IT-Block: Inverted Triangle Block embedded U-Net for Medical Image Segmentation

Xueyang Li, Yongfeng Huang(✉), Cairong Yan, Lihao Liu
School of Computer Science and Technology
Donghua University, Shanghai 201620, China
weike10015@gmail.com, yfhuang@dhu.edu.cn, cryan@dhu.edu.cn, 2181757@mail.dhu.edu.cn

***Abstract*—Convolutional neural network (CNN) such as U-Net has demonstrated excellent performance for medical image segmentation. However, there are some limitations of its scalability. Specifically, the memory would increase significantly when embedding other functional modules into U-Net. Moreover, the kernel size used in U-Net is unitary, which makes it difficult to obtain the multi-level information and extract the target completely. In this paper, we only use 12 convolutional layers of U-Net as a backbone and design a novel architecture named Inverted Triangle (IT) Block embedded into it to address these problems. The IT-Block consists of Dense Connection, Residual Connection, and Inception, aiming to help the network obtain multi-level features and reuse them comprehensively. Furthermore, we optimize the dice loss to alleviate the butterfly effect, making the training process more stable during the backpropagation. The experimental results state that our framework is superior to U-Net in running time and accuracy.**

***Index Terms*—Convolutional Neural Networks, U-Net, Medical Image Segmentation, Butterfly Effect**

I. Introduction

Image segmentation is the first step of image analysis and one of the most difficult problems in image processing. In the field of medical imaging, medical image segmentation has always been the key to supporting for subsequent processes such as registration, cancer diagnosis, treat planning, surgery simulation and reconstruction. However, organs and tissues are varied in shape and size, which increases the difficulty on segmentation.

Recently, many significant researches have shown that Convolutional Neural Networks (CNNs) achieve the state-of-the-art in different medical image segmentation tasks [1], [2], [3], [4], [5], [6]. One of the most well-known architectures in the medial imaging is U-Net [7]. The reasons why U-Net is widely used and modified are as following: (1) an elegant architecture, it consists of downsampling path and corresponding upsampling path. Almost the same number of layers on both paths results in a symmetrical architecture approximately. The two paths of U-Net are bridged through skip connections, such a design fuses the feature maps from different layers, recovering the spatial information lost during downsampling [8], aiming to get a high-resolution and representative feature map, which can generate mask precisely; (2) the adaptivity for small datasets takes very little time for iterative inference. It is the key point where real-time performance required. Besides, massive amounts of annotated training data may not be appropriate in the fields of medical imaging. That is because the process of labeling medical images is time-consuming and must be done by domain experts [9].

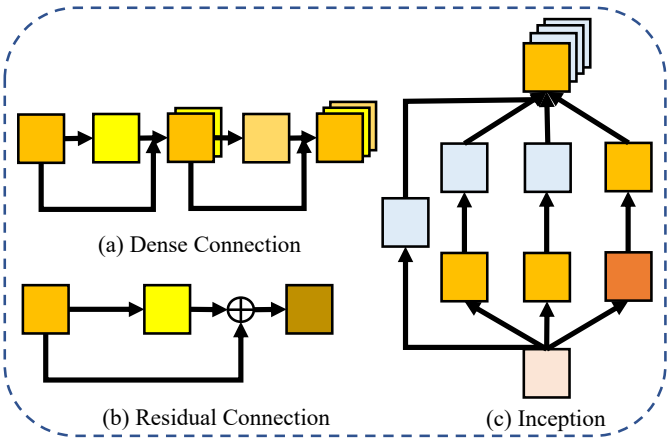

Fig. 1. A brief description of Dense Connection, Residual Connection, and Inception. Please note that the ways to fuse features are different. Residual Connection uses adding operation, but the other two make a fusion via concatenation.

However, the defects that spoil the perfection of U-Net can be summarized in two aspects: (1) it is difficult to make some subtle changes in the architecture to improve performance or be compressed as a functional module integrated into other backbones. More specifically, making the network deeper may obtain better results, while the gradient vanishing is more likely to occur during backpropagation [10], failing to update the parameters; (2) the fixed kernels (3×3) used in U-Net cannot make the network obtain large-structure and tiny-structure information simultaneously. Therefore, extending U-Net by increasing the number of layers directly is not suitable. The target is to take advantage of U-Net and add new elements to it, creating a robust network with low parameters.

In the improvement of U-Net, Dense Connection [11], Residual Connection [12], and Inception [13] are most often used [14], [15], [16] (Fig. 1). The strengths of them are as following: (1) Dense Connection makes layers directly connected with all of their previous layers. Such a design allows the network to take the features from all layers into account. Multi-level feature information is more helpful for the classifier to extract targets than that of the single last-level feature information that is always used in standard

CNNs. Furthermore, Dense Connection enables gradients to flow smoothly during backpropagation, alleviating the gradient vanishing so that the network can be trained easily; (2) The more layers network equips, the more features of different levels can be extracted. But the deeper network also brings two problems (gradient vanishing and degradation [12]). Residual Connection is equivalent to performing the identity mapping, solving the degradation problem when the network goes deeper. Besides, deepened residual networks can be easier to optimize than deep networks produced by overlaying simply; (3) Inception provides a wider network with Multi kernels $(1 \times 1, 3 \times 3, 5 \times 5)$ on parallel levels, allowing the module to freely choose better features. Even more importantly, the 1×1 kernels are implemented to compute reductions before the expensive 3×3 and 5×5 convolutions, decreasing the parameters. The question is how to integrate these three operations into functional modules embedded in U-Net to improve its performance on medical image segmentation and simplify its structure.

Motivated by the drawback of U-Net and previous modification work on it, we test the performance of U-Net (e.g. running time, accuracy) by changing its depth and determine the depth interval of U-Net (the number of layers) that is appropriate for doing segmentation (Fig. 2). Based on that, we propose a block consisting of Dense Connection, Residual Connection, and Inception, aiming to optimize U-Net through embedding the block into shallower U-Net (lower depth). We term this block Inverted Triangle (IT) Block due to its style. In summary, the main contributions are as following: (1) a discussion on the appropriate depth interval of U-Net; (2) easy and flexible implementation of the IT-Block; (3) an optimization of Dice loss; (4) an efficient embedded block that can optimize the standard backbone to excel in challenging medical image segmentation tasks.

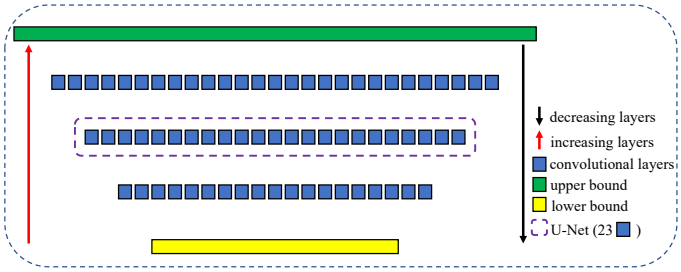

Fig. 2. 23 convolutional layers of U-Net [7]. Please note that there are no other operations such as max-pooling and skip connections in the diagram. Convolutional layers also include up-convolutional layers. Increasing and decreasing layers are only based on convolutional layers. Upper bound and lower bound are the limits of layers that U-Net can achieve, making the segmentation results acceptable.

II. Related Work

A. CNNs in medical image segmentation

Within medical imaging, relevant researches have shown the capabilities of Convolutional Neural Networks (CNNs) to solve the challenging segmentation tasks in the past few years. Fully Convolutional Network (FCN) [17], such an elegant CNN architecture could obtain the probability of each pixel rather than the scalar of the whole image, because the fully connected layer is replaced with the convolutional layer. Moreover, one different strategy in the training period is that FCN has the ability to use the whole image (original image and label image) as inputs, rather than feeding all possible separate sub-images centered on each labelled pixel like the earlier methods [18]. The breakthrough made by FCN promoted the development of CNNs in medical image segmentation tasks. Based on FCN, Zhou et al. [19] combined 2D FCN with 3D Majority voting algorithm, achieving great performance in Three-Dimensional segmentation task of human torso. The upsampling layer as an important part of FCN can restore the feature maps to higher resolution, but this process would lead to inaccurate positions of each pixel, resulting in bad segmentation. To solve this problem, some researchers used MRF [20] to further optimize the segmentation results. FCN has good scalability as a backbone. Olaf Ronneberger et al. [7] extended FCN to a symmetrical architecture like letter U and added the skip connections to fuse the feature between upsampling path and downsampling path, achieving the great performance on the cell tracking challenge. X. Li et al. [14] proposed H-DenseUnet with mixed dense connections, reducing the memory consumption of GPU during the training step and excelling in Liver MICCAI 2017. W. Chen et al. [21] used the skip connections to bridge two U-Nets into a stacked U-Net, which can deal with small datasets and be used in prostate segmentation without a pre-train model. N. Ibtehaz et al. [22] proposed some modifications to improve the performance of standard U-Net. Specifically, they took advantage of Res-Net [12] and Inception [13], adding them into standard U-Net. Inspired by the idea of recurrent segmentation [23], W. Wang et al. [24] created the Recurrent U-Net that can operate in environments where computational power and the amount of training data are limited.

Several deep learning techniques have recently focused on 3D segmentation. Cicek et al. [25] proposed a 3D version of U-Net to implement 3D image segmentation by inputting continuous 2D slices. F. Dubost et al. [26] trained a regression network with a fully convolutional architecture which is combined with a global pooling layer to aggregate the 3D output into a scalar. C. Huang et al. [27] presented a 3D Universal U-Net for multi-organ segmentation problem, filling the gap of flexible multi-domain learning in image segmentation. The limit of 3D convolutional neural networks is memory consumption, because 3D images often need more parameters trained than that of 2D images. This problem was addressed by R. Brügger et al. [28], they designed a partially reversible U-Net architecture that reduces memory consumption substantially. This excellent architecture can recover each layer's outputs from the subsequent layer, which eliminates the requirement for storing activations during backpropagation.

B. Progress on receptive fields

In deep learning, it is sometimes necessary to increase the receptive filed for segmentation tasks to improve performance. Specifically, Large receptive fields can be achieved in three ways but with negative costs: (1) increasing pooling layers but losing some spatial information due to reducing the size of images; (2) enlarging the size of convolutional kernel directly but resulting in parameters growing; (3) making the network deeper but leading to gradient vanishing. Dilate convolution [29] is widely used to increase the receptive fields without losing information like the pooling operation. Furthermore, it has the capability to capture the contextual information from multiple scales. However, the fixed rate of dilation used in [29] makes the convolutional kernel too sparse, which causes local information cannot be completely covered. The first attempt to solve this problem made by P. Wang et al. [30]. They proposed a simple hybrid dilation convolution (HDC) framework, using a range of dilation rates to substitute the fixed rate of dilation, which avoids the gridding problem. To increase the receptive field, larger kernel is not necessarily needed. It is because larger kernel can be replaced by multiple smaller kernel, which can be seen as imposing a regularization on the larger kernel and keep the parameter low [31].

III. Breaking down the problem

A. Rethinking the capability of U-Net

In general, the depth of the network can affect its parameters and ability to extract features. Our target is that the Inverted Triangle (IT) Block can be embedded in the backbone to improve its performance and keeps parameters low. Technically, embedding the IT-Block into the original backbone directly is inappropriate, which would cause a significant increase in parameters. Hence, the key point that we focus on is how to keep the parameters low by decreasing the number of convolutional layers without sacrificing the performance of the original architecture as much as possible. This leads to the exploration of determining the appropriate depth interval of U-Net. To be more specific, there are 23 convolutional layers in U-Net [7]. We need to test the lower bound of appropriate depth interval (the lower number of convolutional layers in U-Net), which can make sure the results segmented by U-Net with lower depth are acceptable. After depth interval is determined, we choose a model with a lower depth from it as the backbone.

We use four datasets including cells [7], Skin, Lung, and Nuclei [16] as our training-validation dataset and run a 5-fold cross-validation due to the small size of each dataset. The dice coefficient (1) is used to compare the similarity of two sets (segmented mask and its corresponding ground truth). We calculate its mean and std (standard deviation) via (11) (12) on each datasets. For the training period, the loss function used in this part has not been optimized (6). The more details about source of each dataset and implementation can be achieved in section IV (Experiments and Results) of this paper.

$$Dice = \frac{2\left|S_{ij} \bigcap G_{ij}\right|}{\left|S_{ij}\right| + \left|G_{ij}\right|} \tag{1}$$

Where S_{ij} is the segmented image. G_{ij} is the ground truth corresponding to S_{ij}. The medical segmentation tasks in our experiments are binary classification problem, so the ground truth G_{ij} is the 0-1 matrix. $\bigcap$ makes matrix S_{ij} multiply matrix G_{ij} point by point, which means multiplying points at the same position in these two matrices. Please note that this operation is different from the multiply in Linear algebra. $|*|$ represents the scalar that is summed up every point in the matrix.

The kernels of U-Net are not consistent (e.g. $3\times3\times64$ in 1^{st} convolutional layer and $3\times3\times128$ in 3^{rd} convolutional layer). That means decreasing one layer from different levels would lead to different changes in parameters. Hence, we change the number of layers and parameters to shape a backbone (Fig. 3). To start with, we reduce the layer of U-Net from top to bottom to find the lower bound of the appropriate depth interval. For the convolutional layer attached to skip connections, we remove them together. Secondly, we also test the highest bound of the appropriate depth interval by increasing the convolutional layer based on U-Net. Fig. 3 shows the results evaluating the effects of layers and parameters. From 12 convolutional layers to 30 convolutional layers, there is no obvious difference in performance on each dataset. Specifically, 27 convolutional layers do the best and 26 convolutional layers do the worst on cells dataset, but the difference is only about MD $0.87368 - 0.86377 = 0.0099$ and SD $0.0121 - 0.0392 = -0.0271$. The same phenomenon also appears on the other three dataset. (e.g. Lung: best 17 layers - worst 12 layers = MD $0.74102 - 0.73004 = 0.011$ and SD $0.0731 - 0.0259 = 0.0472$. Skin: best 22 layers - worst 12 layers = MD $0.52361 - 0.50892 = 0.0147$ and SD $0.0429 - 0.0418 = 0.0011$. Nuclei: best 26 layers - worst 15 layers = MD $0.90842 - 0.88192 = 0.0264$ and SD $0.1365 - 0.1784 = -0.0419$.). Theoretically, we cannot test all datasets but the experimental results reflect that 23 convolutional layers are not the best choice of U-Net. Finally, 26 convolutional layers are set to the upper limit of depth interval. This is because compared to the classical U-Net (23 convolutional layers), the parameter growth rate of 26 convolutional layers is only $\frac{31.81\times10^6-31.03\times10^6}{31.03\times10^6} \times 100\% = 2.51\%$ but that of 27 convolutional layers have reached $\frac{34.17\times10^6-31.03\times10^6}{31.03\times10^6} \times 100\% = 10.12\%$. 12 convolutional layers (U-Net-12) marked by an arrow in Fig. 3 are set to the lower limit of depth interval, which is appropriate for being the backbone. Its advantages are as following: (1) the parameters are only about 1/20 of U-Net ($\frac{1.75\times10^6}{31.03\times10^6} \times 100\% = 5.64\%$), which can support more parameters space for IT-Block or other functional modules; (2) The segmentation performance of U-Net-12 is very close to that of classical U-Net, which makes it possible to exceed classical U-Net with minor changes. The details about U-Net-12 are shown in Fig 4.

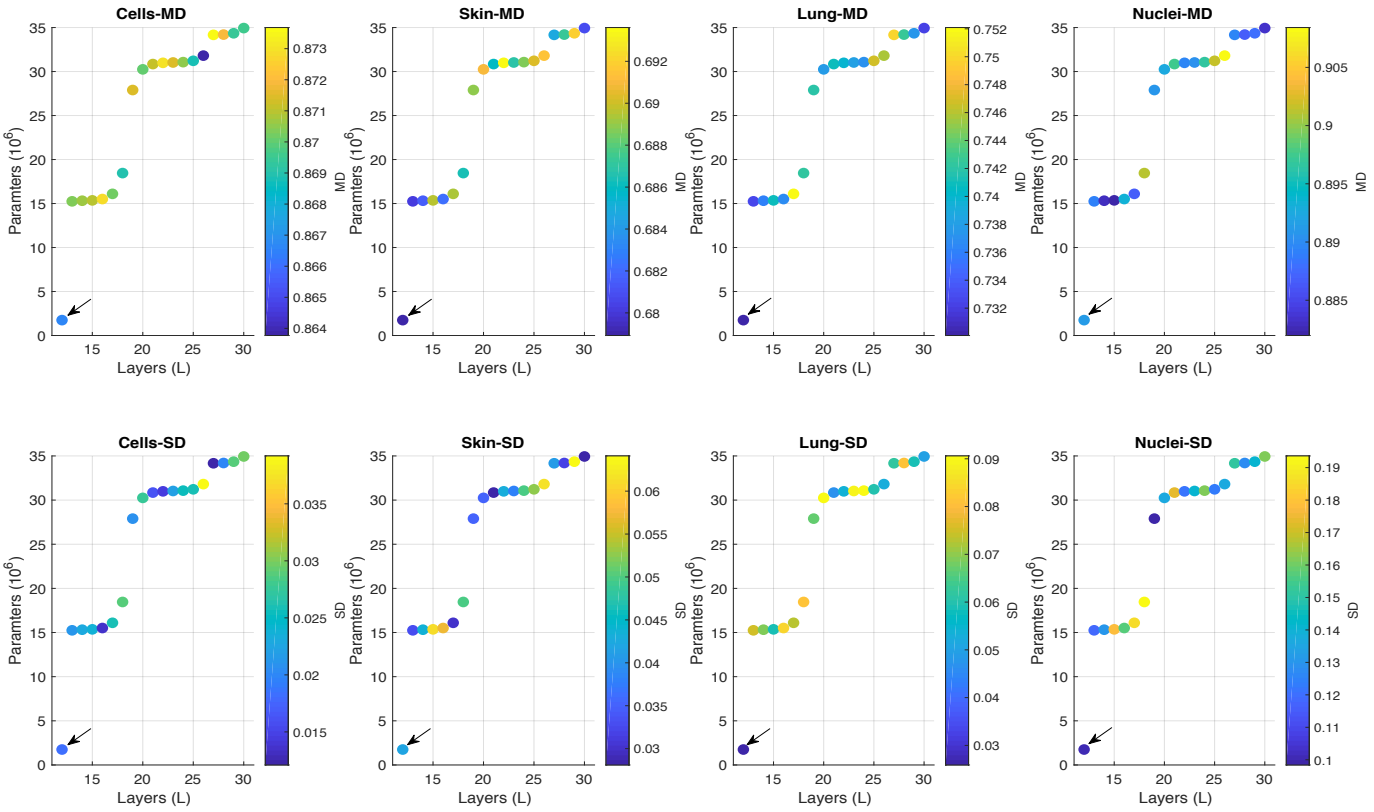

Fig. 3. SD (12) and MD (11) of variant U-Net (different depth) on four datasets including Cells, Skin, Lung, and Nuclei. The calculated dice in this part is standard dice coefficient (1). The color bar on the right side of each graph represents the value of SD and MD. We use an arrow to highlight the well performed backbone with only 12 convolutional layers. The layers we discussed only includes convolutional layers and up-convolutional layers in U-Net.

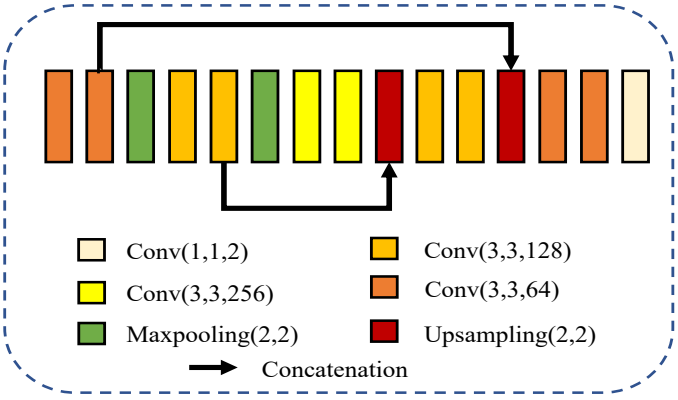

Fig. 4. The details about parameters of each convolutional layer in U-Net-12. Number 12 means that there are 12 convolutional layers including the convolutional layer and the upsampling layer (Transposed Convolution). The last convolutional layer is not counted because its parameters only occupy a small part of the entire network.

B. Optimized Dice Loss

In medical image segmentation, Dice loss [32] is one of the most popular loss functions, generated by the dice coefficient (6). However, such an extreme case where annotated target and segmented target are both small would lead to unstable training, which makes the final results not precise enough. The root of this problem is the use of division operation in the dice coefficient (1). The Division is an operation that needs to be handled carefully when the denominator is small, because the butterfly effect may occur. For example, formula $\frac{2.6812}{0.001} = 2618.2$, if the denominator turns into 0.0011, formula $\frac{2.6812}{0.0011} = 2380.2$. That is to say, when the denominator changes slightly (only changed by 0.0001), the quotient of the formula changes dramatically. Even more badly, the butterfly effect would make the network difficult to find the optimal solution because of the great changing gradient during backpropagation. To detail this, we try to write the dice coefficient in a differentiable form (2) in terms of its calculation and give its partial derivative form (3).

$$d = \frac{2ab}{a+b} \tag{2}$$

$$\frac{\partial d}{\partial a} = \frac{2b^2}{(a+b)^2} \tag{3}$$

Where d represents a differentiable form of dice coefficient. a is a predicted image and b is the ground truth. It could be seen that the denominator in (3) is squared, resulting in generating a smaller scalar (e.g. $0.1^2 = 0.01$), which increases the probability of the butterfly effect. During the backpropagation, updating parameters cannot avoid calculating partial dervatives. To illustrate that, we formally assume net_l is the last layer of the network and w_{ij} is the j^{th} parameter

of the i^{th} layer. According to the chain rule of derivation, we can obtain

$$\frac{\partial DiceLoss}{\partial w_{ij}} = \frac{\partial DiceLoss}{\partial net_l} \cdots \frac{\partial net_{i+1}}{\partial w_{ij}} \tag{4}$$

Thus, $\frac{\partial DiceLoss}{\partial w_{ij}}$ can be seen as proportional to $\frac{\partial DiceLoss}{\partial net_l}$.

$$\frac{\partial DiceLoss}{\partial w_{ij}} \propto \frac{\partial DiceLoss}{\partial net_l} \tag{5}$$

Again according to (2) and (3), parameter w_{ij} would also be affected by the butterfly effect through $\frac{\partial DiceLoss}{\partial net_l}$.

To address that, we propose LDice loss function (7) that is similar to Dice loss (6).

$$DiceLoss = 1 - Dice \tag{6}$$

$$LDiceLoss = -log(Dice) \tag{7}$$

Dice is the dice coefficient (1). The benefit of $log(\cdot)$ is to restrict the denominator by decreasing the power of numerator when calculating partial derivatives of Dice (2), so that the butterfly effect caused by small changes in the denominator is alleviated.

$$\frac{\partial log(d)}{\partial a} = \frac{b}{a(a+b)} \tag{8}$$

According to (3) and (8), now we assume $a_1 = 0.1, b_1 = 0.2, a_2 = 0.1, b_2 = 0.21$ to calculate the rate of changing gradient (9), obtaining $r_{(3)}$ is 3.149% and $r_{(8)}$ is 1.613%. This shows a certain compression on the butterfly effect.

$$r = \left| \frac{g_2 - g_1}{g_1} \right| \times 100\% \tag{9}$$

Where g_i represents the gradient calculated by (3) or (8).We compare Dice loss and LDice loss as the loss function of U-Net [7] on four datasets including cells [7], Skin, Lung, and Nuclei [16]. In Fig. 5, it can be observed that the curve of LDice loss is smoother than that of Dice loss, which makes the training stable. Much more importantly, stable training could help the network find the optimal solution easily, obtaining a lower loss. The experimental results verify that our proposed LDice loss is effective to alleviate the butterfly effect induced by Dice loss. The more implementation details can be achieved in section IV (Experiments and Results).

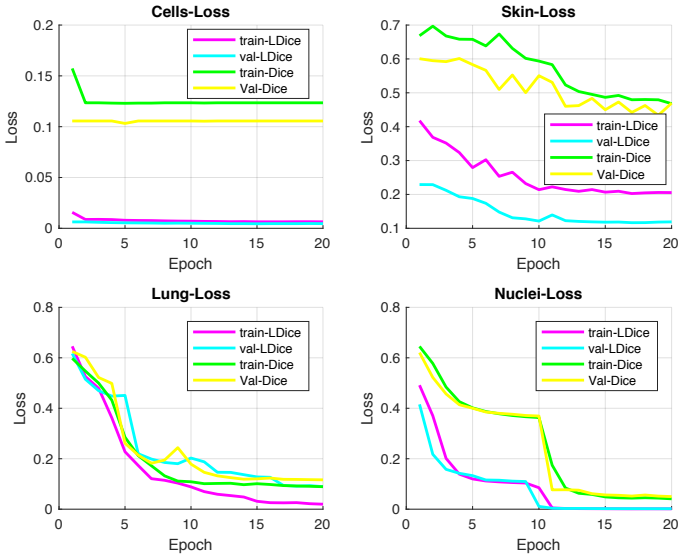

Fig. 5. The train-validation loss comparison of Dice (6) and LDice (7) on Cells, Skin, Lung, and Nuclei.

C. IT-Block

It is undeniable that U-Net excels in medical image segmentation, but also has some limitations. To sum up, the shortcomings of U-Net are mainly reflected in two aspects: (1) the fixed kernel size used in U-Net cannot capture multi-level information; (2) the way of feature fusion is only using concatenation. Furthermore, we discover that 12 convolutional layers are appropriate for this U-shape architecture after a discussion on the depth of U-Net. Specifically, the performance of U-Net-12 is not much different from U-Net but its parameters are only about 1/20 of U-Net. Our target is to add some new elements to U-Net-12 to form a better architecture. Such a design outperforms U-Net in segmentation accuracy, parameters, and running time for training.

Motivated by the shortcomings of U-Net and our target, we present a novel block, named Inverted Triangle (IT) Block because of its shape (see in Fig. 6). It consists of Dense Connection [11], Residual Connection [12], and Inception [13]. The major advantage of the proposed IT-Block is to take advantage of those three operations, aiming to help the network obtain multi-level features and reuse them comprehensively. Formally, the x_l is the output of IT-Block, and the x_{l-1} is the input of IT-Block. The relation between x_l and x_{l-1} is defined in (10).

$$x_l = D(C(I(x_{l-1}) + R(C(I(x_{l-1}))))) + x_{l-1} \tag{10}$$

Where $I(\cdot)$, $R(\cdot)$, and $D(\cdot)$ denote the non-linear calculation including Conv, ReLU [34], and BN [10] in the inception block, the residual block (Fig. 6(c)), and the dense block (Fig. 6(b)). (e.g. Inception block is at the top of Fig. 6(a).). $C(\cdot)$ represents the concatenation operation that fuses feature maps by channel. $+$ is the residual connection, adding feature maps point by point without increasing the channel. For the block we used in the experiment, 1×1 convolutional layer is attached to the end of IT-Block to reduce parameters. $D(\cdot)$ in the dense block actually includes BN-ReLU-Conv(1×1)-BN-ReLU-Conv(3×3). In general, the number of IT-Blocks could be chosen alternatively. However, guided by the target of low parameters, two IT-Blocks are a more reasonable choice. The more details about the position of the two IT-Blocks would be discussed in section IV (Experiments and Results).

IV. Experiments and Results

A. Datasets and Evaluation Metrics

There are many types of medical images such as CT, MRI, microscopy and so on. In order to test as many types of medical images as possible, we select four public medical image datasets including cells [7], Skin, Lung, and Nuclei

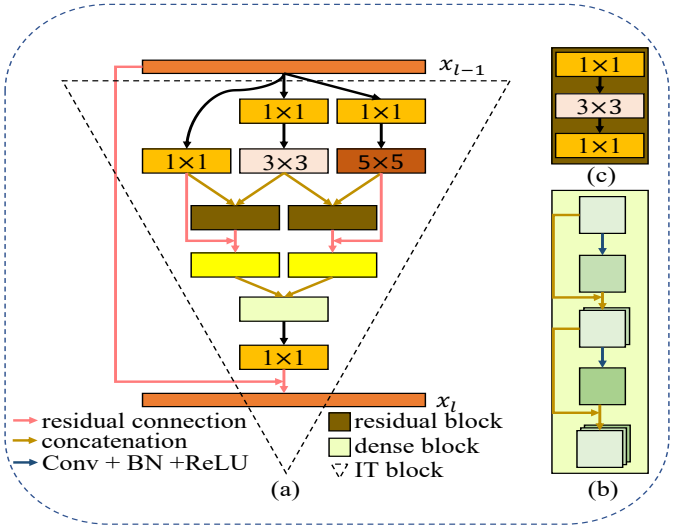

Fig. 6. The Brief description of IT-Block (a), dense block (b), and residual block (c).

TABLE I
DETAILED INFORMATION OF DATASETS

Data name	Source	Image Size	Modality
Cells	ISBI 2012	512×512	microscopy
Nuclei	Data Science bowl 2018	360×360	microscopy
Lung	Kaggle	512×512	CT
Skin Lesion	ISIC 2017	$512 \times 512 \times 3$	dermoscopy

[16]. The size, modality, and source of them are shown in Table I.

In the experiments, three metrics are used, including the number of parameters, the dice coefficient, and the running time for training. Due to the small size of each dataset like the cells dataset used in U-Net [7], we divide these small datasets into five equal parts respectively and apply a 5-fold cross-validation used in [33]. For the dice coefficient, we calculate its mean and std (standard deviation) of each dataset. The MD (Mean Dice coefficient) and SD (Std of Dice coefficient) are defined in (11) (12).

$$MD = \frac{1}{n \times k} \sum_{i=1}^{n} \sum_{j=1}^{k} Dice \tag{11}$$

$$SD = \sqrt{\frac{\sum_{i=1}^{n} (\frac{1}{k} \sum_{j=1}^{k} Dice - MD)^2}{n-1}} \tag{12}$$

where k is the number of images in one subset. n is the fold used in cross-validation. Dice is classical dice coefficient (1).

B. Implementation Details

Preprocessing: To test the adaptability of the network to small datasets, we randomly screened 30 samples from the other three datasets, keeping the same number of samples like cells (only 30 samples) [7]. For saving the memory, we simply resize each image to 256×256 as the input. Before using data augmentation, we divide each dataset into five equal parts (e.g. four parts for training and one part for validation) to prepare for running a 5-fold cross-validation. The datasets (training image and its corresponding labels) are augmented by rotating counterclockwise 90 degrees, 180 degrees, and 270 degrees.

Hyperparameters: In all experiments, the use of hyperparameters is basically the same. Each convolution layer with one stride is followed by BN [10] and ReLU [34] except the last convolutional layer, using 5 batch size and Adam optimizer [35] with the following parameters: $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e-8$. The initial learning rate of $1e-4$ is settled and the sigmoid classifier is used in the last layer. We use he-normal [36] to initialize the weights of the network. The differences are as following: (1) cross-entropy [7] is used as a loss function in rethinking the capability of U-Net and LDice loss is used in experiments about IT-Block; (2) the epochs are different (e.g. 30 epochs in rethinking the capability of U-Net and experiments about IT-Block, 20 epochs in comparison of Dice loss and LDice loss).

Experimental platform and environment: The experiments are conducted on a computer with Intel(R) Core (TM) i7-7700 CPU @ 3.60GHz, Nvidia GeForce GTX 1080 Ti, 16GB RAM, and Samsung SSD 850 EVO 500GB. The operating system is Windows 10(1801). All experiments were run under the Keras framework.

C. Results

The position of IT-Block: There are 14 optional spaces in the structure of U-Net-12 (Fig. 4). Formally, we use i and j to denote the position of two IT-Blocks (e.g. IT-Block (1,14) represents one of the IT-Blocks is located in the first space of U-Net-12 and the other one is located in the 14th space of U-Net-12). To maintain the symmetry of U-Net-12 like U-Net [7], we place the two IT-Blocks symmetrically on the up-down sampling path. Consequently, a total of 7 combinations were obtained. According to the results (seen in Table II), although the running time and parameters of IT-Block (3,12) are not the best among these combinations, there are not much different from the best scores (e.g. IT-Block (1,14) and IT-Block (2,13)). Furthermore, it almost provides better MD (11) and SD (12) than the other six combinations in segmenting cells, lung, skin, and nuclei. In the case of keeping symmetry of the network, this experiment verifies the 3rd space and the 12th space of U-Net-12 is the best choice for two IT-Blocks. Additionally, we discover that when the parameters of two combinations are the same (e.g. IT-Block (1,14) and IT-Block (2,13), IT-Block (4,11) and IT-Block (5,10)), their running time, MD (11), and SD (12) are similar. That is to say, it is possible that the position of IT-Block is not the key to affect the performance of U-Net-12. The changing parameters caused by inserting IT-Blocks at different positions is the main factor.

Comparison with U-Net: In this part, we use the same loss function (LDice loss (7)) for training, aiming to state that our framework (U-Net-12+IT-Block (3,12)) achieves significantly better results than that of U-Net. This is not only because of

TABLE II
THE COMPARISON OF RUNNING TIME, AVERAGE DICE COEFFICIENT AND ITS STANDARD DEVIATION FOR 5-FOLD CROSS-VALIDATION IMPLEMENTED BY U-NET AND U-NET-12 WITH IT-BLOCK.

Models arrow: normal	Parameters arrow: excellent	Time	Cells		Skin		Lung		Nuclei	
			mean	std	mean	std	mean	std	mean	std
U-Net [7]	31.0317×10^6	156s	0.8829	0.0201	0.6894	0.0361	0.7452	0.0824	0.8935	0.1352
U-Net-12 + IT-Block(1,14)	2.3845×10^6	132s	0.8887	0.0241	0.6987	0.0342	0.7527	0.0903	0.9124	0.1361
U-Net-12 + IT-Block(2,13)	2.3845×10^6	132s	0.8892	0.0237	0.6993	0.0346	0.7522	0.0895	0.9129	0.1357
U-Net-12 + IT-Block(3,12)	$\mathbf{2.4052 \times 10^6}\uparrow$	**134s**↑	**0.9086**↑	**0.0172**↓	**0.7485**↑	**0.0364**↑	**0.7861**↑	**0.0692**↓	**0.9317**↑	**0.1238**↓
U-Net-12 + IT-Block(4,11)	2.4259×10^6	136s	0.8873	0.0309	0.6915	0.0389	0.7481	0.0831	0.8949	0.1423
U-Net-12 + IT-Block(5,10)	2.4259×10^6	136s	0.8878	0.0295	0.6921	0.0381	0.7484	0.0827	0.8946	0.1427
U-Net-12 + IT-Block(6,9)	2.4675×10^6	140s	0.8923	0.0454	0.7295	0.0435	0.7671	0.0736	0.9089	0.1358
U-Net-12 + IT-Block(7,8)	2.5089×10^6	142s	0.8979	0.0231	0.7341	0.0292	0.7725	0.0783	0.9147	0.1821

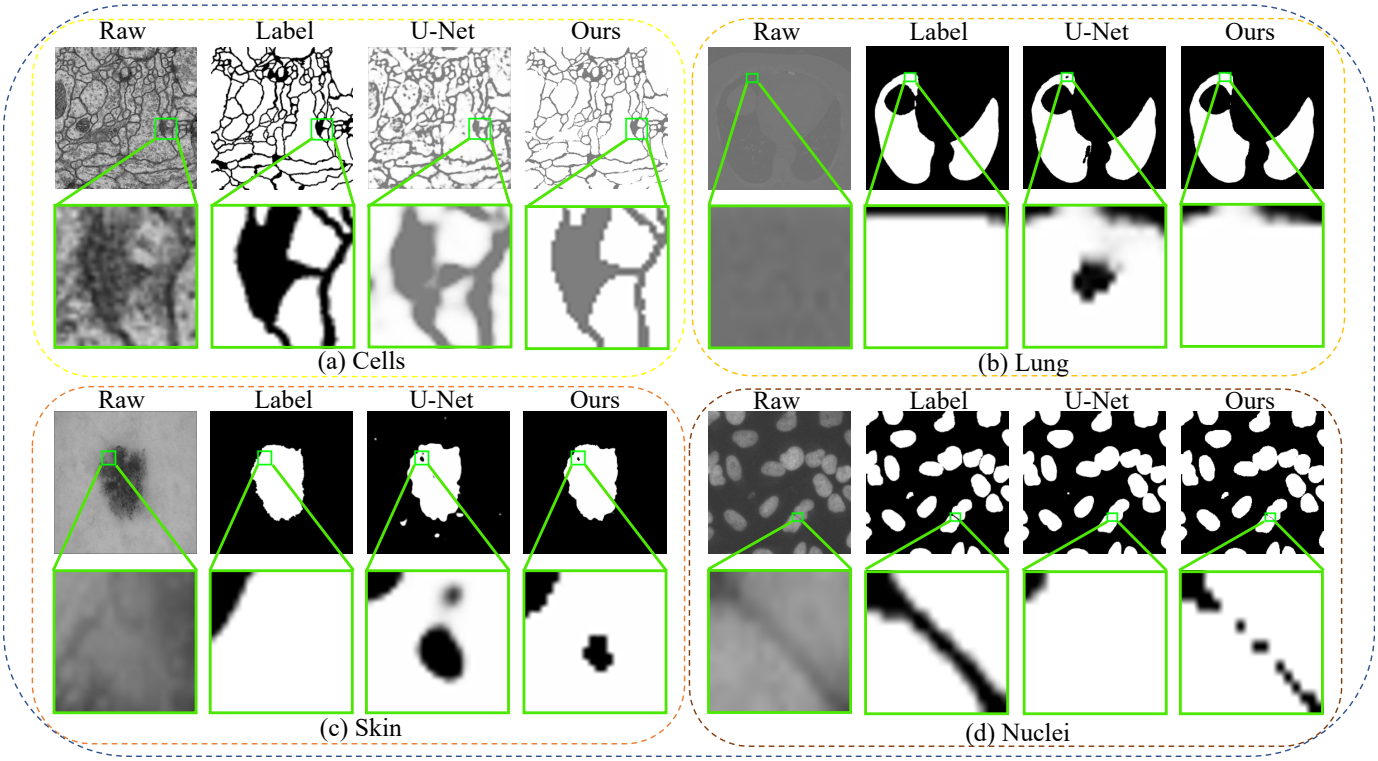

Fig. 7. Some results processed by U-Net and U-Net-12 with IT-Block (3,12) (ours). We use the green rectangle to mark a slight part of each image and magnify the part inside the green rectangle to the same size as its corresponding image.

the optimized Dice loss. In terms of Table II, it could be seen that there is obvious improvement in MD (11), parameters, and running time when compared to U-Net. Although U-Net-12+IT-Block (3,12) gets a worse Skin SD (12), it is close to the Skin SD of U-Net, and this is acceptable in view of its comprehensive promotion on running time, accuracy, and parameters. As the slight region magnified in Fig. 7, it is clearly seen that the tissues of cells, lung, skin, and nuclei are more likely to be misclassified by U-Net. More specifically, there is much different from the mask generated by U-Net and its corresponding label in the slight part of tissues, whereas the mask generated by U-Net-12+IT-Block (3,12) is closer to the label. This result indicates that the fixed kernel (3×3) used in U-Net cannot capture multi-level information. It is easy to lose smaller information, resulting in poor segmentation of slight tissues. This result also proves that the proposed IT-Block could indeed help the network obtain multi-level features and reuse them comprehensively, which makes the network have better capabilities for processing slight objects in the medical image.

V. CONCLUSION

In this paper, we propose a new and simple block (IT-Block) that can be embedded in U-Net, namely the use of Dense Connection, Residual Connection, and Inception to help the network obtain multi-level features and enhance the reuse of them significantly. Furthermore, we introduce a novel loss function (LDice loss) that could alleviate the butterfly effect

caused by Dice loss during backpropagation. Even more interestingly, we discover that U-Net with only 12 convolutional layers (U-Net-12) is not much different from classical U-Net (23 convolutional layers) in segmentation accuracy but keeps parameters low. Based on that, we use U-Net-12 as a backbone, assembling IT-Block and LDice loss to it. Such a new framework outperforms U-Net in four different image segmentation tasks. Future works will aim at how to extend the IT-Block into a whole network that can further optimize its performance on medical image segmentation.

REFERENCES

[1] D. Kwon, J. Ahn, J. Kim, I. Choi, S. Jeong, Y. Lee, J. Park, and M. Lee, "Siamese u-net with healthy template for accurate segmentation of intracranial hemorrhage," in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, October 2019, pp. 848–855.

[2] S. Kohl, B. Romera-Paredes, C. Meyer, J. De Fauw, J. R. Ledsam, K. Maier-Hein, S. M. A. Eslami, R. D. Jimenez, and O. Ronneberger, "A probabilistic u-net for segmentation of ambiguous images," in *Advances in Neural Information Processing Systems 31 (NIPS)*, December 2018, pp. 6965–6975.

[3] W. Wang, J. Chen, J. Zhao, Y. Chi, X. Xie, L. Zhang, and X. Hua, "Automated segmentation of pulmonary lobes using coordination-guided deep neural networks," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, April 2019, pp. 1353–1357.

[4] Y. He, A. Carass, Y. Liu, B. M. Jedynak, S. D. Solomon, S. Saidha, P. A. Calabresi, and J. L. Prince, "Fully convolutional boundary regression for retina oct segmentation," in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, October 2019, pp. 120–128.

[5] A. V. Dalca, J. Guttag, and M. R. Sabuncu, "Anatomical priors in convolutional networks for unsupervised biomedical segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[6] K. C. L. Wong and M. Moradi, "Segnas3d: Network architecture search with derivative-free global optimization for 3d image segmentation," in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, October 2019, pp. 393–401.

[7] O. Ronneberger, P. Fisher, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, October 2015, pp. 234–241.

[8] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, "The importance of skip connections in biomedical image segmentation," in *Deep Learning and Data Labeling for Medical Applications (DLMIA)*, October 2016.

[9] S. Mehta, E. Mercan, J. Bartlett, D. Weaver, J. G. Elmore, and L. Shapiro, "Y-net: Joint segmentation and classification for diagnosis of breast biopsy images," in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, September 2018, pp. 893–901.

[10] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on International Conference on Machine Learning (ICML)*, July 2015, pp. 448–456.

[11] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 2261–2269.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[13] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[14] X. Li, H. Chen, X. Qi, Q. Dou, C. Fu, and P. Heng, "H-denseunet: Hybrid densely connected unet for liver and tumor segmentation from ct volumes," *IEEE Transactions on Medical Imaging*, vol. 37, no. 12, pp. 2663–2674, 2018.

[15] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, 2018.

[16] Z. zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep Learning and Data Labeling for Medical Applications (DLMIA)*, September 2018, pp. 3–11.

[17] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[18] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in Neural Information Processing Systems 25 (NIPS)*, December 2012, pp. 2843–2851.

[19] X. Zhou, T. Ito, R. Takayama, S. Wang, T. Hara, and H. Fujita, "Three-dimensional ct image segmentation by combining 2d fully convolutional network with 3d majority voting," in *Deep Learning and Data Labeling for Medical Applications (DLMIA)*, October 2016, pp. 111–120.

[20] M. Shakeri, S. Tsogkas, E. Ferrante, S. Lippe, S. Kadoury, N. Paragios, and I. Kokkinos, "Sub-cortical brain structure segmentation using f-cnn's," in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, 2016, pp. 269–272.

[21] W. Chen, Y. Zhang, J. He, Y. Qiao, Y. Chen, H. Shi, E. X. Wu, and X. Tang, "Prostate segmentation using 2d bridged u-net," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019.

[22] N. Ibtehaz and M. S. Rahman, "Multiresunet : Rethinking the u-net architecture for multimodal biomedical image segmentation," *Neural Networks*, vol. 121, pp. 74–87, 2020.

[23] Z. Tu and X. Bai, "Auto-context and its application to high-level vision tasks and 3d brain image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1744–1757, 2010.

[24] W. Wang, K. Yu, J. Hugonot, P. Fua, and M. Salzmann, "Recurrent u-net for resource-constrained segmentation," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[25] O. Cicek, A. Abdulkadir, S. S.Lienkamp, and O. Ronneberger, "3d u-net: Learning dense volumetric segmentation from sparse annotation," in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, October 2016, pp. 424–432.

[26] F. Dubost, G. Bortsova, H. Adams, A. Ikram, W. Niessen, M. Vernooij, and M. Bruijne, "Gp-unet: Lesion detection from weak labels with a 3d regression network," in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, September 2017, pp. 214–221.

[27] C. Huang, H. Han, Q. Yao, S. Zhu, and S. K. Zhou, "3d u2net: A 3d universal u-net for multi-domain medical image segmentation," in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, October 2019, pp. 291–299.

[28] R. Brügger, C. F. Baumgartner, and E. Konukoglu, "A partially reversible u-net for memory-efficient volumetric image segmentation," in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, October 2019, pp. 429–437.

[29] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *International Conference on Learning Representations (ICLR)*, May 2016.

[30] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2018, pp. 1451–1460.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, May 2015.

[32] F. Milletari, N. Navab, and S. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *International Conference on 3D Vision*, October 2016, pp. 565–571.

[33] A. Paul and D. P. Mukherjee, "Mitosis detection for invasive breast cancer grading in histopathological images," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4041–4054, 2015.

[34] Y. B. X. Glorot, A. Bordes, "Deep sparse rectifier neural networks," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, April 2011, pp. 315–323.

[35] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, May 2015.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Decemeber 2015, pp. 1026–1034.