Computers, Materials & Continua CMC, vol.62, no.3, pp.1233-1247, 2020

Empirical Comparisons of Deep Learning Networks on Liver Segmentation

Yi Shen[1], Victor S. Sheng[1, 2, *], Lei Wang[1], Jie Duan[1], Xuefeng Xi[1], Dengyong Zhang[3] and Ziming Cui[1]

Abstract: Accurate segmentation of CT images of liver tumors is an important adjunct for the liver diagnosis and treatment of liver diseases. In recent years, due to the great improvement of hard device, many deep learning based methods have been proposed for automatic liver segmentation. Among them, there are the plain neural network headed by FCN and the residual neural network headed by Resnet, both of which have many variations. They have achieved certain achievements in medical image segmentation. In this paper, we firstly select five representative structures, i.e., FCN, U-Net, Segnet, Resnet and Densenet, to investigate their performance on liver segmentation. Since original Resnet and Densenet could not perform image segmentation directly, we make some adjustments for them to perform live segmentation. Our experimental results show that Densenet performs the best on liver segmentation, followed by Resnet. Both perform much better than Segnet, U-Net, and FCN. Among Segnet, U-Net, and FCN, U-Net performs the best, followed by Segnet. FCN performs the worst.

Keywords: Liver segmentation, deep learning, FCN, U-Net, Segnet, Resnet, Densenet.

1 Introduction

Liver segmentation is an important step before lesion detection and diagnose, but manually segmenting livers from medical images is time-consuming. With the soaring of deep learning recent year, many deep learning works have been proposed for automatic liver segmentation. Liver segmentation is analogy to image semantic segmentation, an important branch in the field of AI and computer vision.

Different from image classification, semantic segmentation needs to determine the category of each pixel for accurate segmentation. Therefore, deep learning networks are required to transform the feature map extracted from input images to their original size. The fully convolutional neural network proposed by Long et al. [Long, Shelhamer and

[1] School of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou, China.

[2] Department of Computer Science, University of Central Arkansas, Conway, Arkansas, USA.

[3] Huan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, China.

*Corresponding Author: Victor S. Sheng. Email: ssheng@uca.edu.

Darrell (2015)] is the first work in semantic segmentation that can process arbitrarily sized input images. Specifically, it removes the fully-connected layer and introduces the deconvolution layer to return the feature map extracted from an input image to its original size. After this, the structure of FCN is widely used in semantic segmentation, Segnet Badrinarayanan et al. [Badrinarayanan, Kendall and Cipolla (2017)] conducts the image segmentation in an encoder-decoder manner. Its encoder is identical to the 13 convolution layers in VGG-16, and its decoder will map the feature map extracted by the encoder to the full input resolution feature map. In medical image segmentation, U-Net Ronneberger et al. [Ronneberger, Fischer and Brox (2015)] also employs the encoder-decoder structure to conduct segmentation. It develops the lateral connection to concatenate the feature from the encoder phase to the decoder phase. The effectiveness of the lateral connection is demonstrated in medical image segmentation. Besides, it can help the model achieve good results with fewer images.

With the increment of the number of layers, neural network becomes hard to train. However, the depth of a neural network is crucially important and the abstract level of extracted features can be increased by increasing the number of stacked layers. He et al [He, Zhang, Ren et al. (2016)] designed a residual structure, which adds the input of the non-linear layer to its output. With the residual structure, gradient can be quickly delivered to the previous layer, which makes the depth of the network can be much deeper than before. Different from the residual structure, Huang et al. [Huang, Liu, Van Der Maaten et al. (2017)] concatenates each layer to all the other layers in the same block. Simonyan et al. [Simonyan and Zisserman (2014)] obtained a state-of-the-art performance on liver and lesion segmentation based on the Densenet.

In this paper, we firstly make some adjustments to Resnet and Densenet, since they are originally designed for image recognition, not for image segmentation. Then, we conduct experiments to investigate the performance of FCN, Segnet, U-Net, Resnet and Densenet on liver segmentation. The main purpose of this paper is to find the best deep learning approach for liver segmentation.

2 Five popular deep learning networks

The emergence of AlexNet Krizhevsky et al. [Krizhevsky, Sutskever and Hinton (2012)] made deep learning become a hot topic. Many deep learning approaches have been proposed after AlexNet. The development on deep learning mainly includes the depth increment of deep learning network, the enhancement of convolution module functions, new functional units, and many different real-world applications. We will briefly introduce five popular deep learning approaches in the following subsections.

2.1 FCN

Fig. 1 shows the structure and the schematic diagram of fully convolutional networks (FCN). The max-pooling layer of FCN is commonly used in convolution neural network to reduce the computation complexity and overcome overfitting. After several pooling operations, the feature map will be 16 or 32 times smaller than the size of an input image. What we need in the semantic segmentation is an output image with the same size, where each pixel has an assigned label. Therefore, FCN introduces deconvolution to transform the

final feature map extracted from a medical image to an output image with the same size.

The simplest way to transform the final feature map extracted from an image to an output image with the same size is to directly enlarge 32 times. However, this manner will loss the segmentation detail. Therefore, FCN Long et al. [Long, Shelhamer and Darrell (2015)] deconvoluted the output of the fourth layer and the third layer, carry out 16 times and 8 times up-sampling respectively, and then combine them to generate a prediction for each pixel. This will retain more spatial information in the original input image. Finally, the feature map is classified pixel by pixel, and the result is a little more subtle.

In our experiments, we adapt the up-sampling strategy of FCN-8s to carry out 32-fold up-sampling for pool5 (refer to Fig. 1) features. And then each point of 32x upsampled features will make a softmax prediction to obtain segmentation.

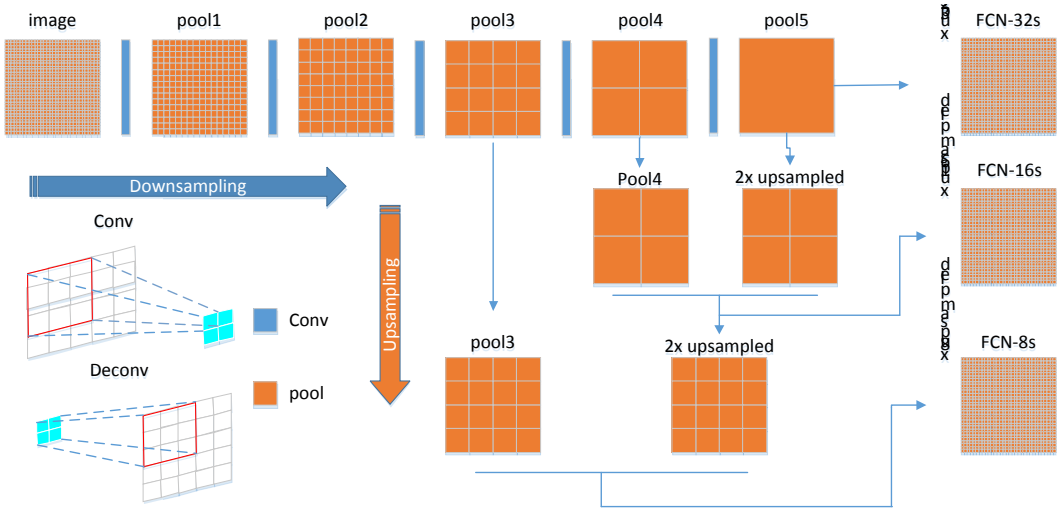

Figure 1: The FCN network structure and its schematic diagram

2.2 U-Net

U-net is a semantic segmentation network based on FCN, which performs well in medical image segmentation, and is the cornerstone of medical image segmentation. Medical images usually have fuzzy boundary, complex gradient and large gray range, so medical image segmentation needs more high-resolution information. The U-net structure combines the information of bottom layers with top layers, and erases the problem of information insufficient during up-sampling through the low-resolution information after multiple down-sampling. Its underlying characteristics are important for model training with a small medical image dataset. Because the underlying information can provide contextual semantic information of the target of segmentation in a whole image, which is helpful for the classification of objects. High level information is directly transferred from encoder to decoder at the same height after concatenate operations, which can provide more detailed features for segmentation, such as gradients.

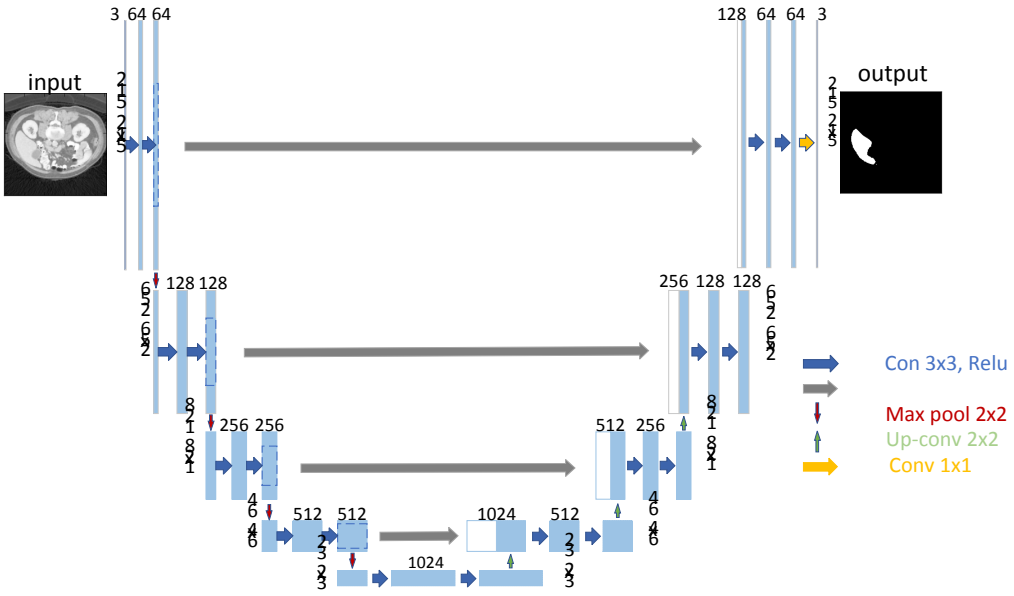

Figure 2: A U-Net network structure

In the up-sampling stage, comparing with FCN, U-Net adopts a completely different feature fusion method: lateral connection! Different from point-by-point addition, U-Net concatenates features together in channel dimensions to form thicker features. U-Net combines low-resolution information (providing the basis for object classification recognition) with high-resolution information (providing the basis for accurate segmentation and positioning), which is perfect for medical image segmentation.

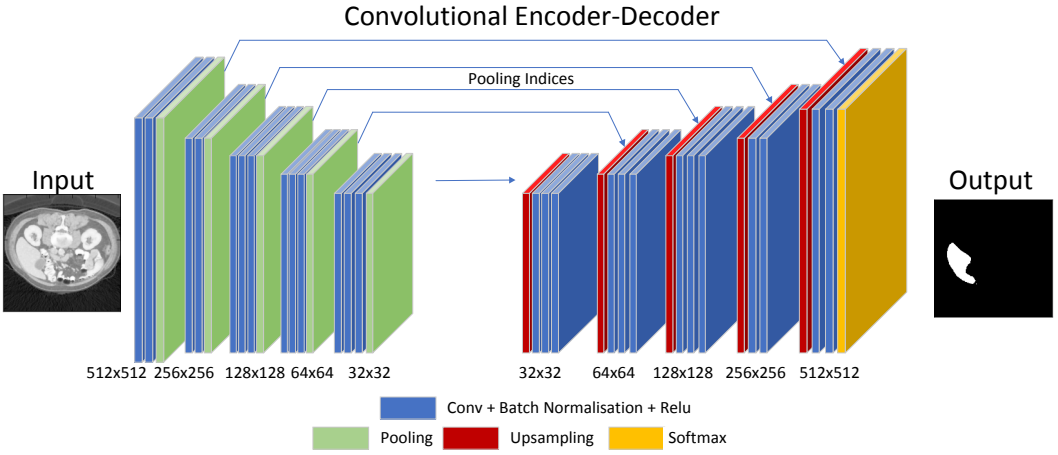

Figure 3: A Segnet network structure

2.3 Segnet

Fig. 3 shows that the whole structure of Segnet is composed of an encoder and a decoder. The encoder is the 13 identical convolution layers in VGG-16, while its decoder is the symmetric structure of its encoder. Segnet uses same convolution layers to extract features in the process of encoding. In the decoding process, Segnet uses convolution to enrich the image information after up-sampling operations. In Segnet, each convolutional layer is followed by a Batch Normalization layer and a ReLu activation layer.

The main difference between Segnet and FCN lies in up-sampling the decoding process. From the above structure, we can see that each pooling layer saves pooling indices and is transferred to a later symmetric up-sampling layer. The process of up-sampling is as follows.

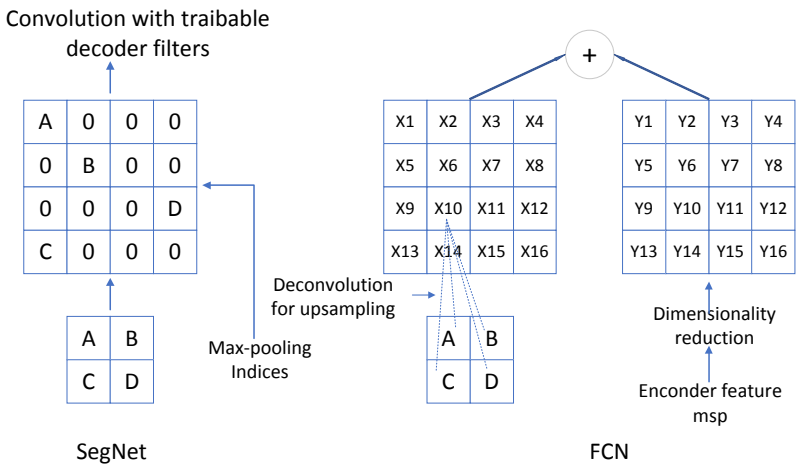

Figure 4: The comparison of up-sampling operation between Segnet and FCN

The left sub-figure of Fig. 4 is the up-sampling method proposed in Segnet. It maps the feature map 'ABCD' to the new feature map by the coordinates of the max-pooling previously saved, and pads zeros for other positions. The right sub-figure of Fig. 4 is the up-sampling method used in FCN. It conducts a deconvolution operation on map 'ABCD' to obtain a new feature map, and then it is added to the previous corresponding encoder feature map. The FCN network just replicates the encoder characteristics, while the Segnet network replicates the maximum pooling component. In terms of memory usages, Segnet is more efficient than FCN.

By comparing the whole network structure of the models introduced, Segnet has fewer training parameters, faster speed and lower memory requirements than previous neural networks.

2.4 Resnet

In ImageNet classification, it has been demonstrated that the error is obviously reduced with the increment of the depth of deep learning networks. That is, the depth of a deep learning network is crucial to its performance. This is because when the number of layers of the network is increased, the network can extract more complex feature patterns. However, the deep learning network is hard to train. This is because the global

distribution gradually approaches the upper and lower limits of the value interval of the nonlinear activation function. This leads to the disappearance of the gradient of the lower neural network in the back propagation. This is the essential reason for the slower and slower convergence of training a deep neural network.

Before Resnet, convolutional networks rarely exceed 20 layers. Resnet solves the problem of gradient dispersion well and makes a deeper network easier to train. Fig. 5 simply shows a typical residual block used in Resnet. Since the network is directly connected to the network of the layer above, the gradient can be propagated better.

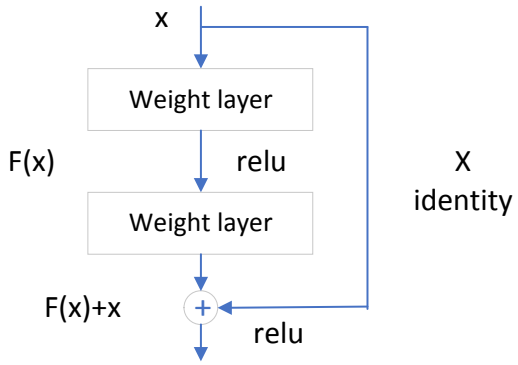

Figure 5: A residual block

Table 1: Three different structures of Resnet

Layers	Output size	ResNet-50	ResNet-101	ResNet-152
Conv_1	512 × 512	7 × 7 conv, stride 2		
Pooling	256 × 256	3 × 3 max pooling, stride 2		
Conv_2_x	256 × 256	$\begin{bmatrix} 1\times1, 64\ conv \\ 3\times3, 64\ conv \\ 1\times1, 256\ conv \end{bmatrix}\times 3$	$\begin{bmatrix} 1\times1, 64\ conv \\ 3\times3, 64\ conv \\ 1\times1, 256\ conv \end{bmatrix}\times 3$	$\begin{bmatrix} 1\times1, 64\ conv \\ 3\times3, 64\ conv \\ 1\times1, 256\ conv \end{bmatrix}\times 3$
Conv_3_x	128 × 128	$\begin{bmatrix} 1\times1, 128\ conv \\ 3\times3, 128 conv \\ 1\times1, 512\ conv \end{bmatrix}\times 4$	$\begin{bmatrix} 1\times1, 128\ conv \\ 3\times3, 128 conv \\ 1\times1, 512\ conv \end{bmatrix}\times 4$	$\begin{bmatrix} 1\times1, 128\ conv \\ 3\times3, 128 conv \\ 1\times1, 512\ conv \end{bmatrix}\times 8$
Conv_4_x	64 × 64	$\begin{bmatrix} 1\times1, 256\ conv \\ 3\times3, 256\ conv \\ 1\times1, 1024\ conv \end{bmatrix}\times 6$	$\begin{bmatrix} 1\times1, 256\ conv \\ 3\times3, 256\ conv \\ 1\times1, 1024\ conv \end{bmatrix}\times 23$	$\begin{bmatrix} 1\times1, 256\ conv \\ 3\times3, 256\ conv \\ 1\times1, 1024\ conv \end{bmatrix}\times 36$
Conv_5_x	32 × 32	$\begin{bmatrix} 1\times1, 512\ conv \\ 3\times3, 512\ conv \\ 1\times1, 2048\ conv \end{bmatrix}\times 3$	$\begin{bmatrix} 1\times1, 512\ conv \\ 3\times3, 512\ conv \\ 1\times1, 2048\ conv \end{bmatrix}\times 3$	$\begin{bmatrix} 1\times1, 512\ conv \\ 3\times3, 512\ conv \\ 1\times1, 2048\ conv \end{bmatrix}\times 3$
Pooling	16 × 16	7 × 7 average pooling, stride 2		
Classification Layers	1 × 1000	1000 D Fully Connected Layer		

In Resnet, when the residual is 0, the accumulation layer only does identity mapping at this time. This ensures the network performance will not decline. In fact, the residual won't be 0, and will make the accumulation layer learn new features based on the input

features, and thus will get a better performance. This is like a kind of a short circuit in a circuit, so it's called a short circuit connection.

Tab. 1 presents five different deep types of Resnet with 50 layers, 101 layers, and 152 layer respectively. The leftmost column of the table shows that Resnet is composed of five parts: conv1_x, conv2_x, conv3_x, conv4_x, and conv5_x. Since the size of feature maps keep the same in the same layer while it is different between layers and the operation on each layer are all convolutions, the author uses the conv n_x to denote each layer. In our experiments, we use 101-layer Resnet. Most importantly, since original Resnet could not perform image segmentation, in order to use Resnet for liver image segmentation, we introduce deconvolution like FCN to transform the final feature map extracted from a medical image to an output image with the same size after removing its fully connection layer.

2.5 Densenet

The core of Densenet is the same with that of Resnet: using short paths concatenate features in early layers with features in later layers. One obvious difference between Densenet and Resnet is that each network layer of Resnet is connected by summation, while Densenet is done by concatenating. In Densenet, the input of each layer includes the outputs of all previous network. We use L to denote the output of one layer, so the formulation of L equals to $\mathbf{K}\times$(L-1)+K_0, where K is the growth rate which represents the number of channels in each layer and K_0 is the number of channels in input.

Tab. 2 shows four different structures of Densenet. In our experiments, Densenet-121 is used. The numbers (such as 6, 12, 24 and 16) in the third column are corresponding growth rates, representing the number of feature maps output from each layer in each dense block. Each bottleneck layer starts with a 1×1 convolution to merge information of each channel and reduce the input feature map, and then tends to do 3×3 convolution. According to the design of the dense block, subsequent layers can get the input from all preceding layers, so the input channel after concatenating is still large. In order to further compress parameters, we add atransition layer between every two dense blocks to do the convolution operation of 1×1.

Densenet improves the transmission efficiency of information and gradient in the network. Each layer can get the gradient directly from the loss function and get the input signal directly, so that the deeper network can be trained. This network structure also has the effectiveness of regularization. Other networks focus on improving their performance from depth and width.

Densenet is dedicated to improving the performance from the perspective of feature reuses. Its network is thin and the number of parameters is controlled. Multiple bottleneck designs tend to have obvious levels, and the number of feature graphs tend to go up layer by layer to ensure the expressive ability of output features; Fewer pooling layers and more down sampling improve the propagation efficiency; There is no dropout in the network, and the regularization is carried out by using BN and a global average pooling, speeding up the training speed. Since original Densenet could not perform image segmentation, in order to use Densenet for liver image segmentation, we introduce

deconvolution like FCN to transform the final feature map extracted from a medical image to an output image with the same size after removing its fully connection layer.

Table 2: Four different structures of Densenet

Layers	Output size	DenseNet-121	DenseNet-169	DenseNet-201
Convolution	512×512	7×7 conv, stride 2		
Pooling	256×256	3×3 max pooling, stride 2		
Dense Block (1)	256×256	$\begin{bmatrix} 1 \times 1\ conv \\ 3 \times 3\ conv \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1\ conv \\ 3 \times 3\ conv \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1\ conv \\ 3 \times 3\ conv \end{bmatrix} \times 6$
Transition Layer (1)	128×128	1×1 conv, 2×2 average pooling, stride 2		
Dense Block (2)	128×128	$\begin{bmatrix} 1 \times 1\ conv \\ 3 \times 3\ conv \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1\ conv \\ 3 \times 3\ conv \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1\ conv \\ 3 \times 3\ conv \end{bmatrix} \times 12$
Transition Layer (2)	64×64	1×1 conv, 2×2 average pooling, stride 2		
Dense Block (3)	64×64	$\begin{bmatrix} 1 \times 1\ conv \\ 3 \times 3\ conv \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1\ conv \\ 3 \times 3\ conv \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1\ conv \\ 3 \times 3\ conv \end{bmatrix} \times 48$
Transition Layer (3)	32×32	1×1 conv, 2×2 average pooling, stride 2		
Dense Block (4)	32×32	$\begin{bmatrix} 1 \times 1\ conv \\ 3 \times 3\ conv \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1\ conv \\ 3 \times 3\ conv \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1\ conv \\ 3 \times 3\ conv \end{bmatrix} \times 48$
Pooling	16×16	7×7 average pooling, stride 2		
Classification Layer	1×1000	1000 D Fully Connected Layer		

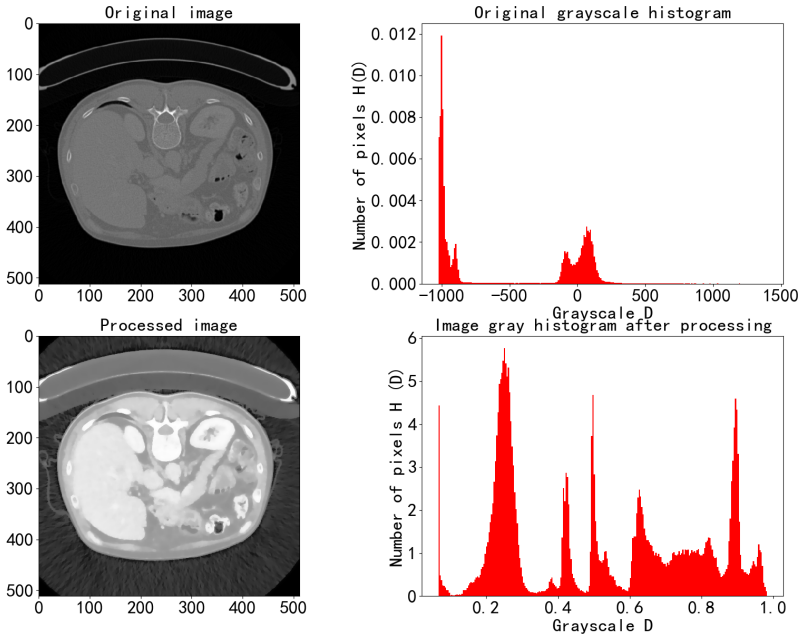

Figure 6: The histogram comparison of an image before and after preprocessing

3 Experiments

In this section, we will conduct experiments to investigate the performance of the five network structures on liver image segmentation.

3.1 Preprocessing

Since original medical images have a wide range number of pixels, and the contrast between livers and its surrounding organs is relatively low, we first perform image contrast enhancement and denoising. Specifically, we first use a linear transformation to adjust the brightness and the contrast of the original medical images. Then, by stretching the distribution range of pixel intensity, the number of pixels of each image is reallocated to a roughly equal number of pixels within a certain gray level. Finally, we use the method of a total variation (the integral of the norm of image gradients) to remove noise and retain main details such as edges. In our experiments, we randomly choose 70% images for training and the rest as test data.

Fig. 6 shows the histogram of an image before and after image processing. Due to the particularity of CT images, the histogram of the original image is very different from that of the processed image. It can be seen that the preprocessed image has a higher brightness and a sharper contrast.

3.2 Parameter settings and loss functions

In the experiment, we use an exponential attenuation learning rate. That is, the learning rate can adjust and change by itself according to the descending speed of training to prevent vibration. The equation of the exponential decay learning rate is defined as follows.

$$lr = lr_0 * gamma\hat{}(globalstep/decaysteps) \tag{1}$$

where lr is the current learning rate; lr_0 is the initial learning rate; gamma is the learning rate decay coefficient (generally between 0 and 1); global step is the number of iterations; and decay steps are the decay rate. In the experiment, the initial learning rate is set as 3e-5, and the learning rate attenuation rate is 0.90, the learning rate will be updated for each epoch. The batch size is set as 2, and the total number of rounds of training is 50000. The selected optimizer is Adam optimizer, which is an adaptive learning optimizer and can make the network convergence faster compared with the traditional gradient descent method, and can quickly jump out of the local optimum to find the global optimum.

Due to the particularity of medical images, some slices of medical images may be tangent to the boundary of the target organ. The area of the tangent part is very small, so it will lead to a low contrast and greatly affect the segmentation accuracy. For this problem, ordinary loss function may convergence. So, we adopt dice as a loss function, and its mathematical equation is $-\frac{2(a \cdot b)}{|a|^2+|b|^2}$, where a is the label image and b is the predicted image by the network. Dice loss is converted from the Dice coefficient, which can measure the similarity between the auto-segmented results and the ground truth. During training we reverse the value of dice and the small dice will generate a big loss.

We standardize trained images to improve the balance of background and foreground, minimizing the loss for training and improving the efficiency of convergence. The pixels

of each image is labeled as either 0 or 1 after preprocessing. We expect to train a segmentation deep learning model with this loss function to reduce the gap between corresponding labelled images and original images in general.

3.3 Experimental equipment and data set

Our experiments run on Ubuntu 16.04 with CPU i7 6700K, NVIDIA GTx1080ti GPU, 32 G memory. All deep learning approaches are developed based on TensorFlow 1.70. The data set we used was provided by the liver segmentation competition, containing 131 CT sequences. The resolution of the CT slices is 512×512.

3.4 Experimental results

We investigate the performance of the five deep learning approaches in terms five popular evaluation measures (i.e., Dice's similarity coefficient (DSC), Volume Overlap Error (VOE), Relative Volume Difference (RVD), Average Symmetric Surface Distance (ASD) and Root-Mean-Square Deviation (RMSD)). The definition of the five evaluation indices are as follows.

DSC measures the ratio of the intersecting area between the segmentation result (S) and the corresponding ground truth (T), which is defined as follows:

$$\mathrm{DSC}(S,T) = \frac{2|S \cap T|}{|S|+|T|} \tag{2}$$

VOE is similar to DSC, in which multiplication is replaced with subtraction operation to represent the error rate, defined as follows:

$$\mathrm{VOE}(S,T) = 1 - \frac{|S \cap T|}{|S \cup T|} \tag{3}$$

RVD measures the difference between the segmentation result (S) and the corresponding ground truth (T), defined as follows:

$$\mathrm{RVD}(S,T) = \frac{|S|-|T|}{|T|} \tag{4}$$

ASD is another way to evaluate the difference between the segmentation result (S) and the corresponding ground truth (T), defined as follows:

$$\mathrm{ASD}(S,T) = \frac{1}{|B_S|+|B_T|} \times \left(\sum_{x \in B_S} d(x, B_T) + \sum_{x \in B_T} d(x, B_S)\right) \tag{5}$$

where BS and BT represent the outline of the segmented liver region and of the corresponding ground truth liver region respectively. $d(x, B_T)$ represents the shortest distance between any pixel x and BS. That is, $d(x, B_T) = \min_{b_T \in B_T} \| x - b_T \|$, where $\| \bullet \|$ is an Euclidean distance.

RMSD measures the deviation between the segmentation result (S) and the corresponding ground truth (T), defined as follows:

$$\mathrm{RMSD}(S,T) = \sqrt{\frac{1}{|B_S|+|B_T|}} \times \sqrt{\sum_{x \in B_S} d^2(x, B_T) + \sum_{x \in B_T} d^2(x, B_S)} \tag{6}$$

Our experimental results are shown in Tab. 3. Note that the unit of DSC, VOE and RVD is in percentage and the unit of ASD and RMSD is mm.

Tab. 3 shows that Densenet-121 has the best performance in terms of DSC and VOE, and takes the second place in terms of RVD, ASD, and RMSD, while Resnet takes the first place in terms of RVD, ASD and RMSD. However, Resnet loses to U-Net in terms of DSC and VOE. To sum up, Resnet_101 and Densenet_121 perform the best, followed by U-Net. They all perform better than Segnet and FCN. Between Segnet and FCN, Segnet performs better. FCN performs the worst because the boundary segmentation accuracy of FCN is not high. Although all the five deep learning approaches in our experiment can perform a reasonable segmentation in terms of DSC, all other evaluation indexes show that segmented results need to be further refined.

Table 3: Experimental results

Model	DSC (%)	VOE (%)	RVD (%)	ASD (mm)	RMSD (mm)
FCN-8s	81.23±1.20	24.30±4.79	24.25±6.01	16.04±1.40	21.01±1.62
U-Net	90.82±1.32	14.28±2.36	21.22±2.56	6.79±2.79	13.62±5.07
Segnet	89.46±0.95	18.31±1.39	52.39±1.20	9.61±1.18	16.97±2.11
Resnet	90.45±1.27	15.11±2.22	**5.56±6.06**	**5.84±0.88**	**9.42±0.81**
Densenet	**91.44±0.87**	**13.40±1.47**	11.20±4.60	6.80±0.06	12.56±0.75

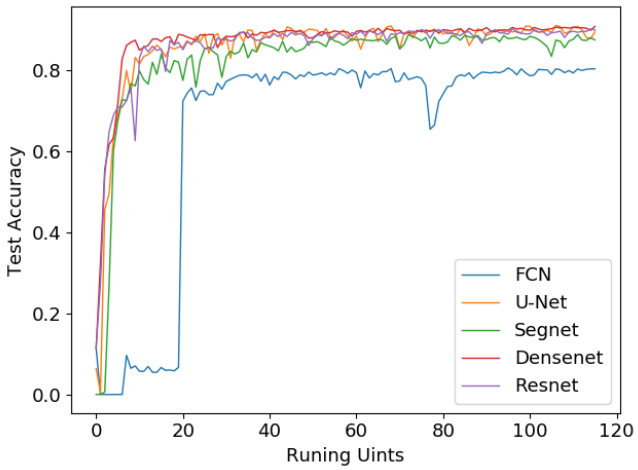

Figure 7: The pixel classification accuracy during testing of each approach against the number of running units

We also conduct experiments to investigate the performance of the five deep learning approaches in terms of the pixel classification accuracy on testing images against the number of running units (note that one unit is 300 rounds). Our experimental results are shown in Fig. 7. From Fig. 7, we can clearly see that Densenet almost always achieves the highest performance at the different training stage and it converges to a stable performance quickly. Its performance tends to be stable at 20 running units. The performance of Resnet is similar to that of U-Net. However, comparing to U-Net, Resnet has some fluctuations and is not very stable at the first 20 running units. Despite Segnet lags behind Densenet, Resnet and U-Net, but it performs consistently better than FCN.

Fig. 8 visually shows the segmentation results generated by FCN, U-Net, Segnet, Resnet and Densenet under the different number of rounds from 300, 3000, 20100, to 40000 rounds. The ground truth segmentation is presented in the last column. From Fig. 8, we can find that Densenet and the Resnet can converge more faster than any other models. They almost converge at 3000 rounds. After 20100 or 40000 rounds, all the deep learning networks are converged. Besides, it is obvious that the segmentation result of FCN is slightly worse.

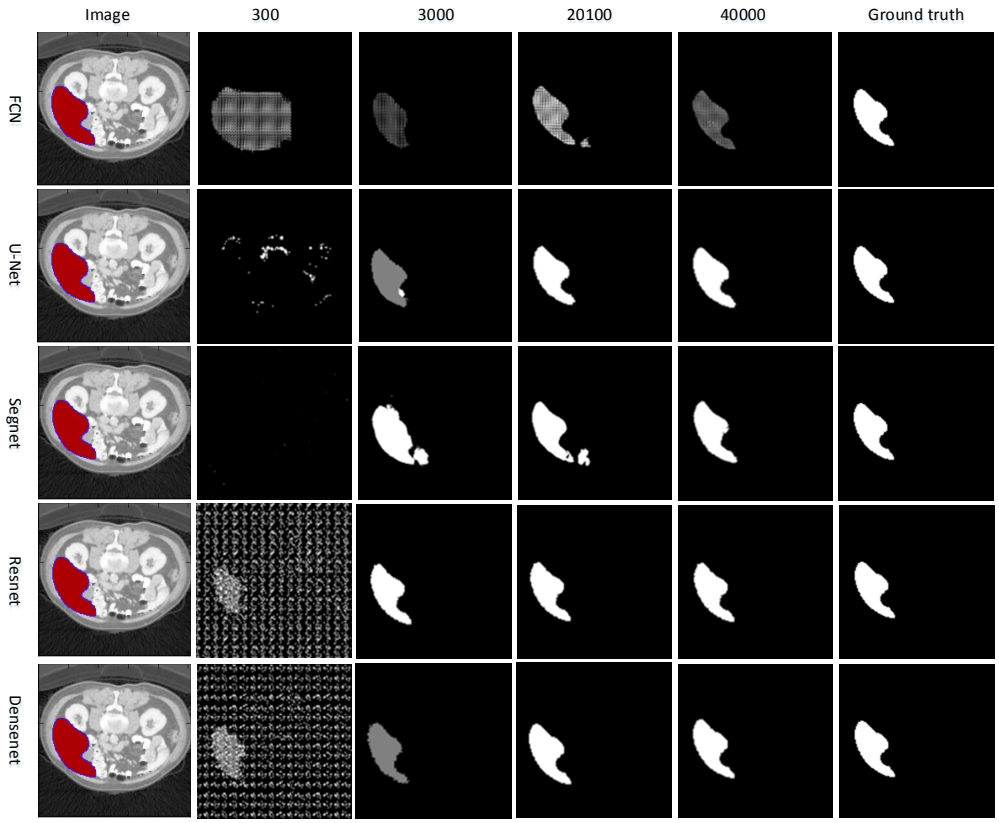

Figure 8: The visualization of segmentation results under the different number of rounds for the five deep learning approaches

To sum up, Resnet and Densenet not only have an excellent segmentation precision, but also have a good convergence rate. Although the data sets used in our experiments are from various CT images of different patients and have different pathological features of different sizes, both Densenet and Resnet still perform well.

4 Discussions

In this paper, we empirically studied the performance of five popular deep learning approaches on live segmentation, i.e., FCN, U-Net, Segnet, Resnet and Densenet.

FCN can completely restore an image to its original size by using convolution and deconvolution operations, satisfying pixel-to-pixel segmentation. It provides a basic idea for deep learning to solve semantic segmentation and inspires many excellent new network

structures. It has no limit to the size of input images with a flexible structure, saving time and space. However, the disadvantages of FCN are also obvious. For example, it is not sensitive to the details of images. This problem is induced by two points. The first one is that the FCN only use one single deconvolution layer to up-sample feature maps. The second one is that it doesn't take the relationship between pixels in the process of classifying into consideration. That is why FCN performs the worst among the five approaches.

Many papers on medical image segmentation are improved by U-Net. This indicates the importance of U-Net. Compared with FCN, U-Net adopts another strategy in the up-sampling stage. It simply concatenates the encoder's feature map to the up-sampling feature map in each stage to form a thicker feature. The network structure of U-Net has a high practicality and is able to learn from relatively small datasets. That is why it has been successfully applied in medical image segmentation. Our experimental results showed that the accuracy and the efficiency of U-Net is higher than FCN and Segnet.

The innovation of Segnet lies in the way of its decoder up-sampling. Its decoder uses the pooled index calculated in the maximum pooled step of the corresponding encoder to perform nonlinear up-sampling. This method eliminates the need for learning oversampling. The feature graph after up-sampling is sparse, so a trainable convolution kernel is then used for convolution operation to generate a dense feature graph. According to its network structure, Segnet has fewer training parameters, faster speed and lower memory requirements than FCN and U-Net. Moreover, it's up-sampling form can be used in other networks. Its segmentation results are not as good as U-Net, but much better than FCN.

Resnet has introduced a residual network structure, through which the residual network can make the network layer deeper and relatively improve its performance. After removing its fully connection layer, we introduce deconvolution like FCN to transform the final feature map extracted from an medical image to an output image with the same size. Our experimental results show that this up-sampling part can segment liver images very well. Its performance on liver segmentation is better than U-Net.

Densenet improves the transmission efficiency of information and gradient in the network. Each layer can get the gradient directly from the loss function and get the input signal directly, so that the deeper network can be trained. This network structure also has the effect of regularization. Other networks focus on improving the network performance from depth and width. However, Densenet is committed to improving the network performance from the perspective of feature reuses. This network is thin and the number of parameters is controlled. Multiple bottleneck designs tend to have obvious levels, and the number of feature graphs tend to go up layer by layer to ensure the expressive ability of output features. Less pooling layer and more use of down sampling improve its propagation efficiency. There is no dropout in the network, and the regularization is carried out by using BN and global average pooling. Therefore, its training speed is speeded up. That is why its accuracy and efficiency are the best among the five networks.

5 Conclusions

In this paper, we conducted experiments to investigate the performance of five popular deep learning approaches on live segmentation, i.e., FCN, U-Net, Segnet, Resnet and Densenet. Since original Resnet and Densenet could not perform image segmentation, in

order to use both Resnet and Densenet for liver image segmentation, we introduce deconvolution like FCN to transform the final feature map extracted from an medical image to an output image with the same size by removing its fully connection layer. Our experimental results show that Densenet performs the best on liver segmentation, followed by Resnet. Both perform much better than Segnet, U-Net, and FCN. Among Segnet, U-Net and FCN, U-Net performs the best, followed by Segnet. FCN performs the worst.

Acknowledgments: This research has been partially supported by National Science Foundation under grant IIS-1115417, the National Natural Science Foundation of China under grant 61728205, 61876217, the "double first-class" international cooperation and development scientific research project of Changsha University of Science and Technology (No. 2018IC25), and the Science and Technology Development Project of Suzhou under grant SZS201609 and SYG201707.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Badrinarayanan, V.; Kendall, A.; Cipolla, R. (2017): Segnet: a deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481-2495.

Bishop, C. M. (1995): *Neural Networks for Pattern Recognition*. Oxford University Press.

Eigen, D.; Fergus, R. (2015): Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2650-2658.

Farabet, C.; Couprie, C.; Najman, L.; LeCun, Y. (2013): Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915-1929.

Glorot, X.; Bengio, Y. (2010): Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249-256.

He, K.; Zhang, X.; Ren, S.; Sun, J. (2016): Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778.

Höft, N.; Schulz, H.; Behnke, S. (2014, September): Fast semantic segmentation of RGB-D scenes with GPU-accelerated deep neural networks. *Joint German/Austrian Conference on Artificial Intelligence*, pp. 80-85.

Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K. Q. (2017): Densely connected convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700-4708.

Ioffe, S.; Szegedy, C. (2015): Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167.

Jégou, S.; Drozdzal, M.; Vazquez, D.; Romero, A.; Bengio, Y. (2017): The one hundred layers tiramisu: fully convolutional densenets for semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 11-19.

LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E. et al. (1989): Backpropagation applied to handwritten zip code recognition. *Neural Computation*, vol. 1, no. 4, pp. 541-551.

Long, J.; Shelhamer, E.; Darrell, T. (2015): Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431-3440.

Krizhevsky, A.; Sutskever, I.; Hinton, G. E. (2012): ImageNet classification with deep convolutional neural networks. *Proceedings of Advances in Neural Information Processing Systems*, pp. 1097-1105.

Papandreou, G.; Chen, L. C.; Murphy, K.; Yuille, A. L. (2015): Weakly-and semi-supervised learning of a DCNN for semantic image segmentation. arXiv:1502. 2734.

Ripley, B. D. (2007): *Pattern Recognition and Neural Networks*. Cambridge University Press.

Ronneberger, O.; Fischer, P.; Brox, T. (2015): U-Net: convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234-241.

Simonyan, K.; Zisserman, A. (2014): Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.

Socher, R.; Lin, C. C.; Manning, C.; Ng, A. Y. (2011): Parsing natural scenes and natural language with recursive neural networks. *Proceedings of the 28th International Conference on Machine Learning*, pp. 129-136.

Sutskever, I.; Hinton, G. E.; Krizhevsky, A. (2012): ImageNet classification with deep convolutional neural networks. *Proceedings of Advances in Neural Information Processing Systems*, pp. 1097-1105.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S. et al. (2015): Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9.

Venables, W. N.; Ripley, B. D. (2013): *Modern Applied Statistics with S-PLUS*. Springer Science & Business Media.

Yu, F.; Koltun, V. (2015): Multi-scale context aggregation by dilated convolutions. arXiv:1511.07122.

Zeiler, M. D.; Fergus, R. (2014): Visualizing and understanding convolutional networks. *Proceedings of European Conference on Computer Vision*, pp. 818-833.