A Real-time Character-level Subtitle Detector based on Faster R-CNN

Haipeng Chen, Qing Song*
Department of Automation, Beijing University of Posts and Telecommunication, Beijing 100876, China
chen6282@bupt.edu.cn, priv@bupt.edu.cn

Abstract. Faced with a deluge of video data, automatic subtitle text detection and recognition using computer vision technology can be extremely useful. This paper proposes a real-time character-level subtitle detector, which detects subtitle text with high accuracy and efficiency in a single forward pass. It also provides an efficient method to build dataset aimed at video subtitle text detection problem. This paper’s method can achieve both 99.5% precision and recall for text detection, and the whole detection and recognition process can achieve 45 fps.

1. Introduction
With the arrival of big data era, the volume of information is constantly increasing, which makes the information retrieval more challenging. Faced with a deluge of video data, how to quickly and effectively analyze and search video content becomes the priority research area. There are still lots of companies or institutions using manual inspection to get the information of video content, which is obviously inefficient and costly. It is a natural idea that let the computers do the video content analysis instead of humans.

Recently, computer vision technology has achieved great success, which becomes an important method for video analysis. To fill the gap between this general image information and rich semantic information that can demonstrate the video content clearly, it is no surprise that we need to utilize the subtitle information. Subtitle shows the dialogue and narration of video in the form of text, thus has extremely rich high-level semantic information for people to understand the video content better.

There are numerous methods for text detection, which can be roughly grouped into three categories:

Character based. Individual characters are detected first by trained network, then merged into words based on their spatial position and semantic association. Neumann and Matas [3] pose the character detection problem as an efficient sequential selection from the set of Extremal Regions (ERs). A sequential classifier is adopted for ERs corresponding to characters. The classification is broken down into two stages to improve computational efficiency. Yin et al. [4] use MSERs (Maximally Stable Extremal Regions) tree to extract character candidates, then construct text candidate by single-link clustering algorithm. An AdaBoost classifier is trained to decide whether a text candidate corresponding to true text or not.

Word based. Words are directly detected with the similar methods of general object detection. Max Jaderberg et al. [5] propose an end-to-end text spotting pipeline using Edge Boxes and a weak aggregate channel features (ACF) detector to extract word bounding box proposals. These proposals
are filtered with a random forest classifier and a CNN is trained for bounding box regression. He et al. [6] propose deep direct regression network (DDRN) that directly outputs values corresponding with the position and size of the text from a given point. The text bounding boxes or text line segments got are grouped by a line grouping method.

**Text-line based.** Zhang et al. [7] train a Text-Block FCN to predict the salient map of text regions in a holistic manner. Then extract the character components within the text blocks by MSER to generate text lines. A small version of the Text-Block FCN called Character-Centroid FCN is ultimately employed to filter non-text lines.

This article proposes a real-time character-level subtitle detector that fits several languages using region-based convolutional neural network (R-CNN) [1]. It directly outputs the coordinates of character bounding boxes and classification results of these characters by cascade of a region proposal network [2] and the following fine classification network.

This paper’s key contributions are summarized as follows:
1. A large-scale subtitle detection dataset is set up called Sub-Priv. It consists of over 200,000 images with over 2 million labeled subtitle characters, which is widely sampled from various open source videos and labeled using our automatic method.
2. A quick and efficient trainable network based on region proposal method is developed for subtitle detection. A lightweight network with high accuracy is found by comparative experiments.
3. Some useful methods are found to combine the prior knowledge with the neural network including designed loss function, anchor matching strategy, etc.

2. Methods

2.1. Dataset

There is no public dataset aim to this specific problem, thus this paper sets up a synthetic subtitle dataset to train the text detection and recognition model.

The data generation pipeline can be summarized as follows (Fig. 1). Firstly, write the text to be generated to subtitle file in SRT format. Since the subtitle text location can be manually set by SRT file, it’s easy to obtain the text lines and corresponding labels.

![Fig. 1 Data preparation pipeline.](image)

To get the accurate character bounding boxes, we simultaneously get the frame before and after subtitle loading while interpreting video frame. Background can be eliminated by simply subtracting these two images, then we can get character localization information through traditional image processing method based on pixel threshold. Ultimately, we write all this ground truth information into xml file in form of PASCAL VOC dataset. [8]
However, there is long tail effect in natural text data, some uncommon characters occur much less than what we need. Therefore, we also add some illogical sentences consist of these uncommon characters whose frequency of occurrence obey normal distribution. The final dataset meets the requirement of data equilibrium, each character has 60~100 instances with different color, font, size and etc.

2.2. Pipeline
The pipeline of our approach is illustrated in Fig. 2. Given an image, it is cut in batch first because the text line usually exists within a certain height range. It can reduce the computation and speed up learning by sampling small crops. A series of anchor boxes are set to match the ground truth. We use 3 scales and 1 aspect ratio, yielding 3 anchors at each sliding position. We can get a set of rectangular object proposals via region proposal network (RPN), followed by ROI Pooling layer and 2 fully-connected layers for text detection and localization.

We perform a text and non-text recognition in this phase and send the primary result to the following fine-grained classification network for subsequent multi-class classification to obtain a better performance.

![Fig. 2 Pipeline of our subtitle text detection and recognition method.](image)

2.3. Loss Function
The loss function for our text detection contains a binary classification loss, a box regression loss with transcendental information and a repulsion loss. [9]

For training region proposal network, we assign a binary class label (of being text or not) to each anchor. Anchor with the highest Intersection-over-Union (IoU) overlap with a ground-truth box or that has overlap higher than 0.7 with any ground-truth box is assigned with positive label. Anchor with IoU lower than 0.3 for all ground-truth boxes is assigned with negative label. Others are ignored.

Our loss function for detection is defined as:

$$L[p_i, \{t_i\}] = \sum_{i} L_{cls}(p_i, p_i^*) + \lambda \sum_{i} L_{reg}(b_i, b_i^*) + \mu \sum_{i} L_{rep}(P_i, P_i)$$  \hspace{1cm} (1)

Here, $i$ is the index of the anchor in one mini-batch and $p_i$ stands for the probability of anchor $i$ being a text. The ground-truth label $p_i^*$ is 1 if the anchor $i$ is positive and is 0 if the anchor $i$ is negative. $b_i$ is the predicted bounding-box of anchor $i$ and $b_i^*$ is the ground-truth box associated with a positive anchor. $P_i$, $P_j$ are proposals sampled from two different subsets, which will be explained later.

The classification loss $L_{cls}$ is log loss over 2 classes. The regression loss is a smooth L1 loss plus a transcendental loss, which is defined as:

$$L_{reg} = smooth \, L1 \, Loss + \alpha \times L_{trans}$$  \hspace{1cm} (2)
\(\alpha\) is the weight parameter for transcendental loss. As texts are aligned horizontally in subtitle, for a series of predicted bounding-boxes whose central heights are close, we set their average height as baseline and force them to get closer to the baseline lengthwise. Therefore, we set up a transcendental loss for supervision, which is defined as:

\[
L_{\text{trans}} = \frac{1}{2} \sum_{i} \sum_{j} \log\left(\frac{\left(y_{\text{min}} + y_{\text{max}}\right)}{2} - y_{\text{base}}\right)^2
\]

E is subset divided by different baseline and \(i\) is the index of bounding-box in subset \(E\). This loss punishes the deviation of predicted bounding-boxes in the same line.

In practice, we find there are false positives caused by crowded text. Through visualization in Fig. 3, we observe that errors usually occur when a predicted box partly covers the text or bounds the union of several overlapping ground-truth objects. Moreover, these errors usually have relatively high confidences, thus leading to top-ranked false positives.

\[
\text{Fig. 3 The visualization of detection errors. Red boxes are correct predicted bounding boxes, while yellow boxes are false positives. The errors usually occur when a predicted box partly covers the text or bounds the union of several overlapping ground-truth texts.}
\]

To address this problem, we introduce the repulsion loss that is proved effective for this occasion. The repulsion loss is made up of three components, defined as:

\[
L_{\text{rep}} = L_{\text{attr}} + \alpha \cdot L_{\text{repGT}} + \beta \cdot L_{\text{repBox}}
\]

\(L_{\text{attr}}\) is the attraction term which requires a predicted box to approach its designated target. \(L_{\text{repGT}}\) and \(L_{\text{repBox}}\) are the repulsion terms which require a predicted box to keep away from other surrounding ground-truth objects and other predicted boxes with different designated targets. \(\alpha\) and \(\beta\) are weight parameters.

We only use \(L_{\text{repBox}}\) here for simplicity and efficiency, which repels each proposal from others with different designated targets. The proposal set \(\rho^+\) is divided into \(|\mathcal{D}|\) mutually disjoint subsets based on the target of each proposal: \(\rho^+ = \rho_1 \cap \rho_2 \cap \ldots \cap \rho_{|\mathcal{D}|}\). Then for two proposals randomly sampled from two different subsets, \(P_i \in \rho_i\) and \(P_j \in \rho_j\) where \(i, j = 1, 2, \ldots, |\mathcal{D}|\) and \(i \neq j\). The loss is defined as:

\[
L_{\text{repBox}} = \frac{\sum_{i \neq j} \text{Smooth} l_1(IoU(B_i^p, B_j^p))}{\sum_{i \neq j} \text{Identity}(IoU(B_i^p, B_j^p) > 0) + \epsilon}
\]

2.4. Detection and Recognition

**Detection.** The detection module is based on faster R-CNN, using ResNet-18 backbone. As the text boxes can be small, we remove the last down-sample layer. Some convolutional layers’ kernel sizes are also decreased to accelerate the detection phase.

Anchor setting is adjusted to suit our problem, and we also adjust the anchor matching method by adding some matching rules to improve recall. When a ground-truth text has no anchor matched, we manually decrease the threshold. When a ground-truth text has too much anchors matched, we manually increase the threshold. This adjustment method is programed into a piecewise function.
Recognition. The recognition module is a light-weight CNN. When we get the predicted bounding-box, we expand the box and cut it in origin image, resize to 112*112 and send it to the recognition network. As input size is small, this process spends little time.

3. Experiments

3.1. Training Details
Our method is implemented on Caffe. For detection, we use ResNet-18 backbone, trained with 720 * 320 image using stochastic gradient descent (SGD) without flip. Momentum and weight decay are set to 0.9 and 5 × 10⁻⁴ respectively. Learning rate is initially set to 1e-3 for first 40k training iterations, and decayed to 1e-4 for last 20k training iterations. All the experiments are carried out on a PC with one Titan X GPU. Text recognition is performed on a classification network based on mobilenet_v1. The inference time of detection phase and recognition phase is 0.012s and 0.006s. The whole end-to-end process costs 0.022s, which is 45 fps.

3.2. Ablation Study
We apply our approach on the former mentioned subtitle dataset. As the ablation study of loss function shown in Table 1, 2, transcendental prominently improve both precision and recall, repulsion loss is also useful as it removes false positives. Because of the regularity of subtitle text, transcendental information can help model perform much better.

Table 1. Ablation study of loss function for detection.

| + Trans Loss | + Rep Loss | Precision(IoU:0.5/0.6/0.7) | Recall(IoU:0.5/0.6/0.7) |
|--------------|------------|----------------|-------------------------|
| √            | √          | 99.5/97.3/91.3 | 99.4/97.3/91.3          |

Table 2. Ablation study of loss function for recognition.

| + Trans Loss | + Rep Loss | Top-1 | Top-5 |
|--------------|------------|-------|-------|
| √            | √          | 93.0  | 94.1  |

3.3. Setting Experiment

Table 3. Performance of different anchor setting (IoU:0.7).

| setting      | anchor scales | aspect ratios | Precision | Recall |
|--------------|---------------|---------------|-----------|--------|
| 1 scale, 1 ratio | 16²           | 1.0           | 94.1      | 94.0   |
| 1 scale, 3 ratios | 16²           | 0.8, 1.0, 1.25 | 94.3      | 94.3   |
| 3 scales, 1 ratio | 8², 16², 32² | 1.0           | 97.0      | 96.8   |
| 3 scales, 3 ratios | 8², 16², 32² | 0.8, 1.0, 1.25 | 97.0      | 96.8   |

4. Conclusion
This paper has presented a simple and effective real-time character-level subtitle text detector, which is also useful for other video-related text detection problem such as barrage with slight modification. The whole system is easy to build using existing project. An efficient data generation method is proposed for subtitle text detection. Moreover, we introduce a series of valid methods and tricks such as repulsion loss to address this problem better. In future, we may fuse the detection and recognition module without losing precision and speed, making it unified and more elegant.
References
[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. TPAMI, 2015.
[2] Ren S, Girshick R, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6):1137-1149.
[3] Neumann L, Matas J. Real-time Lexicon-free Scene Text Localization and Recognition.[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2016, 38(9):1872-1885.
[4] Yin X C, Yin X, Huang K, et al. Robust Text Detection in Natural Scene Images[J]. 2014, 36(5):970-983.
[5] Jaderberg M, Simonyan K, Vedaldi A, et al. Reading Text in the Wild with Convolutional Neural Networks[J]. International Journal of Computer Vision, 2016, 116(1):1-20.
[6] He W, Zhang X Y, Yin F, et al. Deep Direct Regression for Multi-Oriented Scene Text Detection[J]. 2017.
[7] Liao M, Shi B, Bai X, et al. TextBoxes: A Fast Text Detector with a Single Deep Neural Network[J]. 2016.
[8] Everingham M, Winn J. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Development Kit[J]. International Journal of Computer Vision, 2006, 111(1):98-136.
[9] Wang X, Xiao T, Jiang Y, et al. Repulsion Loss: Detecting Pedestrians in a Crowd[J]. 2017.
[10] Girshick R. Fast R-CNN[J]. Computer Science, 2015.
[11] Yao C, Bai X, Liu W, et al. Detecting Texts of Arbitrary Orientations in Natural Images[C]// Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.
[12] 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, 2016, 39(11):2298-2304.
[13] Hu H, Zhang C, Luo Y, et al. WordSup: Exploiting WordAnnotations for Character based Text Detection[J]. 2017.
[14] Zhang Z, Zhang C, Shen W, et al. Multi-Oriented Text Detection with Fully Convolutional Networks[J]. 2016.
[15] Zhou X, Cong Y, He W, et al. EAST: An Efficient and Accurate Scene Text Detector[C]// IEEE Conference on Computer Vision & Pattern Recognition. 2017.
[16] Tian S, Pan Y, Huang C, et al. Text Flow: A Unified Text Detection System in Natural Scene Images[J]. 2016.
[17] Liao M, Shi B, Bai X. TextBoxes++: A Single-Shot Oriented Scene Text Detector[J]. IEEE Transactions on Image Processing, 2018, 27(8):3676-3690.
[18] He P, Huang W, He T, et al. Single Shot Text Detector with Regional Attention[J]. 2017:3066-3074.
[19] Yao C, Bai X, Sang N, et al. Scene Text Detection via Holistic, Multi-Channel Prediction[J]. 2016.
[20] Dai Y, Huang Z, Gao Y, et al. Fused Text Segmentation Networks for Multi-oriented Scene Text Detection[J]. 2017.