Research and Implementation of Multi-scene Image Semantic Segmentation based on Fully Convolutional Neural Network

Fangzhou Yu *

Department of Informatics, Beijing University of Technology, Beijing 100124, China.

* yufangzhou940609@163.com

Abstract. With the rapid development of deep neural networks, image recognition and segmentation are important research issues in computer vision in recent years. This paper proposes an image semantic segmentation method based on Fully Convolutional Networks (FCN), which combines the deconvolution layer and convolutional layer converted from the fully connected layer in the traditional Convolutional Neural Networks (CNN). The multi-scene image data set of the label is model-trained, and the training model is applied to pixel-level segmentation of images containing different targets, and the test results are visualized by writing test modules and the segmentation results of the test set images are colored. The experimental process uses two training modes with different parameters to achieve faster and better convergence, and Mini Batch also are used to adapt to the training of big data sets during training. Finally, through the comparison between the segmentation results of test set and the Ground Truth image, it is proved that the full convolutional neural network training model has a higher validity and Robustness for segmentation of some targets in different scene images.

Key words: Computer vision; Deep learning; FCN; Semantic Segmentation.

1. Introduction

In recent years, the development of deep learning has been very rapid, and the development of computer vision has also ushered in a new upsurge. Image segmentation has always been a very widely used technique in image processing. In image processing, in order to better study the target, it is necessary to separate the different targets from the background for separate analysis, such as in the scene of automatic driving, the roads, vehicles, pedestrians, etc. in the current traffic image. That is the task of image Semantic segmentation. The traditional segmentation method without deep neural network involved has disadvantages, and the model trained by deep learning can be predicted based on a large number of different scenarios, and the segmentation result obtained will be more accurate.

The traditional CNN can be applied to image-level classification and regression tasks, because usually CNN will connect several fully connected layers after convolution, and map the feature map generated by the convolution layer into a fixed-length feature vector. The FCN converts the fully connected layer in the traditional CNN into a convolutional layer, so that the image can be classified at the pixel level. The FCN can accept input images with different size, and use the deconvolution layer to up sampling the feature map of the last volume base layer to produce a correspondingly sized output, so that a prediction can be generated for each pixel.

This paper mainly studies the application of FCN in the field of image semantic segmentation in different complex scenes, and implements and trains an end-to-end, point-to-point network for pixel-level prediction of test images. The model is trained using the Scene Parsing Challenge Dataset provided by MIT. The data set contains 150 semantic categories to achieve simultaneous multi-target training in different scenarios. This article implements 8 layers of Fully Convolutional Networks based on CNN's classic network VGGNet19. To address the problem that FCN has a high demand for GPU, this paper uses Mini Batch to reduce the occupation rate of video memory during neural network training to adapt to the lower performance environment. In addition, in order to make the training of neural network faster and more effective, this paper proposes a two-stage training mode using different learning rates. The experimental results show that the proposed method can effectively increase the training speed and accuracy of the image semantic segmentation model, and has a high effectiveness and Robustness for the segmentation results of traffic road scenes.

2. Multi-Scene Image Segmentation Model based on FCN

2.1 VGG Convolutional Neural Networks

The deep convolution network consists of an input layer, an output layer and a plurality of hidden layers. The hidden layer can be divided into a convolution layer, a Pooling layer, a Relu layer, and a fully connected layer. The input layer is a vector containing RGB information of the training data set image. The convolution layer uses filter to convolve the previous result and extract higher-level features. The pooling layer extracts the maximum eigenvalue in a certain area of the image while down-sampling by reducing the amount of data processing. The mapping process of images in CNN is a forward propagation process. In order to avoid the problem of insufficient expression ability of linear models, each layer needs to join the Relu layer when transmitting, and the Relu layer is the activation function of neurons. For the l-th layer neurons A^l, the Relu function is:

$$A^l = \max\left(0, W^l A^{l-1} + b^l\right) \tag{1}$$

Where W is the mapping weight matrix of the current layer and b is the bias term of the current layer. Each node of the fully connected layer is connected to all nodes of the previous layer to combine the features extracted from the front. The output layer is the predicted result of the image. CNN uses Back Propagation to iteratively update parameters W and b. Firstly, the algorithm uses the actual output of the sample and Ground Truth to obtain the Loss function and the Cost function, and then propagates the Cost function back to each layer. Finally, the Gradient Descent is used to adjust the W and b in the network along the negative gradient direction of the Cost function. According to Gradient Descent, the changes in W and b of the layer l are:

$$\left.\begin{aligned} \Delta W^l &= \frac{\alpha}{\mathrm{m}} \sum_{i=1}^{m} \frac{\partial C_i}{\partial W^l} \\ \Delta b^l &= \frac{\alpha}{\mathrm{m}} \sum_{i=1}^{m} \frac{\partial C_i}{\partial b^l} \end{aligned}\right\} \tag{2}$$

Where α is the learning rate, m is the number of input data, and C is the Cost function. After several iterations and after updating the parameters in each iteration, the Cost function is brought close to the minimum.

The VGG network uses 3*3 filters in each convolutional layer and uses 2*2 kernels in each pooling layer to improve the depth of the network and ensure the depth of the network. It also Reduces the amount of calculations and increases the accuracy. The network structure of VGGnet19 is shown in Fig. 1:

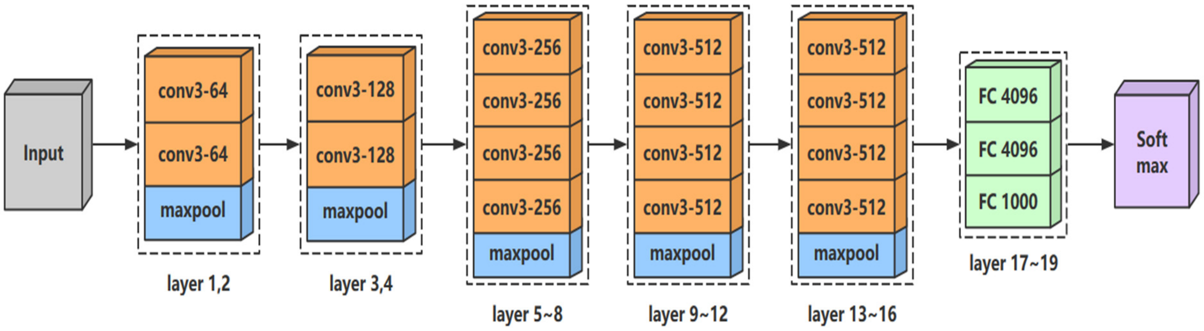

Figure 1. Network configuration of VGGnet19.

2.2 Construction of FCN based on VGG

In the application of image semantic segmentation, FCN can perform dense prediction without full connection layer, and can obtain higher segmentation precision without any pre-processing. The input image of the FCN can be of any size. Due to the existence of the fully connected layer, the traditional CNN can only have a fixed input image size. Compared with the traditional CNN-based image classification method, FCN also increases the processing speed.

The VGG19-based FCN converts the last three layers in the VGG network into three convolutional layers. The first of the three convolutional layers uses 4096 filters which size are 7*7, and the second uses 4096 filters which size are 1*1, and the third uses filters with the number of target classification(150) which size are 1*1. Layers 6 and 7 are one-dimensional vectors of length 4096, respectively, and layer 8 is a one-dimensional vector of length of target classification number (150). By changing the fully connected layer to a convolutional layer, the three layers can use the top 5 layers of trained W and b, and have kernels belonging to this layer. The image obtained at the last level of the convolution is a heat map. In order to be able to obtain a segmented image of the input image size, the heat map needs to be upsampling, that is, the deconvolution layer. The complete VGG19-based FCN structure is shown in Fig. 2:

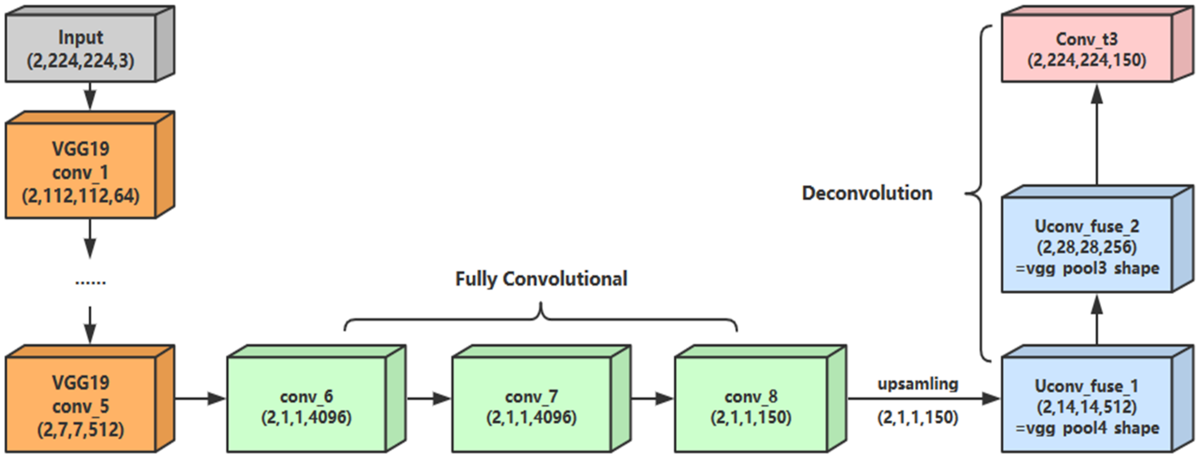

Figure 2. Complete construction of FCN based on VGG.

The deconvolution layer can be regarded as the reverse use of the convolutional layer, the back propagation process of the original convolutional layer is used as the forward propagation of the deconvolution layer, and the forward propagation operation of the convolutional layer is applied to the deconvolution in the process of backpropagation. In addition to this, the deconvolution layer can also be implemented by interpolation.

In the last layer of the FCN, the Softmax function is required to classify the pixel information. The definition of the Softmax function is given as follows:

$$\mathrm{so}\mathit{ft}\max(x)_{\mathrm{i}} = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{3}$$

Where x_i is the output of the pre-stage output unit of the classifier and i represents the category index. $\mathrm{so}\mathit{ft}\max(x)_{\mathrm{i}}$ represents the ratio of the index of the current element to the exponential sum of all elements, which is the probability of the element. In order to compare with the Ground Truth image, the training effect is obtained for the gradient descent. After the logits of the neural network are converted into probability values by the Softmax function, it is also required to perform cross-entropy calculation with the labels. The calculation is given as follows:

$$Loss = -\sum_{\mathrm{i}} t_i \ln y_i \tag{4}$$

Where t_i represents the true value and y_i represents the determined Softmax value. When t_i equals 1, the loss function became $Loss = -\ln y_i$. In the back propagation of each iteration, the result calculated by the Loss function needs to be used to update parameter, and achieve the gradient drop.

3. Experiment

The experimental data set of this paper uses the image parsing Challenge dataset provided by MIT. The data set consists of two parts, of which 20,200 images are for training, and 2000 images are for validation, and the Ground Truth images of these images are included. The image size of the dataset is different, including indoor and outdoor scenes, totaling 150 semantic categories. The experimental environment is given as follows: operating system Ubuntu-16.04-LST-amd64; graphics card NVIDIA GeForce GTX 1060; 6G memory. The FCN setup and test module is implemented using the neural network algorithm library of Python 3.6 and TensorFlow 0.12. Since the data set used for training is very large and contains images of multiple scenes, we use batchsize=2 during training. Only two images are trained in each iteration to make the parameters update faster, ensuring that Links is more effectively converged and avoiding local optimization. Simultaneous use of mini batch ensures that the neural network can be trained normally in a lower GPU performance environment.

Since the batch is applied during training, the input image needs to be preprocessed into a 4-dimensional vector of H*W=224*224. The training process uses segmentation training. In the first stage, in order to ensure that the gradient converges rapidly from a large value, a large learning rate $\alpha = 10^{-4}$ is used; in the second stage, in order to enable the gradient to fall to the lowest point without oscillating near the minimum value, a smaller learning rate $\alpha = 10^{-5}$ is used. After 5 hours of training in the first phase and 18 hours of training in the second phase, the Loss visualization results during the training are shown in Fig. 3:

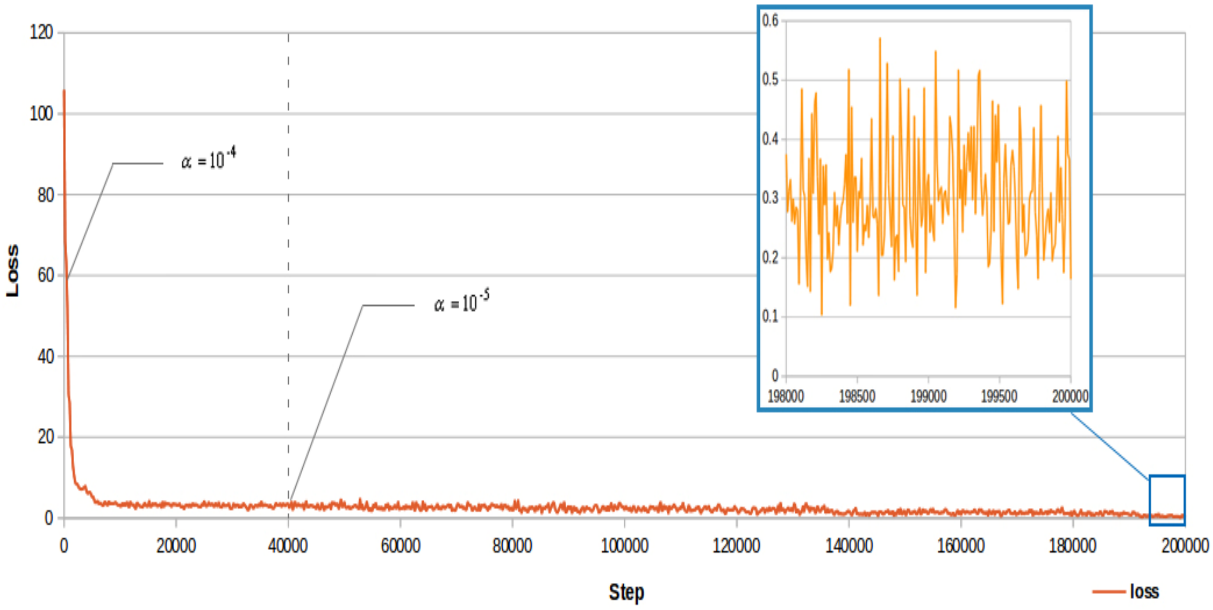

Figure 3. The result of Loss function in the course of piecewise training.

It can be seen from Fig. 3 that the segmentation training is effective for the convergence of the Loss value. After 23 hours of training, the Loss is stable at around 0.3.

In the test module, images are randomly selected from the data set for testing. Visualize segmentation results colored by Terrain color strip, are shown in Fig. 4:

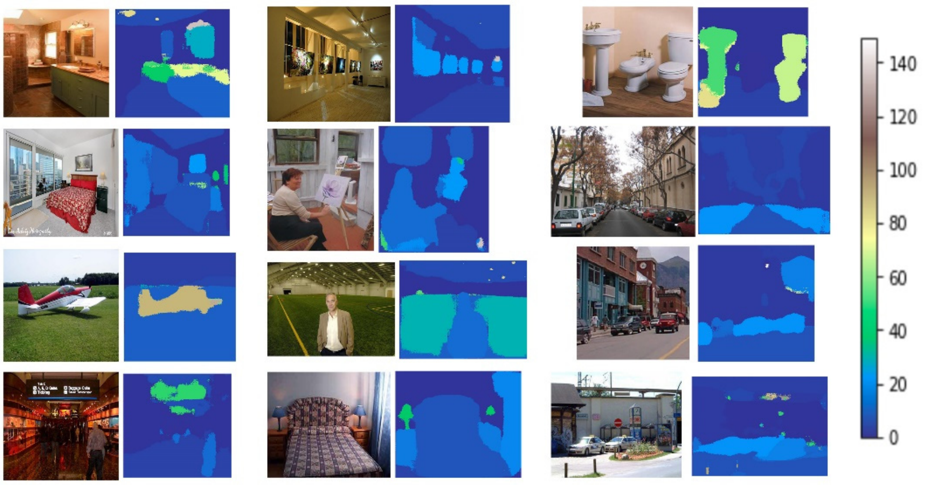

Figure 4. Colored segmentation results of randomly selecting images from the dataset for testing.

In order to verify the accuracy of the model for different scene segmentation, the image semantic segmentation metric Mean Intersection over Union (MIoU) is used to measure the test results of different scene images. The MIoU standard calculates the intersection and union ratio of the two sets of Ground Truth and Predicted Segmentation, and then calculates the average accuracy of each class. The calculation formula is given as follows:

$$MIoU=\frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij}+\sum_{j=0}^{k}p_{ji}-p_{ii}} \tag{5}$$

The MIoU metrics of the test results in different scenarios can be found that the segmentation accuracy of the traffic road scene is high. The MIoU of calculation results are shown in Table 1:

Table 1. MIoU's values of the output images based on different scenes.

Scene type	Bathroom	Bedroom	Parking Apron	Conference Room	Traffic Road	Forest	Museum
MIoU	65.82	65.96	59.72	62.47	72.39	59.87	63.91

The FCN needs to be iteratively optimized continuously. In the case of a short training time, the segmentation result of the trained model cannot achieve sufficiently high precision. The accuracy of the segmentation result is proportional to the training time.

4. Conclusion

Based on the VGG19 network, this paper implements and improves the deep neural network FCN applied to multi-scene image semantic segmentation. Through the model trained by the network, images of different scenes can be segmented at the pixel level. Since the FCN discards the full layer and adds the deconvolution layer, the network supports training images of any size. The network does not require complicated pre-processing and subsequent processing work, and only needs continuous iterative optimization to obtain better segmentation accuracy. The experimental results show that the optimization method in this paper can effectively increase the training speed and accuracy of the image semantic segmentation model, and the MIoU metric results show that the model trained by the network has a strong effectiveness for the segmentation of traffic scenes.

References

[1]. Long J, Shelhamer E, Darrell T. Fully Convolutional Networks for Semantic Segmentation [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 39(4):640-651.

[2]. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition [J]. Computer Science, 2014.

[3]. Garcia-Garcia A, Orts-Escolano S, Oprea S, et al. A Review on Deep Learning Techniques Applied to Semantic Segmentation [J]. 2017.

[4]. Tian Zhuangzhuang, Zhan Ronghui, Hu Jiemin, et al. SAR ATR based on convolutional neural network [J]. Journal of Radars, 2016, 5(3): 320－325.

[5]. GUO Shuxu, MA Shuzhi, LI Jing, et al. Fully convolutional neural network for liver segmentation in CT image. Computer Engineering and Applications, 2017, 53(18): 126-131.

[6]. Sabokrou M, Fayyaz M, Fathy M, et al. Deep-Anomaly: Fully Convolutional Neural Network for Fast Anomaly Detection in Crowded Scenes [J]. Computer Vision & Image Understanding, 2016:S1077314218300249.

[7]. Feng Z, Jie Y, Yao L. Patch-based fully convolutional neural network with skip connections for retinal blood vessel segmentation[C]// IEEE International Conference on Image Processing. 2017.