Research on Edge Detection Method Based on Improved HED Network

Li Lou\textsuperscript{1}, Shasha Zang\textsuperscript{1}

\textsuperscript{1}Computer Science and Technology, Xi’an Shiyou University, Xi’an, Shaanxi, 710065, China
zssjkb@126.com

Abstract. Aiming at the problem of rough and fuzzy edges generated by the current edge detection technology based on HED network, an improved edge detection method for HED network is proposed. First, the useful information captured by each convolutional layer of HED becomes rougher as the size of the acceptance field increases. The improved HED network makes use of all the information of the convolution layer to capture more targets in a larger range, or make the local boundaries of the targets possible. The improved HED makes full use of the multi-scale and multi-level information of the target to obtain high-precision and high-quality edge maps, which lays a good foundation for image segmentation.

1. Introduction

As the most basic computer vision problem, edge detection has a history of about 50 years. Traditional methods for edge or non-edge classification are based on local brightness, color, gradient and texture or other artificial design features\cite{1}. Generalized Boundaries from Multiple Image Interpretations is an article on TPAMI in 2014, which can be used as a representative. However, the edge is usually rich in semantic information, and it is difficult to obtain satisfactory results only through local cues.

Convolutional neural networks have also been used for edge detection in recent years, such as DeepEdge, N4-Fields, CSCNN, Deep Contour and HED. In the deep learning method, Ganin proposed N4-Fields through CNN and nearest neighbor search. Shen divided the contour data into sub-classes and fitted the sub-classes by learning model parameters. Hwang used contour detection as a pixel classification problem, used DenseNet to extract features for each pixel, and then used SVM for classification\cite{2}. Xie proposed HED to achieve image-to-image training and prediction. The network model is based on VGG16, which implements multiple side outputs and fuse these outputs to obtain edge detection results.

In summary, these CNN-based methods often only use the characteristics of the last layer of each convolution stage (VGG16 has 5 stages), ignoring the middle layer, which contains important details that other layers do not. By visualizing the output of different convolutional layers in the convolutional neural network, it is observed that the middle layer contains a lot of useful details, so this paper proposes an improved HED network to efficiently use the CNN features of each convolutional layer.
2. Edge detection

2.1. Comparison of edge detection algorithms
Early edge detection algorithms were built on the basis of image gradient operations, using the first-order or second-gradient information of the image to extract the edge of the image. Representative methods include Sobel operator and Canny operator. This type of gradient-based method has good real-time performance, but it is not robust, and it is easily affected by factors such as noise and lighting. With the development of the field of machine learning, many methods based on manual features have been proposed[3]. Although the edge detection algorithm based on manual features is also promising, its limitation is that the low-level feature information obtained is difficult to characterize the high-level information; most edge detection methods based on CNNs only use the last one of the convolutional network. Layer, ignoring the shallow details in the deep feature information, and it is easy to cause the network to fail to converge and the gradient disappears.

2.2. Holistically-nested edge detection (HED)
Holistically-nested edge detection (HED) performs image-to-image prediction through a deep learning model. The model uses a fully convolutional neural network and a deep-supervised network, and the overall nested edge detection (HED) There are two key issues: (1) Training and prediction based on the overall image, inspired by a fully convolutional neural network, used for image-to-image classification (the system takes the image as input and directly generates the edge image as output); 2) Nested multi-scale feature learning, inspired by deep-supervised networks, performs deep supervision to "guide" early classification results. The model is based on a fully convolutional neural network framework and attempts to use deep supervision technology and multi-scale learning technology to solve the edge blur problem[4].

2.2.1. HED network structure
The schematic diagram of the HED network structure is shown in figure 2:
2.2.2. HED loss function

(1) Training

Let the training set of HED be \( S = \{ (X_n, Y_n), n = 1, ..., N \} \), where \( X_n = \{ x^{(n)}_j, j = 1, ..., |X_n| \} \) represents the original input image and \( Y_n = \{ y^{(n)}_j, j = 1, ..., |X_n| \} \) represents the binary label map of \( X_n \), so \( y^{(n)}_j \in \{0, 1\} \), \( |X_n| \) are the number of pixels in an image.

Assuming that all the parameter values of the VGG-16 network are \( W \), if there are \( M \) side output layers in the network, then the parameter value of the defining molecule is \( w = (w^{(1)}, ..., w^{(M)}) \), and the objective function of HED regarding the side output layer is defined as:

\[
L_{side}(W, w) = \sum_{m=1}^{M} \alpha_m \ell_m^{side}(W, w^n)
\]  

Where \( \alpha_m \) represents the weight of the loss function of each side output layer, which can be adjusted according to the training log or both are 1/5. \( \ell_m^{side}(W, w^n) \) is the loss function of the output layer of each side, the loss function is a cross-entropy loss function of class balance:

\[
\ell_m^{side}(W, w^n) = -\beta \sum_{j \in Y^-} \log \Pr(y_j = 1 | X; W, w^n) - (1 - \beta) \sum_{j \in Y^+} \log \Pr(y_j = 0 | X; W, w^n)
\]  

Among them, \( \beta \) is suitable for the balanced balance weight of positive and negative samples for balanced edge detection, where \( \beta = \frac{|Y^-|}{|Y^-| + |Y^+|} \), \( 1 - \beta = \frac{|Y^+|}{|Y^-| + |Y^+|} \). Represents the number of \( |Y^-| \) non-edge pixels, then \( |Y^+| \) represents the number of edge pixels.

\( Y_{side} = \Pr(y_j = 1 | X; W, w^n) = \sigma(\alpha^{(m)}_j) \) represents the edge value predicted by the \( m \)-th output layer at the \( j \)-th pixel, and \( \sigma() \) is the Sigmoid activation function.

As shown in the figure, the fusion layer is expressed as a weighted sum of \( M \) side output layers, that is, \( \hat{Y}_{side} = \sigma \left( \sum_{m=1}^{M} h_m \alpha^{(m)}_{side} \right) \), and the loss function of the fusion layer is defined as:

\[
L_{fuse}(W, w, h) = \text{Dist}(Y, \hat{Y}_{fuse})
\]  

Where \( \text{Dist}(Y, \hat{Y}_{fuse}) \) represents the cross-entropy loss function.

Finally, the objective function when training the model is to minimize the sum of the side output layer loss \( L_{side}(W, w) \) and the fusion loss \( L_{fuse}(W, w, h) \):

\[
(W, w, h)^* = \arg \min (L_{side}(W + L_{fuse}(W, w, h)))
\]  

(2) Test

Given a picture \( X \), HED predicts \( M \) side output layers and a fusion layer:

\[
(\hat{Y}_{fuse}, \hat{Y}_{side}^{(1)}, ..., \hat{Y}_{side}^{(M)}) = \text{CNN}(X, (W, w, h)^*)
\]  

The output of HED is the average of the side output layer and the fusion layer:

\[
\hat{Y}_{HED} = \text{Average}(\hat{Y}_{fuse}, \hat{Y}_{side}^{(1)}, ..., \hat{Y}_{side}^{(M)})
\]  

2.3. Improved HED network

2.3.1. Improved to VGG16

(1) Remove all fully connected layers and the fifth pooling layer. The purpose of removing the fully
connected layer is to obtain a fully convolutional network, and to remove the fifth pooling layer, to avoid the step size being too large and increase the edge positioning error.

(2) Each convolutional layer in VGG16 is connected to a 1 * 1 * 21 convolution kernel and is divided into 5 stages. The features of each stage are combined through elwise layers to obtain fusion features.

(3) The merged features are then subjected to a 1 * 1-1 convolution kernel, and then deconvolution.

(4) The loss function of each stage is the cross information entropy, and finally the features of the five stages are concatenated to obtain the final features.

2.3.2 Optimized loss function
Set all areas marked as 0 as the background, define the mean value of the mark greater than $\eta$ as the boundary, and define the mean point between the two as the fuzzy point, and do not calculate the loss function.

So the calculation expression of the loss function of a single pixel is as follows:

$$L(X; W) = \begin{cases} 
\alpha \cdot \log(1 - P(X; W)) & \text{if } y_i = 0 \\
0 & \text{if } 0 < y_i \leq \eta \\
\beta \cdot \log P(X; W) & \text{otherwise}
\end{cases} \tag{7}$$

In the formula,

$$\alpha = \lambda \cdot \frac{|Y^+|}{|Y^+| + |Y^-|}$$

$$\beta = \frac{|Y^-|}{|Y^+| + |Y^-|} \tag{8}$$

$Y^+$ and $Y^-$ represent positive and negative samples, respectively, and hyperparameters use $\lambda$ to balance positive and negative samples. The sum of $X_i$ is the CNN feature vector of pixel i, and $y_i$ is the edge probability value of pixel i. $P(X)$ represents the standard Sigmoid function, and W represents all the parameters to be learned by the network structure in this paper. Therefore, the loss function after optimization is as follows:

$$\text{Loss}(W) = \sum_{k=1}^{K} \sum_{i=1}^{I_k} L(X_{fuse}; W) + L(X_{fuse}; W) \tag{9}$$

In the formula, $X^{(i)}$ represents the CNN feature vector of the kth stage, $X_{fuse}$ is the feature vector of the final fusion layer, $|I|$ is the number of all pixels of the image, and K represents the total training stage (5 in this article).

2.4. Comparison with HED
The size of the receptive field of each layer of convolution in VGG16 is different. The advantage of the improved HED over HED is that its mechanism can better learn multi-scale information, which comes from different levels of convolution feature. In the improved HED, the deep-level features are relatively rough, and a strong response can be obtained for larger targets and part of the edges of the targets. At the same time, the shallow features can supplement the deep features with sufficient details[5].

3. Experimental scheme design and result analysis

3.1. Experimental scheme design
The experimental environment selected for configuration in this article is: The computer system is Windows10. The experiment is based on PyCharm development tools, using Python programming language and Caffe architecture. The basic network used by default is designed and implemented by VGG16.

In the training process, the initialization of the 1 * 1 convolutional layer at each stage uses a Gaussian distribution with zero mean, the standard deviation is 0.01, and the deviation initialization is set to 0. The initial value of the 1 * 1 convolutional layer kernel in the final fusion stage is 0.2, and the deviation is still 0. Stochastic gradient descent (SGD) randomly selects 10 images as samples in each iteration.

The parameters $\eta$ and $\lambda$ of the loss function are set according to the training data set. For a given edge probability map, a threshold needs to be set for binarization. There are two strategies to choose from. The first is to set a fixed threshold for all the images in the data set, called the optimal data set scale (ODS). The other is to set an optimal threshold for each image, called optimal single image (optimal image scale, OIS). Each strategy calculates its F-Measure.

$$\text{F-measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$  \hspace{1cm} (10)

In the formula: Precision is the precision rate, Recall is the recall rate.

3.1.1. Model training

BSDS500 is a data set used by scholars in the field of edge detection. It consists of 200 training images, 100 verification images, and 200 test images. Each image will be marked by 4-9 annotators. In this paper, the training and verification set are used to train the adjustment network, and the test set is used to evaluate the performance of the detector. The data amplification method is the same as HED. This paper also uses the BSD500 dataset as training data. During training, set $\eta = 0.5$, $\lambda = 1.1$.

| Method      | ODS   | OIS   |
|-------------|-------|-------|
| Canny       | 0.611 | 0.676 |
| SE          | 0.743 | 0.763 |
| OEF         | 0.746 | 0.770 |
| Deep Contour| 0.757 | 0.776 |
| Deep Edge   | 0.753 | 0.772 |
| HFL         | 0.767 | 0.788 |
| HED         | 0.788 | 0.808 |
| Improved HED| 0.806 | 0.823 |

3.2. Analysis of experimental results

The detection indicators of the edge detection model mainly include: Optimal Data Set Scale (ODS) and Optimal Image Scale (OIS). Among them: ODS refers to the detection result when all pictures in the test set use a fixed threshold; OIS refers to the detection result when using the best threshold for the current picture for each image[6].

The comparison between this algorithm and other algorithms is shown in Table 2:

| Algorithm | ODS   | OIS   |
|-----------|-------|-------|
| Canny     | 0.611 | 0.676 |
| HED       | 0.788 | 0.808 |
| Improved HED | 0.806 | 0.823 |
In order to further explore the effectiveness of the network structure proposed in this paper, this paper uses VGG16 to implement some hybrid networks, connecting the rich feature-side outputs to some convolution stages, while connecting the HED side outputs to other stages. The experiment is only trained and tested on the BSDS500, and in the single-scale state, the performance evaluation results are shown in the following table:

| Improved HED Stage | HED Stage | ODS  | OIS  |
|--------------------|-----------|------|------|
| 1,2                | 3,4,5     | 0.792| 0.810|
| 2,4                | 1,3,5     | 0.795| 0.812|
| 4,5                | 1,2,3     | 0.790| 0.810|
| 1,3,5              | 2,4       | 0.794| 0.810|
| 3,4,5              | 1,2       | 0.796| 0.812|
| —                  | 1,2,3,4,5 | 0.788| 0.808|
| 1,2,3,4,5          | —         | 0.798| 0.815|

The last two lines represent the evaluation results of HED and Improved HED. From other results, we can see that all these hybrid networks have better performance than HED, so it can prove the importance of the improved HED strategy proposed in this paper.

It can be seen from the comparison that the model in this paper improves HED, and the performance has improved to a certain extent. In addition, the comparison of the edge map output by the model in this paper and other algorithms is shown in the following figure:
It can be seen from the comparison that the edge map output by HED is rough and fuzzy, and the processing of details is not good, while the improved HED can combine high-level information and low-level feature information to improve the accuracy and robustness of edge detection. The edge map retains more detailed information and the lines are clearer.

4. Conclusion
In this paper, we propose an edge detection method that uses richer convolution features. Because objects in natural images have different scales and aspect ratios, learning rich hierarchical representations is essential for edge detection. The model in this paper can accurately extract the edges of the objects in the picture, improve the effectiveness and integrity of target edge detection, and lay a good foundation for image segmentation.

Acknowledgments
This work is supported by Xi'an Shiyou University Graduate Innovation and Practical Ability Training Program Funding and National Natural Science Foundation of China Youth Fund (F020517).

References
[1] Perlo J, Mulder C, Danieli E, et al. Digital processing of images of extruded rubber profiles for process control MRI[J]. Measurement, 2016, 82:466-475.
[2] XIE S,TU Z.Holistically-nested edge detection[J]. International Journal of Computer Vision, 2017, 125(1/2/3):3-18.
[3] LIU Y,CHENG M,HU X,et al.Richer convolutional features for edge detection[C]. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition.Piscataway,NJ:IEEE,2017:5872-5881.
[4] Ren J,Zeng X.An image segmentation method of improved sobel operator[J].Boletin Tecnico/Technical Bulletin,2017,55(1):193-199.
[5] Zhang Yueyuan, Zeng Qinghua, Li Yineng, et al. Improved image edge detection algorithm based on Canny [J]. Navigation and Control 2019, 18 (1): 84-90.
[6] Li Lili. Image edge detection algorithm and its application in traffic video segmentation [D]. Chongqing: Chongqing Jiaotong University, 2014.