Natural Scene Chinese Character Text Detection Method Based on Improved CTPN

Wenxian Zeng¹ Qinglin Meng¹* Shuqing Zhang¹

¹ Hebei University of Economics and Business, Shijiazhuang City, Hebei Province, China
*Corresponding author’s e-mail: 2583813020@qq.com

Abstract. Text detection in natural scenes, due to differences in size, font, line direction, lighting conditions, text weakness and image background complexity, plays an important role in the research field and remains a challenging and important topic. We have improved the CTPN text detection network and changed the Side-refinement detection box to determine the scaling mechanism. And based on the experiment, change the LSTM network to the GRU neural network. In the dataset of Chinese character text game in natural scene released by Meituan, it reached 0.78 F1-Measure, which reached 0.89 and 0.61 F1-Measure on the ICDAR 2013 and 2015 data sets respectively. Compared to the 0.88 and 0.61 F1-Measure in the CTPN article, there is a big improvement.

1. Introduction
With the rapid development of Internet technologies and portable mobile devices, more and more application scenarios require the use of textual information in images. At present, natural scene text detection has become a research hotspot in the field of computer vision and pattern recognition, document analysis and recognition. Effective scene text detection can enhance the performance of many multimedia applications, such as mobile visual search, content based image retrieval and automatic symbol translation.

However, the research on Chinese character text detection in natural scenes is less and has great research significance. Because Chinese character natural scene data sets are few, writing is random, and many fonts are similar, the related research is less and the recognition effect is not good. However, many natural scene images require Chinese character recognition. If the Chinese character recognition technology in natural scenes breaks through, it will be a content-based image retrieval technology, automobile driverless technology, intelligent transportation system, and visual perception assisting technology for the Chinese character field. And other fields of technology have made tremendous contributions. In addition, the breakthrough and development of this technology will play a more important role in promoting other fields, with far-reaching theoretical research significance and broad application prospects.

In recent years, the better text detection method is the CTPN (Connecti-onist Text Proposal Network) proposed by Zhi Tian in 2016. This method combines the convolutional neural network with the bidirectional cyclic neural network (BiLSTM) for the first time, and makes clever use of text. The semantic sequence characteristics of the information reached 0.88 and 0.61 F1-Measure on the ICDAR 2013 and 2015 data sets, respectively.
This paper has improved on the basis of CTPN and applied it to the natural scene Chinese character data set, which achieved good results.

The content of this paper is arranged as follows: Section 1 introduces the existing methods of text detection and related work; Section 2 introduces the idea of CTPN network; Section 3 introduces the core algorithm improvement model of this paper; Section 4 shows the experiment and results; Section 5 summarizes the results.

2. Related Work

Literature [1] proposed a hierarchical text detection strategy. This method first uses CNN to extract features, obtains seed text from the obtained maximum stable extreme value region, and locates other degraded text regions according to the seed text, and then uses random forest combination. The context information of the text line finely classifies the text candidate area. In [2], the Faster R-CNN is improved. The Inception RPN method is used to obtain the text candidate region, then, a text detection network is used to remove the background region. Finally, the overlapping detection results are voted to obtain the optimal detection result. In [3], the RNN is introduced into the scene text detection for the first time. The CNN is used to obtain the depth feature, then, the fixed width Anchor is used to detect the text proposal area, and the feature corresponding to the same line Anchor is input into the RNN for classification. Finally, the correct text suggestion areas are merged. This method benefits from the use of sub-blocks (Block, Anchor) to represent the text, so it can also solve the problem of text direction change to a certain extent.

In [12], for the classification of words, CNN and RNN are jointly trained. First, the image features are extracted by standard CNN and expressed as feature vectors by Map-to-sequence. Then, the scene text is learned by using bidirectional LSTM (BLSTM). Spatial context information, Finally, the features are encoded and the final prediction results are obtained. This method combines the detection and recognition models to obtain the best text detection results in the current end-to-end model. In [4], the SSD (Single shot detector) detector can cope with the problem of text detection in any direction by adding direction information. This method predicts text fragments and their connection relationships in multiple scales, and converts text information into two locally detectable information. That is: text level or word level Segments and Links between Segments. The innovation is to add these Links to the network to learn, so that the network automatically learns which Segments belong to the same text line (or word). For the first time, the literature [5] uses a Fully Convolutional Network (FCN) to process images from the pixel layer. This method first uses the Text-block FCN for pixel-level calibration to obtain the probability that each pixel belongs to the text. Further, a text area salient map is obtained, and finally a text candidate region is obtained based on the salient map. Literature [6] proposed a Cascaded Convolutional Text Network (CCTN), which uses cascading to detect text. The specific processing steps mainly include: First, a Coarse-CNN is used to detect the rough. The text area is then judged whether the obtained text area detection result needs to be further processed (Re-fine), and if necessary, processed by Fine-CNN to obtain a more detailed text line for output. In [7], a simple and efficient text detection framework based on Full Convolutional Neural Network (FCN) and Nonmaximum Suppression (NMS) is proposed. This method first outputs the text region pixel level through the full convolutional neural network. The result is detected, and then the above result is obtained by a non-maximum suppression algorithm.

3. Introduction to CTPN Network

Before the CTPN model was proposed, the better model was Faster R-CNN [8]. At that time, many text localization algorithms optimized it. However, Faster R-CNN did not consider the characteristics of the text itself. Text lines generally exist as horizontally long rectangles, and each word in the text line has an interval, and there is semantic association between the texts. In response to this feature, CTPN thinks of a novel idea. They split the task of text detection. The first step is to detect a part of the text box and determine if it is part of a text. After all the small text boxes in a picture are detected, the small text boxes belonging to the same text box are merged, and after combining, a complete and
large text box can be obtained, and the text detection task is completed. Therefore, in the process of detection, CTPN introduces a mathematically similar "differential" idea, as shown in Figure 1 [3], first detecting a small, fixed-width text segment. In the post-processing part, these small text segments are connected to obtain a text line.

![CTPN differential thought map](image1)

**Figure 1. CTPN differential thought map**

The specific implementation process of CTPN is:
- Use the first 5 Conv stages of VGG16 to get the feature map, the size is W*H*C;
- Use the 3*3 sliding window to extract features from the feature map in the previous step, and use these features to predict multiple anchors. Here, the anchor definition is the same as the definition in the previous fastener-rcnn, that is, defining the target candidate area;
- Input the features obtained in the previous step into a bidirectional LSTM, output the result of W*256, and then input the result into a 512-dimensional fully connected layer (FC).
- Finally, the output obtained by classification or regression is mainly divided into three parts. According to the above figure, from top to bottom, it is 2k vertical coordinates: indicating the height of the selection box and the coordinates of the center y axis; 2k scores: indicating k anchors Category information indicating whether it is a character; k side-refinement indicates the horizontal offset of the selection box. In the experiment, the horizontal width of the anchor is 16 pixels, that is, the unit of the minimum selection box is "16 pixels";
- Using an algorithm constructed with text, the resulting elongated rectangle (shown in Figure 2 [3]) is then merged into a sequence box of text.

![CTPN final test results](image2)

**Figure 2. CTPN final test results**

4. Improved CTPN network

Our improvements include improvements to the Side-refinement detection frame merging mechanism, taking height information into the detection location and combining, and also changing the BiLSTM network to GRU, thereby accelerating network training and application runtime and improving network efficiency.

4.1. Side-refinement merge mechanism improvement

There are many authors' implementation methods in CTPN that require special attention: Detecting Text in Fine-scale proposals (selecting the anchor, which is the candidate "rectangular differential box"), Recurrent Connectionist Text Proposals (using the context) The RNN process of this information), Side-refinement (text construction, combining multiple proposals into a straight line). Among them, the task of the Side-refinement stage is to combine and summarize the positioned "small rectangles" to obtain the position information of the required text information. The last reserved small rectangle is the case where score>0.7 is required, that is, the small red rectangles in the following figure are merged, and finally a large yellow rectangle is generated, as shown in Figure 3 [3].
Figure 3. Results of Side-refinement

The main idea is that each two similar proposals (that is, candidate areas) form a pair and merge different pairs until they can no longer be merged. Judging two proposals, Bi and Bj can form a pair with Bi->Bj and Bj->Bi; and because the specified return box has a width of 16 pixels, it will cause some position errors, so here defines Side-refinement, the defined formula is as follows [3]:

\[ o = \left( x_{side} - c_x^a \right) / w^a \]  \[ [1] \]

\[ o^* = \left( x_{side}^* - c_x^a \right) / w^a \]  \[ [2] \]

Among them, the band * is represented as GroundTruth. Xside represents the left or right boundary of the regression, cxa represents the abscissa of the center of the an-chor, and wa is a fixed width of 16 pixels. So the definition of O is equivalent to a scaled scale that stretches the result of the box after the regression, thus better matching the position of the actual text.

However, in the case of the block character of Chinese text, CTPN only defines the left or right boundary of the positioning frame, and since the Chinese text in the natural scene is generally at the same height position, this article sets the Side-refinement definition, taking into account the text height bounding box, forming a new definition of the following scaling ratio O.

\[ o = \left( x_{side} - c_x \right) / \left( y_{side} * w^a \right) \]  \[ [3] \]

\[ o^* = \left( x_{side}^* - c_x^a \right) / \left( y_{side}^* * w^a \right) \]  \[ [4] \]

Among them, the band * is represented as GroundTruth. Yside represents the left or right boundary of the regression. The rest is the same as the original paper. Yside considers the text height information within the zoom ratio, and combines the more consistent height information of the Chinese text of the scene, so that the combined detection frame is more accurate at the text height. The test result of Chinese text is shown in Figure 4. The green box is the improved CTPN test result, and the red box is the CTPN test result. It can be seen that CTPN does not perform well when detecting vertical text, and the improved CTPN considers the height information (yside), and the detection performance is better improved when detecting Chinese text.

Figure 4. Comparison of CTPN (green) and CTPN (red) test results

4.2. BiLSTM changed to GRU

In the CTPN paper, the Recurrent Connect-ionist Text Proposals stage uses the bidi-rectional LSTM pair to feature the feature extraction layer of the VGG16 and the feature of the 3*3 sliding window, and then performs feature sequence prediction, and finally obtains a feature sequence with a depth of 256. When the GTX 1060 GPU performs network training on the Chinese character text data set published by Meituan, the training loss is slowed down, and the loss rate changes as shown in Figure 5, and still trained for 17 hours in the GPU-accelerated environment.
Due to the slow decline in the training process and the slow convergence, this paper adjusts the BiLSTM network to a GRU (Gated recurrent unit) network in the improved CTPN network [9]. The comparison between LSTM and GRU is shown in Figure 6 [9]:

GRU is a good variant of the LSTM network. It is simpler and more effective than the LSTM network, so it is also a very manifold network. It can still solve the long dependency problem in the RNN network. Three gate functions are introduced in LSTM: input gates, forgetting gates, and output gates to control input values, memory values, and output values. There are only two gates in the GRU model: the update gate and the reset gate. The structure of the two doors can greatly reduce the complexity of the model and the amount of parameters in the training process, so that the training loss rate is accelerated, the training time is shortened and the effect is stable. The training time is 3 hours under the GTX 1060 GPU, and the loss rate of the training process is shown in Figure 7.

5. Experimental description and comparison
This section mainly includes an introduction to the Chinese character text dataset, a comparative experiment on ICDAR 2013 and ICDAR 2015, a comparison of the CTPN and the improved CTPN, and an evaluation indicator, which includes statistics P, R, F, and run time T.

5.1. Data set and positioning effect display
The training and test datasets used 25,000 copies of the natural scene Chinese character text game data released by Meituan on March 18, 2019. Each image is made up of different individuals, completely
different, using different devices, different locations, different times and different environments. The data set is mainly Chinese text, and the label content is relatively complete. Each picture is marked with the position and text of a single character, as well as the position and text of each string. Among them, 20,000 pictures were used for training, 2000 sheets were used for verification, and 3000 sheets were used for testing.

The model formed by the improved CTPN network training is used to detect the position of the Chinese text. The effect is shown in Figure 8:
algorithm to the scene Chinese character data set, ICDAR 2013, ICDAR 2015, which has achieved good results.

References
[1] Xu H L, Su F. A robust hierarchical detection method for scene text based on convolutional neural networks. In: Proceedings of the 2015 IEEE International Conference on Multimedia and Expo. Turin, Italy: IEEE, 2015. 1-6.Xu H L, Su F. A robust hierarchical detection method for scene text based on convolutional neural networks. In: Proceedings of the 2015 IEEE International Conference on Multimedia and Expo. Turin, Italy: IEEE, 2015. 1-6.
[2] Li H P, Doermann D, Kia O. Automatic text detection and tracking in digital video. IEEE Transactions on Image Processing, 2000, 9(1): 147-156.
[3] Tian Z, Huang W L, He T, He P, Qiao Y. Detecting text in natural image with connectionist text proposal network. In: Proceedings of the 14th European Conference on Computer Vision. Cham, Switzerland: Springer, 2016. 56-72.
[4] Shi B G, Bai X, Belongie S. Detecting oriented text in natural images by linking segments. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA: IEEE, 2017. 3482-3490.
[5] Zhang Z, Zhang C Q, Shen W, Yao C, Liu W Y, Bai X. Multi-oriented text detection with fully convolutional networks. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA: IEEE, 2016. 4159-4167.
[6] He T, Huang W L, Qiao Y, Yao J. Accurate text localization in natural image with cascaded convolutional text network. arXiv preprint arXiv: 1603.09423, 2016.
[7] Zhou X Y, Yao C, Wen H, Wang Y Z, Zhou S C, He W R, et al. East: an efficient and accurate scene text detector. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA: IEEE, 2017. 2642-2651.
[8] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 39(6):1137-1149.
[9] Jun Y C, Caglar G, KyungHyun C, et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In: Presented in NIPS 2014 Deep Learning and Representation Learning Workshop.2014, 2361-2368.
[10] Yin, X.C., Yin, X., Huang, K., Hao, H.W.: Robust text detection in natural scene images. IEEE Trans.Pattern Analysis and Machine Intelligence (TPAMI) 36, 970-983 (2014).
[11] Neumann, L., Matas, J.: Efficient scene text location and recognition with local character refinement(2015), in International Conference on Document Analysis and Recognition (ICDAR).
[12] Busta, M., Neumann, L., Matas, J.: Fastext: Efficient unconstrained scene text detector (2015), in IEEE International Conference on Computer Vision (ICCV)
[13] Zhang, Z., Shen, W., Yao, C., Bai, X.: Symmetry-based text line detection in natural scenes (2015), in IEEE Computer Vision and Pattern Recognition (CVPR)
[14] He, T., Huang, W., Qiao, Y., Yao, J.: Text-attention convolutional neural net works for scene text detection. IEEE Trans. Image Processing (TIP) 25, 2529–2541(2016)
[15] Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images (2016), in IEEE Conference on Computer Vision and Pattern Recognition (CVPR)