Computer Vision: Final Project
FCN with ResNet for Semantic Segmentation

1st JiangZhuoyang *58119125*
email: 1050251573@qq.com
2nd LiKaixin *58119132*
email: 1321604615@qq.com
3rd Luyangyang *58119128*
email: 2864756803@qq.com

***Abstract*—After investigating the classical papers on the direction of semantic segmentation, among the rich types of models and various implementation methods, we have determined two research directions: one is to trace the pioneering idea of semantic segmentation model based on deep learning method, and the other is to study some improvement ideas of deep learning semantic segmentation model. In the direction of traceability, we focus on FCN. We know that his greatest contribution is to expand the classification method of CNN model at the image level to the pixel level, improve the defects of CNN classification model, and propose a "coding decoding" model. In the direction of improvement, we explore the improvement method of RESNET, which improves the coding structure of ordinary FCN. Finally, the "fcn_resnet" model is obtained by training on the ade data set for result evaluation. On the test set, the segmentation result of the model reaches the demand goal. Compared with other more advanced models, this paper summarizes the areas where this model can be improved in detail, and puts forward the direction of future research work.**

***Index Terms*—Semantic segmentation, FCN, ResNet, DeepLearning**

I. Division of Labor

- 58119125JiangZhuoyang:(35%) Proposed the research theme and research route, completed the writing of the motivation part of the paper, completed the theoretical statement and summary of CNN / FCN in the method statement part of the paper, completed the writing of the model training, result analysis and summary part of the paper, sorted out the content, and wrote the abstract of the paper. Put forward the route of code practice and participate in code debugging. Completed the writing and production of report content and PPT.
- 58119132LiKaixin:(32.5%) completed the theoretical statement and summary of RESNET in the method statement part of the paper, and participated in the proofreading and content arrangement of the paper. The main part of the code is written, and then important code debugging is carried out.
- 58119128LuYangyang:(32.5%) completed the writing of the Research Report of the paper, searched the background data, formed the writing of the background part of the paper, and participated in the proofreading of the paper.

II. Introduction

A. Background

How do humans describe the scene? We might say "there is a table under the window" or "there is a light on the right side of the sofa". The key to image understanding is to decompose a whole scene into several separate entities, which also helps us reason about the different behaviors of the target.

Of course, target detection methods can help us draw the borders of certain entities, but human's understanding the scene can detect each entity with pixel-level fineness and mark precise boundaries. We have begun to develop self-driving cars and intelligent robots, which require a deep understanding of the surrounding environment, so accurate segmentation of entities has become more and more important.

B. Task description

Our task is semantic segmentation, which takes visual scenes as input and scene element categories as output. This is a classification problem and requires the location of specific elements in the image. For the computer, the location depends on the position information of the pixels to complete, so this is a classification task of pixels.

C. Motivation

In the image classification task, the artificial neural network model works well, so we thought about whether it can also be transferred to the pixel-level classification task, and the key lies in how to migrate.

We first investigated some of the existing classic semantic segmentation models: FCN first completed the construction of the pixel-level neural network classification model, and proposed the structure of convolutional coding plus sampling decoding. U-net [6] is similar to it, and has more specific applications in the field of medical imaging. SegNet [1] uses reverse maximum pooling to improve upsampling, eliminating the need for upsampling learning and saving training costs. PSPNet [8] proposes pyramid pooling, which combines multi-size information. A series of deeplabs [2] proposed to expand the receptive field with hole convolution, and to post-process the segmentation results with conditional random fields.

So many models make us feel too complicated, so we grasped two key points that we think: one is tracing the source,

FCN, as a classic of semantic segmentation models, has very important learning significance. The second is to explore how improved ideas are proposed. Here we choose to explore how the idea of ResNet was introduced.

FCN's exploration is mainly task-driven, because the image-level classification task has two flaws, one is limited to the image level, and the other is that the input size is not changeable.There are two ways to improve. One is the improvement in specific areas, and the other is the improvement based on the network structure. The introduction of ResNet is the second, it is not only an excellent deep learning idea, but also adapted to the coding structure of FCN.

III. Methods and Model

In this section, We need to figure out the flaws of traditional CNN image classification first. Then understand how FCN's pioneering method solves CNN's deficiencies. Finally, we will introduce one improvement idea which can be Applied on FCN.

A. CNN's Defects

Firstly, we need to understand the functional defects of CNN in order to better explain what problems the FCN solves later.

1) Description of CNN:

Before using the neural network models to solve the pixel-level task of semantic segmentation of images, some mature classification networks (AlexNet, VGG net and GoogLeNet) have been used for image-level classification tasks. Before introducing the full convolutional network, I think it is necessary to explain the structure (not training) of the CNN-based classification network to understand its functional defects, so as to better explain what problems the full convolutional network solves [3].

2) Structure of CNN:

The classification network based on CNN is mainly connected with several fully connected layers after the convolutional layer.

The convolutional layer will generate a feature map with three dimensions of length, width and depth. For the convenience of description, we will call it the 3D feature map hereinafter. We call the 3D feature map generated by the last convolutional layer the final 3D feature map. The 3D feature map is essentially a three-dimensional space matrix, each element of the matrix is a feature, we call them feature pixel values.

From a functional point of view, a series of fully connected layers that will be accessed later will map the final 3D feature map into a fixed-length feature vector. In the classification network, the final 3D feature map obtained by convolution is fully connected to a feature vector whose dimensions are purely category.

The reason why we can finally get a vector is because in the process of obtaining the full connection of the feature vector of this category, we need to use N convolution kernels that are consistent with the length, width and depth of the final feature map. We call this fully connected convolution kernel. The N fully connected convolution kernels are sequentially convolved with the final 3D feature map. Each fully connected convolution kernel collapses the length, width and depth of the feature map into one value. We convolves N times through N convolution kernels obtaining N feature pixel values and forming a new N-dimensional depth. These N feature pixel values are a kind of probability which can be ultimately used to describe category information, so we call them category feature values.

It is called fully connected because of these fully connected convolution kernels. They performed a weighted summation of all the feature pixel values of the final 3D feature map, then added bias and input the nonlinear function to output a category feature value. This is a kind of fully connected process.

3) Structure of CNN:

There are two main defect of CNN, we will introduce it below to pave the way for the introduction of FCN.

(a) Limitations of image level:

For CNN-based classification networks, in the process of full connection, the length and width dimensions of the features disappear, leaving only the channel dimension to describe category information, that is, each category feature value summarizes the information of the entire information of input image. Therefore, CNN-based classification networks can only analyze image-level features.

(b) The scale is not flexible:

For a CNN-based classification network, in the fully connected process, the fully connected convolution kernel used is completely matched with the size of the final 3D feature map. As a parameter, the fully connected convolution kernel is obtained by training, and it is difficult to change its size simply. Once the size of the input image changes, the size of the final 3D feature map will change accordingly. The size of the fully connected convolution kernel must correspond to the scale of the final 3D feature map, and it can only be retrained.

B. FCN's Solution

1) Motivation of FCN:

FCN classifies the pixels of the image, thereby solving the problem of image segmentation at the semantic level. [5] Based on the CNN-based classification network, FCN has made two main changes:

(a) Fully Convolution:

Convert the fully connected layer of the traditional CNN-based classification network into a series of fully convolutional layers, as shown in the following figure:

In this way, we will not collapse the width and height dimensions, and the final 3D feature map obtained can

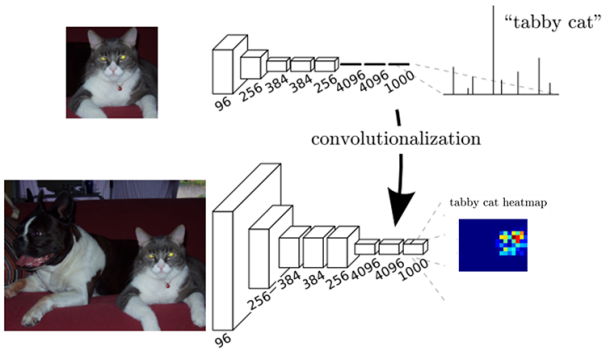


Fig. 1. Fully connected to full convolution

represent the high-level feature information of each region of the input image, instead of only representing the overall information of an image like the category feature vector of CNN.

(b) Upsampling:

The final 3D feature map we obtained through full convolution can already represent the high-level feature information of different areas of the input image. Assuming that these high-level feature information is sufficient to describe the category, now we only need to assign the high-level information back to each pixel.

We need to introduce a series of deconvolution layers to upsample the final feature three-dimensional image obtained from the last volume base layer to restore it to the same size of the input image. This preserves the width and height spatial information in the original input image. Thus, a prediction can be generated for each pixel.

2) Structure of FCN:

The structure of FCN is mainly divided into encoding process and decoding process.The encoding process is used to extract the high-level feature information in the image, and the decoding process is mainly used to restore the size of the 3D high-level feature map back to the size of the original image, and assign the feature pixel value in the high-level feature map back to each pixel of the original image .

As shown in the figure below, the part above the dotted line is the encoding process, and the part below the dotted line is the decoding process.

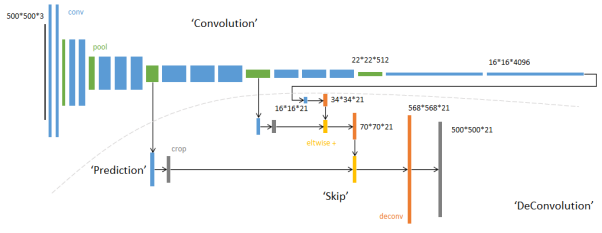


Fig. 2. FCN structure

(a) Encoding structure:

The encoding process mainly uses a convolutional network with fully convolutional layers. When constructing each layer of a convolutional network, there are two important steps: multi-channel convolution and pooling.

(i) Multi-channel convolution:

The process of convolution is actually the process of connecting two layers of neural networks. The convolution kernel is the weight parameter that needs to be trained. It is worth emphasizing that the convolution kernel has three dimensions, length, width and depth.

The kernels' function in the neural network is to perform weighted summation of some feature pixel values of the feature map of the previous layer to obtain a value. This value can be input into a feature pixel value in the next layer after bias and ReLu.

Each convolution has several convolution kernels. The number of convolution kernels depends on the number of channels in the output layer. There are several channels in the output layer and there are several convolution kernels. Each convolution kernel itself also has a depth, which needs to be consistent with the number of channels in the input layer.

The length and width scale of the convolution kernel determines the length and width scale of the output layer. The specific formula is as follows:

$$N = \frac{W-F+2P}{S} + 1$$

Where N is the length and width scale of the output layer. W is the length and width scale of the input layer. F is the length and width scale of the convolution kernel. P is the size of the padding value. S is the step size.

(ii) Pooling:

The purpose of pooling is to directly reduce the feature dimension, increase the receptive field, and reduce the optimization difficulty and parameters at the same time.

The purpose of pooling is to directly reduce the feature dimension, increase the receptive field, and reduce the optimization difficulty and parameters at the same time.

There are many ways to operate pooling, and the more common ones are average pooling and maximum pooling.

Maximum pooling extracts and retains the maximum value of the feature value in the pooling area, and removes other feature values. Mean pooling calculates and retains the average value of the characteristic values in the pooling area. Both reduce the size of the feature space.

It is generally believed that if the area mean (mean pooling) is selected, the characteristics of the overall data can often be retained, and the background information can be better highlighted; if the area maximum (max pooling) is selected, the texture characteristics can be better preserved.

In FCN's classic paper, the author deals with a 21-category classification problem. In their FCN model, a 16*16*4096 3D feature map of high-level features was obtained through a fully convolutional network. In order to represent 21 categories of information, we also need to use 21 1*1*4096 convolution kernels to convolve the entire 3D feature map into a 16*16*21 3D feature map.

When we train our own FCN model, we also need to determine the number of convolution kernels in the last full convolution according to the number of categories.

(b) Decoding structure:

The decoding process is mainly to gradually restore the width and height dimensions of the final 3D feature map obtained by the full convolution to the original image size. At the same time, it is necessary to introduce a multi-scale feature fusion method. Mainly involves two key technologies:

(i) Transpose-convolution (Deconvolution):

When the computer performs a convolution operation, it converts the convolution kernel into an equivalent operation matrix to omit the movement operation. And convert the input to a vector.

The output vector is obtained by multiplying the input vector and the convolution operation matrix. The output vector can be reshaped to get our two-dimensional output characteristics. The specific operation is shown in the figure below:

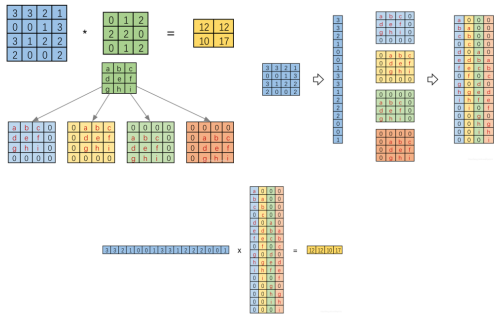


Fig. 3. convolution

Transposed convolution is to restore the last step of the operation in the above figure, transpose the final convolution operation matrix, and multiply it with the result of the convolution to achieve the purpose of restoring the scale, as shown in the following figure:

Fig. 4. Transpose

The above is the calculation principle in the computer, the actual mathematical operation can be equivalently represented by the following figure:

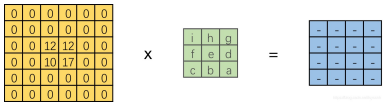


Fig. 5. Transpose-convolution

(ii) Multi-scale Feature Fusion:

FCN also introduces the method of multi-scale feature fusion and skipping, which cuts and adapts the 3D feature maps of each scale level in downsampling, and adds to fuses them with the 3D feature maps in the upsampling process.

This can make the results of shallower features more refined, and the results of deeper features more robust.

3) Advantages of FCN:

The FCN model converts the fully connected layer into a fully convolutional layer, and restores the feature map of the obtained high-level features to the scale of the original image.

It solved the two problems of CNN classification networks: only image-level tasks can be solved, and the input image scale is not flexible.

The output result of FCN is a 3D feature map with the number of categories as the depth and the size of the original image as the length and width. This is equivalent to assigning a category feature vector with the number of categories as the dimension to each pixel of the original image. Therefore, each pixel can be classified.

At the same time, the up-sampling process of FCN is a restoration of down-sampling. In the whole process, there will not be a situation where the size of the input feature map and the parameter size are bound together like the fully connected layer. That is to say, for different scales of the input image, the parameter scale of FCN can be fixed, which solves the problem of immutability of the input scale of the classification network based on CNN.

4) An Improvement Idea:

Like many image neural networks, the FCN encoding process also faces the problem of uncertainty in the optimal depth. Insufficient network depth and degradation caused by purely deepening of the network are both possible reasons that make the FCN model training poorly effective.

Therefore, we can improve FCN from the perspective of optimizing the depth structure of the FCN model.

C. ResNet's Improvement

1) Why we need ResNet:

From experience, the depth of the network is crucial to the performance of the model. When the number of network layers is increased, the network can extract more complex feature patterns, so the deeper the model is, the better the results can be theoretically obtained. It can also be seen from figure

6 that the deeper the network is, the better the results are. However, the experiment finds that the deep network presents a Degradation problem: when the network depth increases, the network accuracy becomes saturated or even decreases. This phenomenon can be seen directly in figure 7: a 56-layer network performs worse than a 20-layer network. This would not be an overfitting problem, because the training error of the 56-layer network is also high. We know that the deep Web has problems with gradient disappearance or explosion, which makes deep learning models difficult to train.

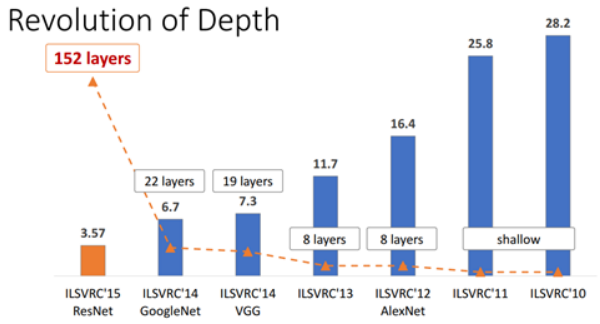


Fig. 6. ImageNet classification Top-5 error

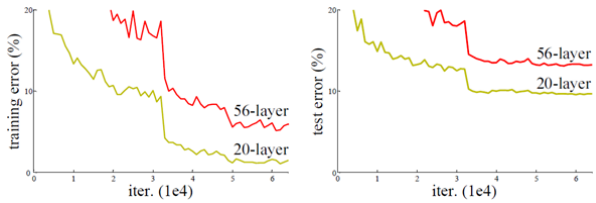


Fig. 7. Error of 20-layer and 56-layer networks on CIFAR-10

At the very least, the degradation of deep networks indicates that deep networks are not easy to train. But let's consider the fact that now you have a shallow network, and you want to build the deep network by stacking new layers up. At one extreme, these additional layers learn nothing but just copy the characteristics of the shallow network, which is Identity mapping. In this case, the deep network should perform at least as well as the shallow network and should not degrade. Based on this idea, residual learning is proposed to solve the degradation problem.

2) Residual unit description:

Residual network [7] still let the nonlinear layer satisfy $F(x, w_h)$, and then from the input directly into a short connected to the output of the nonlinear layer, make the whole map into $y = F(x, w_h) + x$, that is the core of the residual network formula, in other words, the residual error is to establish an operation of the network, any used the operation of the network can be referred to as residual network.

For an accumulation layer structure (composed of several layers), when input is x, the learned feature is denoted as H(x). Now we hope it can learn the residual F(x)=H(x)-x, so in fact the original learning feature is F(x)+x. The reason for this is that residual learning is easier than raw feature learning. When the residual is 0, the accumulation layer only does the identity mapping at this time, at least the network performance will not decline, in fact, the residual will not be 0, which will also enable the accumulation layer to learn new features based on the input features, so as to have better performance. The structure of residual learning is shown in figure 8.

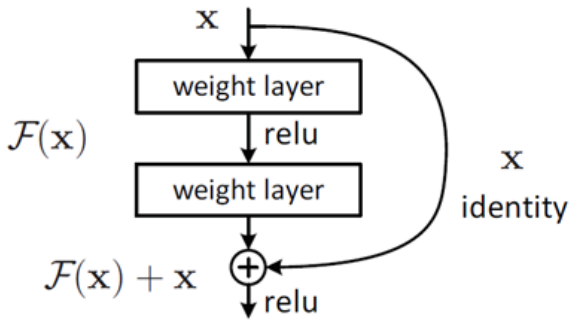


Fig. 8. Residual learning unit

3) Construction of ResNet:

(a) Principle of residual element

Residual unit can be expressed as:

$$y_l = x_l + F(x_l, w_l) \tag{1}$$

$$x_{l+1} = f(y_l) \tag{2}$$

x_l is the input of layer l, $w_l = \{w_{l,k} | 1 \le k \le K\}$ is the parameter of layer l, K is the layer number of residual element, x_{l+1} is the output of layer l and the input of layer $l+1$, F is the residual function, f is the ReLU activation function.

Based on the above formula, we can obtain the learning characteristics from shallow l to deep L as follows:

$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i, w_i) \tag{3}$$

For any depth of L, the above equation (3) shows some good properties.

- The feature x_L of layer L can be divided into two parts. The first part is the shallow network representation x_l plus a residual function mapping $\sum_{i=l}^{L-1} F(x_i, w_i)$, indicating that the model is in the form of a residual in any unit.
- For the feature x_L of any depth L, it is the sum of all the previous residual modules, which is completely opposite to the simple network without short connection. The reason is that the feature x_L of the network at layer L without shorting is the result of a series of vector multiplications, that is, $\prod_{i=0}^{L-1} w_i x_i$ (in the case of ignoring batch normalization and activation functions).

Again, the above equation shows very good back propagation property, assuming that the loss is ϵ, according to the chain rule, we can get

$$\frac{\partial \epsilon}{\partial x_l} = \frac{\partial \epsilon}{\partial x_L}\frac{\partial x_L}{\partial x_l} = \frac{\partial \epsilon}{\partial x_L}(1 + \frac{\partial}{\partial x_l}\sum_{i=l}^{L-1} F(x_i, w_i)) \tag{4}$$

As can be seen from the above equation, gradient $\frac{\partial \epsilon}{\partial x_l}$ is composed of two parts, one is the information flow $\frac{\partial \epsilon}{\partial x_L}$ without any weight weighting, and the other is the $\frac{\partial \epsilon}{\partial x_L}(\frac{\partial}{\partial x_l}\sum_{i=l}^{L-1} F(x_i, w_i))$ with the weighted layer. The linear characteristics connected by the two parts guarantee that information can be directly transmitted back to the shallow layer. At the same time, the formula also indicates that for small batch, gradient $\frac{\partial \epsilon}{\partial x_l}$ is unlikely to disappear, because it is usually not always 1 for small batch, so this means that even if the weight is very small, the gradient will not be 0, and there is no problem of gradient disappearance.

(b) Construction of ResNet

ResNet network is modified based on VGG19 network, and residual unit is added through short-circuit mechanism, as shown in figure 10. The changes are mainly reflected in ResNet directly using the convolution of stride =2 for downsampling, and replacing the full connection layer with the Global Average Pool layer. An important design principle of ResNet is that when the size of feature map is reduced by half, the number of feature map is doubled, which maintains the complexity of network layer. As can be seen from figure 9, compared with ordinary networks, ResNet increases the short-circuit mechanism every two layers, which forms residual learning, where dotted lines represent changes in the number of feature map. ResNet at 34-Layer, shown in figure 9, can also build deeper networks as shown in figure 10. As can be seen from figure 10, for ResNet at 18-layer and 34-layer, residual learning between two layers is carried out. When the network is deeper, residual learning between three layers is carried out. The three-layer convolution kernel is 1x1, 3x3 and 1x1 respectively. It is worth noting that the number of feature maps in the hidden layer is relatively small, and is 1/4 of the number of output feature maps.

Next, we will analyze the residual unit construction in the network in detail. ResNet uses two residual units, as shown in figure 6. The image on the left corresponds to the shallow network, while the image on the right corresponds to the deep network. For short-circuit connections, the input can be directly added to the output when the input and output dimensions are the same. But when the dimensions are inconsistent (corresponding to a doubling of the dimensions), these cannot be added directly. There are two strategies :(1) zero-padding is used to add dimensions. In this case, downsamping should be performed first, and stride=2 can be used for pooling. (2) New mapping shortcut is adopted, generally 1x1 convolution is adopted, which will increase parameters and the amount of calculation. In addition to using identity mapping directly, short-circuit connections can of course be made using Shortcut projection.

IV. Training

In this section, we will introduce the data set we used to train resnet-fcn model. Then we will make a brief introduction to the model training process step by step.

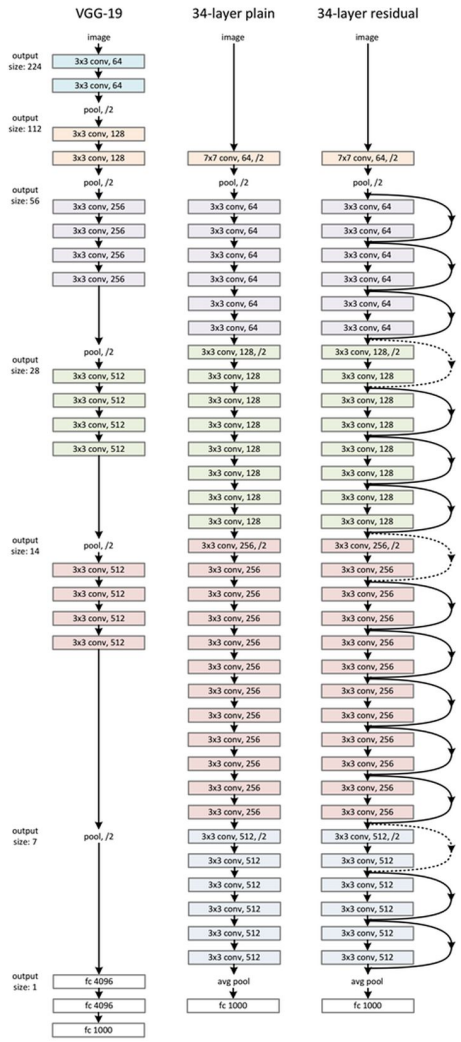


Fig. 9. ResNet network structure diagram

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig. 10. ResNet at different depths

A. Data Set

In this experiment, we used the ADE20k data set. ADE20k has more than 25,000 images (20ktrain, 2k val, 3ktest), which are densely annotated with open dictionary tag sets. For 2017 Places Challenge 2, 100 things and 50 stuff categories covering 89% of all pixels were selected.

1) Dataset Composition:

We have the data set with 4 parts.

- Training set: 20000 images

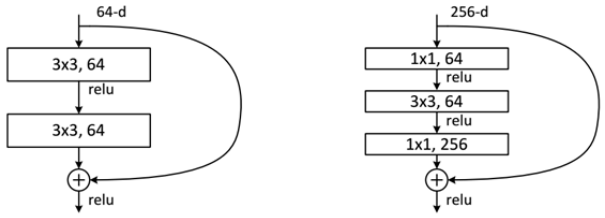


Fig. 11. Different residual units

All Images are fully annotated with objects and, many of the images have parts too.

- Validation set: 2000 images
  Fully annotated with objects and parts
- Training set: 210 images
  Testing images will be released later. So that, we segment 210 pictures from the training set.
- Consistency set: 210 images
  64 images and annotations used for checking the annotation consistency.

2) Pictures and Notes:

Each folder contains images separated by scene category (the same scene category as the location database). For each image, the objects and parts are stored in two different png files. All objects and some instances are sparsely annotated.

3) Files under Each Image:

- $.jpg$:
  RGB image.
- $\_seg.png$:
  This is the object segmentation mask. This image contains information about the segmentation mask of the object classes, and also separates each class into instances. Channel R and G encode object class mask. Channel B encodes the instance object mask. The function m of loadAde20K extracts two masks.
- $\_segparts\_N.png$:
  This is the part segmentation mask, where N is a number (1,2,3,...), indicating the level in the part hierarchy. Parts are organized in a tree, where objects are composed of parts, parts can also be composed of parts, and parts of parts can also have parts.
- $\_.txt$ ::
  This is a text file describing the content of each image (description objects and parts). This information is redundant with other files. But it also contains information about the properties of the object.

B. Training Construction

We will train the resnet-fcn [4] model with the construction steps.

1) Take the ResNet101 or ResNet50 as initialization.
2) Construct FCN-32s decoder whose step size of deconvolution is 32:
   The decoder will directly generate a small segmentation map predicted from the small feature map, and then directly up-sampled to a large image with step of 32.
3) Construct FCN-16s decoder whose step size of deconvolution is 16:
   The decoder add a deconvolution based on FCN-32s decoder. Upsampling is completed in two times, so that the step is changed into 16. Before the second upsampling, the prediction results of the fourth pooling layer are merged. Use skip structure to improve accuracy. Then up-sample it with
4) Construct FCN-8s decoder whose step size of deconvolution is 8:
   This time,upsampling is completed in three times, which further integrates the prediction results of the third pooling layer.

V. Result

A. Quantitative analysis

Based on several evaluation functions for semantic segmentation results, we test the semantic segmentation results of FCN model based on Resnet50 encoder.

TABLE I
QUANTITATIVE EVALUATION RESULTS

Model	Evaluation criteria			
	PA	MPA	MIoU	FWIoU
FCNResNet50	0.924	0.925	0.882	0.887
DeepLab	0.935	0.973	0.928	0.901
PspResNet	0.956	0.967	0.920	0.942

Among the four evaluation results,PA means Pixel Accuracy, which illustrate the ratio of correctly classified pixel points to all pixel points. MPA calculate the ratio of the correct number of pixel points in each category to all pixel points in that category and average. These two evaluation criteria reflect the model effect of pixel-level classification tasks on pixel classification itself. It can be seen that FCNResnet has a generally good classification accuracy of each pixel, which can reach more than 90%.

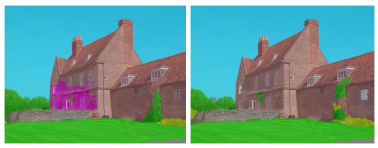
MIoU is a standard metric for calculating the proportion of intersection and union between two sets. In image segmentation, the two sets are the Ground Truth and the predicted segmentation. This can be converted to the ratio of TP (intersection) to TP,FN, and FP (union). FWIoU can be understood as a weighted sum of IoU of each class according to the frequency of occurrence of each class.These two evaluation criteria are specific to the relation between the Ground Truth and the predicted segmentation.It can be seen that if further analysis is made on the result types between the predicted value and the true value, FCN also performs fairly well, and IoU is also improved after category weight is added.

However, if we compare it with other more advanced models, we choose Deeplab and PSPresnet here, and we find that FCNResnet still has room for improvement. Although FCNResnet has met our requirements purely from the perspective of task requirements, there are still many areas for improvement in the design from the perspective of functional details and model training process.

B. Qualitative analysis

For specific images, the visualized results of several semantic segmentation models are as follows. It can be seen that the segmentation results of FCNresnet have generally met our requirements.

(a) FCNResNet101 (b) FCNResNet50



(c) DeepLab (d) PspResNet

Fig. 12. Result visualization

In addition to the segmentation of holistic objects in large regions, it can also segment small fragmented objects. However, compared with the other two more advanced semantic segmentation models, the most obvious problem of FCNresnet, which can be seen from the visualization results, is that it is more likely to produce fragmentation segmentation errors of some local regions in integral objects. We believe that this local error is related to the insufficient scope of receptive field.

VI. Conclusion

After qualitative and quantitative analysis of the results, we summarize the advantages and disadvantages of the model as well as the improvement direction.

A. Advantage

First of all, our model reflects FCN's improvement on CNN, which is also the purpose of our study on FCN. Details are as follows:

1) Expand the task level to the pixel level.
2) Make the input size variable.

Secondly, our model is even better with the introduction of ResNet because it has the following advantages:

1) Deepen the number of network layers and improve the segmentation accuracy of the network.
2) More jump connections can be added in the middle of the network, so that multi-scale segmentation can be better combined with the background semantic information of the image.
3) ResNet has the advantages of fast convergence and reduced model data volume.
4) ResNet makes the model more easy to train, which can not only prevent model degradation, but also prevent gradient disappearance. Loss does not converge.

B. Disadvantage

Although the model improves the effect of the basic FCN to some extent, our model still has the following shortcomings due to some shortcomings of FCN structure design:

1) The results obtained are still not precise enough. Although 8 times up-sampling is much better than 32 times, the up-sampling result is still fuzzy and smooth, not sensitive to the details in the image.
2) The classification of each pixel fails to fully consider the relationship between pixels. The spatial regularization step used in the segmentation method based on pixel classification is ignored, which lacks spatial consistency.

C. Future Work

After comparing with more advanced models and combining with the defects of the models, we have come up with the following improvement ideas. The follow-up work mainly includes the following directions:

1) Optimize network details, such as improving convolution of down-sampling, improving receptive field, or improving methods used for up-sampling.
2) Post-processing of results, such as post-processing with conditions at the airport.
3) Post-processing of results, such as post-processing with conditions at the airport.

D. Feeling

All in all, our experiment result was not bad and we learned a lot from it.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[3] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018.

[4] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pages 239–248. IEEE, 2016.

[5] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[7] Andreas Veit, Michael J Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. *Advances in neural information processing systems*, 29:550–558, 2016.

[8] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.