Natural scene text detection based on multiscale connectionist text proposal network

Min Huang¹, Chaohao Lan¹ ², Wei Huang¹, Yang Tao²
¹School of Software Engineering, South China University of Technology, Guangzhou, People’s Republic of China
²Guangzhou Robust Ltd, Guangzhou, People’s Republic of China
E-mail: lanchaohao@qq.com

Abstract: The technique of recognising text in natural scene pictures is widely used in social production. For the existing identification methods, it is difficult to accurately identify in complex environments. The accuracy of the detection determines the efficiency of the identification. A text detection method based on Multiscale Connectionist Text Proposal Network is proposed. The Multiscale-Region Proposal Network regresses and classifies the extracted region to obtain the final candidate region. Taking a large number of commodity image samples as a dataset, the multi-scale joint text proposal network is used to detect and locate the text content area in the image. The experimental results show that the proposed algorithm improves the detection accuracy in complex environments.

1 Introduction

With the popularity of high intelligence mobile devices and the rapid development of the Internet, scene recognition becomes a research hotspot. In addition to speech recognition [1] and other perceptions [2], there is also image-based computer vision. Object-detection [3] is an important research object in the field of computer vision, and text detection [4] has also become an extremely important issue. The scene text includes many things, such as billboards, road signs and product packaging. The detection and identification of such texts requires consideration of a number of factors. The text has a wide variety of colours, sizes and fonts, and is affected by illumination and angle. In addition, because the environment in which the scene image is located is complex and diverse, the background in which the text is located is also difficult to determine.

Currently, the traditional methods of ordinary text detection include Stroke Width Transform (SWT) [5] based on connected component analysis [6, 7] and Maximally Stable Extremal Region (MSER) [8]. As an improved method of SWT, MSER can detect text feature regions more accurately, but the detection accuracy in complex environments is still not high enough. With the wide application of neural networks, more and more people use deep learning models to solve the problem of text detection. The current mainstream deep learning method is based on the target detection of Faster Region with Convolutional Neural Networks (RCNN) [9] and TextBoxes [10]. Among them, Faster RCNN is an algorithm based on target detection. It consists of VGG16 (Visual Geometry Group) [11] and RPN [9] networks, which can identify targets of different sizes. TextBoxes is an improved text image detection and localisation algorithm based on SSD [12], using VGG16 and consisting of six custom convolutional networks, each of which provides a fixed-size target marquee output. As a classic target detection algorithm, Faster RCNN is difficult to converge to a better value when the text image is detected. The reason is that the selection of the text candidate box in the Faster RCNN is based on the Region Proposal Network (RPN) network, which is more conducive to the identification of clear boundaries, and the target object whose boundary does not change drastically, and the text content as a target of dramatic horizontal change is difficult to recognise for Faster RCNN. In order to adapt to the characteristics that the length and width of the text content do not match, TextBoxes modifies the length of the default marquee and adjusts the convolution kernel to adapt to the changing characteristics of the text. However, TextBoxes does not effectively recognise text for overexposed areas, and the recognition of words with excessive spacing between characters is not efficient.

2 Multi-scale regional proposal network based on RPN

Aiming at the above problems, this paper proposes a natural scene text recognition algorithm based on Res-VGG16 depth feature extraction, named Multiscale Connectionist Text Proposal Network (MS-CTPN).

Based on CTPN [13], the MS-CTPN mainly includes the following three parts: a deep residual connection network Res-VGG16, a bidirectional recurrent neural network (Bi-RNN) and a Multi-scale Region Proposal Network (MS-RPN). As a key part of MS-CTPN, MS-RPN is the focus of this paper. It is actually a special computing network and consists of four key components: a text candidate box proposal module, a candidate box classification module, a candidate box regression module and a Region of Interest (ROI).

2.1 Text candidate box proposal

The recognition of the text content box by the RPN is similar to the target detection, that is, the text content can be regarded as a specific target, and a candidate box that may be selected to the target is provided. Calculate the vertical gap (regression score) between the candidate box and the real box, and whether the candidate box contains the real box (classification score), and judge whether the correct candidate box is found according to the final score. However, the biggest difference between the text box and the common target detection is that the similar objects in the target detection usually have relatively clear boundaries, and the text content as the target is difficult to have a clear boundary. In other words, the boundary of the text content as the target (horizontal direction) can vary drastically, depending on the length of the text content itself. Therefore, it is necessary to improve the candidate frame proposal method in the original target detection, so that the model can adapt to the text content boundary of multiple scales.

In order to adapt to the selection of text content at various scales, combined with the dramatic change of text content in the horizontal direction, this paper uses a method that can cover the general selection of text content. The method of splicing out a complete selection box by using a plurality of small selection...
boxes of fixed width, that is, changing the number of small selection boxes to achieve variable text content size, as shown in Fig. 1. A small box with a fixed width only needs to predict the vertical coordinate (y-axis coordinate) and height value (height in the y-axis direction) of the corresponding box of the pixel. It is not necessary to predict the lateral coordinate to reduce the difficulty of prediction. The text area with varying bounding length can be effectively obtained by combining the adjacent small box.

In Fig. 1, the anchor (small candidate frame) of the corresponding position is first generated by the region sequence feature, and then the anchor is positioned by the coordinates of the current position feature point. The coordinates of the current feature point are taken as the centre point coordinates of the anchor, and in the case where the hold width is fixed, the height value is changed to acquire anchors of different sizes of the current feature point. Each anchor corresponds to two sets of data: one is the position data of the anchor, including the coordinates of the vertical axis and its height; the second is the label in the marquee where the anchor is located. The tag indicates whether the box contains text content in the real data. If it exists and the overlap rate exceeds 0.7, the tag of the anchor is considered to be 1 (positive class), otherwise it is set to 0 (negative class). Obtaining the predicted anchor data requires the use of a real anchor, which is an anchor of the same width obtained from a real text region slice, with the ordinate and height not fixed. Based on the ratio of the area overlap with the real anchor, it can be determined whether the anchor can meet the requirements of containing the text content. Finally, all the anchor data (including the anchors of the positive and negative classes) can be obtained, and the anchor data of the sequence features of the entire region is ready.

### 2.2 Text candidate frame classification

After the anchor data is obtained, it is only necessary to compare the tag corresponding to the anchor data with the value corresponding to the feature vector in the region, and the difference between the predicted value of the feature vector of the corresponding region and the real region can be obtained. Then use this to guide the model to iterative learning in a certain direction. Therefore, the work of this part is mainly two points: align the anchor data with the feature vector; calculate the gap between the anchor data tag value and the predicted value of the feature vector.

There are two categories marked in the anchor data, that is, whether the anchor contains text content (positive or negative). The dimension of the anchor data is \((N \times W^2 \times H^2) \times A \times 2\), which means that the images of the N batches are converted into A anchors corresponding to \(N \times W^2 \times H^2\) feature points, where \(W^2 \times H^2\) represents the product of the width and height of the original region sequence features (the total number of feature points). However, the dimension of the region sequence feature data generated in the Bi-RNN portion is \(N \times W^2 \times H^2 \times h\), and the output data of the Bi-RNN layer needs to be fully connected to the dimension equivalent to the anchor data. Considering that each region sequence feature point will generate A anchors, the final anchor data and data features should be mapped to \((N \times W^2 \times H^2) \times A \times 2\) dimensions.

Therefore, the connection method as shown in Fig. 2 is adopted, where \(K\) is \(N \times W^2 \times H^2\).

The vector outputted by the fully connected layer will be cross-entropy calculated with the corresponding target anchor of the same dimension to obtain the corresponding category difference value. The training process is used as part of the loss function to guide the model to iterative learning until the difference between the output vector and the corresponding target anchor converges to a certain minimum value.

### 2.3 Text candidate box regression

The text candidate box regression module needs to use the location information and category information corresponding to the data anchor, wherein the category information and the classification process are processed in the same way, but the location information needs further processing. The position of the data anchor is determined by the position of the corresponding feature point and the height and length of the anchor itself, but the proposed data anchor does not coincide with the real data anchor. Actually, the corresponding regression value cannot be obtained by calculating the difference between the feature point position and the position of the proposed data anchor because the position of the feature point and the position of the proposed data anchor are one-to-one. Therefore, when acquiring the data anchor, the position information of the proposed anchor needs to be converted by the position of the real anchor, and the difference between the proposed data anchor and the real data anchor is represented by a converted score value \(\tau\). The conversion formula is as follows.

\[
\begin{align*}
    r_x &= \frac{x_a - x^*}{w^*}, \quad r_y = \frac{y_a - y^*}{h^*} \\
    r_w &= \log \left(\frac{w_a}{w^*}\right), \quad r_h &= \log \left(\frac{h_a}{h^*}\right)
\end{align*}
\]

In the formula, \(r_x\) represents the \(x\) coordinate difference score of the real anchor’s \(x\) coordinate (centre point) and the proposed anchor, \(r_y\) represents the difference score between the \(y\) coordinates, and \(r_w\) represents the difference score of the anchor width, and \(r_h\) represents the difference score of the anchor height. \((x^*_a, y^*_a)\) denotes the centre coordinates of the proposed anchor, and \((w^*_a, h^*_a)\) denotes the width and height of the proposed anchor (the real anchor with the * sign). Obviously, since the width values of all anchors are fixed, it can be concluded from (1) and (2) that \(r_x\) and \(r_w\) are always 0, that is, only the regression score is calculated for the \(y\)-axis coordinate and height. By first deriving the difference score between the proposed anchor and the real anchor, and then calculating the regression value between the feature vector and the proposed anchor, the regression score between the feature vector and the real text content box can be calculated. The next step is to map the proposed data anchor and feature vector to the same dimension as in the process of calculating the classification score, so that each feature point has a corresponding proposed anchor to calculate the regression value. Therefore, it is necessary to map the dimension of the region sequence feature of the Bi-RNN output from \(N \times W^2 \times H^2 \times h\) to \((N \times W^2 \times H^2) \times A \times 4\). The reason why it is multiplied by 4 is because it needs to be aligned with the centre point coordinates, width and height regression values \(t \rightarrow (r_x, r_y, r_w, r_h)\) in the proposed data anchor. In this paper, the difference regression method is used to calculate the regression difference between the feature vector and the proposed anchor, that is, the value of the feature vector is subtracted from the value of the corresponding data anchor. Finally, the optimiser iterative optimisation converges the subtracted difference to a minimum point.

### 2.4 Interest area extraction

After the previous three points of work, data preparation, classification and regression of the proposal network, the
classification loss and the regression loss need to be integrated, and the model loss in (3) can be obtained.

\[
\text{model_loss} = \text{cls_loss} + \text{reg_loss}
\]  

\[
\text{cls_loss} = - \sum_{t} \left[ a_t \log(z_i^{fci}) + (1 - a_t) \log(1 - z_i^{fci}) \right]
\]

\[
\text{reg_loss} = w \times (\Delta z - \delta)
\]

where \( z_i^{fci} \) is the \( i \)-th eigenvalue of the fully connected layer in the classification, and \( \Delta z \) is the output of the fully connected layer in the regression, and \( a_t \) represents the tag value of the \( i \)-th anchor. \( z \) is the sigmoid function that can output the probability distribution of the eigenvalues, and \( \delta \) is the four regression values of the proposed anchor \( [r_x, r_y, w, h] \). Through the iterative learning method, the model loss converges to the local minimum, and finally a comprehensive minimum value can be obtained in classification and regression, even if the proposed small text candidate box classification and regression reach a local best level. Since the first few steps are cut, the anchor size of the anchor that needs to be merged into a complete text box.

Due to the use of smaller fixed-size anchors, the number of proposed anchors is very large. Therefore, before the proposed anchors are merged, the anchors need to be screened, and only the positive anchors with an overlap ratio of 0.7 or more are retained. These filtered proposal anchors are then inversely transformed and converted to the size of the original image size. Finally, all adjacent or overlapping (possibly overlapping in the vertical direction) proposal anchors are merged using the text line construction method [14].

### 3 Experiment and analysis

The experimental dataset uses the natural scene text image multi-category dataset generated by SynthText [15]. SynthText uses a Ultrametric Contour Map (UCM) [16]-based segmentation algorithm to segment the original scene graph into multiple continuous color regions. Since the text content in the natural scene text image is generally on the surface of the background image, in order to obtain the same effect, SynthText adopts a pixel-level depth image extraction algorithm based on the deep convolution network. According to Random Sample Consensus (RANSAC) [17], a plane perpendicular to the normal color region normal vector in the scene graph is fitted, and then the text is placed on the plane, and the natural scene text image in which the text is naturally integrated into the background can be obtained. The dataset contains the main data types such as generated text image, text content, text box position and category label. The dataset contains a total of 12,000 samples, of which 10,000 are used as training data and 2000 are used as test data. (Fig. 3)

The evaluation indicators used in text detection experiments mainly include model loss, classification loss, regression loss, total loss and intersection over union. The model loss is the sum of classification loss and regression loss, while the total loss is the model loss plus the regular term. The smaller the loss, the more accurate the prediction result in the model training process. The classification loss, regression loss and model loss are shown in (3) through (5), and the total loss is calculated as follows:

\[
\text{total_loss} = \text{model_loss} + \lambda \sum_{w} |w| 
\]

where \( \lambda \) is a regular coefficient, and the larger the value, the larger the penalty. \( w \) is the weight of a layer in the model, and the network layer with a relatively large weight value is punished.

The intersection over union is a classic evaluation index applied in the target detection, which refers to the overlap ratio of the candidate box generated by the model and the ground truth box. In other words, the intersection of the candidate box and the real box is the value of the union. In the text image positioning task, the text content needs to be detected and positioned as the target, and the evaluation formula for the target candidate frame positioning accuracy is as follows:

\[
\text{IoU} = \frac{\text{area}(cb) \cap \text{area}(gt)}{\text{area}(cb) \cup \text{area}(gt)}
\]

where \( cb \) represents the candidate box; \( gt \) represents the real target box; \( area \) represents the area of the selection box.

In order to construct the MS-CTPN model, the parameters are divided into three parts: Res-VGG16 parameter setting, Bi-RNN parameter setting and RPN parameter setting. In the Res-VGG16 basic network, it mainly consists of a convolution layer and a pooling layer. The convolutional layer uniformly uses the convolution kernel \( (3 \times 3) \) and the same mode (keep the original feature size), and the number of convolution kernels between each set of convolutions is multiplied until the fourth set of convolution kernels is 512. The pooling layer connects each set of convolution operations, using a maximum pooling operation of \( 2 \times 2 \) range and \( 2 \times 2 \) stride, so that the feature vector height and width dimensions between adjacent two convolutions are reduced by half. After the fifth set of convolutions, there is also a \( 3 \times 3 \) convolution layer with a height and width stride of 1 as a feature map, and a mapping operation is performed on the features extracted by Res-VGG16.

The Bi-RNN part contains a bidirectional RNN network, each consisting of W LSTM (Long Short-Term Memory) cells, and the \( W \) value is determined by the size of the specific input image. Assuming that the input image size is \( 224 \times 224 \times 3 \), after Res-VGG16 and a feature mapping convolutional layer, a feature vector of \( 14 \times 14 \times 512 \) can be obtained. Then there are 14 LSTM cells corresponding to each, and each LSTM cell also contains the same number of \( 512 \) hidden nodes as the number of mapped feature vector channels.

The proposed anchor width for the RPN section is set to 16 and the height varies from 11 to 237. Each proposed anchor height varies in a factor of 1.4 from the previous one, and finally 10 proposed anchors are available. When calculating the classification loss and the regression loss between the candidate frame and the real marquee, the output dimension of the Bi-RNN layer is adjusted to connect the fully connected layers with widths of 20 (classification labels \( \times 10 \) offer anchors) and 40, respectively. Finally, the classification score and regression score of each pixel of the corresponding sequence region feature vector are obtained.

The remaining hyperparameters of MS-CTPN also include: learning rate, learning rate, rate of data, data epoch and so on. The learning rate is set to 0.0001, and is attenuated according to the attenuation rate of 0.95 during the training process. The attenuation condition is set such that the total loss of the model does not decrease during every 20 iterations. Since the algorithm itself cannot process multiple batches of proposed anchors at the same time, the data batch size is set to 1, the number of data rounds is set to 2, and the maximum number of iterations is set to 20,000.

Fig. 4 shows the changes in model loss, classification loss, regression loss and total loss of the three models of Faster RCNN, TextBoxes and MS-CTPN proposed in the test set, respectively, during the iteration of the test set 10,000 times. The blue line in the figure represents the error curve of the TextBoxes. It can be seen that the TextBoxes have a better effect than the Faster RCNN. Compared with the previous two methods, MS-CTPN is faster than
can get more accurate text content sequence region features, and the model as a whole can obtain better recognition effect.

4 Conclusion

This paper has proposed a multi-scale text content location method. The text content region feature in the text image was extracted by using Res-VGG16, and the region feature was serialised by Bi-RNN. Finally, the obtained sequence region feature was proposed by the MS-RPN region proposal network for the text candidate box. By minimising the regression loss and classification loss of the sequence region feature vector, the associated multiple size proposal data anchors can be obtained. According to the size difference between the feature vector and the real text image, the adjacent proposed data anchors were combined to obtain a text content prediction box on the real image. Finally, the prediction accuracy of the model can be verified by comparing the internal structure adjustment of the model, and the proposed MS-CTPN model can be found to be effective and the accuracy is relatively higher by comparing the final prediction results.

5 Acknowledgements

The project was partly sponsored by Guangdong Province Science and Technology Planning Projects (Grant: 2016B070704010), and Guangdong Province Science and Technology Planning Projects (Grant: 2016B010124010).

6 References

[1] Huang, M., Li, X., Wu, S., et al.: ‘Speech scenario adaptation and discourse topic recognition on mobile smartphone’, 2015 Int. Conf. on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 7–9 December 2015, pp. 486–491
[2] Huang, M., Bai, Y.-D., Liang, Q.-P.: ‘A context speculation method based on fuzzy logic inference in crowd-sensing process’. Int. Conf. on Wireless Communication and Sensor Network, WCSN2014, Wuhan, China, 13–14 December 2014, pp. 411–417
[3] Girshick, R., Donahue, J., Darrell, T., et al.: ‘Rich feature hierarchies for accurate object detection and semantic segmentation’. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 2014
[4] Yin, X.C., Yin, X., Huang, K., et al.: ‘Robust text detection in natural scene images’, IEEE Trans. Pattern Anal. Mach. Intell., 2014, 36, (5), pp. 970–983
[5] Epbinstone, B., Oke, E., Wexler, Y.: ‘Detecting text in natural scenes with stroke width transform’. CVPR 2010: Proc. of the 2010 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Washington, DC, 2010, pp. 2963–2970
[6] Chen, H.Z., Tsai, S.S., Schroth, G., et al.: ‘Robust text detection in natural images with edge-enhanced maximally stable extremal regions’. ICF 2011: Proc. of the 2011 IEEE Conf. on Image Processing, Piscataway, NJ, 2011, pp. 2609–2612
[7] Yi, C.-C., Tian, Y.-L.: ‘Localising text in scene images by boundary clustering, stroke segmentation, and string fragment classification’, IEEE Trans. Image Process., 2012, 21, (9), pp. 4526–4528
[8] Matus, J., Chum, O., Urban, M., et al.: ‘Robust wide baseline stereo from maximally stable extremal regions’, Image Vis. Comput., 2004, 22, (10), pp. 762–767
[9] Ren, S., He, K., Girshick, R., et al.: ‘Faster R-CNN: towards real-time object detection with region proposal networks’, IEEE Trans. Pattern Anal. Mach. Intell., 2015, 39, (6), pp. 92–99
[10] Liao, M., Shi, B., Bai, X., et al.: ‘Textboxes: a fast text detector with a single deep neural network’. arXiv:1611.08779v1
[11] Simonian, K., Zisserman, A.: ‘Very deep convolutional networks for large-scale image recognition’, arXiv preprint arXiv:1409.1556, 2014
[12] Liu, W., Anguelov, D., Erhan, D., et al.: ‘SSD: single shot multibox detector’. European Conf. on Computer Vision, Amsterdam, The Netherlands, 2016, pp. 22–27
[13] Tian, Z., Huang, W., He, Y., et al.: ‘Detecting text in natural image with connectionist text proposal network’, 2016
[14] Yu, C.C., Hsu, W.H.: ‘Automated text detection system in natural scenes using bottom-up-line construction’
[15] Gupta, A., Valdalski, A., Zisserman, A.: ‘Synthetic data for text localisation in natural images’. Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016
[16] Pont-Tuset, J., Arbelaez, P., Barron, J.T., et al.: ‘Multiscale combinatorial grouping for image segmentation and object proposal generation’. IEEE Trans. Pattern Anal. Mach. Intell., 2016, 39, (1), pp. 128–140
[17] Chum, O., Matas, J., Kittler, J.: ‘Locally optimized RANSAC’. Joint Pattern Recog. Symp., Magdeburg, Germany, 2003, vol. 2781, pp. 236–243