C^{2}TD: A FAST AND ROUBUST TEXT DETECTOR VIA CENTRE LINE REGRESSION AND CENTRE BORDER PROBABILITY

A PREPRINT

Junjie Wang*
Aiyunxiao research
wangjunjie@aiyunxiao.com

June 3, 2019

ABSTRACT

Text detection is an essential step of scene text or documents text recognition system. Recently, instance segmentation based methods and anchor-based methods have achieved great success in text detection. However, predicting texts throughout an image or separating very close small text are still challenges. In this paper, we present a fast and robust methods that suitable for predicting arbitrary long text and small dense texts. Our FPN network outputs center-border probabilities and the top/bottom offsets at the same time. The center-border probabilities make it possible to separate dense texts without further processing. By filtering the probabilities maps with a higher probability, we can obtain the center line, the corresponding map positions of which are densely predicts top/bottom offsets. The post-processing need no heavy computation, which makes it suitable for real word use. Codes are available at: http:xxx

1 Introduction

Text detection still encounter two problems. First, predicting text lines throughout a image confuses some text detection methods like EAST [1],TextBoxes [2]. These methods directly predict the cordinates of bounding boxes. Due to the limitation of receptive field and the extreme large aspect ratio of text lines, they struggle to predicted a complete bounding boxes. Second, heavy Computation hampers real word use. In order to separate each text instances, PixelLink [3] predicted 8 link direction, and PSENet [4] apply the progressive scale expansion mechanism. However, this steps aggravate computation complexity. mn

Some methods like advanced and corner ? predicted the key points of a textline. For example, advanced EAST predicted the head and tail. When the training data were weakly labelled, the method may fail to predict the key point, which lead to a failure of predicting the whole text line.

To solve these problems, we introduce our methods named C^{2}TD, which is based on centre line regression and centre-border probability. The method was mainly inspired by TextMountain [5] and FCOS [6]. TextMountain predicted text center-border probability (TCPB), and claim that the label rules can separate text instances which can not be easily achieved by semantic segmentation. Then, the method also use instance segmentation to get the final boxes.

FCOS is anchor-free common object detector, which show that eliminating predefined anchor boxes also yeids a promising performance, while make the whole pipeline simpler. FCOS works by predicting top/bottom/left/right offsets as each foreground pixel, which is similar to EAST. When two objects have intersections, the prediction can be ambiguous. Hence, a single-layer branch to predict the *center-ness* was added. By only using the centre positioin to predict the bounding boxes, the method discards many low-quality prediction.

Our method predicted a centre border probability map. By filtering the probabilities map with a threshold, a center line was obtained. Based on our observation, predicting the center line is more roubust than predicting the border line. And

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

the centre line also served as a fuction to line the whole object. Similar to CTPN [7], we only regress two offsets of a text line, because the limitation of receptive field, and rely on the centre line to *link* the results.

2 Approach

In this section, we first show the label rules of centre border probability. Then the regresion methods. At last, we introduce a yet simple post processing method. Pay attention to that any feature pyramid network (FPN) and multile scale prediction should work fine with this methods, we omit this part. For efficiency, the FPN only outputs 1/4 feature maps in our experiments.

2.1 Label rules

In Text moutain, the centre border probability ranges from 1 to 0 at foreground pixel. When we used 1/4 score map, some samll text lines which are only 10 pixels height may have only one line labelled as 1. We make the probability range from 1 to 0.5 in our experiments.

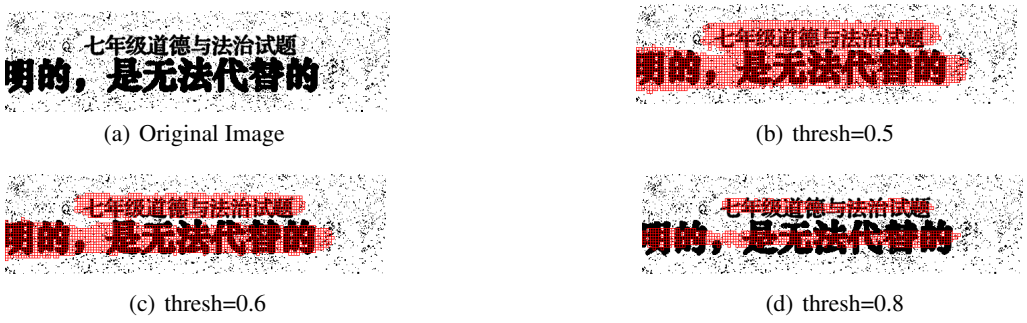
Plot some figures to explain that this lables rules works.

The probability target at position y is defined as:

$$step = 0.5/|y_c - y_b| \tag{1}$$

$$p_y = 1 - step * |y_c - y| \tag{2}$$

where y_c is centre postion of the text line, and y_b is the border position. One channel of a FPN outputs the probability, and trained with smooth l1 loss.



(a) Original Image (b) thresh=0.5

(c) thresh=0.6 (d) thresh=0.8

Figure 1: An sample shows setting different threshold of the centre-boder probalibity leads to different separability. The 1/4 socre map are projected back to original image. When set the threshold to 0.5, the two texts instance are not separable.

When testing, we set a high threshold, typically 0.8, to filter out the border pixels. To solve the imblance problems of foreground and background, Online Hard Example Mining (OHEM) is also applied [8]. By convention, we set the negative-positive ratio to 3.

2.2 Regression the cordinates

We only densely predict cordinates along one direction. For horizontal text lines, only the top and bottom offsets are predicted, which is similar to CTPN. Because the limitation of receptive field, predicting the other two cordinates are not accurate.

The regression targets are encoded as:

$$t^* = t - y * stride \tag{3}$$

$$b^* = b - y * stride \tag{4}$$

where t and b are top/bottom coordinates, y is vertical location at the feature map

2.3 Network outputs and loss

The network outputs 3 score maps.

2.4 Post process

The post process is quiet simple. By filtering the probabilities maps with a higher probability, we can obtain a text center line map. We use Opencv methods to cluster active pixels in each instance. Some more efficient cluster methods using GPU methods may be implemented in the future. By using the active pixels, the final cordinates are avaraged by these densely prediction.

3 Experiments

4 Conclusion

References

[1] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. EAST: An Efficient and Accurate Scene Text Detector. 2015.

[2] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. TextBoxes: A Fast Text Detector with a Single Deep Neural Network. 2016.

[3] State Key, Computer Science, Dan Deng, Haifeng Liu, and Xuelong Li. PixelLink: Detecting Scene Text via Instance Segmentation. 2017.

[4] Li Xiang, Wenhai Wang, Wenbo Hou, Ruo Ze Liu, Lu Tong, and Yang Jian. Shape robust text detection with progressive scale expansion network. 2018.

[5] Yixing Zhu and Jun Du. TextMountain : Accurate Scene Text Detection via Instance Segmentation.

[6] Hao Chen. FCOS: Fully Convolutional One-Stage Object Detection.

[7] Zhi Tian and Weilin Huang. Detecting Text in Natural Image with Connectionist Text Proposal Network. pages 1–16.

[8] Abhinav Shrivastava. Training Region-based Object Detectors with Online Hard Example Mining.