Project 3: Scene Text Recognition

Yanwei Fu

Abstract

(1) This is project 3 of our course. The deadline is 5:00pm, June 15, 2021. Please upload the report via elearning.

(2) The goal of your write-up is to document the experiments you've done and your main findings. So be sure to explain the results. Generate a single pdf file of your projects and turned in along with your code. package your code and a copy of the write-up pdf document into a zip or tar.gz file and named as Final-Project-*your-student-id*_your_name.[zip|tar.gz], and Final-Project-test-label-*your-student-id*_your_name.[zip|tar.gz]. Also put the names and Student ID in your paper. We provide the testing images, while the testing label is with-held for the evaluation. We care your performance in this task.

(3) You are open to use anything to help you finish this task.

(4) About the deadline and penalty. In general, you should submit the paper according to the deadline of each mini-project. The late submission is also acceptable; however, you will be penalized 10% of scores for each week's delay.

1 Dataset

The dataset is uploaded in

http://www.sdspeople.fudan.edu.cn/fuyanwei/download/pj-data-yikai/

In your servers, you can use wget to download the dataset. This might be faster, since all the servers are in Fudan campus. For example,

wget http://www.sdspeople.fudan.edu.cn/fuyanwei/download/pj-data-yikai/test.zip

Of course, you can download the file in your local machine, and upload to your servers by 'scp' command.

2 Background

Scene Text Recognition (STR) refers to the recognition of text information in pictures of natural scenes. One related concept is Optical Character Recognition (OCR), which refers to analyzing and processing the scanned document image to identify the text information. STR is a harder task compared with OCR, since the text in the natural scenes have extremely rich form:

1. The characters can have different sizes, fonts, colors, brightness, contrast, etc.
2. The text line, if exists, may be horizontal, vertical, curved, rotated, twisted and other styles.
3. The text area in the image may also be deformed (perspective, affine transformation), incomplete, blurred, etc.

4. The background of natural scene images is extremely diverse. For example, text can appear on a flat surface, a curved surface, or a wrinkled surface; there is a complex interference texture near the text area, or a texture similar to the text in the non-text area, such as sand, grass, fence, brick wall, etc.

5. Multi-language text also exists.

To achieve the goal of scene text recognition, one traditional way is split the task into two sub-tasks: first detect/segment the area where text may appear and then recognize the text information in the detected/segmented area.

For general object detection, researchers have designed some powerful models, like SSD[7], YOLO[11], Faster R-CNN[12]. However, directly using them on the scene text detection often result in a unexpected performance. The major reason is that scene text is often a fine-grained object with varying line length and ratio of length to width.

There have been many efforts to tackle the detection task. CTPN[16] regard the text line as a character sequence rather than a single independent target in object detection task. Text characters of each line are mutually context, which can be utilized in the training process to improve the performance. RRPN [10] introduces the rotation factor in the bounding box. In this framework, the ground truth of a text area is expressed as a rotating five tuples (x, y, h, w, θ) where θ is omitted in the traditional object detection models. DMPNet[9] use quadrilateral instead of rectangle to handle the varying style of scene text. FTSN[2] regards the first step as a segmentation task instead of a detection task, which leads to more precise detected bounding boxes. EAST[17] directly provides pixel-level prediction to eliminate intermediate steps such as candidate proposal, text region formation and word partition. Other efforts include SegLink[13], PixelLink[3], Textboxes[6]/Textboxes++[5], WordSup[4].

After detecting the area of interest, the second step is recognizing the text from the detected area. CRNN[14] is currently a popular recognition model for STR. It contains CNN feature extraction layer and BLSTM sequence feature extraction layer, which can carry out end-to-end joint training. It uses BLSTM and CTC components to learn the contextual relationship within characters, thereby effectively improving the accuracy of text recognition and making the model more robust. In the prediction process, the front end uses a standard CNN network to extract the features of the text image, and uses BLSTM to fuse the feature vectors to extract the context features of the character sequence, and then obtain the probability distribution of each column of features. Finally, the text sequence is obtained through prediction by the transcription layer. RARE[15] utilize a spatial transformation network for better recognize distorted text.

Some efforts also made to design an end-to-end scene text detection and recognition model, including FOTS[8], STN-OCR[1], etc.

3 Task Description

3.1 Objective

In this project, you will try to solve the task of scene text recognition. Given an input scene image, the objective is to localize a set of bounding boxes and their corresponding transcriptions. You can use models introduced in the above, or design your own model. *Note that you can only use the provided training set to train your network from scratch.* After training, provides inference results on the test set. Write a report to illustrate your algorithm and experimental results on the training set. Your grade will depend on your report and performance on the test set.

3.2 Dataset

You are provided a dataset whose images are natural scene images containing text of 10 different languages, including Arabic, Bangla, Chinese, Devanagari, English, French, German, Italian, Japanese and Korean. The training data is unbalanced for the different languages. You will use 7932 images for training your network, and use 1990 images for evaluation. The images mainly contain focused scene text, however, some unintentional text may appear in some images. The text in the scene images of the dataset is annotated at word level. A ground-truth word is defined as a consecutive set of characters without spaces, i.e. words are separated by spaces, except in Chinese and Japanese where the text is labeled at line level. Each ground-truth word is labeled by a 4-corner bounding box and is associated with a Unicode transcription. For each image, we have a txt file to store the bounding box and transcription, where each line corresponds to one text block in the image. The format is:

$$x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, \text{script,transcription}$$

where the former are the bounding box coordinates and the last is the target transcription. The "script" can be ignored. Your will provide the txt file of each test image in the similar format for evaluation (ignore the "script").

3.3 Evaluation Protocol

The f-measure is used as the metric for ranking your results. A correct (true positive) result if it has a correct detection of a text box with correct transcription. A detection is counted as correct if the detected bounding box has more than 50% overlap (intersection over union) with the ground truth box. For example, suppose a ground truth bounding box is $A(g)$ with transcription $t(g)$, and a detected bounding box is $A(d)$ with transcription $t(d)$. Then a positive match is counted if (g, d) verifies the following condition:

$$\frac{A(g) \cap A(d)}{A(g) \cup A(d)} > 0.5 \wedge t(g) = t(d)$$

where the equation between $t(g)$ and $t(d)$ means the edit distance between the two transcriptions is zero.

At the whole test set level, the evaluation metrics are computed cumulatively from all the test images. Denote the set of positive matches as M, the set of expected words as G and the set of filtered results as T. Then the f-measure is computed as follows:

$$P = \frac{|M|}{|T|}$$

$$R = \frac{|M|}{|G|}$$

$$\text{f-measure} = \frac{2 \cdot P \cdot R}{P + R}$$

References

[1] Christian Bartz, Haojin Yang, and Christoph Meinel. Stn-ocr: A single neural network for text detection and text recognition. *arXiv preprint arXiv:1707.08831*, 2017.

[2] Yuchen Dai, Zheng Huang, Yuting Gao, Youxuan Xu, Kai Chen, Jie Guo, and Weidong Qiu. Fused text segmentation networks for multi-oriented scene text detection. In *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018.

[3] Dan Deng, Haifeng Liu, Xuelong Li, and Deng Cai. Pixellink: Detecting scene text via instance segmentation. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[4] Han Hu, Chengquan Zhang, Yuxuan Luo, Yuzhuo Wang, Junyu Han, and Errui Ding. Wordsup: Exploiting word annotations for character based text detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[5] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *IEEE transactions on image processing*, 2018.

[6] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[7] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *European conference on computer vision*, 2016.

[8] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[9] Yuliang Liu and Lianwen Jin. Deep matching prior network: Toward tighter multi-oriented text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[10] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*, 2018.

[11] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[12] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 2015.

[13] Baoguang Shi, Xiang Bai, and Serge Belongie. Detecting oriented text in natural images by linking segments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[14] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2016.

[15] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[16] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *European conference on computer vision*, 2016.

[17] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017.