End-To-End System Recognition Based on Improved Faster RCNN+CTC Text Detection

Ying Wang[a], Baolong Guo[b,*], Zhe Huang[c] and Cheng Li[d]

School of Aerospace Science and Technology, Xidian University, Xi' a 710071, China

[a]katy_wying@163.com, [b]blguo@xidian.edu.cn, [c]huangz@stu.xidian.edu.cn, [d]licheng812@stu.xidian.edu.cn

*Corresponding author

Keywords: Text Detection, Text Recognition, Faster R-CNN, LSTM, CTC

Abstract: In the current era of rapid development of science and technology information, and the rapid development of Internet technology and mobile terminal equipment, images have gradually become the main source of people's daily communication. The technology of extracting content in the image has become a focus of attention. This article mainly uses improved Faster R-CNN text detection and recognition end-to-end system to extract and recognize text information in images. This algorithm introduces a bidirectional LSTM network in the original Faster R-CNN network to retain the context information of the text, and uses Monte Carlo non-maximum suppression method to judge the slanted text box. This method detects the performance of long text and slanted angle text. There has been a significant improvement. Finally, CTC is used for text recognition.

1. Introduction

Text in images is one of the most important channels for transmitting information. Especially in the era of rapid development of science and technology information, along with the rapid development of Internet technology and portable mobile terminal equipment, images have become the main source of people's mutual communication. Images contain rich semantic information, so the extraction and understanding of text information in images is gradually becoming the hottest research direction in recent years. The extraction of text information is generally divided into two stages, that is, the position of the text area is obtained through the text detection network, the position is cut out by cropping, and then the text recognition result is obtained through a text recognition network. Among them, text detection is the most important first step, and it is also a difficult point in text detection and recognition. The problem has been explored to understand different methods. There are three main categories: sliding window-based methods, connected region-based methods, and deep learning-based methods. This article focuses on the Faster R-CNN algorithm when detecting text: insufficient detection of long text, missed detection of skewed text, etc. And combined with CTC mechanism to achieve end-to-end text recognition.

2. Faster R-CNN

Faster R-CNN contains 13 VGG16 convolutional layers, 13 relu layers, and 4 pooling layers. This module first uses the VGG16 basic convolutional network to directly extract the basic feature map of the image, and uses the obtained sequence feature map as the input of the RPN network and the CNN network; the acquisition of the text candidate area is performed on the RPN network model; then the pooling layer passes Pooling realizes extracting the feature map of RoI at this layer, inputting the feature map to the fully connected layer and determining the target category; finally, the rectangular area of the candidate area is classified and detected, and the precise position of the desired candidate area frame can be obtained. Figure 1 is a schematic diagram of the designed Faster RCNN algorithm.

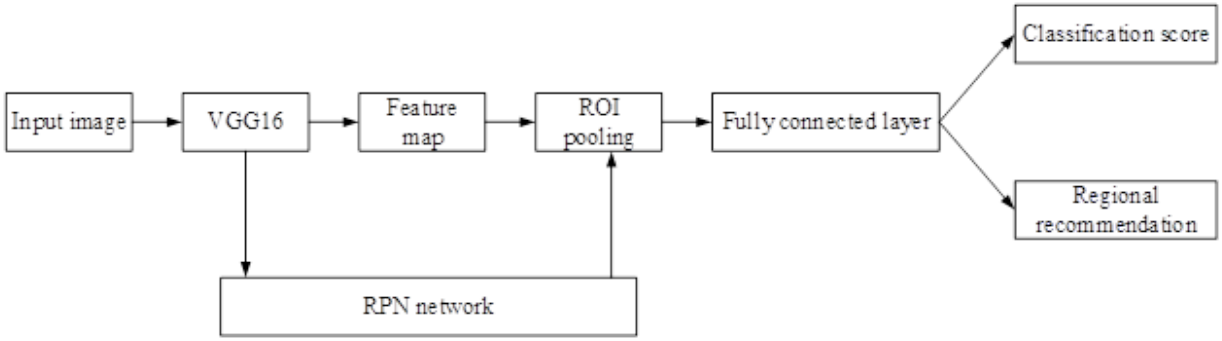

Fig. 1 Faster RCNN algorithm flowchart

3. Improved text detection method

3.1 Sequence feature extraction based on CNN

This article is similar to Faster R-CNN, which extracts sequence feature maps based on CNN. The VGG16 network model used consists of 13 convolutional layers, 3 fully connected layers, and 4 pooling layers, which are divided into 5 blocks in total, and each block is separated by the Pooling layer. In this paper, the convolution layer output of VGG16 is used to obtain the Conv5_3 feature map. The input image size in VGG16 is 224 × 224 × 3, and the number of channels increases sequentially from 64, first to 128, then to 256, and finally to 512. The size of the convolution kernel is set to 3 × 3 and the step size is 1. Each pixel on the left and right padding is used to ensure that the size of the feature map does not change. After obtaining the sequence features, it is used as the input of the LSTM network.

3.2 Context feature extraction based on bidirectional LSTM network

Long Short Term Memory Network (LSTM) is a variant of RNN that can store long-term gradient information, effectively forget and retain information. It solves the problem that RNN cannot handle long-distance dependencies. Figure 2 shows the LSTM unit structure.

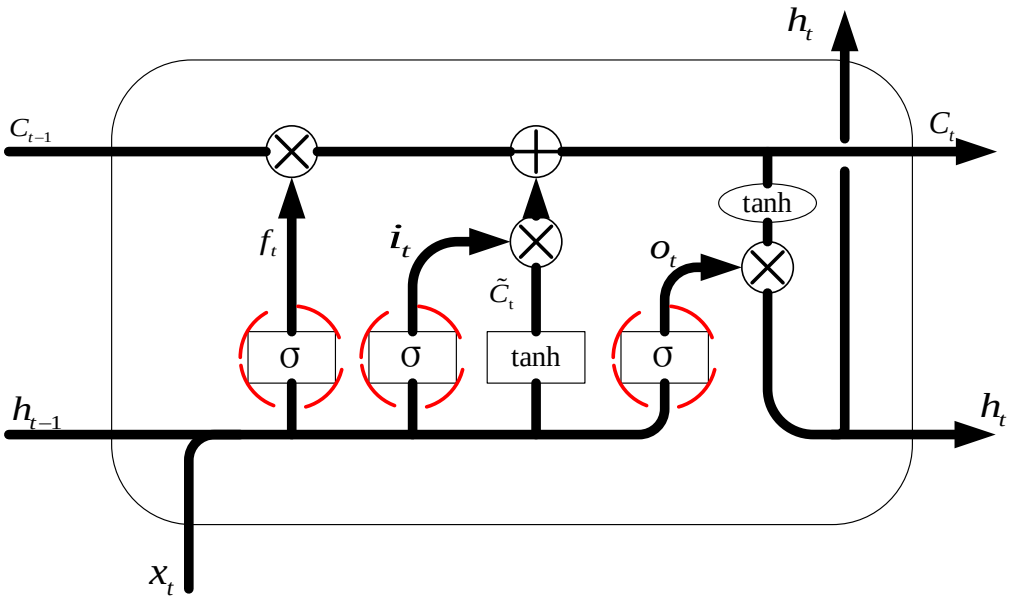

Fig. 2 LSTM unit structure

Forget gate: It is used to control the degree of forgetting the state of the previous unit. Decide which information should be forgotten and which information should be retained. This strategy is implemented through the sigmoid layer of the forget gate. Set input with the state of the unit

corresponding to the forget gate is, the output is a value between 0 and 1, 1 means completely reserved, 0 means completely forgotten. The formula for calculating the forgetting gate is:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

Among them, represents the sigmoid function of the forget gate, represents the output of the last moment, represents the input of the current moment.

Input gate: used to control the input information. The sigmoid layer of the input gate decides which update options are available. Then, the tanh layer creates new candidate values that need to be added to the current cell state and is recorded as Updating the current state of a unit is usually determined by a forget gate and the unit's input gate working together.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{3}$$

Among them, is the current new unit status, is the current new candidate vector state, is the read-write output of the current input gate, is the read -write output of the forget gate.

Output gate: It is used to control the output information. It is also the last step of LSTM, which determines the final network output content based on the content saved in the unit state. Similarly, the sigmoid layer is first run to determine which parts of the unit state to output. Then use the tanh activation function handles the state of the unit. Setting its value between [-1, 1], the output is:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{4}$$

$$h_t = o_t * \tanh(C_t) \tag{5}$$

The input of the bidirectional LSTM network is the sequence feature extracted by CNN, and the dimension of the sequence feature is 512. Then the feature dimension of the bidirectional LSTM network after extracting features in the forward and reverse directions is 1024. This can be used to represent context features. Fig. 3 shows the bidirectional LSTM structure.

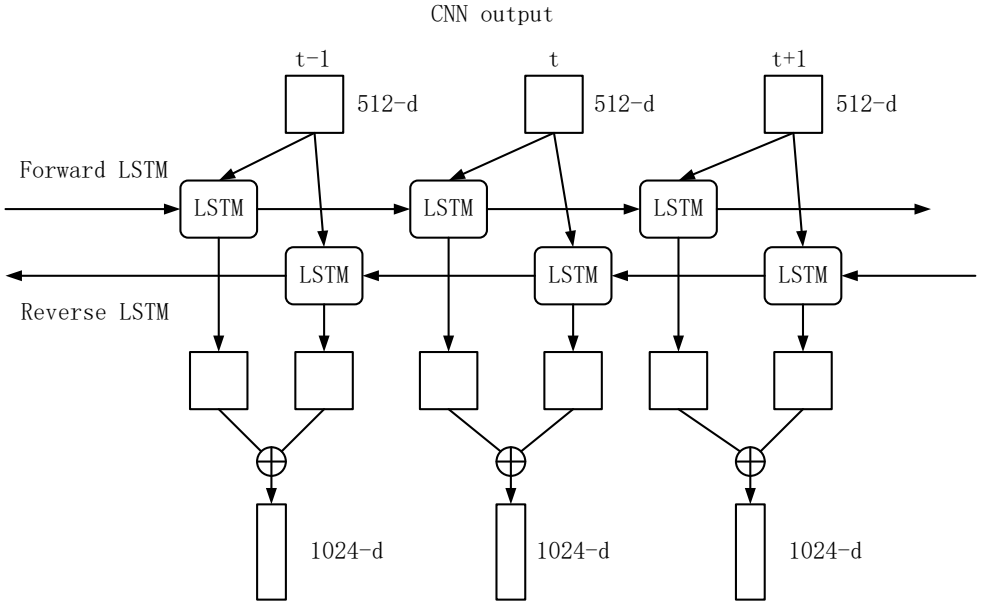

Fig. 3 Bidirectional LSTM structure

3.3 Candidate region extraction based on RPN

Region Proposal Network (RPN) is specially designed in Faster R-CNN to automatically extract a candidate frame. Its basic operation is to directly generate a fully connected feature with a length of 512 on a 3 * 3 sliding window on the conv5-3 convolution feature structure diagram. Then based on

this connected feature, two fully connected layers can be generated. One is used to analyze the coordinates and width and height of each center point and anchor point of the candidate target area; one is used to determine whether the candidate area is foreground or background. The RPN network loss calculation formula is:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{6}$$

$$L_{reg}(t_i, t_i^*) = \sum_{i\in\{x,y,w,h,\theta\}} smooth_{L1}(t_i - t_i^*)$$

$$smooth_{L1}(x) = \begin{cases} 0.5x^2 & if\ |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \tag{7}$$

$$t_x = \frac{x - x_a}{w_a},\ t_y = \frac{y - y_a}{h_a},\ t_w = \log(\frac{w}{w_a}),\ t_h = \log(\frac{h}{h_a}),\ t_\theta = \theta - \theta_a \tag{8}$$

$$t_x^* = \frac{x^* - x_a}{w_a},\ t_y^* = \frac{y^* - y_a}{h_a},\ t_w^* = \log(\frac{w^*}{w_a}),\ t_h^* = \log(\frac{h^*}{h_a}),\ t_\theta^* = \theta^* - \theta_a \tag{9}$$

In formula (6) L_{cls} represents the evaluation classification loss function, L_{reg} represents the evaluation candidate area loss function. Represents the probability value of predicting whether the candidate area is text, p_i^* represents the probability value of the text's true worth. L_{reg} Is defined in equation (7), w here $t_i = (t_x, t_y, t_w, t_h, t_\theta)$ represents the predicted scaling parameters at the time of classification, $t_i^* = (t_x^*, t_y^*, t_w^*, t_h^*, t_\theta^*)$ represents the true translation zoom parameter. In the following equations (8) and (9), $t_x, t_y, t_w, t_h, t_\theta$ are used to represent the coordinates, width, height, and scaling parameters of the center point of the predicted rectangular frame. Respectively, $t_x^*, t_y^*, t_w^*, t_h^*, t_\theta^*$ are used to represent the scaling parameters of the coordinates, width, height, and angle of the center point of the actual rectangular frame. $x_a, y_a, w_a, h_a, \theta_a$ Represent the coordinates, width, height, and angle of the center point of the rectangular area of the text output by the RPN model at the same time. $x^*, y^*, w^*, h^*, \theta^*$ Represent the coordinates, width, height, and angle of the center point of the actual rectangular area of the text, respectively.

3.4 Monte Carlo non-maximum suppression

Non-maximum suppression is an algorithm used to remove non-maximum values. The essence is to search for the maximum value in a local area, and suppress the elements that are not the maximum value. It can also be called a local maximum search. After detecting a rectangular frame of text, a threshold of a certain size is set in order to exclude the overlapping rectangular frames. If the overlap ratio (Intersection over Union (IOU)) meets the set value, it is considered that the subsequent detection results have appeared before, so the subsequent detection results can be excluded. According to this principle, the structure of the first detection is retained until all the results are compared. Repeat the above process in order to get the final text rectangle. The calculation method of IOU is:

$$IOU = \frac{A \cap B}{A \cup B} \tag{10}$$

In the formula, A, B represent two rectangular areas, $A \cap B$ is the intersection area of two rectangles, $A \cup B$ is the sum of the area of two rectangles.

Because images text in natural scenes are not all horizontal, most of them will have a certain tilt angle. In order to be able to calculate the overlap ratio of tilted rectangular frames, the Monte Carlo method is introduced and a sufficiently large sample size is set, then the frequency of the event can be regarded as the probability of the event.

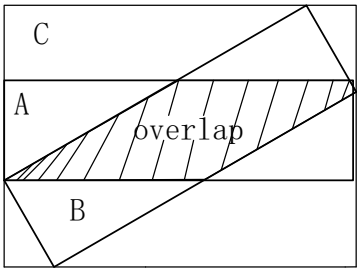

Fig. 5 Example of slanted text

The shaded area in the figure is the overlapping area of two rectangular boxes. The overlapping area of the oblique rectangular boxes is solved using the Monte Carlo method. The first step is to determine whether there is an intersection between the two rectangular frames. If they do not exist, IOU is 0; Second, if the two rectangular frames have an intersection, take the largest circumscribed rectangle of the two overlapping rectangles and record it as C; in the third step, collect 10,000 samples from C, and count the sample points located in the overlapping area, and record it as N; In the fourth step, according to the Monte Carlo principle, the ratio of N to the total sample points can be regarded as the area ratio of the rectangular A, B overlapping area to C.

4. Text area recognition method

The text area recognition uses the CTC transcription mechanism. The CTC transcription mechanism can map the output of LSTM to each position into a string that needs to be recognized. It does not need to train the network by the calculation method of Loss alignment. CTC adds empty characters to the mark symbols, marked as "_", you can directly predict the probability of the output sequence. CTC finds the probability of all possible strings, which is the product of the probability at each position, and then uses the string with the highest probability as the final output if there are empty characters between characters, remove the empty characters and do not merge the same characters after removing them. If there are no empty characters between characters, you need to merge the same characters. Assume that an output from LSTM is $\pi=\{\pi_1,\pi_2,...\ \pi_n\}$, and the corresponding label is $I, m<n$. The purpose of CTC is to convert a function B Maps to I, which is $I=B(\pi)$. Let $y_{\pi t}^t$ be the probability that the output at time t is π_t. Assuming that each output is independent, one of them matches $l=B(\pi)$. The probability of the path is:

$$p(\pi \mid x)=\prod_{t=1}^{T} y_{\pi t}^t \tag{11}$$

The probability mapped to I is:

$$p(I \mid x)=\sum_{\pi \in B^{-1}(I)} p(\pi \mid x) \tag{12}$$

5. Experimental results

This experiment verifies the recognition of long text, text with oblique angle and text content by the algorithm proposed in this paper. The experiment was completed on a computer. Ubuntu 16.04 system, using Python language, Tensorflow framework, CPU is Intel (sR) Core (TM) i5-7500@3.40

GHZ, GPU is NVIDIA GTX1050Ti, using Opencv3.6 image processing library. Training is performed for 50,000 iterations. Fig. 7 (a) is long text detection, and Fig. 7 (b) is with Text detection at oblique angles. Fig. 8 shows the output of the end-to-end text detection and recognition system.

Fig. 7 (a) is long text detection

Fig. 7 (b) is with Text detection at oblique angles.

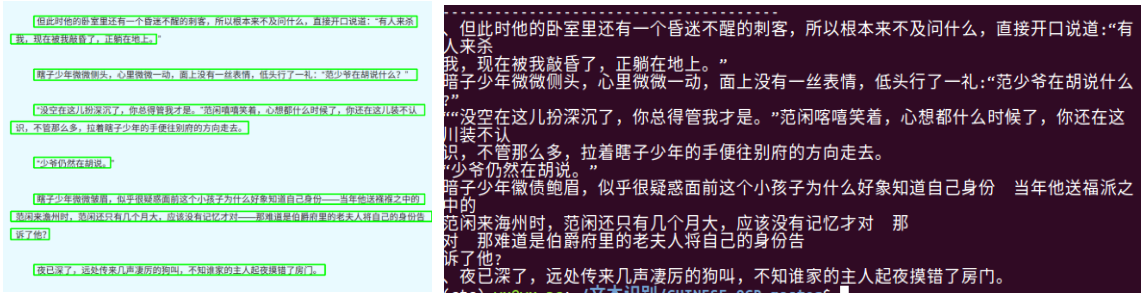

Fig. 8 End-to-end text recognition results

6. Summary

This paper introduces an improved end-to-end text detection and recognition method based on Faster R-CNN + CTC. It extracts text context information through a bidirectional LSTM network, and determines text rectangles through non-maximum suppression based on Monte Carlo methods. Finally, the CTC transcription mechanism is used to identify the text content. From the experimental results, this method can effectively detect long text and oblique text, and can extract characters from image text. However, the shortcoming of this method is that the accuracy of text recognition for images combined with Chinese and English is slightly worse, which needs further research.

Acknowledgments

This research is supported financially by National Natural Science Foundation of China (Grant No. 61571346).

References

[1] Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition [J].Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[2] M. Busta, L. Neumann, J. Matas. Deep textspotter: An end-to-end trainable scene text localization and recognition framework[C]. Proceedings of the IEEE International Conference on Computer Vision. Venice, Italy, 2017: 2204-2212.

[3] A. Graves, F. Gomez. Connectionist temporal classification:labelling unsegmented sequence data with recurrent neural networks [C]. International Conference on Machine Learning. Pittsburgh,Pennsylvania, USA, 2006:369-376.

[4] Shi B, Bai X, Yao C. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 39(11):2298-2304.