A Semantic Segmentation and Edge Detection Model Based on Edge Information Constraint Training

Longlong Wang 1, Fuxiang Liu1* and Jingqing Xu2

1 Key Laboratory of Dynamics and Control of Flight Vehicle, Ministry of Education, Beijing Institute of Technology, Beijing 100081, China
2 Shijiazhuang Campus of Army Engineering University

*13810272951@139.com

Abstract. The purpose of semantic segmentation is to classify the pixels within the target contour. Edge detection is another major basic vision task in machine vision. Today’s most effective semantic segmentation models and contour edge detection models are isolated networks. The edge of the output of the semantic segmentation model is coarse and cannot be directly used. And the output of the edge detection network cannot output the classification information of the pixels inside the contour. In view of the above shortcomings of the existing network, we propose a semantic segmentation model based on edge constraint optimization, so that the output of the semantic segmentation model has more delicate edge information, and the network directly outputs accurate contour edge graphs. The edge information output by the network can be directly used for tasks such as corner detection and center point detection. Experiments show that the mIoU statistics obtained by our model on the validation set of PASCAL VOC2012 can reach 83.9%. At the same time, more detailed edge details can be obtained. This algorithm has high engineering and theoretical research value.

1. Introduction

In recent years, increasingly accurate segmentation models have been developed. Most of these networks are implemented using CNN network layers with complex structures, such as models with feature pyramid pooling layers [1-3]. At present, semantic segmentation models are widely used in fields such as scene recognition, autonomous driving, and medical fields, which have greatly affected our daily lives. Edge detection is a basic visual task. Through edge detection, we can easily extract the position information of the target from the edge map, so as to achieve accurate positioning information. There are many ways to extract edge information or semantic segmentation results with more edge details. The first is to extract edge information based on traditional image processing methods. This method has low computational complexity and is easy to implement, but it is easily affected by the environment such as lighting, and detection failure often occurs.

Compared with the traditional image processing-based edge detection algorithm, this algorithm can accurately detect the edge of the target [4]. The RCF network [5] is a CNN network based on rich deep semantic features. It detects the image boundaries by performing semantic fusion on multi-scale feature maps and edge training. Compared with the first method, the implementation accuracy is very high, but the classification information of the edges cannot be obtained. To solve this problem, the concept of segmentation edge detection (SED) was proposed. The network model based on this
concept can detect semantic edge information well, such as CaseNet [6] and [7] was applied to the medical image field. However, this method cannot classify the pixels inside the contour.

The third method is firstly to obtain the mask of the target through semantic segmentation or instance segmentation, and then extract the boundary through the mask to obtain the contour edge information of the target. However, this model does not specifically trained by the edge information of the target, so the output contours edge do not perform well in detail, and cannot be directly used for edge detection tasks.

In order to optimize the edge information, the PointRend [8] method treats the segmentation task as a rendering method. It performs spatial constraint training on the semantic segmentation network and optimizes the traditional grid training method. Experiments show that the edge information of this network structure can achieve good results [9]. However, this method still requires complex post-processing algorithms to extract the edge information of the target.

In this article, we propose an edge constraint based semantic segmentation algorithm that uses both segmentation ground truth labels and target edge labels as training input labels, and edge labels as semantic segmentation constraints. This algorithm can directly output the edge detection results of different categories of targets. Experimental research shows that our algorithm has a good performance, and it can directly output accurate semantic edge information while obtaining accurate semantic segmentation results.

Subsequent articles will be described in the following structure. Chapter 2 will introduce the existing edge contour detection and semantic segmentation model algorithms. The third chapter introduces the semantic segmentation network model of edge information constraints. The fourth chapter presents some comparative experimental results of our algorithm. Chapter 5 summarizes the conclusions of this article.

2. Related works

The basic architecture of the existing mainstream segmentation network is a network model based on the encoder-decoder structure. The encoder network compresses the features of the image to obtain rich semantic information. This is a feature compression process. The second stage is the decoder network structure. The input of the network is the semantic feature information output by the encoder, and the output of the decoder is the segmentation map.

The decoder here can be understood as reconstructing the semantic information into segmentation mask information. The obtained segmentation mask will classify each pixel in the image. Finally, the contour and position information of the target is obtained. The encoder is actually a feature extraction network consisting of a convolutional layer module and a maximum pooling layer.

In the actual use of the process, we can customize the encoder and decoder respectively. For example, the basic convolutional layer is replaced with Atrous convolution, so that the network can increase the receptive field without increasing parameters [10]. In [11], the author used the residual network connection, which has stronger feature representation ability, so the training speed is improved and the accuracy is higher. In general, the loss functions we use are mainly loss functions such as Dice loss, cross-entropy loss, and Focal loss.

Edge detection is a major problem in computer vision. Due to its wide application in many advanced applications including object detection, object suggestion generation, and image segmentation, edge detection is a core low-level problem in computer vision.

CaseNet [10] network uses Resnet50 as the feature extraction network. This network mainly uses the shallow coarse features at different levels and the semantic features in the deep network layer to fuse. Finally, the output score map with the same shape as the original image is obtained.

\[
I(x, W) = \sum \sum -\beta (P(G_i^x)) \log \left( P(X_i^x) \right) - (1-\beta)(1- P(G_i^x)) \log \left( 1- P(X_i^x) \right)
\]  

(1)

Where \( P(G_i) \) represents the true probability of whether each pixel is an edge pixel. \( P(X_i) \) represents the predict probability of whether each pixel is an edge pixel.
It can be obtained from the above analysis that the semantic segmentation network does not perform specifically train on the edges, so the edges obtained by the semantic segmentation model will not be very fine and the edge contours will be inaccurate, which will affect the final evaluation results. The edge detection network only performs pixel-level classification on the edges of the target. On the other hand, the pixels inside the outline are all backgrounds, which easily cause the problem of imbalance between positive and negative samples.

3. Edge constrained semantic segmentation model

We propose a semantic segmentation network based on edge constraints. The semantic segmentation task is divided into two parts. During semantic segmentation task training, edge contour information is used to constrain the semantic segmentation task so that the network model can output edge information. At the same time, the inner pixels of the obtained edge contour are classified, and the two tasks are constrained with each other to obtain more accurate target segmentation results and edge detection results.

The network structure we proposed is also implemented based on the encoder-decoder architecture. The biggest difference from the encoder-decoder of the original semantic segmentation is the output layer, input layer and the loss function. In the rest of this chapter, we will describe the algorithm's input, output layers and loss function.

3.1. Input label processing

First, we extract the edge information of the ground truth label. The algorithm flow for extracting boundary information is as follows. The first step is to extract different categories of semantic masks from the ground truth mask. For example, VOC2012 is a 20-class semantic segmentation task. The shape of the extract result is $[n, nc, w, h]$, where $nc = 21$ (the background is the 21st category).

In order to deal with the overlap between different categories of masks and increase the proportion of positive samples, we processed the edge graph, as shown in Figure 1. The processing first uses the erosion operation of morphological algorithm, and then the output image is subtracted from the output of step one. And then we get the hard edge ground truth label. At last, we use the Gaussian blur operation in Figure 1 (d) to process the hard edges. After the above three steps of operation, we obtained the soft class-independent edge ground truth label.

As shown in Figure 2, the data input into the model is divided into three parts, the first part is the original input image. The second part is the original semantic segmentation mask of the ground truth label, and the third part is the soft class-independent edge label extracted from the second part. Therefore, the data dimension of the ground truth input to the model is $[n, nc, w, h]$. Taking PASCAL VOC2012 dataset as an example, the number of categories is 20, the background category and segmentation masks needs to be considered, so the number of channels for input labels is $nc = 22$.

3.2. Loss function

In order to optimize the network parameters and make the edge information have a constraint effect on the semantic segmentation task, so that the network can simultaneously output contour edges and segmentation mask maps, we processed the output of the last layer in Figure 2. The last layer not only has a pixel mask, but also a boundary mask.

Therefore, the loss function can be divided into two parts, the first part is semantic segmentation loss, and the second part is class-independent contour edge loss. For the semantic segmentation loss part, we use a loss function that adds dice loss and cross entropy loss together. The formula for this loss function is as follows:

$$S = \sum_{c}^{K} \left\{ \frac{2\sum_{j}^{N} p_{c}^{j} g_{c}^{j}}{\sum_{j}^{N} p_{c}^{j} + \sum_{j}^{N} g_{c}^{j}} + \sum_{j}^{N} \left( g_{c}^{j} \log p_{c}^{j} + (1 - g_{c}^{j}) \log (1 - p_{c}^{j}) \right) \right\}$$

(2)
Figure 1. We use two steps to get the soft edge ground truth label. (a) is the original image, (b) is the true label mask of the original image, and (c) the original true label image is subtracted from the erosion operation output image to obtain the contour boundary image. (d) Gaussian blur processing on the contour border image to get soft edge label.

Figure 2. The input as model training consists of three parts. The first is the input original image. The second part is the original ground truth mask label, his part of the input is used to train the semantic segmentation. The third part is the soft edge label image extracted from the second part. So the input shape of the label is $[n, num\_class + 2, w, h]$ (num_class=20 in VOC 2012 dataset)

Where $S$ represents the segmentation loss function, $N$ is the number of pixels in the sample. In $p_i^c$, $i$ indicates the pixel index, and $c$ is the index of the segmentation category. $g$ represents the ground true label.

We use the following formula to calculate the edge loss.

$$E_{ai} = \begin{cases} \alpha \cdot g_{ai}^c \cdot \log(p_{ai}^c) & \text{if } g_{ai}^c = 1; \\ \beta \cdot (1-g_{ai}^c) \cdot \log(1-p_{ai}^c) & \text{other} \end{cases}$$

$$E = \sum_{i=0}^{N} \sum_{c=1}^{K} E_{ai}$$

Where $p_{ki}^c$ is the predicted value of the kth category edge label, $g_{ki}^c$ is the predicted value of the cth edge label, and $E_{ai}$ is the cth edge loss of the ith pixel. Here, we use a binary cross entropy loss function for the edge loss of each category. The superscript $e$ indicates whether the current edge map pixel is predicted to be the background or edge class. Equation (4) shows that we separately calculate the edge loss of each category, and then sum the losses of all categories to get the total contour edge loss.

In order to solve the unbalanced samples of positive and negative samples during edge constrained training, we use coefficients $a = w \cdot \frac{G}{G^- + G^+}$, $\beta = \frac{G^+}{G^- + G^+}$. In order to highlight the edge information, we use $w$ to represent the weight map of the edge information in the input sample.

In essence, we have optimized the semantic segmentation task. While classifying each pixel as the foreground or background pixels, we also classify the foreground pixels, that is, determine whether the foreground pixels are edge pixels. We hope to realize the optimization of semantic segmentation tasks through this idea.

3.3. Network architecture and training

The schematic diagram of the network structure adopted by our model is shown in figure 3. In the figure, $PE$ and $TE$ represent the prediction value and ground truth of the edge information.
respectively, and $PS$ and $TS$ represent the predicted segmentation value and ground truth segmentation labels respectively. We use an encoder-decoder based network structure for segmentation tasks with edge constraints.

![Network structure diagram](image)

**Figure 3.** Network structure diagram used in this paper.

The resnet50 with $\text{stride}=16$ is used as the encoder of the network, and the convolution layer is replaced by the Atrous convolution layer, so that the encoder has a larger receptive field. In the decoder, the convolutional layer of the upper layer fuses the coarse features of the output of the encoder in different levels. The decoder’s upsampled stride is still 16.

The output layer of the network consists of two types of output layers, of which 21 channels are used for training of semantic segmentation, and semantic segmentation uses cross entropy and dice loss as loss functions. The edge loss uses a binary cross entropy loss for the edge label of each category, so the number of channels used by the edge loss is $21 \times 2 = 42$.

4. The experiment result

In this section, we have designed some experiments to verify the validity of the algorithms proposed in section 3.

We use the PASCAL VOC2012 semantic segmentation benchmark to test the performance of the segmentation tasks of different algorithms under the same mass network structure. The experiments were performed on a workstation with an Intel (R) XEON CPU, 32 GB RAM and four RTX2080Ti GPUs.

Before training, we performed data enhancement operation on the image. Firstly, the image was augmentation by flip, rotation and brightness adjust operation. During the training process, we used the Adam optimizer as the network optimizer. The learning rate was adjusted every 20 steps, the learning rate was adjusted from 0.001 to 0.00001. We use the same validation set to test different network structures. The test output is shown in Table 1. The figure 4 shows several samples randomly selected from the PASCAL VOC2012 test data set, and the prediction output of these samples.

The last column of the image is the and the output of semantic segmentation, and the right column is the boundary map of each type of output. From the experimental results, it can be seen that, because we use the edge constraint training method, on the same validation figures, the detection mIoU is higher than that of segModel [12], and the accuracy is almost close to Deeplabv3 [13], at the same time, compared to the results of the other two models, the semantic segmentation results output by our algorithm have more delicate and accurate boundary information. It can directly output the corresponding category edge information.
Table 1. Test results on different networks in the PASCAL VOC2012 semantic segmentation benchmark dataset

| Method  | Result/mIOU |
|---------|-------------|
| SegModel | 81.8        |
| DeepLabv3 | 84.5       |
| Ours | 83.9        |

Theoretically, after obtaining the mask of the semantic segmentation and edge information, we can fuse the results of the two outputs to optimize the results of the semantic segmentation output. At the same time, the edge information can be directly used to perform the center detection and corner point detection of the target.

5. Conclusion
This paper proposes a segmentation model based on edge constraint training. Our model can directly output segmentation masks with rich detailed edge information, and directly output accurate edge graphs with category information. The experimental analysis results show that compared with other algorithms, the proposed algorithm can obtain relatively excellent semantic segmentation results, and can effectively output the edge information of the target.

Reference
[1] Zhao, Hengshuang, et al. "Pyramid scene parsing network." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
Zhou, Bolei, et al. "Semantic understanding of scenes through the ade20k dataset." *International Journal of Computer Vision* 127.3 (2019): 302-321.

Dai, Jifeng, et al. "R-fcn: Object detection via region-based fully convolutional networks." Advances in neural information processing systems. 2016.

Deng, Ruoxi, et al. "Learning to predict crisp boundaries." Proceedings of the European Conference on Computer Vision (ECCV), 2018.

Liu, Yun, et al. "Richer convolutional features for edge detection." Proceedings of the IEEE conference on computer vision and pattern recognition, 2017.

Liu, Yun, et al. "Semantic edge detection with diverse deep supervision." *arXiv preprint arXiv:1804.02864* (2018).

Hatamizadeh, Ali, Demetri Terzopoulos, and Andriy Myronenko. "End-to-end boundary aware networks for medical image segmentation." International Workshop on Machine Learning in Medical Imaging. Springer, Cham, 2019.

Kirillov, Alexander, et al. "PointRend: Image Segmentation as Rendering." *arXiv preprint arXiv:1912.08193* (2019).

Taghanaki, Saed Asgari, et al. "Deep Semantic Segmentation of Natural and Medical Images: A Review." arXiv preprint arXiv:1910.07655 (2019).

Chen, Liang-Chieh, et al. "Rethinking atrous convolution for semantic image segmentation." *arXiv preprint arXiv:1706.05587* (2017).

Lin, Guosheng, et al. "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition, 2017.

Shen F, Gan R, Yan S, et al. Semantic segmentation via structured patch prediction, context crf and guidance crf[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 1953-1961.

Chen L C, Zhu Y, Papandreou G, et al. Encoder-decoder with atrous separable convolution for semantic image segmentation[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 801-818.