Text Detection in Trademark Images under Semi-Supervision

Xupeng Shi¹*, Shaozhi Wang¹ and Xiao Tan¹

¹Chongqing University of Posts and Telecommunications, Chongqing 400065, China;
*Corresponding author’s e-mail: s181231006@stu.cqupt.edu.cn

Abstract. The texts in trademark image usually contain characters, text lines and other different parts. Therefore, compared with the detectors that can detect text lines directly, the detector that can detect characters is more suitable for the detection of texts in trademark images. However, training character detectors needs a huge number of characters with position annotation, the construction process of this type of dataset will take a lot of time. In order to overcome the limitations of the detectors trained by datasets with word-level annotations and the shortage of datasets with character-level annotations, we propose a text detect method that can detect the single character in trademark images. Our proposed model is pretrained by the synthetic image dataset with character level annotation, then the pretrained model is used to detect the unannotated trademark images dataset to find more annotations for retrain the pretrained model. We also construct a trademark text dataset, which includes 4500 images. Experimental results on our dataset demonstrate our method can achieve the state-of-the-art performance compared to other methods. The results show our detector can detect complicated texts in trademark images.

1. Introduction

The number of infringement cases of the text in trademark images is increasing every year. That means the government or other users need efficient and automate solutions to judge whether the texts in trademarks are legal or not. So, as the first step and the most important part of these automate solutions, trademark text detection is very important.

In essence, the purpose of trademark text detection is to locate the text position in the image, so there is no difference between trademark text detection and text detection in other situations. such as Scene text detection.

Recently, text detectors show good detection effect. Text detectors usually rely on a large number of appropriate datasets for training. Unfortunately, we don’t find open datasets of texts in trademark images, most of the existing text datasets in other scenes do not provide character-level annotations, manually labeling these datasets will take a lot of time.

In this paper, we propose a new text method which can locate the single characters and group the detected characters into a text line. Our detector contains a convolutional neural network which can represent text characters by a single point at the center of their bounding box and the text characters’ size. In addition, our detector contains a text group module. The convolutional neural network is used to localize in-dividual characters in the image, and the text group module is used to group characters into text lines.

In order to make up for the deficiency of character-level annotation, we propose a semi-supervised learning strategy to train our model.

The idea of semi-supervised learning strategy is to pretrain the model by using the synthesized fully annotated character images, then the pretrained model is used to detect the unannotated
trademark images dataset to generate character level detection results, the results will be combined with the unannotated trademark images to make trademark images be a new dataset which is announced partly, then the dataset announced partly is further combined with the annotated synthetic images dataset to re-train the pretrained model to get the final stable character detection model.

we collect text from various trademark images and constructing a new dataset. This dataset contains 4500 images. For evaluation and comparison, we split 4000 images as training set and 500 for testing.

On the proposed dataset, the results demonstrate our trademark text detector can effectively detect the trademark texts and outperform other methods with a large margin.

2. Related Work
The major detectors of text detection including two types: The characteristics of the first type of detectors are the use of handcrafted features such as MSER[1] or SWT[2]. The second type of detectors are deep learning based text detectors.

With the deepening of the research on deep learning based text detectors, the existing detectors can be divided into four categories:

Regression-based text detectors usually adapted from popular object detectors such as Faster R-CNN[3] and SSD[4], unlike general objects, texts in trademark images usually have different types of shapes and different aspect ratios. To handle these problems, CTPN[5] combine CNN and RNN, the detector detect small, fixed width text segments. In the post-processing part, these small text segments are connected to get the text line. RRPN[6] tried to deal the problems more directly by redesign the anchor box and calculation method of IOU of Fast R-CNN[7], these redesign parts make the detector be able to detect text instances in horizontal and oriented forms. In recent, ContourNet[8] designed Adaptive-RPN module to generate higher precision and quality candidate boxes, and Local Orthogonal Texture-aware Module (LOTM) module to decouple the horizontal and vertical text contour detection in candidate boxes.

This type of detectors is design to detect text instances at the pixel level, without the limits of bounding boxes, this type of detectors can detect text instances in horizontal, oriented and curved forms. TextSnake[9] was proposed to detect text instances by predicting the text region and the center line together with geometry attributes; PixelLink[10] make text instances segmented out by linking pixels with the same instance together, segmentation-based text detectors usually suffer from linking different text instances together, PSENET[11] try to solve this problem by designing segmentation-based detector with multiple predictions.

3. The Proposed Method
Our text detector is designed to detect individual character in trademark images and group characters into text lines. The detector contains two parts, firstly we train a deep neural network to predict character regions. Since there is no public trademark text data set available, the model is trained by a semi-supervised strategy. Second we design a text group module to group each character into text lines.

3.1. Architecture
The text detector needs low-level information and high-level information at the same time because the size of the trademark text area vary tremendously, inspired by the idea of U-net[12], we design a fully convolutional network architecture to merge feature maps gradually. Due to some special circumstances in the trademark text, especially in the Chinese text, such as some Chinese radicals are Chinese characters at the same time, so sometimes the text detector will output two or three results in one character, if the detector use NMS, finally the detector will output wrong answers, so we detect characters in trademark images by exploiting the strategy which can detect object without the use of anchor box.

A schematic view of our model is showed in Figure 1. The model contains three parts: feature extractor part, feature-merging part and output part.
The network in the feature extractor part inherits from the resnet50[13] network model, in feature-merging part, conv5 features are up-sampled by deconvolution and merged with conv4 features by a merge layer. The merged conv5-conv4 features are merged with conv3 features and merged conv5-conv4-conv3 features are merged with conv2 features in the same way.

To avoid the using of NMS, the output part is inspired by the idea of Centernet[14], the output part use the feature maps produced by feature-merging part. The final output includes text/non-text classification, the position of the center points of text objects, the length and width of the text objects, and the discretization error caused by the output stride.

Figure 1. The example of our network architecture.

3.2. Label Generation
For each training image, we generate the ground truth label with character-level annotations. The annotations of bounding boxes usually in the form of \( (x_1, y_1, x_2, y_2, c) \) which contain the location of the characters, in the annotations, \( x_1 \) and \( y_1 \) denotes the coordinates of upper left corner, \( x_2 \) and \( y_2 \) denotes the coordinates of lower right corner, \( c \) denotes the category of the bounding box.

According to the annotation, we generate the ground truth which is needed for model training. The ground truth contains three parts:
- Get the corresponding center coordinates \( p \) in the original images, \( p \) is computed as:
  \[
  p = \left( \frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right)
  \]  
  (1)
- Get the center coordinates \( \tilde{p} \) of the feature map after downsample, \( \tilde{p} \) is computed as:
  \[
  \tilde{p} = \left\lfloor \frac{p}{R} \right\rfloor
  \]  
  (2)
where \( R \) denotes the down sampling multiple
- Get the keypoint heatmap \( Y_{xyc} \), \( Y_{xyc} \) is computed as:
  \[
  Y_{xyc} = \exp \left( -\frac{(x - \tilde{p}_x)^2 + (y - \tilde{p}_y)^2}{2\sigma_p^2} \right)
  \]  
  (3)
Where \( \sigma_p \) denotes a standard deviation related to the size of the target

3.3. Semi-Supervised Learning
Due to there is no public dataset of the texts in trademark images, so we use semi-supervised strategy to train our detection model.
We use the composition algorithm to synthesize a dataset $SD$ with character-level annotations, the dataset $SD$ is used to pretrain our detection model $M$, after that, the pretrained model is $M1$. Then we apply $M1$ to the unannotated dataset of text in trademark images to find more annotated data, the results of the detection in unannotated dataset of text in trademark images are a set of candidate character bounding boxes as well as the corresponding detection score:

$$C = \{(c_1, s_1), (c_2, s_2), \ldots, (c_i, s_i), \ldots\} \quad (4)$$

The true positive samples are selected by a confidence threshold:

$$P = \{c_i \mid s_i > S \text{ and } c_i \in C\} \quad (5)$$

where $s_i$ denotes the detection confidence of the $i_{th}$ detected character candidate $c_i$. The $S$ is the detection confidence threshold used to identify the positive samples. Our experiment shows that characters in trademark images can be well searched when $S$ is set to 0.3. If $S$ is too large, we may lose most of the positive samples else if $S$ is too small, we may get a lot of false positive samples.

The results of the detection in unannotated dataset of text in trademark images are used to make the unannotated dataset of text in trademark images be annotated, let the annotated dataset of text in trademark images be $D$. Finally, we integrate dataset $D$ and dataset $SD$ train the model $M1$ to get the final stable model.

3.4. Textline group module
Since it is easier to construct a dataset which characters are grouped into text lines so the texts in our dataset are grouped into text lines. We need to group the characters into text lines to output the final results after the character results are produced.

In fact, the text detection in trademark image is for subsequent text recognition, and different from the text detection task in other scenes, the arrangement of trademark text has certain regularity, so we

4. Experiments

4.1. Dataset
Our dataset contains 4500 images. The images are collected from Intellectual Property Office of Chongqing Intellectual Property Forum, which contain lots of horizontal and multi-oriented text. In addition, our dataset is multi-lingual with mainly Chinese and English text.

4.2. Annotation
The annotation format of our dataset is the same as icdar2017mlt, we use the four vertices of the rectangle to describe horizontal and multi-oriented text, and we use a polygon with eight points to describe curve texts.

To evaluate the method’s performance, the dataset is follow the PASCAL VOC protocol[15], which uses 0.5 IoU threshold to decide true or false positive.

4.3. Implementation Details
we take two steps to train our model: The first step is to train our model on the synthesized dataset, and the second step is to integrate the synthesized dataset and the marked real trademark text dataset for the second training.

In the first step, we synthesized a dataset of 10000 images with character level annotation and train our model with the dataset.

In the process of training, we first freeze the backbone, and set batch size 8, learning rate 0.001, the train epoch is set to 100. second we train the whole model and set batch size 8, learning rate 0.0001. the input images are resized to 512*512, the train epoch is set to 100.

Before the second step, we apply the model we trained in the first step on 4000 trademark images to generate character annotations in trademark images, the detection confidence threshold that is used
to identify the positive samples is set to 0.3. Due to the difference of image size between the synthesized dataset and the trademark image dataset, so we merge four trademark images into one image. Then we mix trademark images dataset and synthesized dataset together.

In the second step, we use the mixed dataset to fine-tune the model we trained in the first step. We set batch size 8, learning rate 0.0001. the train epoch is set to 100. the input images are resized to 512*512. The model generated in this step is the final model.
We use the final model to test the trademark images in the test set, the input images are resized to 256.

4.4. Experimental Results
For our dataset, texts are group into text lines. With text line group module, our detector can group the detected characters into text lines, the results of our method are showed in figure 2. In addition, our method is better than other methods as shown in Table 1.

| Method        | Precision | Recall | F-score |
|---------------|-----------|--------|---------|
| PAN[16]       | 0.802     | 0.592  | 0.681   |
| Seglink[17]   | 0.6802    | 0.4799 | 0.5628  |
| ContourNet[8] | 0.6507    | 0.6891 | 0.6694  |
| Ours          | 0.7887    | 0.9254 | 0.8516  |

Figure 2. Results on our dataset.

5. Discussions
The form of the texts in trademark images is very complex, sometimes the texts can not group into text lines, so our method has more advantages because we adopt the pipeline of detecting single character first and then grouping the detected characters into text lines. This is different from other detectors which rely on the features of text lines.

Our detector can detect multilingual texts in trademark images, but in some languages such Chinese and Japanese, there are huge number of different characters, so we need bigger train dataset than other detectors, It is very important to train our model with semi-supervised strategy.

6. Conclusion
In this paper, in order to detect texts in the trademark images we propose a text detector which can detect the characters in the trademark image in the first, then group the characters into text lines.

Since the character-level datasets are rare, we trained our model with semi-supervised strategy. Also in order to facilitate the contrast experiment and model training, we propose a new dataset with 4500 trademark images.
The experiment shows that our method shows the latest performance compared with other detectors. As future work we hope to combine our text detector with text recognize model to make our detector has higher accuracy and can adapt to text detection tasks in more scenes.

Acknowledgments
This research was funded by Chongqing Graduate Student Research Innovation Project: (CYS19257).

References
[1] J. Matas, O. Chum, M. Urban, T. Pajdla, Image and vision computing 22 (2004) 761-767.
[2] B. Epshtein, E. Ofek, Y. Wexler, Detecting Text in Natural Scenes with Stroke Width Transform, Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, 2010.
[3] S. Ren, K. He, R. Girshick, J. Sun, IEEE Transactions on Pattern Analysis & Machine Intelligence 39 (2017) 1137-1149.
[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S.E. Reed, C. Fu, A.C. Berg, european conference on computer vision (2016) 21-37.
[5] Z. Tian, W. Huang, T. He, P. He, Y. Qiao, european conference on computer vision (2016) 56-72.
[6] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, X. Xue, IEEE Transactions on Multimedia 20 (2018) 3111-3122.
[7] R. Girshick, Fast r-cnn, Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440-1448.
[8] Y. Wang, H. Xie, Z. Zha, M. Xing, Z. Fu, Y. Zhang, ContourNet: Taking a Further Step toward Accurate Arbitrary-shaped Scene Text Detection, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
[9] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, C. Yao, european conference on computer vision (2018) 19-35.
[10] D. Deng, H. Liu, X. Li, D. Cai, Pixellink: Detecting scene text via instance segmentation, Proceedings of the AAAI Conference on Artificial Intelligence, vol 32, 2018.
[11] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, S. Shao, Shape robust text detection with progressive scale expansion network, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9336-9345.
[12] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234-241.
[13] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778.
[14] X. Zhou, D. Wang, P. Krähenbühl, arXiv preprint arXiv:1904.07850 (2019).
[15] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, International journal of computer vision 88 (2010) 303-338.
[16] A. Neubeck, L. Van Gool, Efficient non-maximum suppression, 18th International Conference on Pattern Recognition (ICPR'06), vol 3, IEEE, 2006, pp. 850-855.
[17] B. Shi, X. Bai, S. Belongie, Detecting oriented text in natural images by linking segments, Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2550-2558.