Project 3: Scene Text Recognition

Du Mengfei
The School of Data Science
Fudan University
18307130148@fudan.edu.cn

Abstract

Scene texts include rich semantics and environmental information which may be useful for vision-based applications, and recognizing scene texts have received more and more attention in recent years. In this paper, we try to use an apporach : EAST for object detection and use CRNN with CTC loss and attention based model for text recognition. We finally use the combination of EAST+CRNN as our final method.

1 Introduction

Scene Text Recognition(STR) refers to the recognition of text information in images of natural scenes.Two fundamental tasks are included in STR:text detection and recognition. Text detection aims to determine the position of text from input image, and the position is represented by a bounding box. Text recognition aims to convert image regions containing text into machine-readable strings. Different from the general image classification, the dimension of output sequence for text recognition is not fixed. In many situations, text detection is a preliminary step of text recognition. Recently, many researchers begin to integrate the detection and recognition tasks into an end-to-end text recognition system.

In this paper, we mainly provide a two-stage method with the conbination of EAST and CRNN or other methods.We also try to implement the end to end approach in the later part.And we split the training set to evaluate the methods we have implemented.

2 Related Work

As mentioned above, Scene Text Recognition is a challenging problem.There exists lots of researches in this field. And we will briefly introduce some works about this problem.

2.1 Text Detection

Text Detection aims to figure out areas containing texts. Similar to majority of computer vision tasks, many previous text detection methods are based on handcraft features as well as prior knowledge, and deep learning based methods gradually become the mainstream at around 2015. Generally speaking,methods of text detection can usually be divided into two categories: HandCrafted Feature Extraction and Deep learning[1].

Traditional text detectors focus on developing hand-crafted low-level features to discriminate text and non-text components in scene image,which can be mainly classified into two categories, i.e., sliding window (SW) and connected component (CC) based methods.Wang et al. [2] provided an end-to-end pipeline for STR, where they perform multi-scale character detection via SW classification.Wang et al. [3] applied a convolutional neural network (CNN) model with SW scheme to obtain

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

candidate lines of text in given image, and thus estimate text locations. Epshtein et al. [4] presented SWT operator to compute the width of the most likely stroke for image pixel.

Recently, deep learning has been widely used in semantic segmentation and general object detection.Serveral works achieved great sucess.He et al. [5] presented the cascaded convolutional text networks (CCTN), which uses two networks to implement coarse-to-fine segmentation for scene image.The Single-Shot Detector (SSD) [6] like architecture is used to extract features and perform text/non-text prediction as well as link prediction. The predicted positive pixels are joined together into text instances by predicted positive links. Finally, text bounding boxes are generated directly from the segmentation result without location regression.Tian et al. [7] proposed a connectionist text proposal network (CTPN) to localize scene text. In CTPN, VGG16 backbone is first used for feature extraction, and then a vertical anchor mechanism is developed to predict text locations in a fine scale. Finally, a Bidirectional long short term memory (BLSTM) is applied to connect the fine scale sequential text proposals. Zhou et al. [8] proposed EAST to directly provide pixel-level prediction to eliminate intermediate steps such as candidate proposal, text region formation and word partition.

2.2 Text Recognition

Similar to text detection, text recognition in scene also experiences the transition from traditional means using handcrafted features to deep learning era. Nowadays, the main methods can be classified into three types:character classification based, word classification based and sequence based methods.

Lee et al. [9] presented recursive recurrent neural networks (RNNs) with attention model for text recognition. Kang et al. [10] designed a context-aware convolutional recurrent network for word recognition. Besides a lexicon dictionary, the metadata of the input image, such as title, tags, and comments, are used as a context prior to enhance the recognition rate. Shi et al. [11] proposed a convolutional recurrent neural network (CRNN) for image-based sequence recognition. A standard CNN model is first used to extract a sequential feature representation from input image. Then a bidirectional long-short term memory (LSTM) network is connected with the top convolutional layers to predict a label distribution for each frame of feature sequence. Finally, the connectionist temporal classification (CTC) is applied to find the label sequence with the highest probability conditioned on the per-frame predictions.

2.3 End to end Text Spotting

Text detection and recognition are usually combined to implement text spotting, rather than being treated as separate tasks.

Yao et al. [12] presented a unified framework, where text detection and recognition share both features and classification. Furthermore, the dictionary is generated according to Bing search, whose error correction scheme can be used to enhance the recognition rate Jaderberg et al. [13] also proposed t an end-to-end text spotting system. Word level bounding box proposals are first obtained with high recall, and then filtered by a random forest classifier for improving precision. Two CNNs are used for bounding box regression and text recognition respectively.

3 Method of Text Detection

In this part we will introduce the text detection model EAST[14] that we used for the experiment.

EAST has two stages. The first stage is based on the Fully Convolutional Network (FCN) model, which directly generates text box predictions. The second stage is to generate non-maximum suppression of the generated text prediction boxes ((which can be rotated rectangles or horizontal rectangles)).

3.1 Model Structure

First the network uses a network Pvanet as the base net for feature extraction.And in our model we try to use VGG16, Resnet,and other network architectures to extract image features. Based on the above-

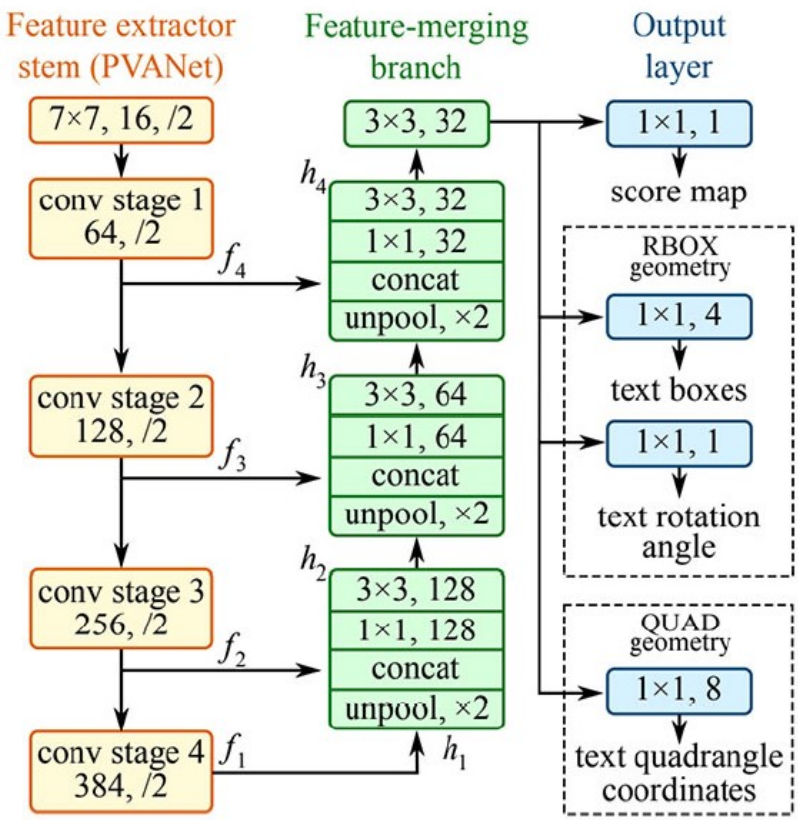

Figure 1: the Network Structure of EAST

mentioned backbone feature extraction network, feature maps of different levels are extracted (their sizes are $[\frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}]$ of inuput-image respectively), so that feature maps of different scales can be obtained. The purpose is to solve the problem of severe scale transformation of text lines. Ealy stage can be used to predict small text lines, and late-stage can be used to predict large text lines. The feature merging layer merges the extracted features. The merging rules here use the U-net method: extracting the top features of the network from the features and merging them down according to the corresponding rules. Network output layer, including text score and text shape. According to different text shapes (can be divided into RBOX and QUAD), the output is also different, the concreate structure is in Figure 1.

3.2 Loss Function

The loss function is composed of two parts: score map loss and geometry lossas .The concrete formula is as follows:

$$L_{loss} = L_s + \lambda_g L_g Lloss = L_s + \lambda_g L_g$$

In order to simplify the training process, this article uses class-balanced cross entropy, the formula is as follows:

$$Ls = balanced - xent(\widehat{Y}, Y^*) = -\beta Y^* log\widehat{Y} - (1-\beta)(1-Y^*)log(1-\widehat{Y})$$

$Y = F_s$ is the prediction of the score graph, and Y^* is the labeled value. The parameter β is the balance factor between positive and negative samples, the formula is as follows:

$$\beta = 1 - \frac{\sum_{y^* \in Y^*} y^*}{Y^*}$$

For geometry loss L_g, it is divided into RBOX Loss and QUAD Loss. RBOX uses IoU loss for the AABB part because it is invariant to objects of different sizes.

For RBX loss:

$$L_g = L_{AABB} + L_\theta = -logIoU(\widehat{R}, R^*) + \lambda_\theta(1 - cos(\widehat{\theta} - \theta))$$

For QUAD loss:

$$L_g = L_{QUAD}(\widehat{Q}, Q^*) = min_{\tilde{Q} \in P_{Q^*}} \sum_{c_i \in C_Q} \sum_{\tilde{c} \in \tilde{Q}} \frac{smoothed_{L1}(c_i - \tilde{c_i})}{8N_{Q^*}}$$

4 Text Recognition

Text recognition branch is used to obtain corresponding sequential character labels from input image or feature map.Characters are seen as classification labels in most methods, and here we utilize two popular text recognition methods :CNN + RNN + CTC loss and CNN + Seq2seq + Attention. The detailed information of CRNN components are shown in Figure 3, including CNN, RNN and CTC.

4.1 CRNN

The total structure of CRNN[15] is shown in Figure 2. The CRNN model is mainly composed of three parts : CNN, RNN, CTC loss. The CNN part is a normal convlutional layers to extract the convolutional feature maps of input images. The recurrent network layer is a deep Bi-LSTM network, which continues to extract text sequence features based on convolutional features.

The CTC method using a "blank" to solve the redundancy problem, the two same consecutive letters will be Merge into one unless it has a "<blank>" inside. In this way, there are many kinds of sequences that can map into the same result. For each sequence π of length T from the input x of Bi-LSTM, we can find the probability of such sequence generation.

$$P(\pi|x) = \prod_{i=1}^{T} p(x_i)$$

Then for a fixed sequence, its generation probability is the sum of all probabilities of sequence that can be mapped to that sequence:

$$P(l|x) = \sum_{\pi \in B^{-1}(l)} p(\pi|x)$$

where $B^{-1}(l)$ means all sequence that can be mapped to that fixed sequence, so in order to maximize the probability of the correct sequence. we can updated the gradient of BiLSTM through back propagation.

4.2 Attention

The network architecture of CRNN + Attention is to add a layer of attention mechanism behind the output layer of the CRNN network. In detail, it is CRNN + GRU. The encoder is composed of CNN and Bi-LSTM model. And the decoder is a GRU. We use the state of all hidden layers of encoder to solve the problem of Context length limitation. When the decoder outputs characters, its hidden vector and the hidden vector in the encoder layer are synthesized as a new hidden vector to operation. The model is as Figure 4.

5 Experiment

The details of our implementation and perfomance are explained in this section.

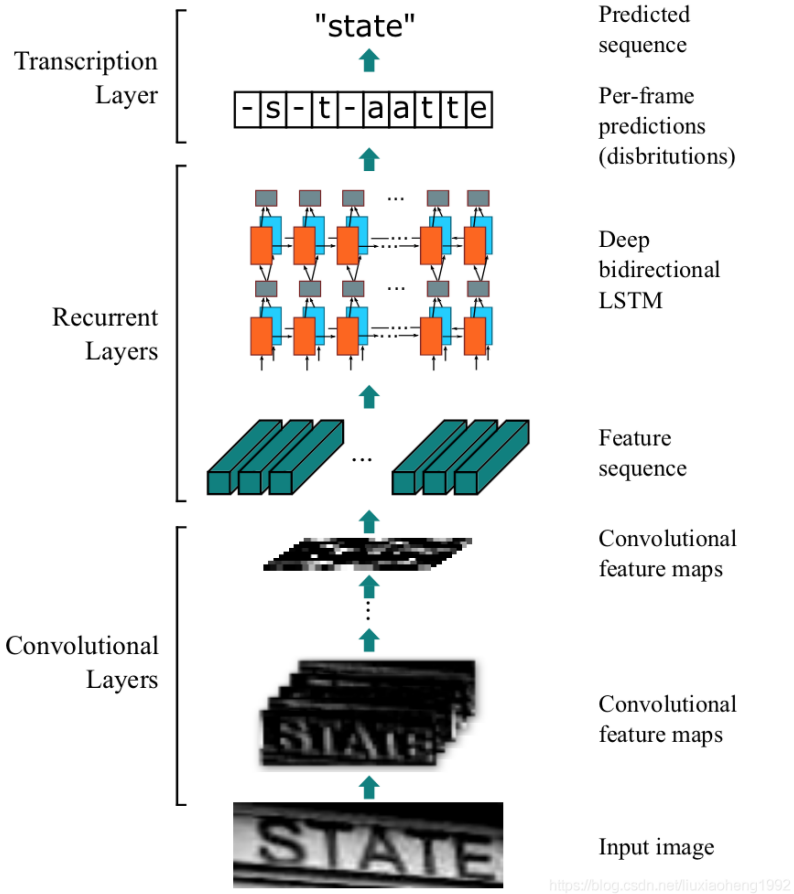

Figure 2: the Network Structure of CRNN

Table 1: Parameters of EAST

Parameter	Value	Parameter	Value
Epoch	600	Weight decay	0.01
Learning rate	10^{-3}	Number workers	8
Batch size	16	Optimizer	Adam

5.0.1 Dataset

The data set of this final project is divided into a training set and a test set. There are 7932 images in the training set and 1990 images in the test set. The training set contains a total of ten languages, they are Arabic, Bangla, Chinese, Devanagari, English, French, German, Italian, Japanese and Korean. Each ground truth of training data is marked by a 4-corner bounding box and is text transcription. Each picture has a txt file with a corresponding name to store the border and transcript, where each line corresponds to a text block in the image. The format is:

$$x_1; y_1; x_2; y_2; x_3; y_3; x_4; y_4; script; transcription$$

5.1 Text Detection

For the text detection , we use EAST model and the concreate parameters can be seen as Table 1.

The losslanscape of the training is as Figure3. Some results of the EAST model are as Figure4:

Table 2: Parameters of CRNN

Parameter	Value	Parameter	Value
Epoch	200	LSTM hidden size	256
Learning rate	10^{-5}	Number workers	8
Batch size	256	Optimizer	Adam

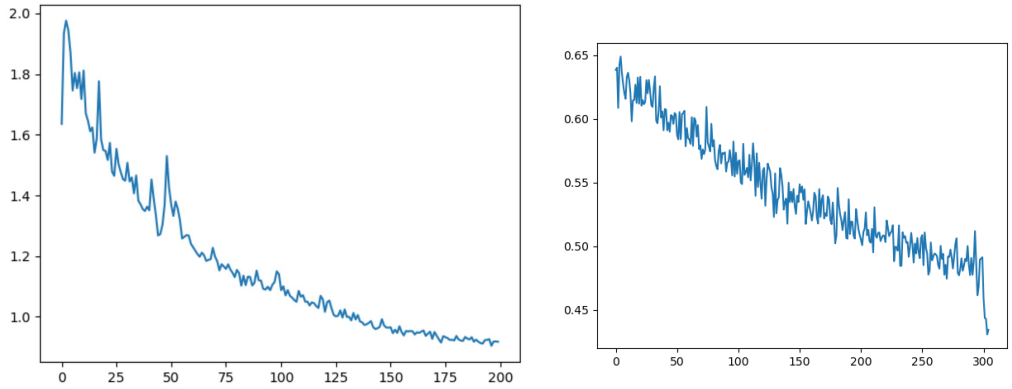

Figure 3: Loss Landscape of EAST

5.2 Text Recongition

For the text recognition, the concreate parameters can be seen as Table 2.

For CRNN, the model can get a 32.89% on the test dataset at 188 epoch.
For Attentin , the model can get a 30.24% on the test dataset at 173 epoch.

6 Conclusion

In this report, we implement one prevelant model of text detection and compare the performance of two different models of text recognition on this data set. We also try to implement an end to end network but the performance is not very good.We finaly select the EAST + CRNN as our model. In future work, we try to explore more effective feature acquisition methods for high-resolution images.

Figure 4: Results of EAST

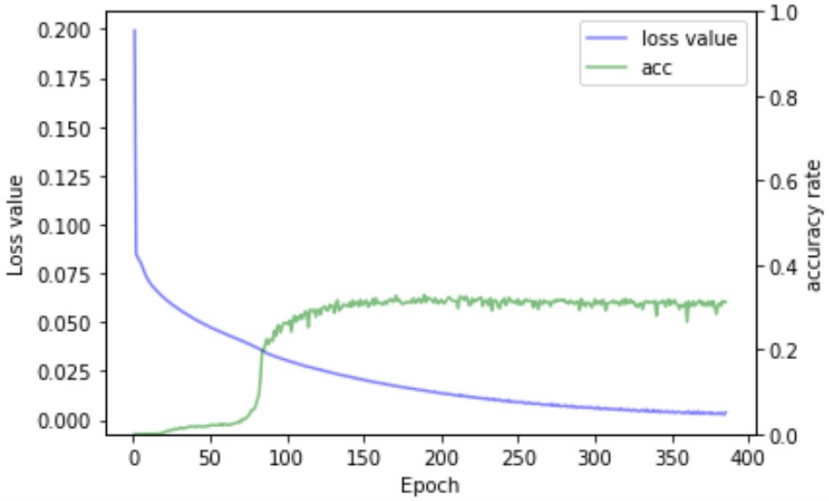

Figure 5: Loss Landscape of CRNN

References

[1] Lin H , Yang P , Zhang F . (2019) Review of Scene Text Detection and Recognition. *Archives of Computational Methods in Engineering*.

[2] Babenko B, Belongie S . (2012) End-to-end scene text recognition. *IEEE international conference on computer vision, pp 14571464*

[3] Wang T, Wu DJ, Coates A, Ng AY . (2012) End-to-end text recognition with convolutional neural networks. *International conference on pattern recognition, pp 33043308*

[4] Epshtein B, Ofek E, Wexler Y . (2010) Detecting text in natural scenes with stroke width transform. *In: Computer vision & pattern recognition, pp 29632970*

[5] He T, Huang W, Qiao Y, Yao J . (2016) Accurate text localization in natural image with cascaded convolutional textnetwork, *pp 110. arXiv :1603.09423*

[6] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S . (2016) SSD: single shot multibox detector. *In: European conference on computer vision, pp 2137*

[7] Tian Z, Huang W, He T, He P, Qiao Y . (2016) Detecting text in natural image with connectionist text proposal network. *In: European conference on computer vision, pp 5672*

[8] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. (2017) East: an efficient and accurate scene text detector. *In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 55515560, 2017.*

[9] Lee CY, Osindero S . (2016) Recursive recurrent nets with attention modeling for OCR in the Wild. *In: IEEE conference on computer vision and pattern recognition, pp 22312239*

[10] Kang C, Kim G, Yoo SI . (2017) Detection and recognition of text embedded in online images via neural context models. *In: Proceedings of association for the advancement of artificial intelligence,pp 41034110*

[11] Shi B, Bai X, Yao C . (2016) An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Trans Pattern Anal Mach Intell 39:22982304*

[12] Yao C, Bai X, Liu W . (2014) A unified framework for multioriented text detection and recognition. *IEEE Trans Image Process 23:47374749*

[13] Jaderberg M, Simonyan K, Vedaldi A, Zisserman A . (2016) Reading text in the wild with convolutional neural networks. *Int J Comput Vis 116:120*

[14] https://github.com/SakuraRiven/EAST
[15] https://github.com/Holmeyoung/crnn-pytorch