Prostate Segmentation using 2D Bridged U-net

Wanli Chen[1,+], Yue Zhang[1,3,+], Junjun He[2], Yu Qiao[2,*], Yifan Chen[1,4,*],Hongjian Shi[1,*], Xiaoying Tang[1,*]

[1] Southern University of Science and Technology
[2] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[3] The University of Hong Kong
[4] The University of Waikato
[+] Equal contribution
[*] Corresponding authors (yu.qiao@siat.ac.cn, yifan.chen@waikato.ac.nz, shihj@sustc.edu.cn, tangxy@sustc.edu.cn)

Abstract. In this paper, we focus on three problems in deep learning based medical image segmentation. Firstly, U-net, as a popular model for medical image segmentation, is difficult to train when convolutional layers increase even though a deeper network usually has a better generalization ability because of more learnable parameters. Secondly, the exponential ReLU (ELU), as an alternative of ReLU, is not much different from ReLU when the network of interest gets deep. Thirdly, the Dice loss, as one of the pervasive loss functions for medical image segmentation, is not effective when the prediction is close to ground truth and will cause oscillation during training. To address the aforementioned three problems, we propose and validate a deeper network that can fit medical image datasets that are usually small in the sample size. Meanwhile, we propose a new loss function to accelerate the learning process and a combination of different activation functions to improve the network performance. Our experimental results suggest that our network is comparable or superior to state-of-the-art methods.

1 Introduction

1.1 Network for Image Segmentation

Convolutional neural network shows a great advantage over traditional methods in computer vision. Recently, fully convolutional network (FCN) [1] has become the main framework for image segmentation task. Specifically, for medical image segmentation, a popular FCN is U-net [2]. U-net has an encoder-decoder structure with concatenation being used to merge features. It is widely used in medical image segmentation [3] [4] because of its efficiency and the adaptivity for small dataset. The main drawback of U-net is that it is difficult to go very deep. To fully exploit the utility of U-net but go deeper, a stacked U-net has been proposed. However, such network is likely to be trapped into a sub-optimal solution because a stacked U-net is more complicated than a single U-net. As such, a stacked U-net is usually employed when there is a pre-train model [5]

[6]or fed with a large number training data (more than 10K) [7]. There is a work concatenating two U-net by two loss [19], but they didn't consider the information sharing in two U-net.In this paper, we propose a bridging architecture between two U-nets. Specially, we connect each decoder layer of the first U-net with the corresponding encoder layer of the second U-net, which directly inputs the features of the previous layers into the latter layers. This process reduces the training cost and exhibits a better performance than a single U-net. By using network bridging, a stacked U-net can deal with small datasets and be used in medical image segmentation without a pre-train model.

1.2 Feature Fusion Methods

To bridge two U-nets, an appropriate method is needed. Network bridging can also be viewed as feature fusion. The main fusion methods can be divided into two categories: addition and concatenation. Addition is an intuitive way. It directly adds features together. This fusion method has been widely used in many computer vision tasks such as ResNet [8] based classification, Feature Pyramid Network (FPN) [9] based detection and FCN based image segmentation. This method can be viewed as highway gradient transfer [10], which will accelerate gradient propagation. Addition will change the distribution of weights, which is pernicious for network initialization. This issue can be alleviated when using concatenation for feature fusion. Representative networks include DenseNet [11] and U-net. We will have a further discussion on concatenation and addition in part 3 of this paper and show that concatenation is superior for our network bridging.

1.3 Activation Functions

To make our network performs better, different activation functions are applied in our network. Activation function is an important component of a neural network. The activation layers adds non-linearity to the weights so that the network can deal with more complex tasks. Previously, sigmoid [12] has been used as the activation function. However, sigmoid will saturate during training. Then ReLU [13] has been proposed to solve the saturation issue. When using ReLU as the activation function, the learning rate should be carefully adjusted because ReLU gets saturated in negative axis, and a big learning rate will "kill" some neurons. To address this efficiency, Exponential ReLU (ELU) [14] has been proposed. ELU does not get saturated in the negative axis immediately. However, saturation still happens when the network gets deeper. When ELU gets saturated, it is no different from ReLU. As such, we can replace some ELU layers with ReLU. Because ELU is not saturated when the negative axis is 0, the replacement could be viewed as "reset" ELU, which will re-activate the subsequent saturated ELU neuron. In this paper, we use ELU and ReLU simultaneously. We also designed a new loss function for medical image segmentation.

To sum up, we have three contributions. Firstly, we propose a new network structure to accelerate the learning process of stacked U-net. Additionally, we investigate the performance of different feature fusion methods. Secondly, we explore the utility of using ELU and ReLU as the activation functions to reach a superior performance. We also extend the result into the general image classification task. Thirdly, we design a new loss function.

2 Related Work

2.1 U-net

The U-net consists of a down-convolutional part and up-convolutional part. The down-convolutional part aims at extracting features for classifying each voxel into one or zero. It consists of repeated application of two 3×3 convolutions. At each downsampling step the number of feature channels is doubled. And the up-convolutional part aims at locating regions of interest (ROI) more precisely. Every step in up-convolutional part consists of an upsampling of the feature map followed by a 2×2 convolution that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3×3 convolutions. Max pooling and ReLU activation was used for the convolution block in U-net.

2.2 PROMISE12

Determination of prostate volume (PV) can help detect pathologic stage of diseases, such prostate cancer. What's more, the accurate prostate specific antigen (PSA) is dependent on the quality of the PV. The accuracy and variability of PV determinations pose limitations to its usefulness in clinical practice. It is also an essential part in clinical to get the the size, shape, and location of the prostate relative to adjacent organs. Recently, this kind of information can be obtained by MRI using the high spatial resolution and soft-tissue contrast. This, combined with the potential of MRI to localize and grade prostate cancer, has led to a rapid increase in its adoption and increasing research interest in its use for this application. Consequently, there is a real clinical and research need for the accurate robust, automatic prostate segmentation methods used as an preprocessing procedure for computer-aided detection and diagnostic algorithms, as well as a number of multi-modality image registration algorithms.

Prostate MR Image Segmentation challenge 2012 (PRPMISE12) was held to compare segmentation algorithms for MRI of the prostate during MICCAI 2012. After MICCAI 2012, the organizer are still receiving and uploading submission [15]. For training data, the MRI images and ground truth segmentation results are all available, shown in Fig. 1. However, the ground truth segmentation results are only available for organizers, who will evaluate the results of participants. In this paper, we use PROMISE12 dataset as our training and validation dataset and we also submitted our testing results to organizers.

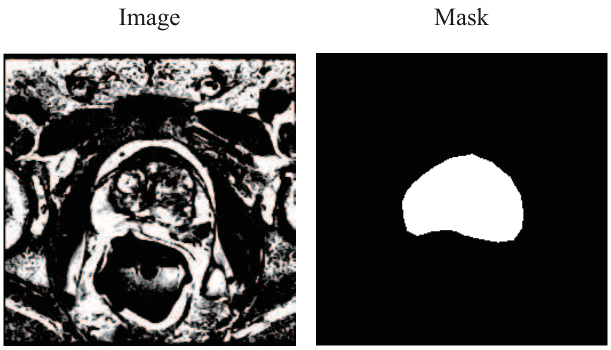

Fig. 1. A sample in PROMISE12. Mask means the ground truth segmentation result, which is only available in training data.

3 Model Setup

3.1 Bridged U-net

Our network is based on U-net, which is a classical encoder-decoder net in medical image application. Based on U-net, a stacked U-net is proposed. The stacked U-net improves network performance by using the first U-net to find a coarse feature and use the second U-net to obtain a fine result. The stacked U-net is, however, not useful for medical image segmentation. It is hard to reach convergence and usually dive into a sub-optimal solution because the increasing complexity of network. To overcome the issue, we propose a network bridging method. Different from the previous stacked U-net which acquires large number training data, bridging two U-nets can reduce the training cost and makes the network fit for medical application where the training data are usually not sufficient. This is because bridging two U-nets can fully use different features in multi levels, which will accelerate the convergence of neural network. Our network structure is shown on Fig. 2. The gray block represents a ELU cluster (2 conv-BN-ELU blocks), and the yellow block represents a ReLU cluster (2 conv-BN-ReLU blocks). The dotted lines represents network bridging. The red lines represents skip connections.

Network Bridging In order to bridge features in two U-nets, we can use addition or concatenation as our bridging method. Both methods are widely used in computer vision. For example, FPN used addition to combine low level features with higher level semantic features, while DenseNet uses concatenation for features combination. To merge features of different levels, we argue that concatenation is more effective. We proved that in the perspective of weights initialization.

In weights initialization, to make sure information flow, we need to keep the variance of input equal to output, $Var\left[x\right] = Var\left[y\right]$, as shown on Fig. 3.

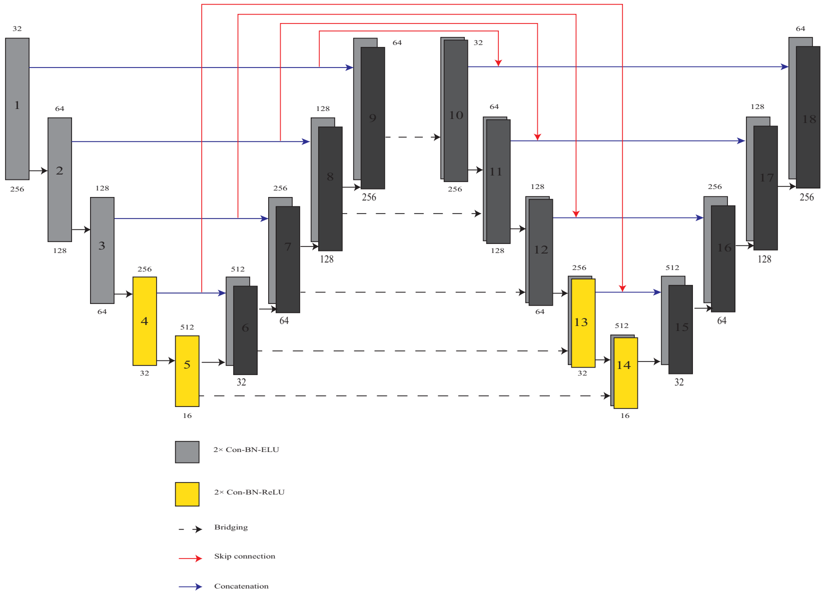

Fig. 2. Bridged U-net architecture. The number above each block represents the number of feature channels. The number inside each block represents the sequence number. The number below each block means the image size.

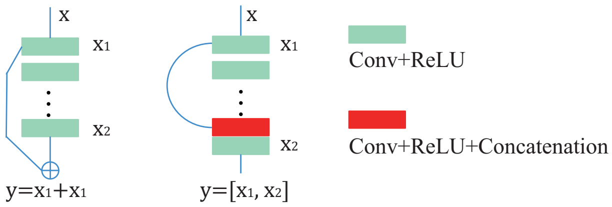

Fig. 3. The comparison of addition and concatenation

Assuming all convolution layers are initialized with gaussian normal filler and using ReLU as activation function and all parameters are independently and identically distributed (IID). According to the fact that addition of two normal distribution is another normal distribution with $\sigma = \sigma_1 + \sigma_2$ and concatenation of two IID normal distribution is another distribution with $\sigma = \sigma_1 = \sigma_2$. Then we obtain:

Addition:

$$Var\left[y\right] = Var\left[x_1 + x_2\right], \tag{1}$$

$$Var\left[y\right] = 2Var\left[x\right], \tag{2}$$

Concatenation:

$$Var\left[y\right] = Var\left[x_1, x_2\right], \tag{3}$$

$$Var\left[y\right] = Var\left[x\right], \tag{4}$$

Table 1. Influence of network bridging and skip connection. vDSC is the abbreviation of volumetric Dice Similarity Coefficient.

Method	Bridging method	Skip connection	Mean vDSC [%]
U-net	None	None	86.73
Stacked U	None	None	85.57
Stacked U	Addition	None	86.99
Stacked U	Concatenation	None	87.85
Stacked U	Concatenation	Concatenation	86.02
Bridged U-net	Concatenation	Addition	**88.12**

Therefore, using concatenation can guarantee the information flow. Thus, we deem concatenation is better for feature fusion. This result is also proved in our network by ablation experiments shown on Table 1. In this table, we only use ELU as our activation function. The results shows that using concatenation and skip connection can improve network performance.

Skip connection Skip connection is an important part for medical image segmentation, which is helpful to improve network performance. In Bridged U-net, we use addition for skip connection. Although concatenation is more effective in the perspective of weight initialization, addition has it own advantages. As it shown in Fig. 2, we connect the two U-nets in their concatenation stage. The reason why not using concatenation is that it will cause redundancy. If using concatenation as our skip connection method, the decoder part of the second U-net have to learn more parameters than the first U-net, which aggravate the learning burden of the second U-net and the network will not converge.

3.2 Activation Function: The Combination of ReLU and ELU

In artificial neural networks, the activation function play an important role. Rectifier liner unit (ReLU) is the most popular activation function for deep neural network [16]. Exponential liner unit (ELU) replace the negative part in ReLU with exponential function, which is helpful to make the average of output close to zero [14]. In our network, we initially use ELU with all layers.

Both ReLU and ELU are widely used in segmentation task. In this work, we find the combination of ReLU and ELU can improve the segmentation performance. Neural networks usually suffer low coverage rate because of vanishing gradient, especially for deep network. ELU provides a buffer in negative axis so that it will not saturate immediately. However, ELU still suffers the saturation problem when network gets deeper.

When ELU saturated to negative values, there will be no difference between ELU and ReLU. Therefore, although ELU is used, the performance of our network is close to the network that only use ReLU.

To overcome this issue, we have proposed a method using ELU and ReLU simultaneously. Because ELU will be saturated when the network going deeper and has the same effect as ReLU, we can simply replace some ELU layers with ReLU. This replacement will not influence the function of these specific layers since ELU is the same as ReLU on that condition. However, this replacement will affect the following layers. The saturated negative values in the following ELU layers will be reset to 0 because of ReLU, which means these following layers are not saturated anymore. We replaced the activation function of 7^{th} - 10^{th} and 25^{th} - 28^{th} convolution layers with ReLU (cluster 4, 5, 13, 14 shown on Fig. 2). The results in shown on Table 2.

Table 2. Network performance using different activation functions. ELU/ReLU only means using ELU/ReLU in all layers. Cluster 1 means replacing the activation function in block 3, 7, 12, 16 shown on Fig. 2 with ReLU. Cluster 2 means replacing the activation function in block 5, 9, 10, 14 shown on Fig. 2 with ReLU. Cluster 3 means replacing the activation function in block 4, 5, 13, 14 shown on Fig. 2 with ReLU. The result shows that it is unwise to add ReLU layer frequently (Cluster 1). Two ReLU blocks should have a relative large interval (Cluster 3).

ELU only	ReLU only	Cluster 1	Cluster 2	Cluster 3	Mean vDSC [%]
Yes					88.12
	Yes				88.07
		Yes			87.56
			Yes		88.10
				Yes	**89.10**

However, it is unwise to add ReLU layers frequently. Two ReLU blocks should have a relative large interval. This is because the ReLU layer should only be added when ELU is going saturated. If adding ReLU frequently, it is hard to guarantee the saturation of ELU. In our network, we gap the two ReLU clusters with 18 convolution layers (9 ELU blocks), which is shown on Fig. 2. Since image segmentation is a pixel-wise image classification task. We want to find whether this method is useful in traditional image classification task.

Then we use CIFAR-100 [17] as our training-validation dataset and VGG-16 [18] as our model. In the training phase, the learning rate is initially set as 0.1 and decay 2 times after 20 epochs. We choose 128 as batch size and 10^{-6} as weight decay. After 250 epochs training, the result is shown on Table 3. In this

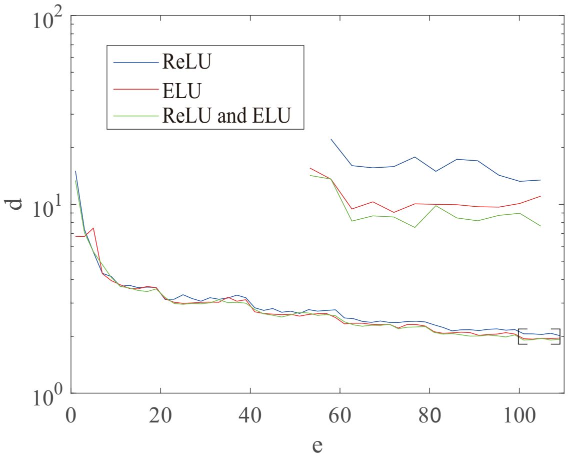

Fig. 4. Validation loss of VGG-16 in CIFAR-100 using different activation functions.

experiment, we also use ELU for all layers but replace the activation function of 4^{th} and 5^{th} convolution layers with ReLU. The results shows that the replacement can improve the network performance. The validation loss curve is shown on Fig. 4, from which we can find that the ELU shows better performance than ReLU initially. Additionally, the performance of ELU and ReLU combination is better than ReLU but worse than ELU on the beginning. The reason is that ELU is not saturated at the beginning. However, with the growth of epochs, the ELU-ReLU combination starts chasing and shows the lowest loss among them. The reason is that ELU starts saturating with the growth of epochs, but ReLU can reset the saturated ELU to 0 so that the following ELU layers are not saturated anymore.

Table 3. Accuracy performance of VGG-16 using different activation functions on CIFAR-100 dataset.

Model	ReLU only	ELU only	ELU and ReLU	Accuracy
VGG-16	Yes			0.7052
		Yes		0.7163
			Yes	**0.7201**

3.3 Cos-Dice Loss Function

Dice loss, as the most popular loss function for medical image segmentation, uses dice similarity coefficient (DSC) to generate training loss. DSC is a statistic used for comparing the similarity of two sets. It is calculated as this:

$$DSC(GS, SEG) = \frac{2\left|GS \cap SEG\right|}{\left|GS\right| + \left|SEG\right|}, \tag{5}$$

where GS represents the gold standard segmentation of a prostate region, SEG represents the corresponding automatic segmentation,and $|GS \cap SEG|$ refers to the overlap region. $|\cdot|$ represents the sum of the entries of matrix. The dice loss is defined as

$$L_{Dice} = 1 - DSC, \tag{6}$$

In medical image segmentation, dice loss is more effective than other loss functions that used in semantic segmentation. Because the number of positives and negatives are highly unbalanced in the task of medical image segmentation. However, the dice loss has its own limitation.

To illustrate that, we have investigated back propagation function. Assume z_j^l is the j^{th} input of l^{th} Layer, a_j^l is the j^{th} output of l^{th} layer, σ is the activation function. Then we obtain

$$a_j^l = \sigma\left(z_j^l\right), \tag{7}$$

$$z_j^l = \sum_k w_{jk}^l a_k^{l-1} + b_j^l, \tag{8}$$

We now focus on the last layer. Suppose δ_j^l is the error and L is the loss function, then we obtain

$$\delta_j^l = \frac{\partial L}{\partial z_j^l} = \frac{\partial L}{\partial a_j^l} \cdot \frac{\partial a_j^l}{\partial z_j^l}, \tag{9}$$

Thus,

$$\delta_j^l \propto \frac{\partial L}{\partial a_j^l} \tag{10}$$

It can be observed that the error is proportional to the partial derivative of loss function to output. Then we can plot the Loss-Intersection graph for dice loss shown in Fig. 5 using blue line. The intersection percent can be regard as the ratio of a_j^l in last layer to ground truth.

According to the equation (9), the back propagated error could be calculated as the gradient of Loss-Intersection curve. According to Fig. 5, we find that gradient of dice loss is not varied. In other words, when error back propagated, there is no much difference between the intersection percentage of 20% and the intersection percentage of 70% considering gradient. This deficiency will cause oscillation when the learning rate decrease. To solve the issue, we need to design a loss function that has larger penalty when the intersection area is small and smaller penalty when the intersection area is big.

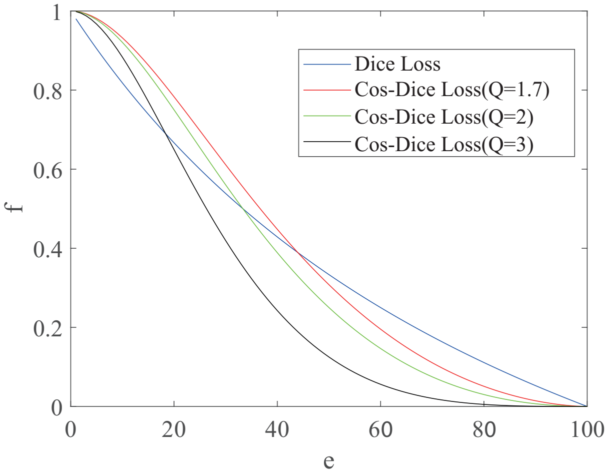

Fig. 5. Loss-Intersection graph of dice loss and cos-dice loss with different factor Q

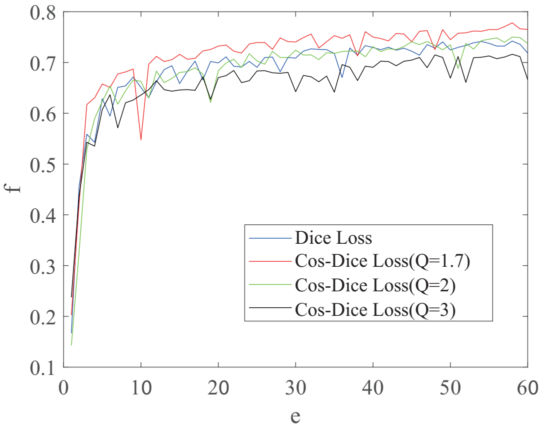

Fig. 6. The performance of dice loss and cos-dice loss with different factor Q

We propose Cos-Dice Loss Function:

$$L_{CosDice} = \cos^{Q}\left(\frac{\pi}{2} \cdot DSC\right), Q > 1. \tag{11}$$

Where Q is an adjustable number. As it shown in Fig. 5, the cos-dice loss is smoother than dice loss when the intersection percentage is large and rougher than dice loss when the intersection percentage is small.

By adding cos-dice loss, we have a more stable result with better performance. Table 4 shows that we have 0.46% gain in performance and the model becomes more stable. Actually, the principle of cosine transform is adding a weight into original dice loss:

$$\frac{\partial L_{CosDice}}{\partial a_j^j} = -Q\cos^{Q-1}\left(\frac{\pi}{2}DSC\right)\cdot\sin^{Q-1}\left(\frac{\pi}{2}DSC\right)\cdot\frac{\pi}{2}DSC' = w\cdot\frac{\partial L_{Dice}}{\partial a_j^j}, \tag{12}$$

where $w = -Q\cos^{Q-1}\left(\frac{\pi}{2}DSC\right)\cdot\sin^{Q-1}\left(\frac{\pi}{2}DSC\right)\cdot\frac{\pi}{2}$. From equation (12) we can see that the back propagated error calculated by cos-dice loss is similar to original dice loss, which makes cos-dice loss maintain the advantages of dice loss. Additionally, this weight is easy to modify by adjusting Q. A bigger Q leads to a smoother loss. However, the performance of network will decrease when Q is too big, as it shown on Table 4. Therefore, Q should be carefully adjusted to obtain the optimal result.

3.4 Implementation Details

Pre-processing PROMISE12 challenge provides 50 training datasets, each dataset contains one 3D prostate MRI image that composed of several 2D slices. We choose 45 datasets for training and 5 datasets for validation. The validation dataset number is 5, 15, 25, 35, 45. We simply resize every slice to 256 × 256 as our pre-processing method. The data augmentation was applied by random flipping, rotation from -10° to 10° to generate more data. The original training set contains 1250 slices. We obtained 5000 images (still a relatively small number for stacked U-net) after data augmentation.

Implementation The proposed method was implemented in Python language, using Keras with Tensorflow backend. All experiments are conducted on a Linux machine running Ubuntu 16.04 with 32 GB RAM memory. Bridged U-net is trained using two GTX 1080 Ti GPUs. We use Adam optimizer with initial learning rate 0.001 and 24 batch size for training.

4 Results

4.1 Validation Results

Table 4. Performance of cos-dice loss with different Q.

Loss	Mean vDSC [%]
Dice Loss	89.10
Cos-Dice Loss $Q = 1.7$	**89.56**
Cos-Dice Loss $Q = 2$	88.77
Cos-Dice Loss $Q = 3$	87.79

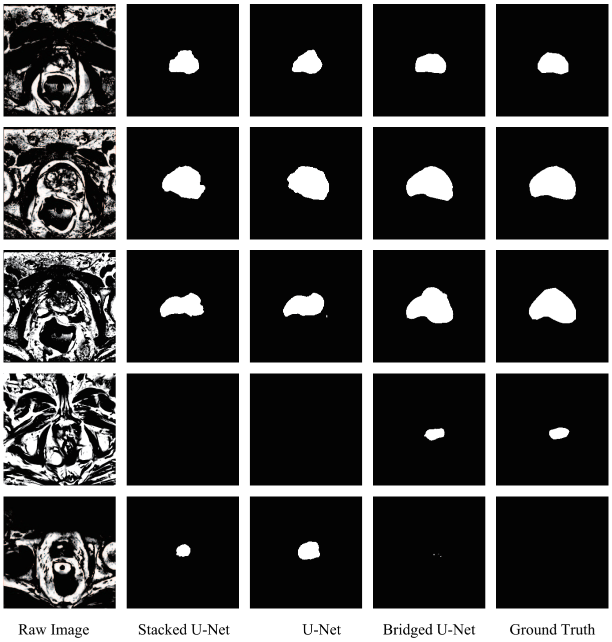

Fig. 7. Segmentation results. From left to right are raw image, the segmentation results of U-net, the segmentation results of a stacked U-net, the segmentation results of Bridged U-net, ground truth respectively

Qualitative Comparison To intuitively compare the proposed method with the U-net and stacked U-net, the segmentation results of some representative and challenging samples are shown in Fig.7. It can be observed that these prostate images have fuzzy boundaries and the pixel intensity distributions are inhomogeneous both inside and outside of the prostate. Additionally, both prostate and non-prostate regions have similar contrast and intensity distributions. All of these phenomenons make the segmentation task difficult.

As shown in the second column and third column in Fig.7, both stacked-U-net and U-net failed to obtain satisfactory result, though the model could detect part of prostate. The results of Bridged U-net are shown in the fourth column of Fig. 7. The fuzzy boundaries are well detected by our proposed method, Bridged U-net. Besides, the segmentation boundary are more continuous and smooth than the competing method. It can also be observed that the Bridged U-net can reduce false negative and false positive rate according to the fourth and fifth rows in Fig. 7.

Quantitative Comparison The statistical results of the three methods are shown in Table 5. We use six ways to evaluate the results and all of parameters show our proposed Bridged U-net performs better than stacked U-net and

Table 5. Quantitative comparison between the proposed method with other methods. Abbreviation: (a) vDSC: volumetric Dice Similarity Coefficient, (b) STD: Standard deviation, (c) HD: Hausdorff Distance, (d) ABD: Average Boundary Distance, (e) RAVD: Relative Absolute Volume Difference. ↑ means the higher value is better. ↓ means the lower value is better.

parameter	Stacked U	U-net	Wnet
Mean vDSC[%] ↑	85.57	86.73	**89.56**
Median vDSC[%] ↑	86.54	86.97	**90.33**
STD vDSC [%] ↓	6.32	5.02	**3.01**
Mean HD [mm] ↓	14.44	11.06	**9.20**
ABD[mm] ↓	1.494	1.699	**1.051**
Mean RAVD [mm] ↓	19.65	18.27	**11.32**

U-net. From the first rows and second row we can see, the average and median vDSC values of our method are highest. Besides, the vDSC standard deviation of Bridged U-net is lowest, demonstrating our method is stable. Considering Hausdorff distance (HD) , average boundary distance (ABD) and relative absolute volume difference (RAVD) shown in last three rows in Table 5, Bridged U-net suffer lowest value compared with other methods, which means Bridged U-net perform best in these parameters. It can be proved that the proposed method obtains significant improvement on the prostate segmentation compare with stacked U-net and U-net.

4.2 Testing Results

Table 6. Quantitative comparison between the proposed method with other methods on testing data. ↑ means the higher value is better. ↓ means the lower value is better.

Team	DSC[%] ↑	HD[mm] ↓	ABD [mm] ↓	RAVD [mm] ↓	Score↑
Ours(Bridged U-net)	**89.96**	5.5788	**1.5938**	7.2674	**86.50**
DenseFCN(DenseNet)	88.98	**5.3219**	1.6619	**6.3514**	86.36
MBIOS(U-net)	88.06	10.1561	2.4928	7.0986	83.66
UdeM 2D(ResNet)	87.42	5.8899	1.954	12.3722	83.45
Ours(Stacked U-net)	87.15	9.6123	14.5539	11.32	81.44

Our testing results have been submitted to MICCAI PROMISE12 grand-challenge website and evaluated by the organizer. The total score for ranking is obtained after calculating each metrics (mean DSC, HD, ABD, RAVD) by comparing testing result with ground truth segmentation results. The result shows that Bridged U-net performs much better than 2D U-net architecture. We have compared our method with other state-of-the-art 2D methods. Team DenseFCN, using dense block to replace convolution blocks of U-net, was ranked 1^{st} in 2D method before we submitted Bridged U-net. Team MIBOS uses U-net architecture while team UdeM 2D uses residual block to replace convolution blocks of U-net. In addition to these methods, we also provide the result of a stacked U-net for reference. From Table 6 we can see that our proposed network performs best in DSC and ABD metrics, additionally, we get the highest total score among 2D methods.

5 Conclusion

In this paper, we proposed network bridging architecture, which makes stacked U-net suitable for small image datasets such as medical image datasets. We also discussed the bridging and skip connection methods and find out that concatenation is better for network bridging while addition is better for skip connection. Besides of this, we proposed ELU and ReLU combination to improve network performance, which is also effective in traditional image classification task. In addition to activation function, we proposed cos-dice loss to solve the oscillation problem during network training. We use MICAAI PROMISE12 dataset to evaluate our network and the result shows that our network performs better than original U-net, stacked U-net and other state-of-the-art methods.

References

1. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 3431–3440
2. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, Springer (2015) 234–241
3. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 3D Vision (3DV), 2016 Fourth International Conference on, IEEE (2016) 565–571
4. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3d u-net: learning dense volumetric segmentation from sparse annotation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer (2016) 424–432
5. Shah, S., Ghosh, P., Davis, L.S., Goldstein, T.: Stacked u-nets: a no-frills approach to natural image segmentation. arXiv preprint arXiv:1804.10343 (2018)
6. Ghosh, A., Ehrlich, M., Shah, S., Davis, L., Chellappa, R.: Stacked u-nets for ground material segmentation in remote sensing imagery. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2018) 257–261

7. Ma, K., Shu, Z., Bai, X., Wang, J., Samaras, D.: Docunet: Document image unwarping via a stacked u-net. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 4700–4709
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
9. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR. Volume 1. (2017) 4
10. Srivastava, R.K., Greff, K., Schmidhuber, J.: Training very deep networks. In: Advances in neural information processing systems. (2015) 2377–2385
11. Huang, G., Liu, Z., Weinberger, K.Q., van der Maaten, L.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Volume 1. (2017) 3
12. Han, J., Moraga, C.: The influence of the sigmoid function parameters on the speed of backpropagation learning. In: International Workshop on Artificial Neural Networks, Springer (1995) 195–201
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
14. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289 (2015)
15. Litjens, G., Toth, R., van de Ven, W., Hoeks, C., Kerkstra, S., van Ginneken, B., Vincent, G., Guillard, G., Birbeck, N., Zhang, J., et al.: Evaluation of prostate segmentation algorithms for mri: the promise12 challenge. Medical image analysis **18** (2014) 359–373
16. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521** (2015) 436
17. Krizhevsky, A., Nair, V., Hinton, G.: Cifar-10 and cifar-100 datasets. URl: https://www. cs. toronto. edu/kriz/cifar. html (vi sited on Mar. 1, 2016) (2009)
18. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
19. Xide X. and Kulis, B. : W-Net: A Deep Model for Fully Unsupervised Image Segmentation arXiv preprint arXiv:1711.08506, 2017.