performance in the dense connection block. In this way, the Dense-Inception blocks are set in the middle of the network model, where the size of the feature map is small, and a large kernel can be replaced by two smaller kernels to reduce computational cost [24]. Formally, let's assume $g_{n\times n}(\cdot)$ denotes a $n\times n$ kernel convolutional layer, $g_b(\cdot)$ denotes the BN layer, and $p_{n\times n}(\cdot)$ denotes the Max-pooling layer. The sign $\circ$ represents concatenation and $f_{IR}(\cdot)$ represents the function of Bottleneck layer followed by the proposed Inception-Res module. Thus, the output of the proposed Inception-Res module is:

$$x_{l+1} = g_{1\times1}(g_{1\times1}(x_l)\circ g_b(g_{1\times1}(p_{3\times3}(x_l)))\circ g_b(g_{3\times1}(g_b(g_{1\times3}(g_{1\times1}(x_l))))))+x_l \quad (2)$$

The output of the $l+1^{th}$ layer in the Dense-Inception block is as follows:

$$x_{l+1} = f_{IR}([x_0, x_1, ..., x_l]) \quad (3)$$

Where $[x_0,x_1,...,x_l]$ refers to the concatenation of the feature maps produced in the layers 0, 1, …, l. Three Dense-Inception blocks are designed in total: one block is set in the analysis path, one is in the synthesis path, and the last one is set in the middle of network. Each Dense-Inception block except the middle one contains 12 proposed Inception-Res modules, and the middle one has 24 Inception-Res modules. The growth rate is used as the channel input of the residual inception module. Due to the concatenation connection, the size of the feature map will not get changed [25].

2.3 Down-sample & up-sample block

The down-sample block and up-sample block are illustrated in Fig. 5 (left) and Fig. 5 (right) respectively. These two blocks have the same architecture except for the convolution and Max-pooling layers in the down-sample block and the deconvolution and up-sampling layers in the up-sample block. They can be viewed as a simplified inception module with three branches. Compared with the standard U-Net architecture, Max-pooling and the Upsampling2D layer are used to reduce and enlarge the size of feature maps respectively, which may lead to feature loss and reduce the accuracy of results. Thus, the main purpose of the proposed down-sampling and up-sampling blocks is to overcome this problem and avoid

feature loss.

Fig. 5 The down-sample block (Left) and the up-sample block (Right)

Formally, let $g_{n}{}^{2}_{\times n}(\cdot)$ denote the convolution layer with 2 strides, $h_{n}{}^{2}_{\times n}(\cdot)$ denotes the convolution transposed layer with 2 strides, $p_{3}{}^{2}_{\times 3}(\cdot)$ denotes the max-pooling layer with 2 strides and $u^{2}(\cdot)$ represents the up-sampling layer with 2 strides. The expression for the down-sample block is:

$$x_{l+1} = g_{1\times1}(g^{2}_{3\times3}(g_{1\times1}(x_l)) \circ g^{2}_{3\times3}(g_{3\times3}(g_{1\times1}(x_l))) \circ p^{2}_{3\times3}(x_l))) \tag{4}$$

and the up-sample block is:

$$x_{l+1} = h_{1\times1}(h^{2}_{3\times3}(h_{1\times1}(x_l)) \circ h^{2}_{3\times3}(h_{3\times3}(h_{1\times1}(x_l))) \circ u^{2}(x_l))) \tag{5}$$

The Whole network model is designed followed by the concept of Encoding-Decoding architecture with skip connections. The encoding process corresponds the analysis path and the decoding process corresponds the synthesis path. In the encoding process, when the training images are transmitted into the model as the inputs, the channel number will increase by double after each Inception-Res block or Dense-Inception block; and the size of feature maps will reduce by half after down-sampling block. In the decoding process, the channel of feature maps will reduce by half after each Inception-Res block or Dense-Inception block; and the size of feature maps will increase by double after up-sampling block. And the training images will return to original as outputs size in final.

3. Performance Evaluation

In order to demonstrate the performance of the DIU-Net model, it is evaluated using three different medical image segmentation tasks. We use Keras programing language, based on TensorFlow to demonstrate the proposed model on a Core i7 7700 processor and 16 GB RAM with a NVIDIA TITAN XP GPU.

3.1 Datasets

The datasets contain three different segmentation tasks, including lung segmentation in CT datasets, blood vessel segmentation and MRI brain tumor segmentation task. The lung segmentation dataset is from the “Finding and Measuring Lungs in CT Data” competition in the Kaggle Data Science Bowl in 2017. It is a collection of CT images, manually segmented lung and measurements in 2/3D. In the experiments only 2D images will be used and measured. The blood vessel segmentation task contains three different datasets including the DRIVE [26], STARE [27] and CHASH_DB1 [28]. In order to enhance the generality of model, these three datasets are randomly merged to create a new dataset which will be used in the proposed model. The MRI brain tumor segmentation dataset was acquired on a 3T scanner at the UMC Utrecht and contain 88 subjects including patients with diabetes, dementia, Alzheimer and matched controls [20]. To fit the dataset into the network each subject was sliced into 2D images based on axial direction. Example image data from these three datasets are shown in Fig. 6. In this figure, the first row is from lungs dataset, the last row is from brain dataset, and the remaining rows are from the retina dataset. The left column shows the original images, the right column shows the ground

truth.

Fig. 6 The lung segmentation datasets, the retina blood vessel segmentation and the brain tumor segmentation datasets

3.2 Evaluation metrics

For quantitative performance evaluation, several evaluation measures will be used as follows: DICE coefficient (DC) [29], Jaccard similarity (JS) [30], accuracy (AC), sensitivity, specificity (SP) [31], F1-score [32], area under curve (AUC) [33] and the 95% confidence interval of AUC. In order to define these measures, we also use the variables True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN), Ground Truth (GT) and Segmentation Result (SR). The GT stands for the segmented region, and all the imaging datasets have been segmented manually by experienced experts, following the standard annotation protocol. These GT contours are references for further segmentation analysis [34]. The SR stands for the segmentation result from the method to be evaluated. The TP represents the pixels exist in both Ground Truth and proposed segmented region. The TN represents the pixels doesn't exist in neither Ground Truth nor segmented region. The FP represents the pixels exit in segmented region only. The FN represents the pixels exit in Ground Truth only. These expressions are shown in Table 1. The area under the curve (AUC) is the size of area under the Receiver Operating Characteristic (ROC); AUC intuitively reflects the classification ability of ROC curve expression. The 95% CI for AUC represents the 95% confidence interval of area under curve.

Table 1 Expressions of different evaluation metrics

3.3 Implementation details

In the proposed model, every convolution/deconvolution layer is followed by a batch normalization layer except the bottleneck layer or the last layer. The ReLU function [35] will be used as the activation function. In terms of the loss function, because of the binary labels in the segmentation tasks, a cross-entropy function will be used in this model. The Adam [36] method will be used as the optimization algorithm. During the experiment stage, various kinds of optimization algorithms were considered. The Adam optimization algorithm is a combination of algorithm Momentum and Adagrad. In training experiments, it had better performance compared with other algorithms. The parameters of the model will be initialized with the He_normal initializer [37] which can make the initial parameters of network much easier to trian compared with other initializers. In these experiments, the Adam algorithm will get employed and the initial learning rate will be set to 1×10^{-5}, $\mathrm{beta\_1} = 0.900$ and $\mathrm{beta\_2} = 0.999$. The network won't get converged if the learning rate is too large or too small. The batch size of training and validating datasets is 8. In each epoch 8 samples will be transposed into the network as the input, and too large batch size will slow down the training or validating speed. The model requires a total of 120 epochs to train with 300 steps per epoch.

In order to make a better comparison with the proposed model, another DIU-Net called DIU-Net-1 is also designed with only one Dense-Inception block in the middle of the network. To evaluate the proposed model, the results of the experiment will be compared with the SegNet, FCN-8s, U-Net and

ResU-Net [8].

3.4 Lung segmentation

The provided lung dataset consists of 267 images and the same number of labels. In this experiment, 90% of the data will be used for training and validating, and the remaining data will be used for testing. In order to get a reliable and stable model, 5-fold cross-validation will be used in this experiment. The size of each original grayscale image is 512×512 pixels and will be resized to 256×256 after image preprocessing. The experimental outputs from the models using 3 images from the lung segmentation dataset are shown in Fig. 7. The first column shows the original image, the second column shows the ground truth, and the remaining columns show the outputs as follows: SegNet, FCN-8s, U-Net, ResU-Net, DIU-Net-1 and DIU-Net. Fig. 8 shows the training accuracy of each model, and Fig. 9 shows the training loss curve of each model. Fig. 10 shows the corresponding validating accuracy and Fig. 11 shows the validating loss curve of each model. The testing results indicates that SegNet and FCN-8s have fuzzy pixels on the edge of segmentation position, the U-Net and ResU-Net show good performance because of the skip connection, but still some pixels are wrongly predicted. Due to the proposed model being deeper and capable of learning more features from datasets, the result shows better performance compared with the other models.

Fig. 7 Experimental outputs for lung dataset using different kinds of methods

Fig. 8 Training accuracy of the different kinds of models

Fig. 9 Training loss of different kinds of models

Fig. 10 Validating accuracy of the different kinds of models

Fig. 11 Validating loss of different kinds of models

Table 2 summarizes the segmentation performance. From Table 2 it shows that DIU-Net network shows better performance under each evaluation index. DICE coefficient is the most direct evaluation index of segmentation accuracy. In this experiment, the average DICE coefficient of DIU-Net is 0.9857, and it is 0.233% higher than the second segmentation performance from ResU-Net's results; And in terms of Jaccard similarity, AC, SE, SP, F1-score and AUC coefficient, the results from DIU-Net are all higher than second segmentation performance as follows: 0.9824%, 0.24%, 1.87%, 0.01%, 0.60% and 0.84%. Thus, with the help of dense connections and inception layers, DIU-Net can be deeper and more effective than the original networks. In order to make the experiment more convincing, the experiments of SegNet, FCN-8s, U-Net and ResU-Net used the same initial method, optimization algorithm, loss function and other initial parameters with DIU-Net. The DIU-Net with three Dense-Inception modules has better segmentation performance than with just one Dense-Inception block. The results show that the module can effectively deepen the depth of the network without introducing gradient disappearance, so that the network can learn more image features.

Table 2 Experimental results of proposed approaches for lung segmentation and comparison against other networks (The bold font is the best value for each column)

3.5 Blood vessel segmentation

The original images to be used for blood vessel segmentation are RGB images, and they can provide the clearest features of blood vessel under the green channel. Thus, preprocessing via normalization is necessary before conducting the segmentation experiment. The whole dataset contains only 136 samples of RGB images including labels. Among the samples, 85% of the data will be used for training and validating, and the remaining 15% data will be used for testing. In order to get a reliable and stable model, 5-fold cross-validation will be used in this experiment. The size of each original RGB image is 565×584 pixels and each image will be resized to 256×256 after image preprocessing. The training accuracy and loss curves for the dataset are shown in Figs. 12 and 13 respectively. And the validating accuracy and loss curves for the dataset are shown in Figs. 14 and 15. The experimental outputs of DIU-Net using the blood vessel dataset are shown in Fig. 16. The first row shows an example from the STARE dataset, the second row uses the CHASE_DB1 dataset and the last row uses the DRIVE dataset, all of these datasets are chosen from the testing set experimental outputs. The first column shows the input images, second column shows the ground truth, and the remaining rows show the segmentation results in the following order: SegNet, FCN-8s, U-Net, ResU-Net, DIU-Net-1 and DIU-Net. From the training curve, in Figure 12, it can be seen that although the DIU-Net requires more epochs to improve the accuracy compared with the other networks, the training and validating accuracy lines show that the DIU-Net gains overall better performance with a larger number of epochs. Because this retina dataset is much more complicated than the lung segmentation dataset, the result of SegNet is fuzzy and therefore we cannot view the result clearly, and the U-Net, FCN-8s and ResU-Net models do not segment all of the blood vessels as illustrated in Figure 16. The DIU-Net model has a much better performance due to the deeper network architecture and anti-gradient vanishing. A deeper network can learn more features and the performance is acceptable.

Fig. 12 Training accuracy of different kinds of models

Fig. 13 Training loss of different kinds of models

Fig. 14 Validating accuracy of different kinds of models

Fig. 15 Validating loss of different kinds of models

Fig. 16 Experimental outputs for retina blood vessel segmentation using different kinds of methods

From the training curve we can determine that the DIU-Net with three dense blocks and DIU-Net-1 with one dense block show better performance than all other models. In addition, the validation accuracy in Fig. 14 demonstrates that the DIU-Net and DIU-Net-1 show better validation accuracy compared with the ResU-Net, U-Net and SegNet. Both of DIU-Nets need more epochs to train but the trained models obtain better results than the other approaches.

In terms of quantitative analysis, Table 3 shows the detail of the evaluation results. As the blood vessel data contains three different kinds of datasets including the DRIVE, STARE and CHASH_DB1, the results of traditional methods are calculated by averaging. From Table 3 it can be concluded that the DIU-Net model shows better performance in terms of the DICE coefficient, Jaccard similarity, accuracy, F1-score and the sensitivity. In particular, the DICE coefficient is 0.15% higher than the results of ResU-Net. In terms of Jaccard similarity, AC, SE, SP, F1-score and AUC coefficient, the results from DIU-Net are all higher than second segmentation performance as follows: 0.31%, 0.15%, 2.84%, 0.09%, 2.11% and 0.60%. Compared with the traditional methods [33, 35], the DIU-Net also achieves better performance. Therefore, the results demonstrate the effectiveness of the proposed method for medical image segmentation tasks. In particular, the blood vessel segmentation task is more complicated than the lung segmentation task, and the DIU-Net with deeper architecture performs better in this task than other networks. The Dense-Inception blocks make the network much deeper and wider in order to gain increased performance.

Table 3 Experimental results of proposed approaches for retina blood vessel and comparison against other networks or traditional methods (The bold font is the best value for each column)

3.6 Brain tumor segmentation

The provided MRI scans brain tumor segmentation dataset consists of 88 subjects, and each subject was sliced into 2D images based on axial direction. After deleting the data without label pixels there are 4937 images and the same number of labels in total. In this experiment, 90% of the data will be used for training and validating, and the remaining data will be used for testing. In order to get a reliable and stable model, 5-fold cross-validation will be used in this experiment. The first column shows the original image, the second column shows the ground truth, and the remaining columns show the outputs as follows: SegNet, FCN-8s, U-Net, ResU-Net, DIU-Net-1 and DIU-Net. Fig. 17 shows the training accuracy of each of these models, and Fig. 18 shows the corresponding training loss curves. Fig. 19 shows the validating accuracy curve of each model and Fig. 20 shows the validating loss curve of each model. The experimental outputs from the models using 3 images from the brain tumor segmentation dataset are shown in Fig. 21.

Fig. 17 Training accuracy of different kinds of models

Fig. 18 Training loss of different kinds of models

Fig. 19 Validating accuracy of different kinds of models

Fig. 20 Validating loss of different kinds of models

Fig. 21 Experimental outputs for retina blood vessel segmentation using different kinds of methods

From the training curve we can determine that all the networks can get the level of accuracy around 0.9800, and the training loss can reduce to about 0.0100. In addition, the validation accuracy in Fig. 19

demonstrates that all the networks can get the level of accuracy around 0.9700. In Fig. 20, the validating loss curve shows the validating loss can reduce under 0.1000. Testing results from Fig. 21 shows that the results of DIU-Net have better segmentation performance than others.

In terms of quantitative analysis, Table 4 summarizes the segmentation performance. From Table 2 it shows that DIU-Net network shows better performance under each evaluation index. In this experiment, the average DICE coefficient of DIU-Net is 0.9892, and the average Jaccard similarity, AC, SE, SP and F1-score of DIU-Net are as follows 0.9940, 0.9970, 0.9284, 0.9986 and 0.9085. In terms of the AUC coefficient and its 95% confidence interval, the average AUC of DIU-Net is 0.9867 and the 95% CI is (0.9847-0.9889), which is 2.57% higher than second segmentation performance. These results verify the feasibility and effectiveness of DIU-Net.

Table 4 Experimental results of proposed approaches for brain tumor segmentation and comparison against other networks or traditional methods (The bold font is the best value for each column)

3.6 Algorithmic Run-time

The overall training and testing time of all the models, in each experiment, are shown in Table 5. From Table 5 it can be seen that the proposed model needs more time to train and test than other networks, and the DIU-Net network with three Dense-Inception blocks needs more time to train and test than the DIU-Net-1 network with only one Dense-Inception block, as this block does make the network much deeper and wider. The use of the trained DIU-Net model is not significantly slower than others and the training time is one of the computational costs. Though the training speed of the DIU-Net is slower than other networks, considering the improved performance, its running time is acceptable.

Table 5 Training/Testing time on various kinds of models

4. Conclusions

A novel FCN architecture is proposed as an extension of the U-Net using Dense-Net and GoogLeNet and the proposed network is called “DIU-Net”. There are four key features in the DIU-Net, namely Inception-Res block, Dense-Inception block, down-sampling and up-sampling block. With the function of Inception-Res block the network can be significantly wider, and the residual connection can make the network easier to train. In the middle part of the network, dense connection in Dense-Inception block makes the network deeper while avoiding the possible gradient disappearance. Meanwhile, densely connection can optimize the network results and reduce the computational load of training. In the whole network model, the improved lower/upper sampling module is used to replace the pooling layer/upper pooling layer in the original U-Net model, which reduces the network change feature ruler. The loss of feature information or the introduction of noise caused by inch time also deepens the depth of the network, so that the network can learn more image features.

The model is evaluated using three different medical image segmentation tasks: lung segmentation, blood vessel segmentation and brain tumor segmentation. The experimental results demonstrate that the proposed DIU-Net model shows better performance compared with existing methods including the

SegNet, FCN-8s, U-Net and ResU-Net models. From the comparison between DIU-Net and DIU-Net-1, the Dense-Inception block does make the network deeper and wider and hence improves the gradient propagation. However, too many Dense-Inception blocks may significantly increase the computational burden and does not provide better results. In this experiment just three blocks are needed for the proposed network model. In a word, it can be concluded that the proposed DIU-Net is a promising approach for semantic medical image segmentation.

The limitation of the proposed DIU-Net model is that in the Dense-Inception block increasing the growth rate may lead to too many parameters, making the model more difficult and slower to train. Another issue is that the test datasets did not include ultrasound images, which contain more speckle noise. Therefore, in the future we will aim to simplify the network structure whilst maintaining its accuracy and effectiveness. And we will also test the proposed model on ultrasound datasets and modify it to make the model more effective on various kinds of medical problems.

Conflict of interest

None.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. U1713216 and the Shenyang Intelligent Robot Laboratory Fund of China under Grant 18-007-0-06.

References

[1] Srinivasan A, Sundaram S (2013) Applications of deformable models for in-dopth analysis and feature extraction from medical images—a review. Pattern Recognition & Image Analysis, 23(2), 296-318. https://doi.org/10.1134/S1054661813020132.

[2] Litjens G, Kooi T, Bejnordi B E, Setio A, Ciompi F, Ghafoorian M et al (2017) A survey on deep learning in medical image analysis. Medical Image Analysis, 42, 60-88. https://doi.org/10.1016/j.media.2017.07.005.

[3] Hongzi Z, Yanyong Z, Mo L, Ashwin A, Kaoru O (2018) Exploring deep learning for efficient and reliable mobile sensing. IEEE Network, 32(4), 6-7. https://doi.org/10.1109/MNET.2018.8425293.

[4] Krizhevsky A, Sutskever I, Geoffrey E Hinton (2017) ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6). https://doi.org/10.1145/3065386.

[5] Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. Computer Science 2014. http://arxiv.org/abs/1409.1556.

[6] Chen L C, Papandreou G, Kokkinos I, Murphy K, Yuille A L (2016) Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis & Machine Intelligence, 40(4), 834-848. https://doi.org/10.1109/TPAMI.2017.2699184.

[7] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al (2014) Going deeper with convolutions. CVPR 2015, 1-9. https://doi.org/10.1109/CVPR.2015.7298594.

[8] He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. CVPR 2016,

770-778. https://doi.org/10.1109/CVPR.2016.90.

[9] Huang G, Liu Z, Weinberger K Q, Laurens V D M (2016) Densely connected convolutional networks. CVPR 2017, 4700-4708. https://doi.org/10.1109/CVPR.2017.243.

[10] Long J, Shelhamer E, Darrell T (2014) Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis & Machine Intelligence, 39(4), 640-651. https://doi.org/10.1109/TPAMI.2016.2572683.

[11] Ronneberger O, Fischer P, Brox T (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. MICCAI 2015, 234-241. http://arxiv.org/abs/1706.01307.

[12] Bi L, Kim J, Kumar A, Fulham M, Feng D (2017) Stacked fully convolutional networks with multi-channel learning: application to medical image segmentation. The Visual Computer, 33(6-8), 1061-1071. https://doi.org/10.1007/s00371-017-1379-4.

[13] Graham B, Laurens V D M (2017) Submanifold sparse convolutional networks. Neural and Evolutionary Computing 2017. https://arxiv.org/abs/1706.01307.

[14] Chen L, Bentley P, Mori K, Misawa K, Rueckert D (2018) Drinet for medical image segmentation. IEEE Transactions on Medical Imaging, PP (99), 1-1. https://doi.org/10.1109/TMI.2018.2835303.

[15] Huang G, Sun Y, Liu Z, Sedra D, Weinberger K (2016) Deep networks with stochastic depth. Computer Vision - ECCV 2016, 646-661. https://doi.org/10.1007/978-3-319-46493-0_39.

[16] Pleiss G, Chen D, Huang G, Li T, Laurens V D M, Weinberger K Q (2017) Memory-efficient implementation of densenets. Computer Vision and Pattern Recognition 2017. http://arxiv.org/abs/1707.06990.

[17] Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. Machine Learning 2015. http://arxiv.org/abs/1502.03167.

[18] https://www.kaggle.com/kmader/finding-lungs-in-ct-data/data.

[19] Staal J, Abramoff M D, Niemeijer M, Viergever M A, Van Ginneken B (2004) Ridge-based vessel segmentation in color images of the retina. IEEE Transactions on Medical Imaging, 23(4), 501-509. https://doi.org/10.1109/tmi.2004.825627.

[20] Menze B H, Jakab A, Bauer S, et al. (2015) The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). IEEE Transactions on Medical Imaging, 34(10), 1993-2024 https://doi.org/10.1109/TMI.2014.2377694.

[21] Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: a deep convolutional encoder-decoder architecture for scene segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1-1. https://doi.org/10.1109/TPAMI.2016.2644615.

[22] Szegedy C, Ioffe S, Vanhoucke V, Alemi A (2016) Inception-v4, inception-resnet and the impact of residual connections on learning. CVPR 2016. http://arxiv.org/abs/1602.07261.

[23] Alom M Z, Hasan M, Yakopcic C, Taha T M, Asari V K (2018) Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. CVPR 2018. http://arxiv.org/abs/1802.06955.

[24] Jégou, Simon, Drozdzal M, Vazquez D, Romero A, Bengio Y (2016) The one hundred layers tiramisu: fully convolutional densenets for semantic segmentation. CVPR 2016. http://arxiv.org/abs/1611.09326.

[25] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. CVPR 2016, 2818-2826. https://doi.org/10.1109/CVPR.2016.308.

[26] Niemeiji M, Staal J, Ginneken B V (2004) Comparative study of retinal vessel segmentati on methods on a new publicly available database. SPIE medical imaging, 5730, 648-656.

https://doi.org/10.1117/12.535349.

[27] Hoover A D, Kouznetsova V, Goldbaum M (2000) Locating blood vessels in retinal imag es by piecewise threshold probing of a matched filter response. IEEE Transactions on Me dical imaging, 19(3), 203-210. https://doi.org/10.1109/42.845178.

[28] Owen C G, Rudnicka A R, Mullen R. (2009) Measuring retinal vessel tortuosity in 10-ye ar-old children: validation of the Computer-Assisted Image Analysis of the Retina (CAIAR) program. Investigative ophthalmology & visual science, 50(5), 2004-2010. https://doi.org/10. 1167/iovs.08-3018.

[29] Dice L R (1945) Measures of the amount of ecologic association between species. Journal of Ecology, 26. https://doi.org/10.2307/1932409.

[30] Jaccard P (2010) The distribution of flora in the alpine zone. New Phytologist, 11(2), 37-50. https://doi.org/10.1111/j.1469-8137.1912.tb05611.x.

[31] Xu B, Wang N, Chen T, Li M (2015) Empirical evaluation of rectified activations in convolutional network. Computer Science. Machine Learning 2015. http://arxiv.org/abs/1505.00853.

[32] Huang H, Xu H, Wang X, Silamu W (2015) Maximum f1-score discriminative training criterion for automatic mispronunciation detection. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 23(4), 787-797. https://doi.org/10.1109/taslp.2015.2409733.

[33] Chang H, Zhuang A H, Valentino D J, Chu W C (2009) Performance measure characterization for evaluating neuroimage segmentation algorithms. NeuroImage, 47(1), 122-135. https://doi.org/10.1016/j.neuroimage.2009.03.068.

[34] Ail A A, Ahmad B R, Ali M, Afshin M, Reza R, Jamileh A, Amir H J, Mohammad B S. (2018) A Hybrid Multilayer Filtering Approach for Thyroid Nodule Segmentation on Ultrasound Images. Journal of Ultrasound in Medicine, 00, 1-12. https://doi.org/10.1002/jum.14731.

[35] Jarrett K, Kavukcuoglu K, Ranzato M A, LeCun Y (2009) What is the best multi-stage architecture for object recognition? Computer Vision 2009, IEEE 12th International Conference on 2009. https://doi.org/10.1109/ICCV.2009.5459469.

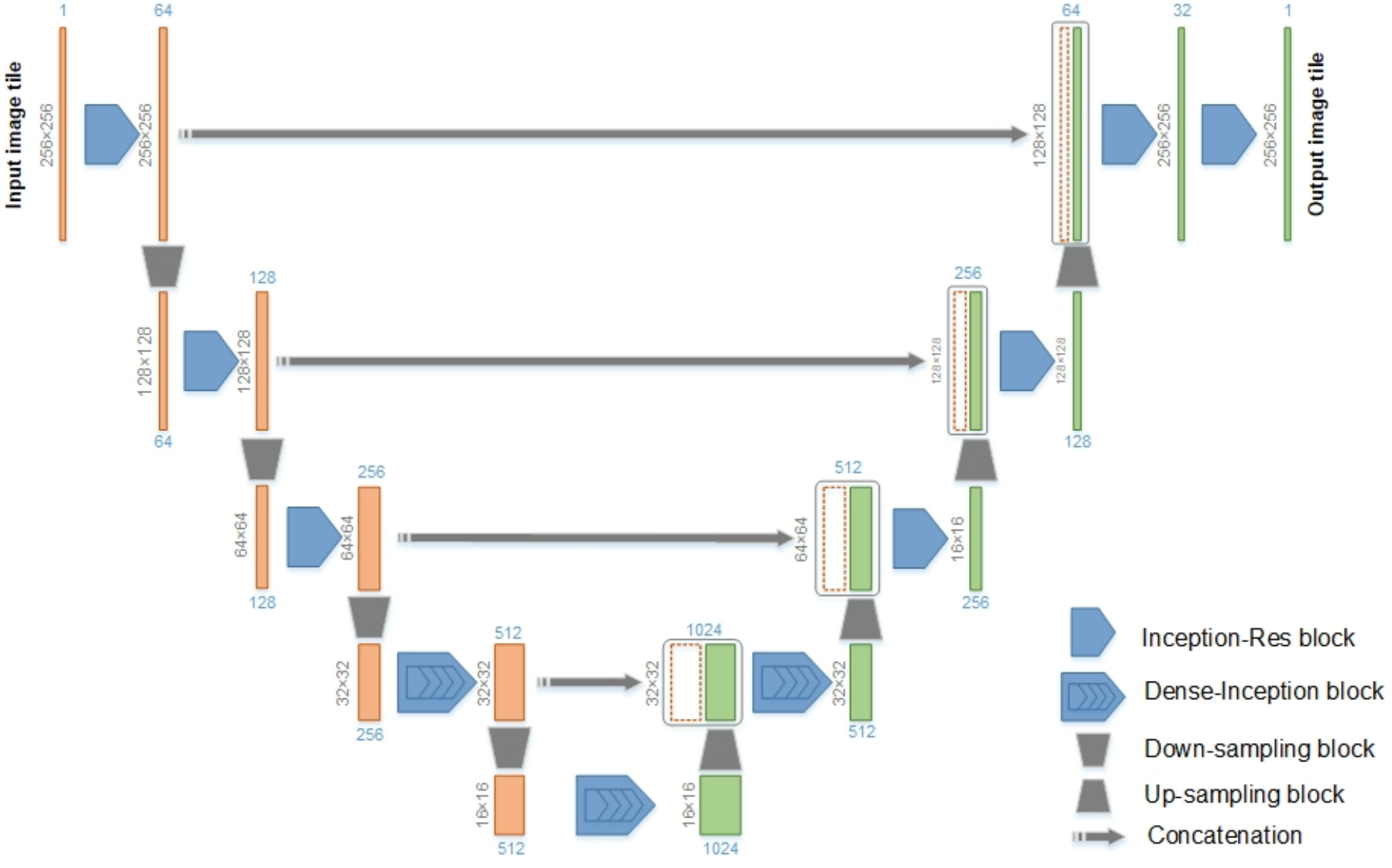
[36] Kingma D P, Ba J (2014) Adam: a method for stochastic optimization. Machine Learning 2014. http://arxiv.org/abs/1412.6980.

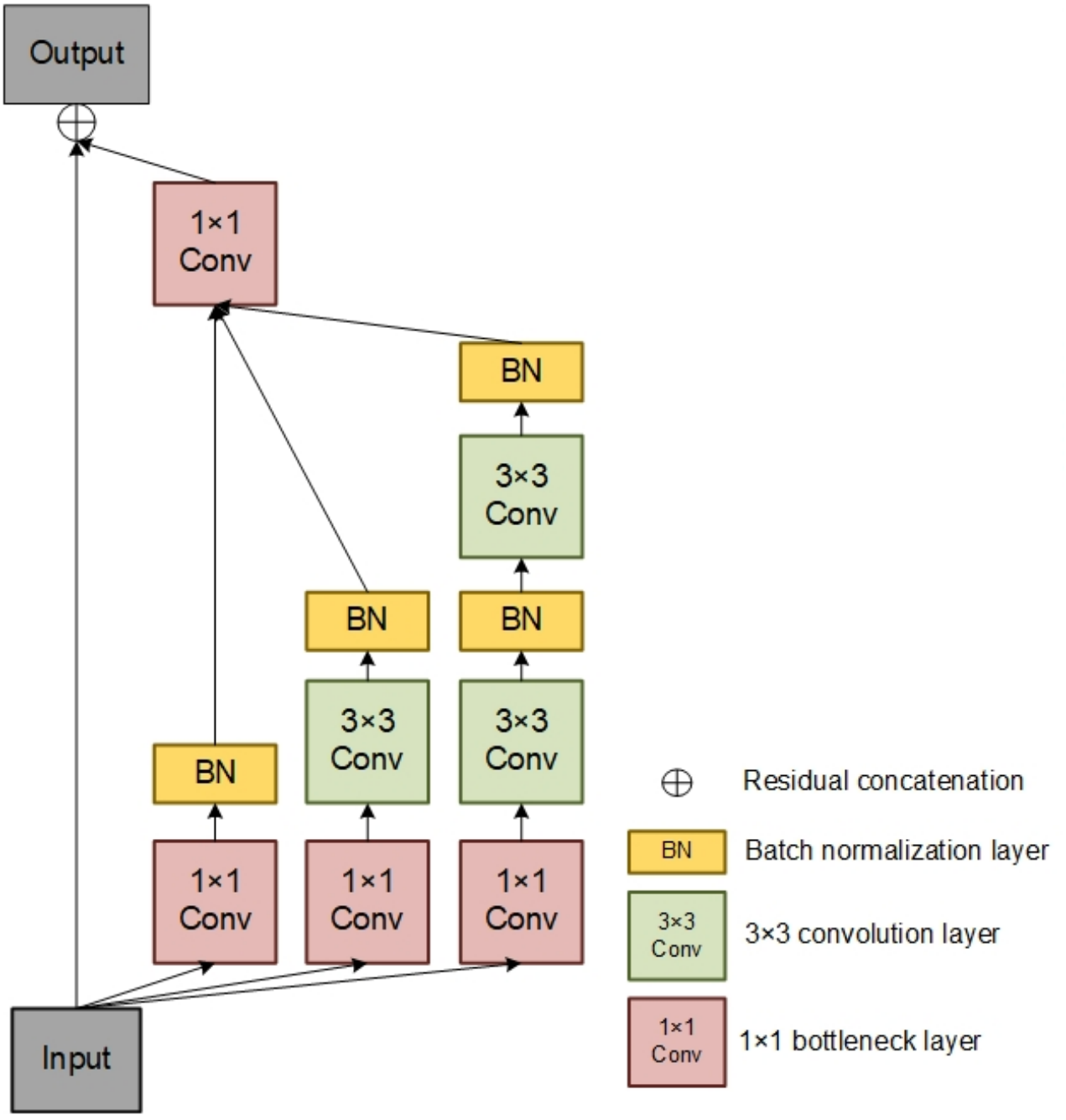
[37] He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: surpassing human-level performance on imagenet classification. IEEE International Conference on Computer Vision (ICCV), 2380-7504. https://doi.org/10.1109/ICCV.2015.123.

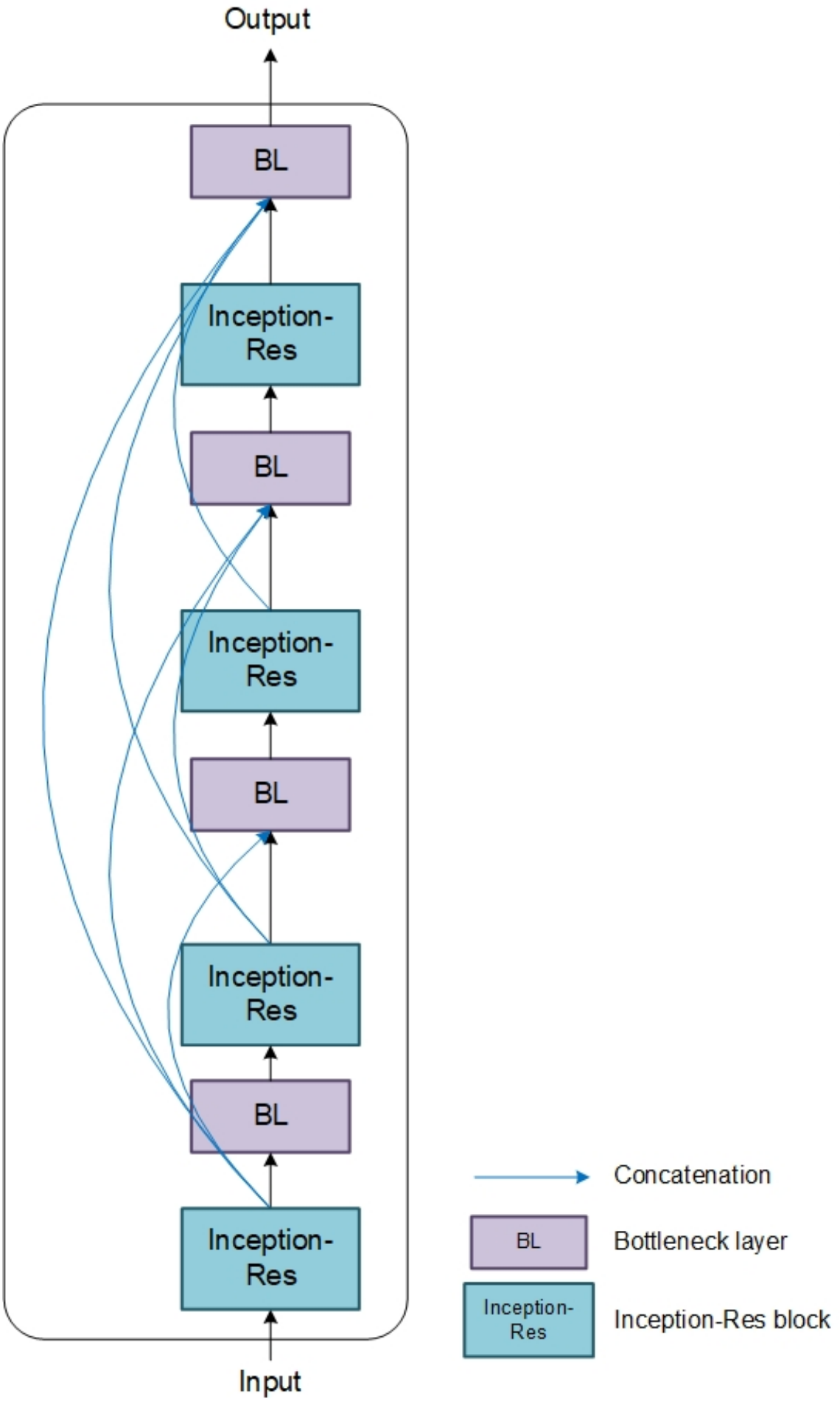
[38] Zhao Y, Rada L, Chen K, Harding S P, Zheng Y (2015) Automated vessel segmentation using infinite perimeter active contour model with hybrid region information with application to retinal images. IEEE Transactions on Medical Imaging, 34(9), 1797-1807. https://doi.org/10.1109/TMI.2015.2409024.

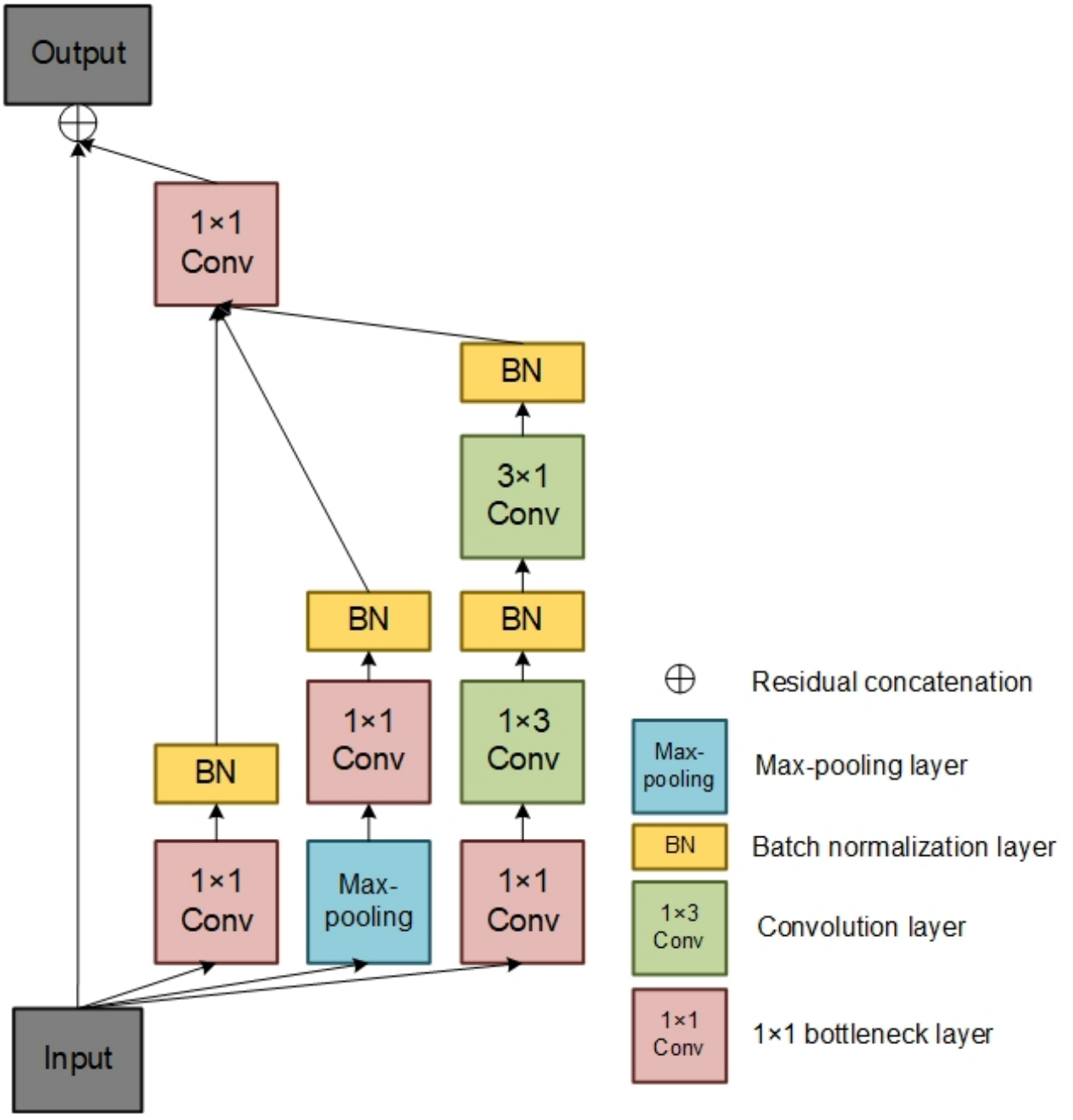
[39] Fraz M, Remagnino P, Hoppe A, Uyyanonvara B, Rudnicka A R, Christopher G, O, Barman S A (2012) An ensemble classification-based approach applied to retinal blood vessel segmentation. IEEE transactions on bio-medical engineering 2012 59(9), 2538-48. https://doi.org/10.1109/TBME.2012.2205687.

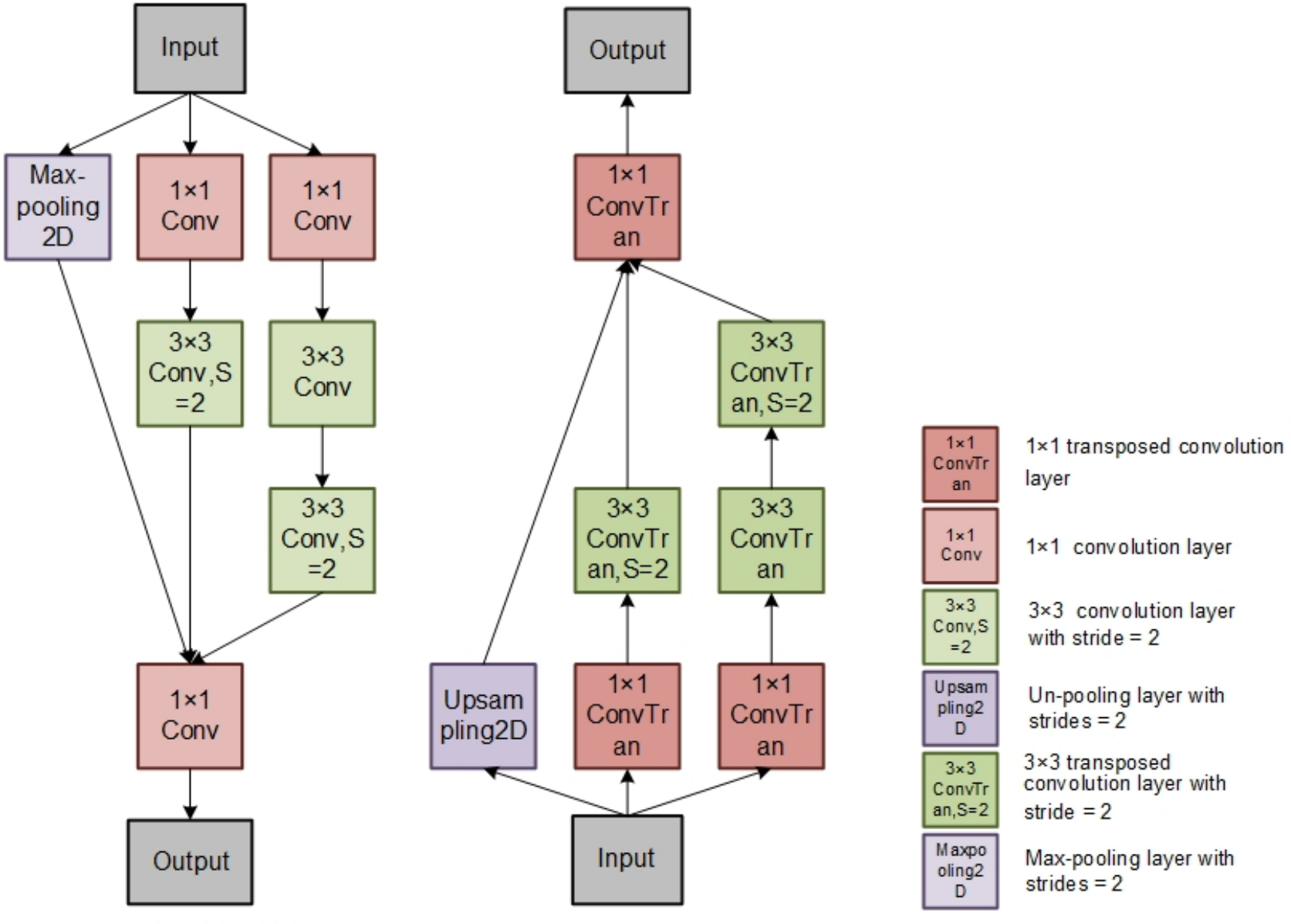
[40] Li Q, Feng B, Xie L, Liang P, Zhang H, Wang T (2016) A Cross-Modality Learning Approach for Vessel Segmentation in Retinal Images. IEEE Trans. Med. Imaging 35 (1): 109-118. https://doi.org/10.1109/TMI.2015.2457891.

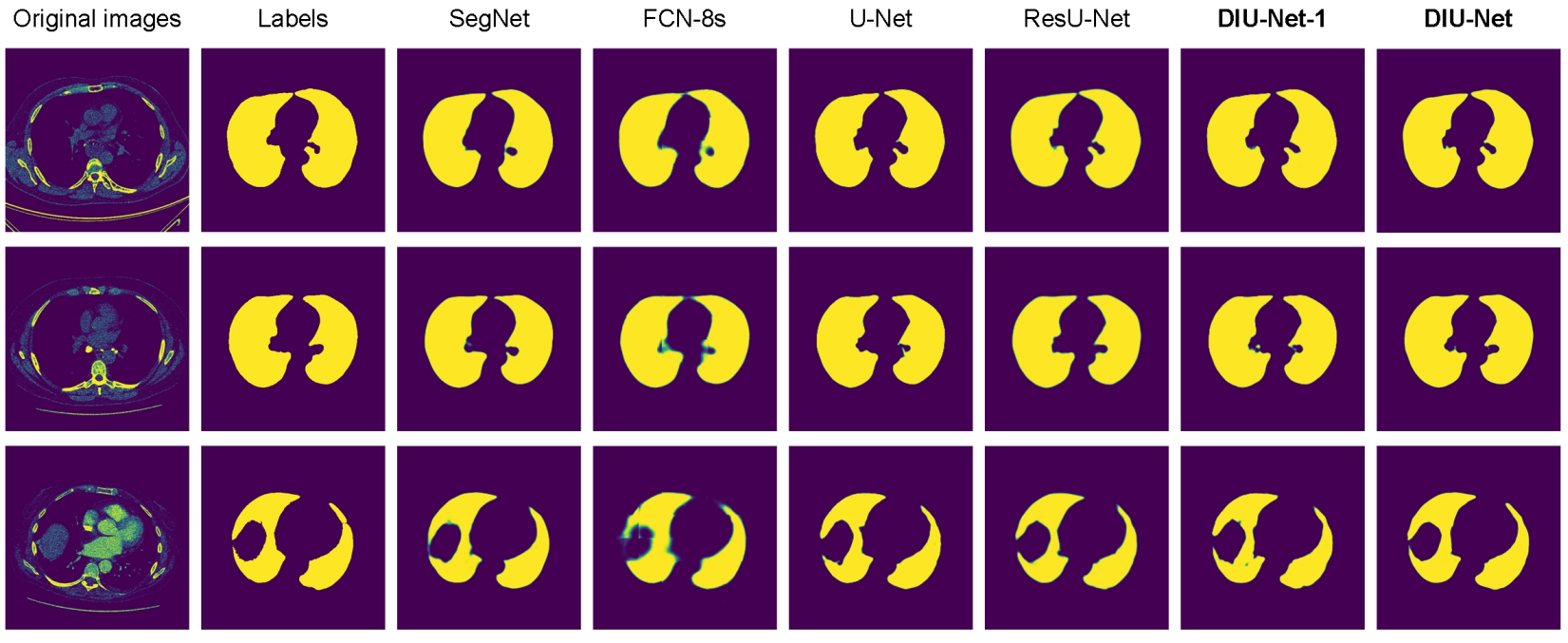

Input image tile
1
256×256
64
256×256
128×128
64
128
128×128
64×64
128
256
64×64
32×32
256
512
32×32
16×16
512
1024
32×32
16×16
1024
512
32×32
256
16×16
512
128×128
128
256
128×128
64
128×128
64
32
256×256
256×256
1
Output image tile
Inception-Res block
Dense-Inception block
Down-sampling block
Up-sampling block
Concatenation

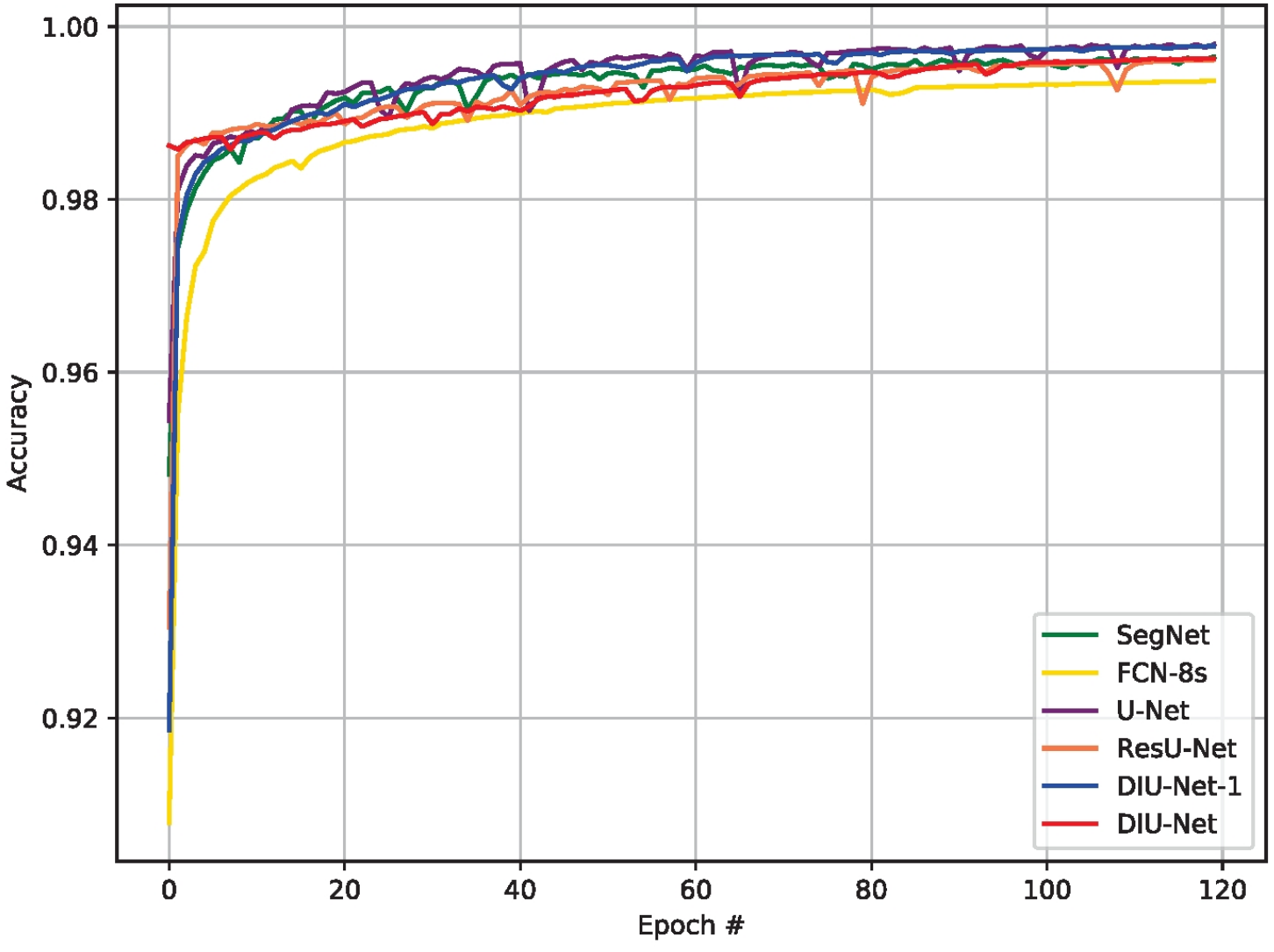

Output
1×1 Conv
BN
3×3 Conv
BN
BN
3×3 Conv
3×3 Conv
BN
1×1 Conv
1×1 Conv
1×1 Conv
Input
Residual concatenation
BN
Batch normalization layer
3×3 Conv
3×3 convolution layer
1×1 Conv
1×1 bottleneck layer

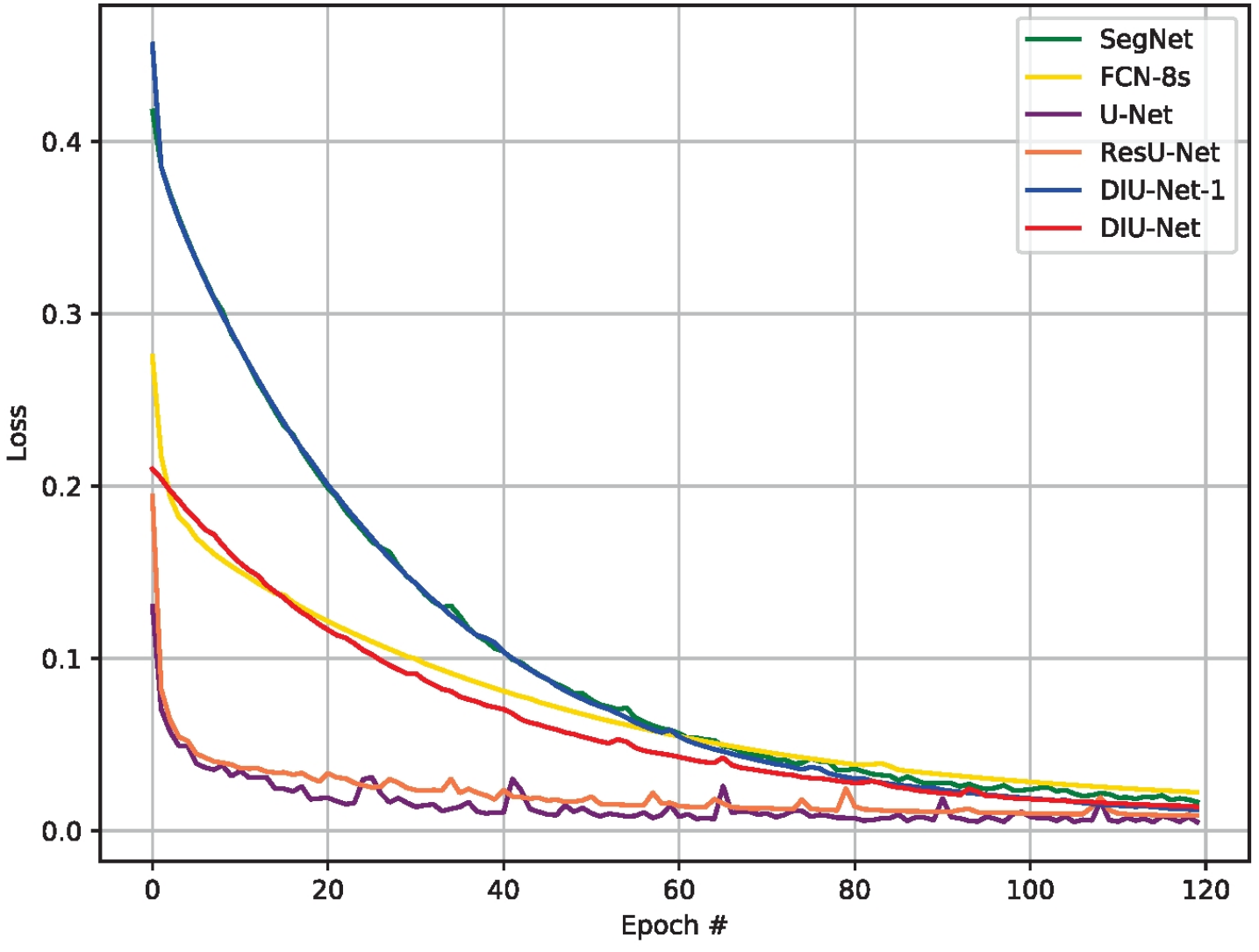

Output
BL
Inception-Res
BL
Inception-Res
BL
Inception-Res
BL
Inception-Res
Input
Concatenation
BL
Bottleneck layer
Inception-Res
Inception-Res block

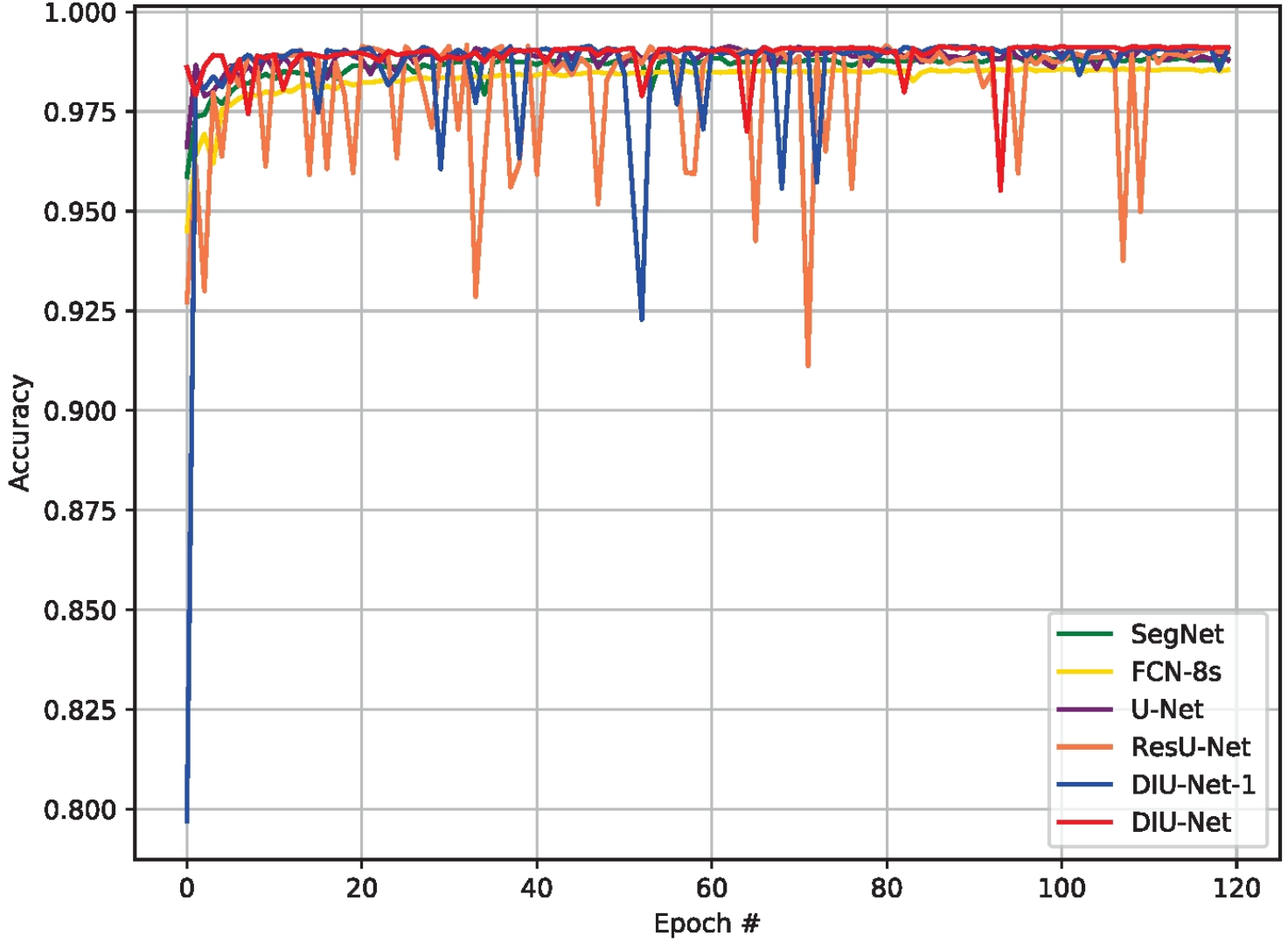

Output
1×1 Conv
BN
3×1 Conv
BN
BN
1×1 Conv
1×3 Conv
BN
1×1 Conv
Max-pooling
1×1 Conv
Input
Residual concatenation
Max-pooling
Max-pooling layer
BN
Batch normalization layer
1×3 Conv
Convolution layer
1×1 Conv
1×1 bottleneck layer

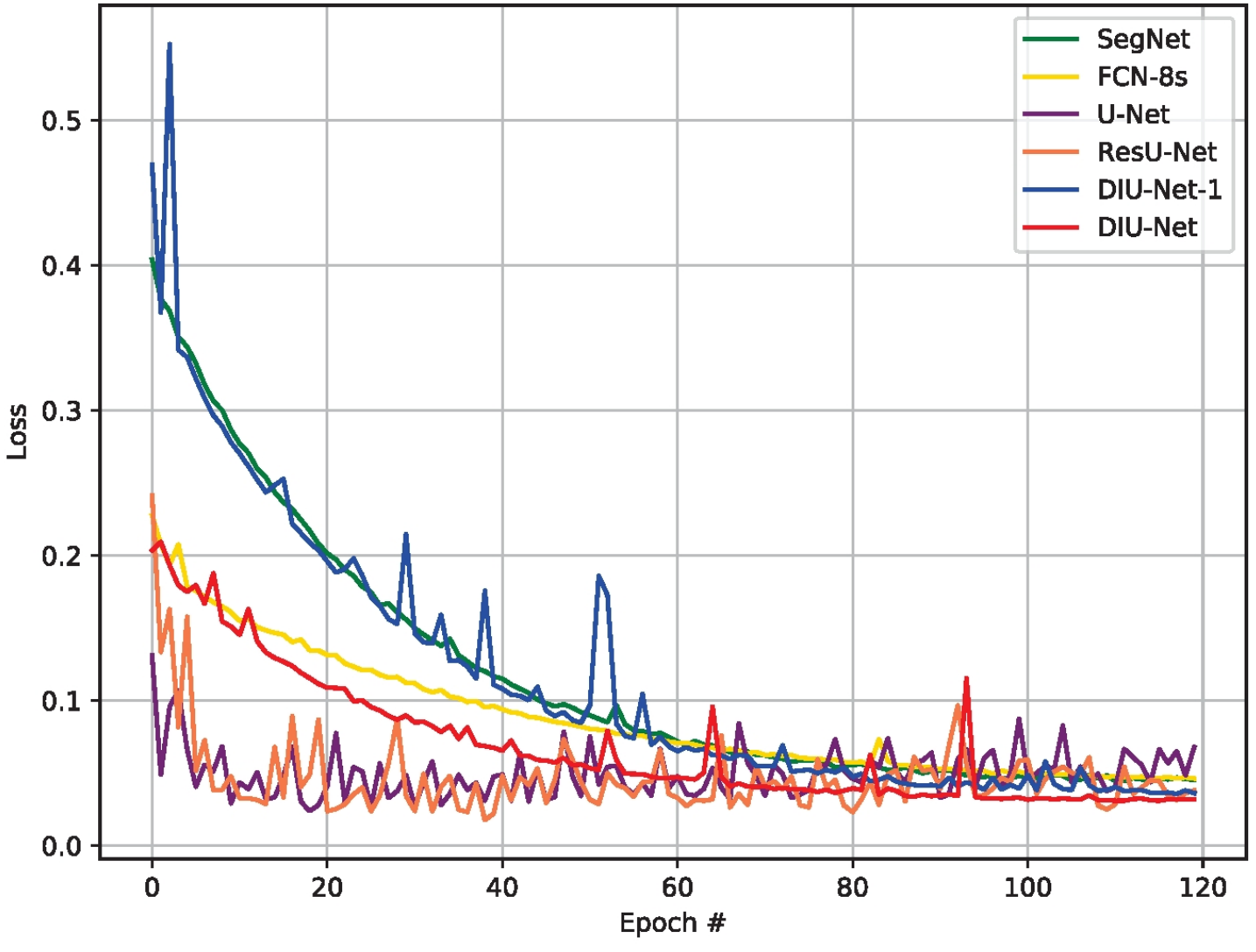

Input
Max-pooling 2D
1×1 Conv
1×1 Conv
3×3 Conv,S =2
3×3 Conv
3×3 Conv,S =2
1×1 Conv
Output
(a) Down-sampling module
Output
1×1 ConvTran
3×3 ConvTran,S=2
3×3 ConvTran,S=2
3×3 ConvTran
Upsampling2D
1×1 ConvTran
1×1 ConvTran
Input
(b) Up-sampling module
1×1 ConvTran
1×1 transposed convolution layer
1×1 Conv
1×1 convolution layer
3×3 Conv,S =2
3×3 convolution layer with stride = 2
Upsampling2D
Un-pooling layer with strides = 2
3×3 ConvTran,S=2
3×3 transposed convolution layer with stride = 2
Maxpooling2D
Max-pooling layer with strides = 2

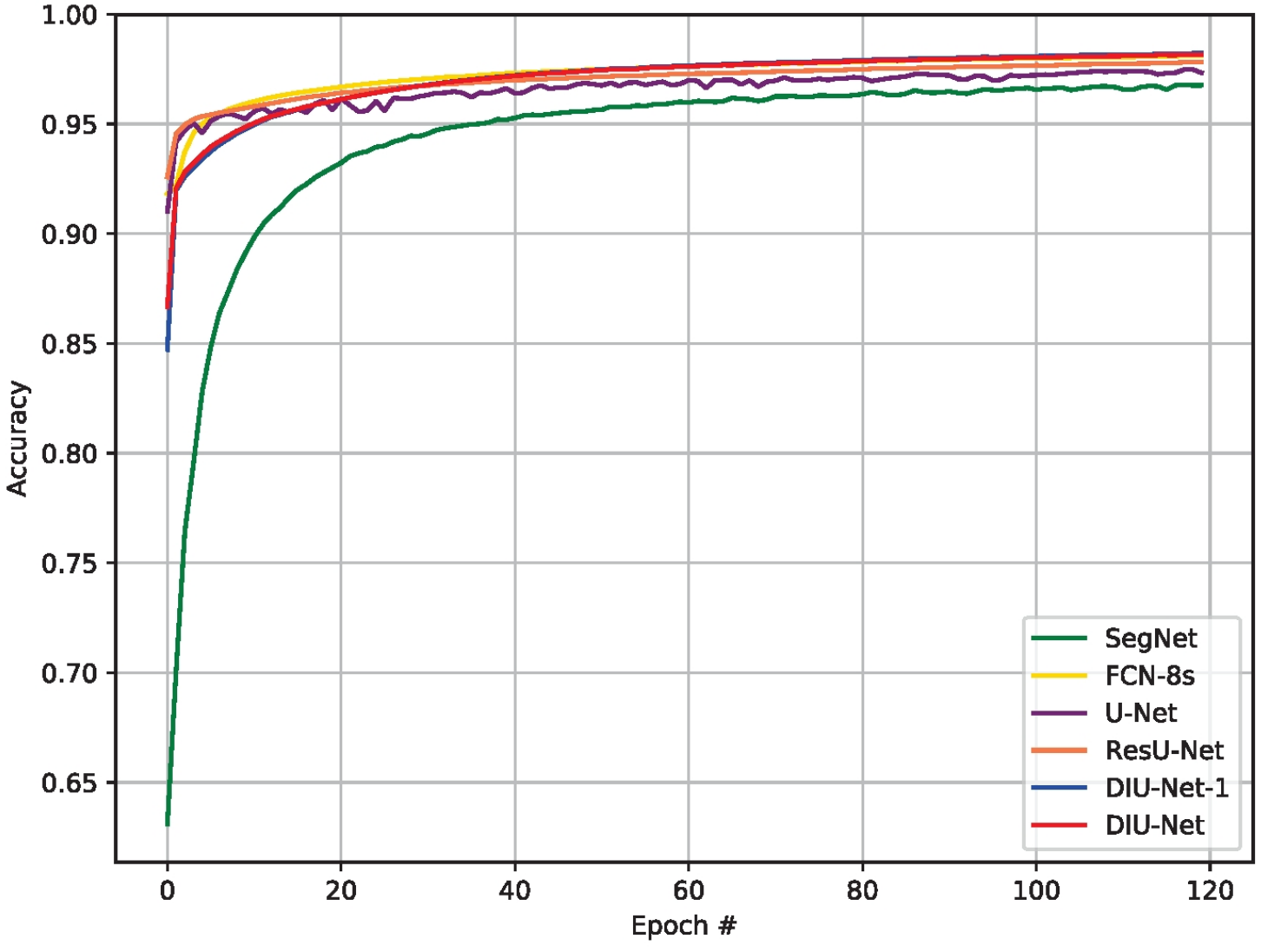

Original images
Labels
SegNet
FCN-8s
U-Net
ResU-Net
DIU-Net-1
DIU-Net

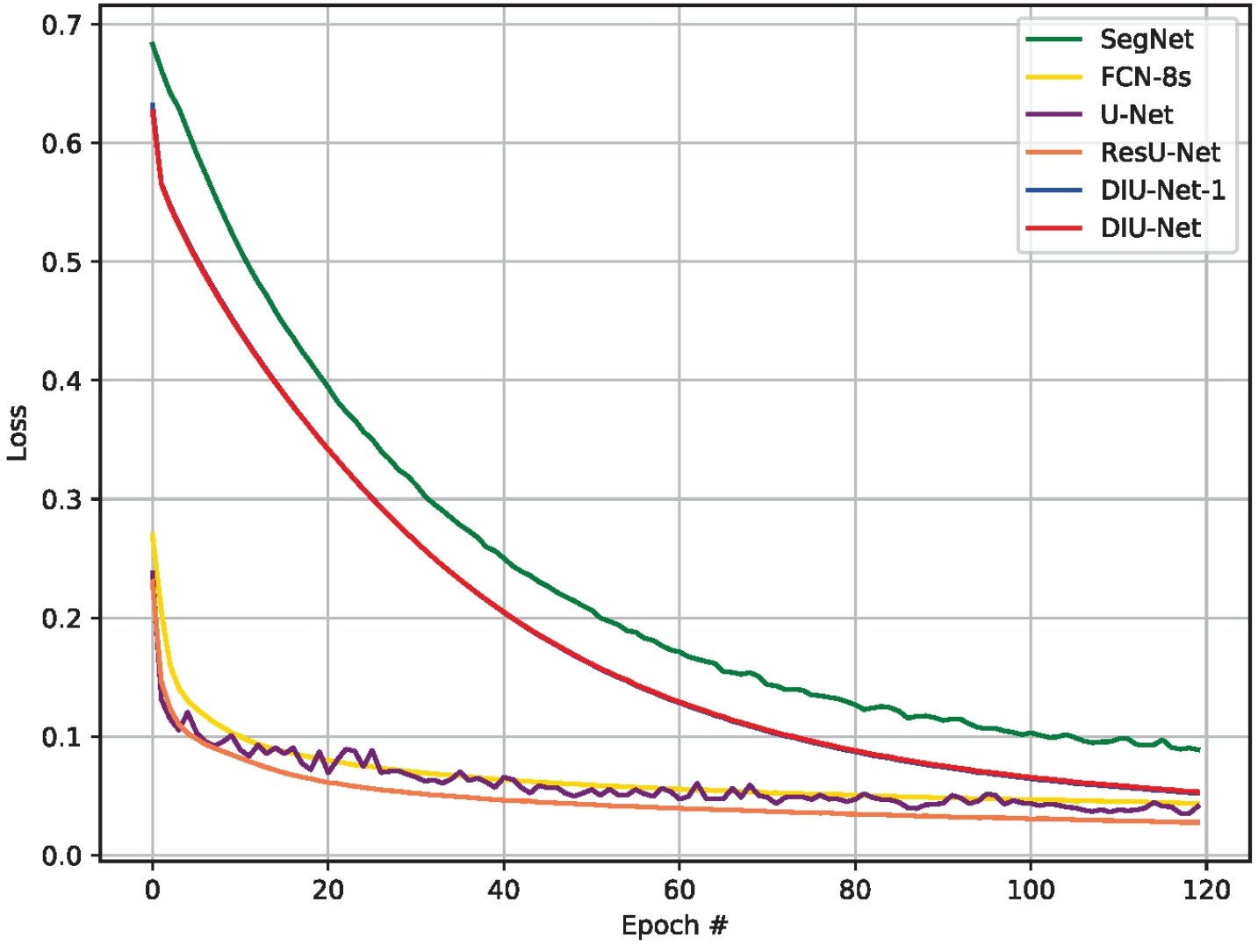

1.00
0.98
0.96
0.94
0.92
Accuracy
0
20
40
60
80
100
120
Epoch #
SegNet
FCN-8s
U-Net
ResU-Net
DIU-Net-1
DIU-Net

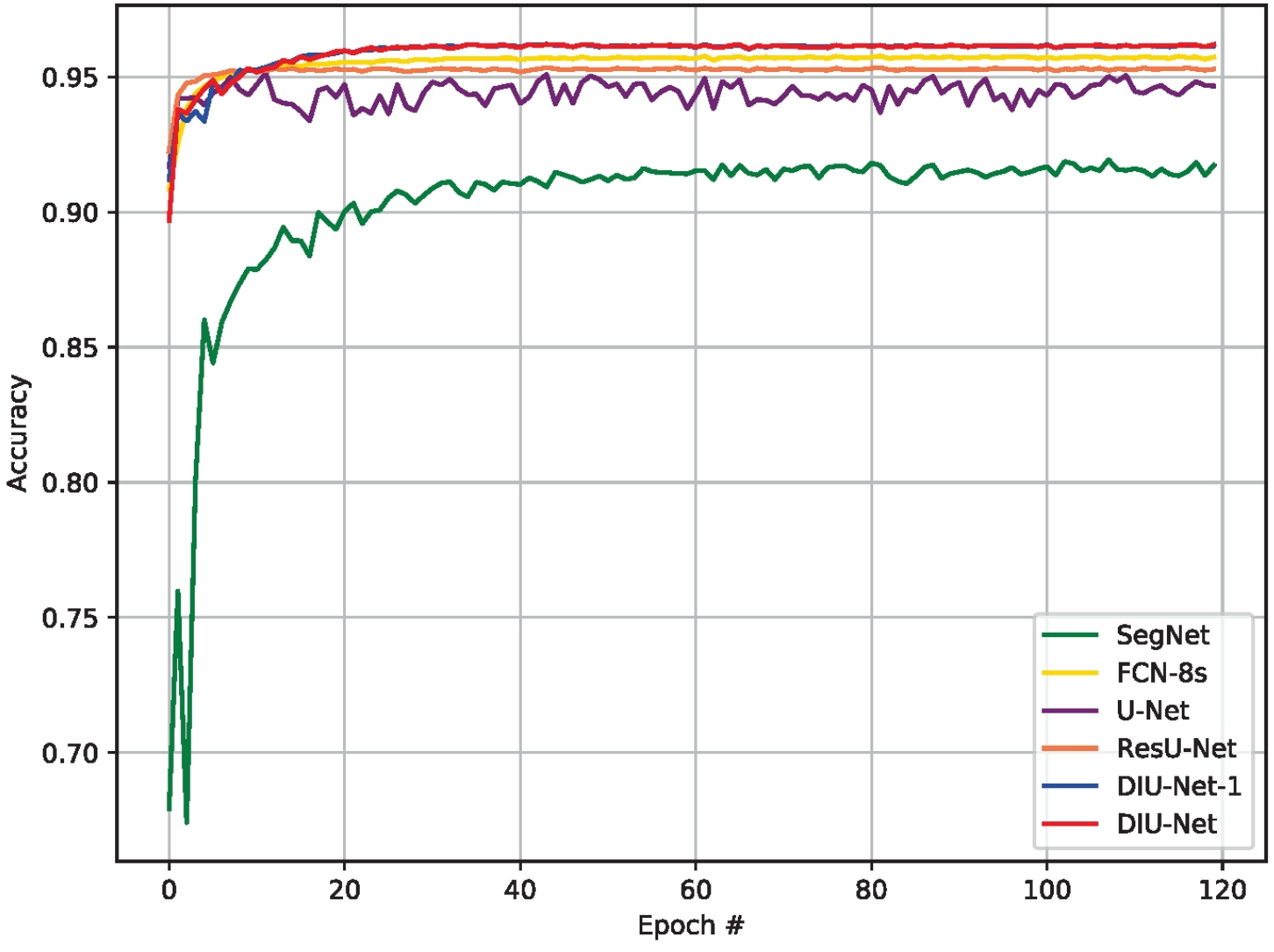

SegNet
FCN-8s
U-Net
ResU-Net
DIU-Net-1
DIU-Net
Loss
0.0
0.1
0.2
0.3
0.4
0
20
40
60
80
100
120
Epoch #

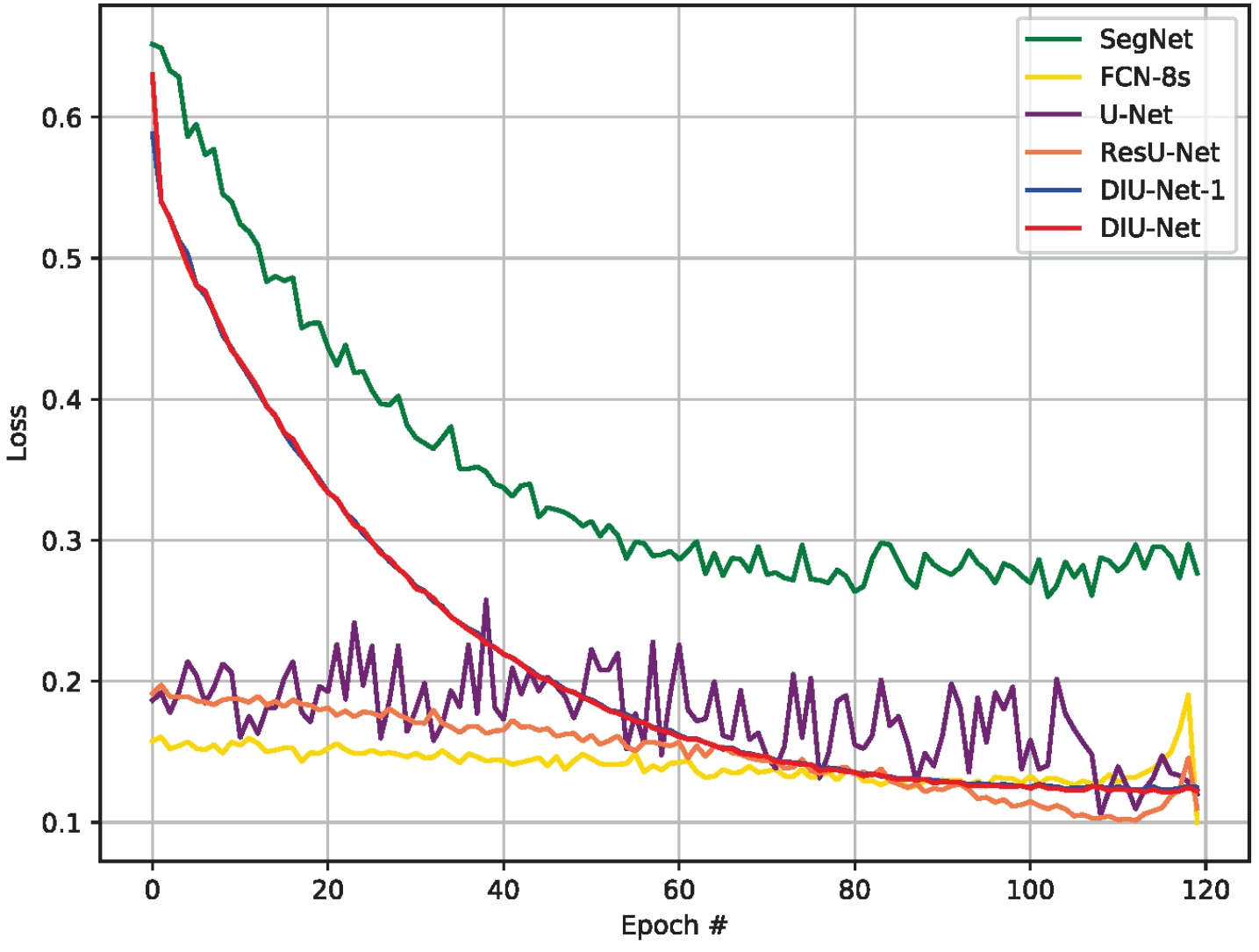

1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
0.800
Accuracy
0
20
40
60
80
100
120
Epoch #
SegNet
FCN-8s
U-Net
ResU-Net
DIU-Net-1
DIU-Net

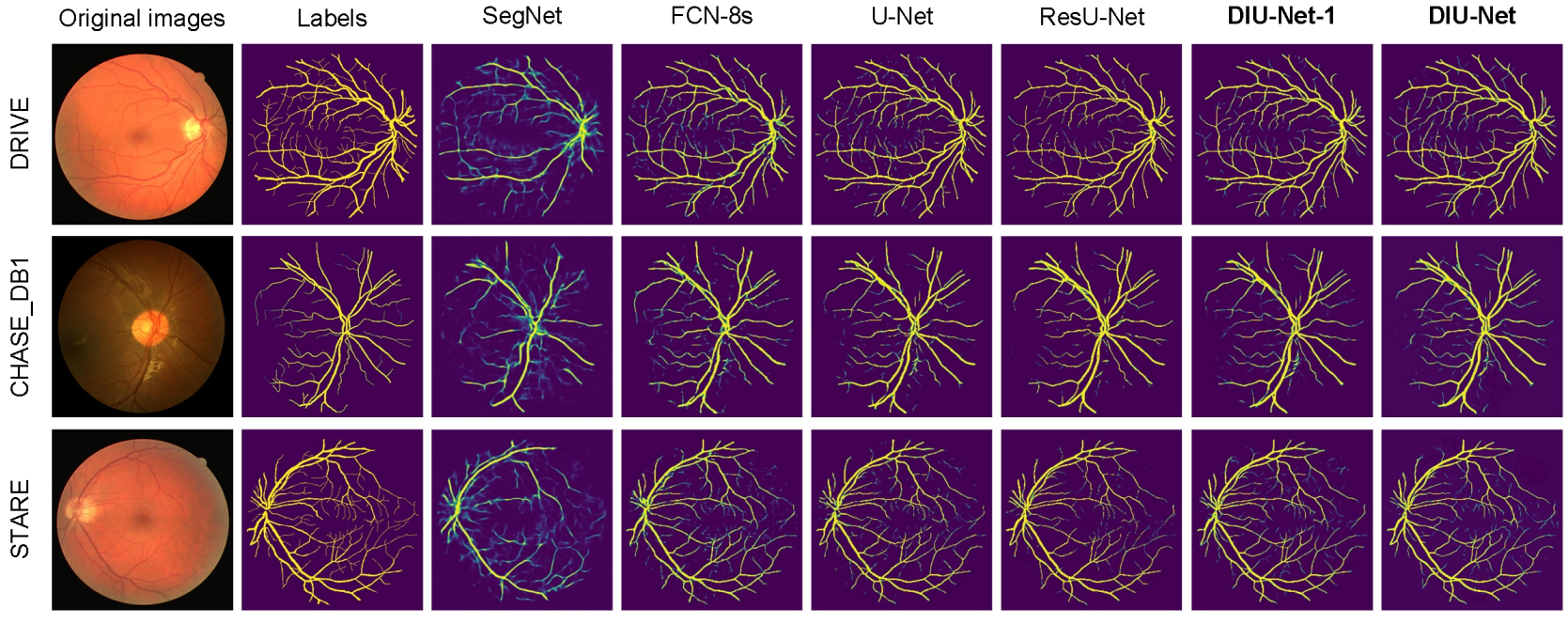

SegNet
FCN-8s
U-Net
ResU-Net
DIU-Net-1
DIU-Net
Loss
0.0
0.1
0.2
0.3
0.4
0.5
0
20
40
60
80
100
120
Epoch #

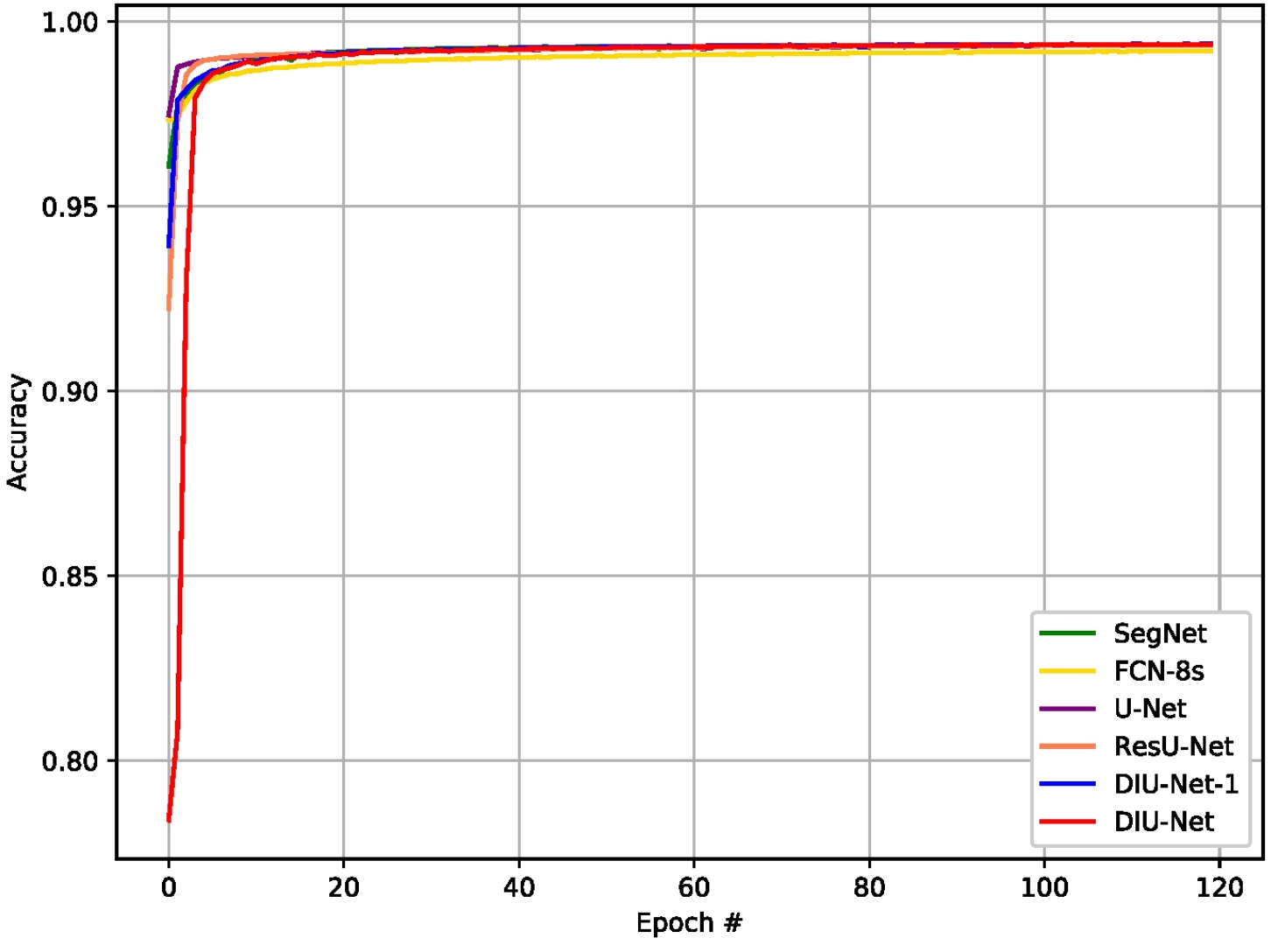

1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
Accuracy
0
20
40
60
80
100
120
Epoch #
SegNet
FCN-8s
U-Net
ResU-Net
DIU-Net-1
DIU-Net

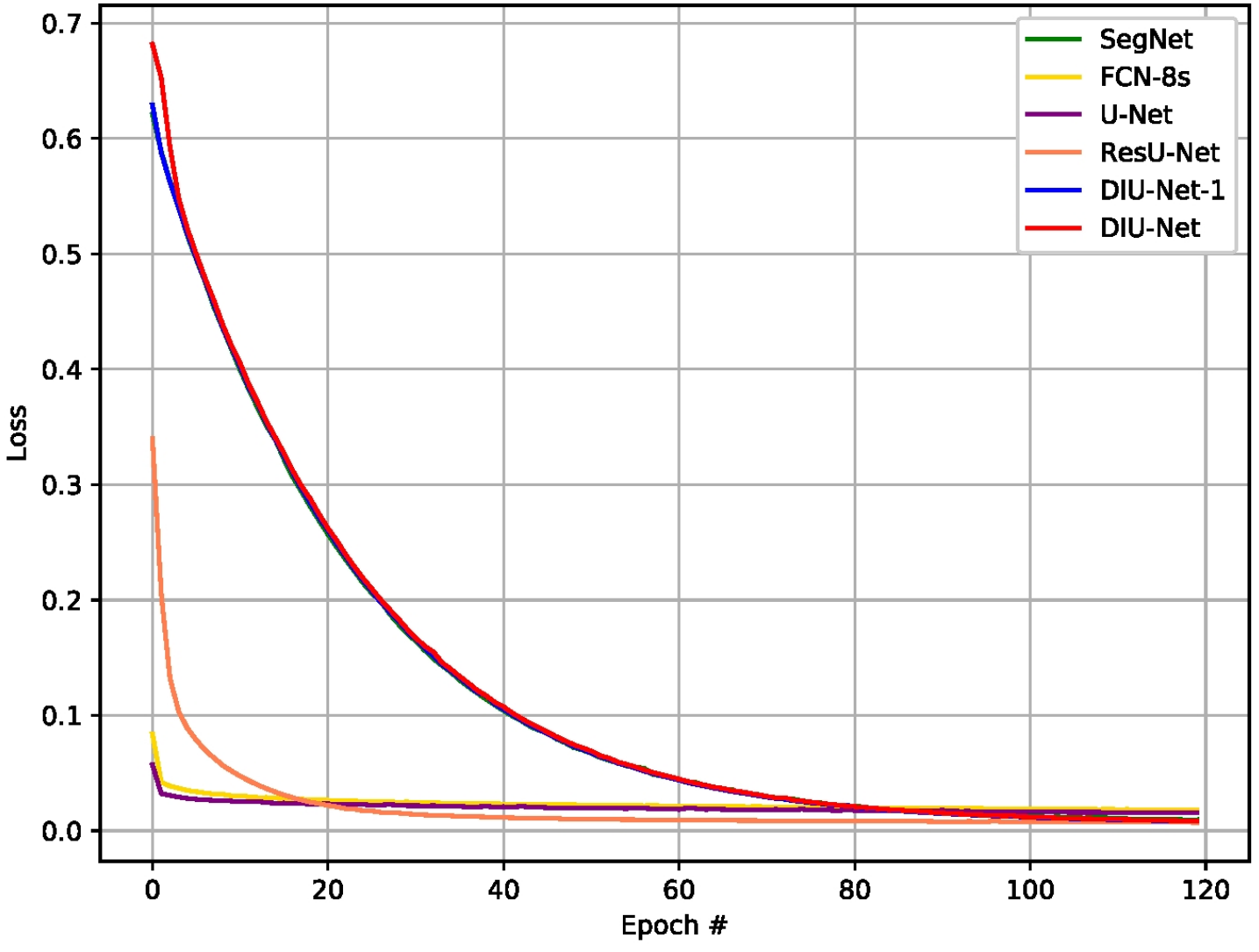

SegNet
FCN-8s
U-Net
ResU-Net
DIU-Net-1
DIU-Net
Loss
Epoch #
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
20
40
60
80
100
120

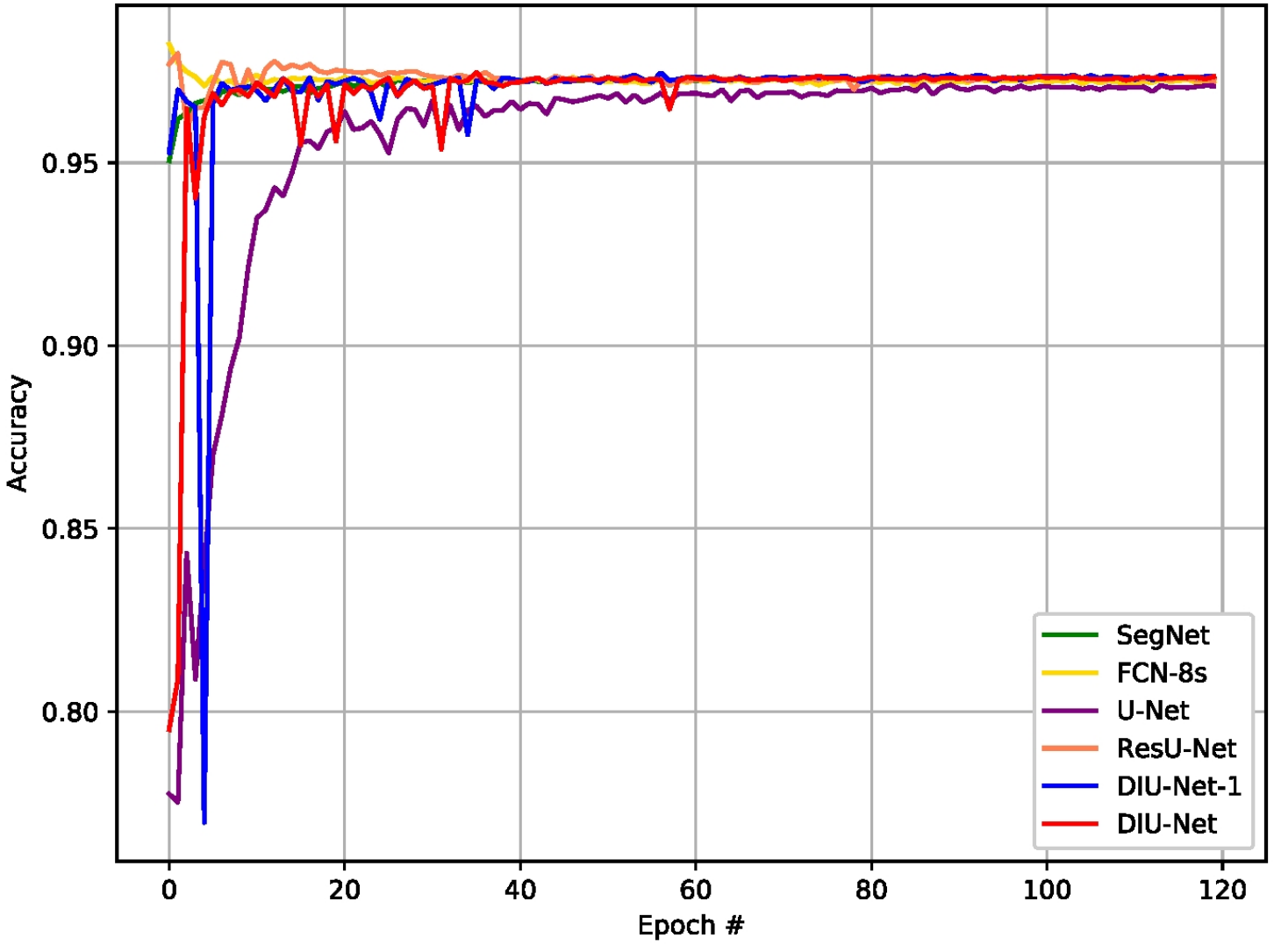

0.95
0.90
0.85
0.80
0.75
0.70
Accuracy
0
20
40
60
80
100
120
Epoch #
SegNet
FCN-8s
U-Net
ResU-Net
DIU-Net-1
DIU-Net

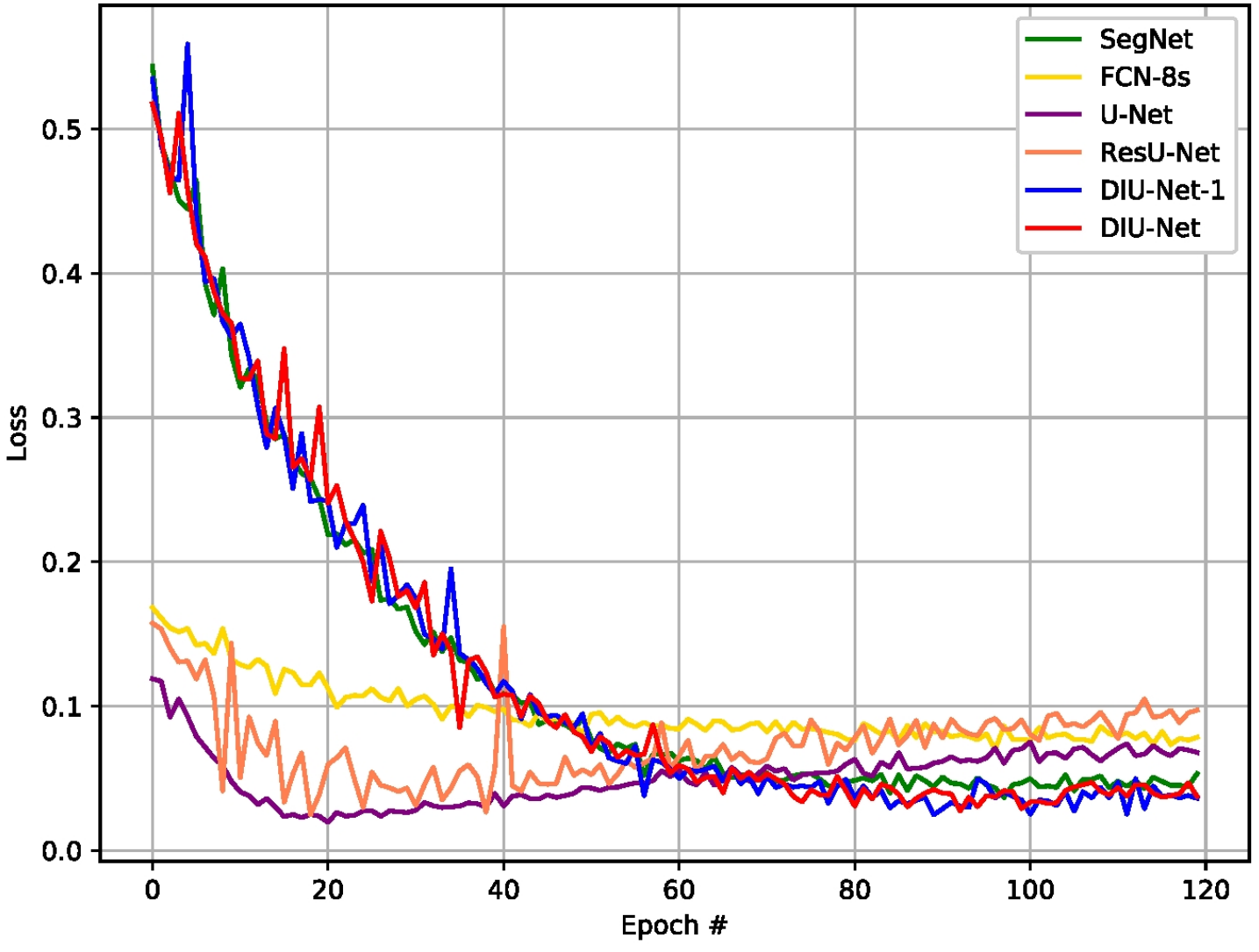

SegNet
FCN-8s
U-Net
ResU-Net
DIU-Net-1
DIU-Net
Loss
0.6
0.5
0.4
0.3
0.2
0.1
0
20
40
60
80
100
120
Epoch #

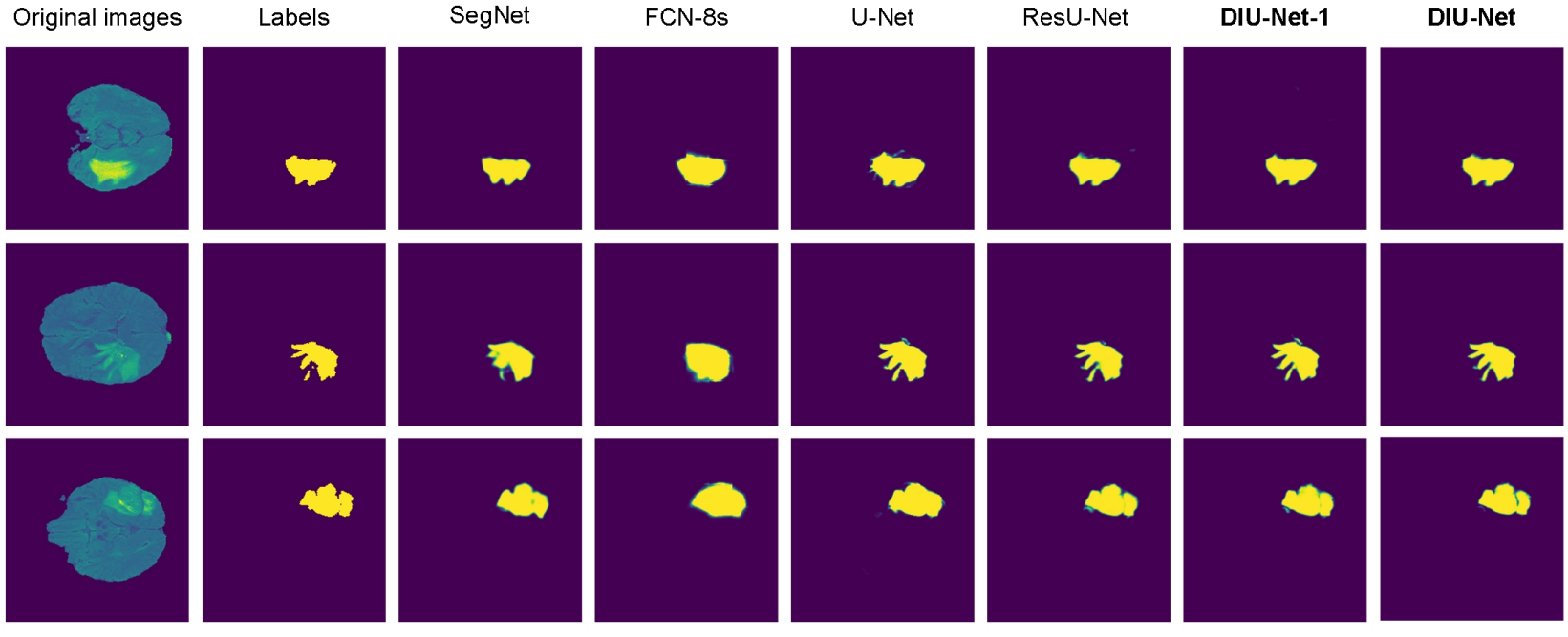

Original images
Labels
SegNet
FCN-8s
U-Net
ResU-Net
DIU-Net-1
DIU-Net
DRIVE
CHASE_DB1
STARE


1.00
0.95
0.90
0.85
0.80
Accuracy
0
20
40
60
80
100
120
Epoch #
SegNet
FCN-8s
U-Net
ResU-Net
DIU-Net-1
DIU-Net


SegNet
FCN-8s
U-Net
ResU-Net
DIU-Net-1
DIU-Net
Loss
Epoch #
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
20
40
60
80
100
120


0.95
0.90
0.85
0.80
Accuracy
0
20
40
60
80
100
120
Epoch #
SegNet
FCN-8s
U-Net
ResU-Net
DIU-Net-1
DIU-Net


SegNet
FCN-8s
U-Net
ResU-Net
DIU-Net-1
DIU-Net
Loss
0.0
0.1
0.2
0.3
0.4
0.5
0
20
40
60
80
100
120
Epoch #


Original images
Labels
SegNet
FCN-8s
U-Net
ResU-Net
DIU-Net-1
DIU-Net

Evaluation metrics	Expression
DICE	$2\frac{GT \cap SR}{\lvert GT \rvert + \lvert SR \rvert}$
JS	$\frac{\lvert GT \cap SR \rvert}{\lvert GT \cup SR \rvert}$
AC	$\frac{TP + TN}{TP + TN + FP + FN}$
SE	$\frac{TP}{TP + FN}$
SP	$\frac{TN}{TN + FP}$
F1-score	$\frac{2TP}{2TP + FP + FN}$

Networks	DICE	JS	AC	SE	SP	F1-score	AUC (95%CI)
SegNet	0.9816	0.9790	0.9894	0.9662	0.9960	0.9753	0.9810 (0.9788-0.9833)
FCN-8s	0.9793	0.9743	0.9870	0.9632	0.9909	0.9689	0.9821 (0.9794-0.9845)
U-Net	0.9826	0.9810	0.9904	0.9641	0.9978	0.9773	0.9810 (0.9776-0.9844)
ResU-Net	0.9834	0.9824	0.9911	0.9619	0.9981	0.9791	0.9820 (0.9788-0.9853
DIU-Net	**0.9857**	**0.9871**	**0.9935**	**0.9846**	**0.9982**	**0.9850**	**0.9903 (0.9890-0.9918)**
DIU-Net-1	0.9823	0.9804	0.9901	0.9616	0.9980	0.9768	0.9799 (0.9769-0.9828)

Methods	DICE	JS	AC	SE	SP	F1-score	AUC (95% CI)
Yitian Zhao [38]	-	-	0.9540	0.7503	0.9817	-	0.9643
Fraz [39]	-	-	0.9491	0.7386	0.9772	-	0.9736
Qiaoliang Li [40]	-	-	0.9566	0.7593	0.9817	-	0.9743
SegNet	0.9128	0.8521	0.9200	0.3631	0.9736	0.4403	0.8196 (0.8078-0.8313)
FCN-8s	0.9523	0.9228	0.9598	0.7645	0.9788	0.7671	0.9713 (0.9630-0.9796)
U-Net	0.9555	0.9287	0.9630	0.7741	0.9814	0.7834	0.9687 (0.9599-0.9775)
ResU-Net	0.9567	0.9309	0.9642	0.7461	0.9854	0.7829	0.9495 (0.9411-0.9580)
DIU-Net	**0.9582**	**0.9338**	**0.9657**	**0.7967**	**0.9863**	**0.8003**	**0.9802 (0.9731-0.9873)**
DIU-Net-1	0.9578	0.9331	0.9654	0.7765	0.9818	0.7951	0.9776 (0.9711-0.9842)

Networks	DICE	JS	AC	SE	SP	F1-score	AUC (95%CI)
SegNet	0.9878	0.9911	0.9955	0.9018	0.9973	0.8710	0.9496 (0.9425-0.9567)
FCN-8s	0.9853	0.9862	0.9930	0.9217	0.9942	0.8195	0.8529 (0.8328-0.8730)
U-Net	0.9872	0.9900	0.9950	0.9233	0.9953	0.8611	0.9495 (0.9424-0.9566)
ResU-Net	0.9891	0.9938	0.9969	0.9232	0.9981	0.9046	0.9607 (0.9528-0.9686)
DIU-Net	**0.9892**	**0.9940**	**0.9970**	**0.9254**	**0.9986**	**0.9058**	**0.9867 (0.9847-0.9889)**
DIU-Net-1	0.9891	0.9939	0.9959	0.9245	0.9941	0.9056	0.9613 (0.9545-0.9681)

Networks	Lungs segmentation		Blood vessel segmentation		Brain tumor segmentation	
	Training time	Testing time	Training time	Testing time	Training time	Testing time
SegNet	3.3h	0.97sec	2.8h	0.87sec	6.2h	0.65sec
`FCN-8s	3.8h	0.97sec	3.2h	0.89sec	6.3h	0.77sec
U-Net	5.0h	1.12sc	4.1h	0.98sc	8.5h	0.89sec
ResU-Net	5.2h	1.18sec	4.7h	1.01sec	8.4h	0.87sec
DIU-Net-1	7.4h	1.34sec	5.0h	1.03sec	9.8h	1.14sec
DIU-Net	7.7h	1.36sec	5.6h	1.07sec	11.2h	1.27sec