Research on Text Detection and Recognition Based on OCR Recognition Technology

Yuming He
Software Engineering
Shanghai Starriver Bilingual School
Shanghai, China

***Abstract*—Optical character recognition (OCR) is an important branch in the field of machine vision. It involves pattern recognition, image processing, digital signal processing, artificial intelligence and other disciplines. It is a comprehensive. It has important use value and theoretical significance in high-tech fields such as word information processing, office automation, machine translation and real-time monitoring system. In the 21st century, with the popularity of smart phones with high-definition cameras, OCR has a new pursuit in its development: more and more people pick up their mobile phones to photograph the things and scenes they see and obtain the text information in the pictures. Therefore, the recognition of characters in natural scenes has become a brand-new topic. In the past, text detection and text recognition algorithms were basically based on artificially designed features and traditional image processing methods. These features and algorithms were difficult to design and needed a lot of professional knowledge and experience support, so the accuracy was not high and they were not generalized. In recent years, with the rapid development of deep learning technology, breakthroughs have been made in the fields of computer vision such as image classification, object detection and semantic segmentation. Deep learning algorithm is a data-driven algorithm. The algorithm based on deep learning can automatically discover and learn the hidden feature rules in a large number of data through iterative training, without too much human intervention, so it has better generalization than traditional image processing related algorithms.**

Keywords—Optical character recognition, Text detection and recognition, Deep learning, Natural scene image

I. INTRODUCTION

With the emergence of various photographic equipment and the enhanced convenience of mobile phone cameras, documents have gradually changed from previous paper materials to a wide variety of electronic ones, such as PDF file, ect. Electronic documents have attracted extensive attention and have been widely used due to its convenience and digital features. However, at present, most electronic documents, like PDF files, are scanned electronic documents, which are composed of pictures and cannot be changed additionally. For example, when using scanned PDF files, users cannot directly highlight or give comments on the documents. Therefore, how to extract text from pictures has become a hot topic. Thus, OCR (Optical Character Recognition) has come into the sight of researchers.

The research scope of OCR technology has expanded from document recognition under simple background to handwriting recognition under natural scenes as the technology processes. The simple backgrounds and the uniform printed characters make it easy to extract text from paper images. The traditional machine learning have achieved satisfactory results. For example, the SVM (Support Vector Machine) proposed by John C.Platt in [1] has been widely applied. However, handwriting recognition is more difficult than ordinary document recognition, because every handwriting word is written differently from people to people. It adds a lot of troubles to the construction of data set and recognition. The complex backgrounds are the most difficult part. Compared to the single background of documents, the background of natural scenes are complicated. They consist cars, pedestrians, etc. It is challenging to recognize text from these complex backgrounds.

Therefore, at present, OCR mainly focuses on the following three facets:

1) Recognition and transformation of traditional documents

2) Recognition of handwriting

3) OCR under natural scenes

II. TEXT DETECTION

Throughout the OCP operation, text detection is more difficult than text recognition. How to accurately abstract the text from its background is not an easy task. Many networks of text detection come from image classification networks under general scenes, such as VGGNet, ResNet, DenseNet, and some special network models, such as FCN(Fully Convolutional Network), STN(Spatial Transformer Network), etc. Faster RCNN is the most frequently used detection network framework. However, compared with objects, text is a relatively small target. Text is not as prominent against its background as are general objects, so copying those detection networks directly often leads to unsatisfactory results. Therefore, based on those detection networks, researchers need to do more work to improve the general detection networks to improve the detection accuracy.

A. Detecting Text in Natural Image with Connectionist Text Proposal Network

In 2015, FastrRCNN, which was improved by Ren et al. on itself, was a great improvement in the field of Object Detection, shortening the detection time of a picture by about 10 times. FastrRCNN changed the search box detection part of FastrRCNN from the original selective search to the RPN network, and integrated it into the whole Detection Network, which greatly accelerated the detection.

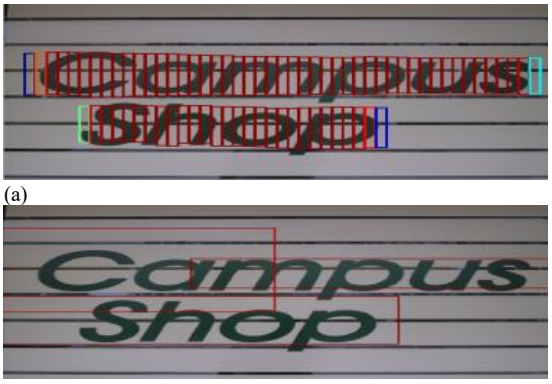
978-1-7281-8304-6/20/$31.00 ©2020 IEEE

Because of the popularity of Faster RCNN, Zhi Tian et al. also planned to apply it to text detection accordingly. In [3], researchers proposed a novel text detection network CTPN Network. The network is improved based on the popular target recognition network Faster RCNN.

The researchers took text detection experiments with Faster RCNN, as shown in Figure 1 (a). They found that Faster RCNN has a good detection effect. It was able to detect every line, but the accuracy on columns is not high. Therefore, in CTPN(Context Text Proposal Network), the main idea of them is to change the setting mode of anchor in Faster RCNN. In Faster RCNN, anchors are set as shown in Figure 2, and different anchors are set according to different width-height ratios. In CTPN, researchers innovatively fix anchors with a width of 16px, and then set 9 anchors with different heights according to a certain proportion. The effect is obvious, as shown in Figure 1 (b) based on Faster RCNN. The detection of rows is effective, and the detection effect of columns is fine and accurate.

Besides the innovation of anchor setting, as shown in Figure 3, the CTPN Network also introduced BLSTM. BLSTM is bidirectional, which is an upgrade version of RNN network. As it known to all, the biggest feature of RNN series networks is that they integrate contextual information for training. In traditional neural networks, the information of each iteration training will not be related except the gradient. However, one input of RNN series network is the information of last training, so RNN series network is context-related network. In CTPN, the anchors of text are also context-related, and it is meaningless to take out a single anchor independently. Therefore, it is obvious an improvement of anchor recognition ' s accuracy by introducing BLSTM.

Up to now, CTPN is the most developed Text Detection Network, which takes the lead of recognition accuracy. However, it does not mean that it is a perfect network. The training and recognition speed of CTPN is not satisfactory, and many researchers are committed to optimizing CTPN at present.



(a)

(b)

Figure 1 Faster RCNN is able to closely detect the text

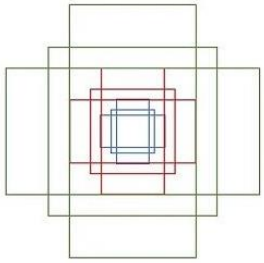


Figure 2 The changed way of how anchor is set in Faster RCNN

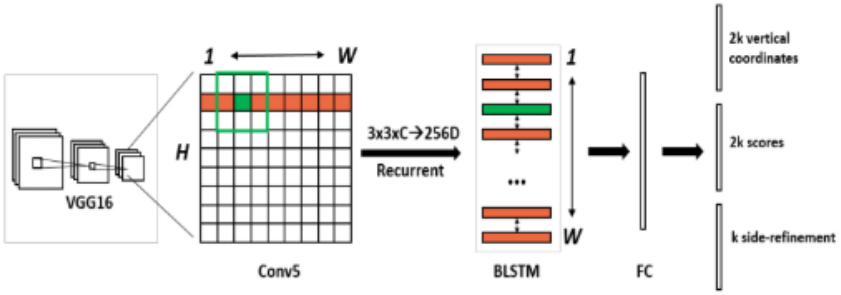


Figure 3 The Whole CTPN

B. EAST: An Efficient and Accurate Scene Text Detector

Traditional text detection methods, as well as some text detection methods based on Deep Learning, are mostly multi-stage. Therefore, we should adjust parameters of each stage separately during training, which not only consumes time, but also accumulates errors of each stage layer by layer, and then the final detection results are not ideal.

In [4], Xinyu Zhou et al. have raised the idea of the EAST detection network. One of its key innovations is to shorten this kind of multi-stage into two-stage: FCN and NMS (Non-Maximum Suppression), which not only reduces the detection time, but also leads to better recognition. Efficient detection is a major feature of the EAST network, which removes many redundant intermediate layers. The network structure is shown in Figure 4, which is divided into three parts.

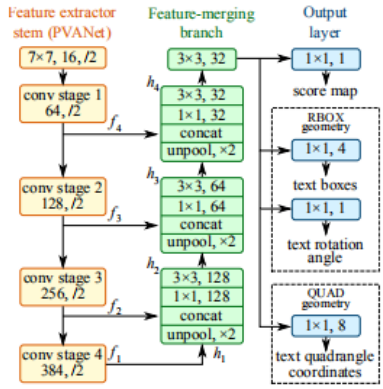


Figure 4 The EAST Network

The first one is the Feature Extractor Layer. Some basic networks, such as VGG and Resnet, are used for feature

extraction. Features are extracted from top to bottom respectively to obtain feature maps of different scales: f4, f3, f2, and f1.

Then, the second is the Feature Merging Layer. In this layer, the feature map obtained from the previous step is sampled from bottom to top and then concat.

Figure 5 The Result of EAST

The last part is the Output Layer. This is to output a score map and the coordinate information of the detection box. The coordinate information can be four regression detection boxes and one angle information, or eight coordinate information of detection boxes.

According to an analysis of the above network structure, it can be found that compared with CTPN, EASR is designed to be simpler and has fewer layers. Therefore, one of the biggest advantages of the EAST network is to enable efficient and fast detection. In addition, the EAST network has another two biggest advantages:

1) Compared with CTPN, the detection accuracy of horizontal text is better, but there are errors and omissions in other directions. The EAST network performs much better than CTPN in detecting a multi-angle text, as shown in Figure 5, the blue box shows the detection result of CTPN, while the green box reveals that of EAST. It shows that the detection results of EAST are better when referring to multi-angle text.

2) The EAST network can detect long texts and word-level texts effectively. However, the EAST network is not good at detecting long texts due to its simple network and limited receptive fields.

C. Detecting Oriented Text in Natural Images by Linking Segments

In [5], a relatively novel network, the SegLink network, is proposed. Similarly, compared with CTPN, the SegLink network solves the problem that CTPN is not effective in detecting multi-angle texts. The main detecting process of SegLink net is shown in Figure 6.

Figure 6 The SegLink Network

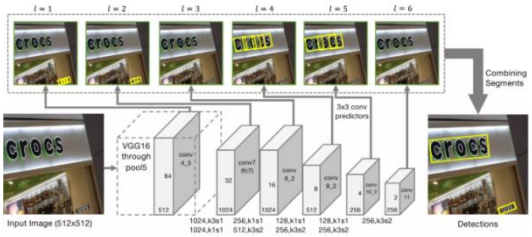


Figure 7 The result of the SegLink Network

1) Segments are detected, which can be a part of the text line. This can also be a character, etc.

2) Then these Segments in the same text line are linked, which explains why it is called SegLink. It stands for Segment and Link. This is also the main innovation [5]. In order to successfully detect the Segments which belong to one text line, and link those of the same text line, the network model needs to learn the following two things:

a) The Location of Segment

b) The Link Relationship between Segments

Network results of SegLink are shown in Figure 7.

a) This net adopts VGG16 as the backbone of the network and replaces the two full connection layers (FC6 and FC7) with two convolutional layers (conv6 and CONV7), followed by four convolutional layers. As for the scale of these six convolutional layers, each layer is only half in size of the previous one.

b) Then take out feature maps of these six convolutional layers of conv_4_3, conv_7, conv_8_2, conv_9_2, conv_10_2 and conv11 and convolve them to get Segment and Link.

(1) The Segment Detection

The Segment Detection, is similar to the SSD model proposed in [8]. Seven channels are obtained by convolving each feature map, two of which are the confidence of the Segment in the text line, and the other five are the offset.

(2) Link Detection

Link Detection can be divided into two situations:

1) In-layer Link Detection

As shown in Figure 8(a), in-layer detection is based on the detection of 8 fields of each Segment under the same feature layer, and the score between two segments is given. The positive score represents the same text, while the negative score reveals the opposite.

2) Cross-layer Link Detection

As shown in Figure8 (b), the cross-layer Link Detection

mainly solves the problem of redundant detection caused by the same text being detected in different layers.

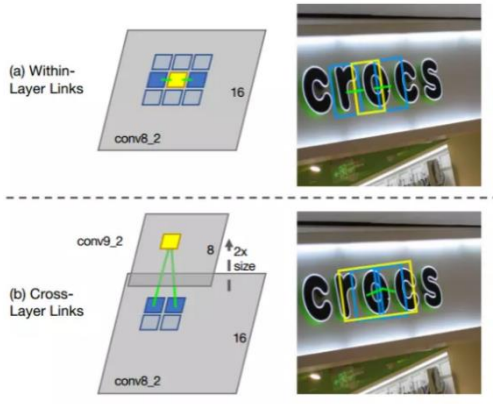


Figure 8 Link Detection

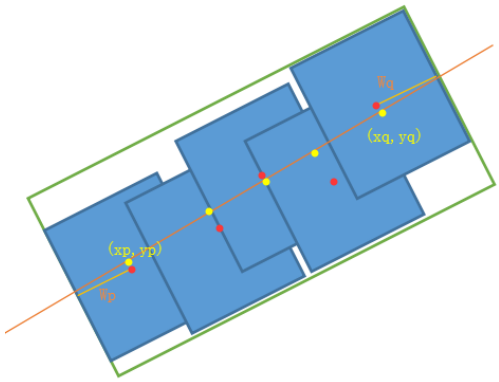


Figure 9 A merging Algorism

3) Finally, a merging algorithm is adopted, as shown in Figure 9. Link the Segment to get the final detection box. The steps of the merging algorithm are as follows:

•In the previous step, we get the Segment of the same text line.

•Conduct a linear regression of the center of these Segments, that is, the red dot in Figure 9, to get a straight line.

• Each Segment is projected vertically to this straight line, and the two points (XP, YP) and (XQ, yQ) in the furthest distance are taken from all projection points, namely the yellow ones in Figure 9.

•The final calculated text box is as follows:

•Center Location: ((xp+xq)/2, (xp+xq)/2)

•Width: the distance between 'p' and 'q', and with half widths of the Segment where these two points are located.

•Height: the average height of all Segments.

It can be found that SegLink performs better than CTPN in detecting multi-angle texts, because the slope of the straight line obtained by the linear regression in the final merging algorithm contains the angle information of the detection box. However, it is precisely because the text box is obtained by a linear regression of straight lines that SegLink cannot detect a curved text well.

III. Text Recognition

Using the previous OCR technology, the text recognition task is divided into two steps: single character segmentation and classification. The traditional way is to cut out a single character by projection, then classify and recognize this single character. It can be found that the robustness of this method is not good, because when characters are cut by projection, it will be affected by intermediate step errors, such as binarization, Gaussian Blu, etc. These errors will gradually accumulate, resulting in an unsatisfactory segmentation, and a worse result of classification and recognition. At present, deep learning is widely used in various fields, so the traditional text recognition method is outdated. It is more popular to add deep learning method in the present day. This series of methods does not need to divide the text explicitly, but to recognize the whole text image.

After converting texts into sequences, an end-to-end text recognition net can be realized by different network and translation, which is better than traditional methods in recognition accuracy and robustness.

At present, there are two trends in end-to-end text recognition tasks: CRNN (Convolutional Recurrent Neural Network) and Attention. The difference between these two methods lies in the output layer. The two main methods both apply the network structure of CNN+RNN, and the difference is how to transform the sequence feature learned by the network into recognition results. The method that CRNN adopts is the CTC (Connectionist Temporal Classification) Algorithm, while the Attention type adopts the Attention Mechanism.

A. CRNN: An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition

Given the text recognition, CRNN, proposed by Professor Baoguang Shi from Huazhong University of Science and Technology, is pioneering. Previously, text recognition was only realized by traditional rejection, character segmentation and recognition of single character, so the result was often unsatisfactory under complicated scenes. However, CRNN is a text recognition network functioning from one end to another and addresses the most difficult terminal-to-terminal OCR problem – how to align sequences with variable lengths. Compared to the traditional method, CRNN not only gets rid of the complicated explicit process of character segmentation, but also tests the robustness of characters better.

The proposal of CRNN is based on an achievement in voice recognition. In [6], researchers put forward using CTC Loss (Connectionist Temporal Classification Loss) to recognize speech sequences. At the transformation phase, referring to the advantage of CTC that is able to process speech sequences with variable lengths, CRNN applies CTC Loss to recognize text sequences with variable lengths.

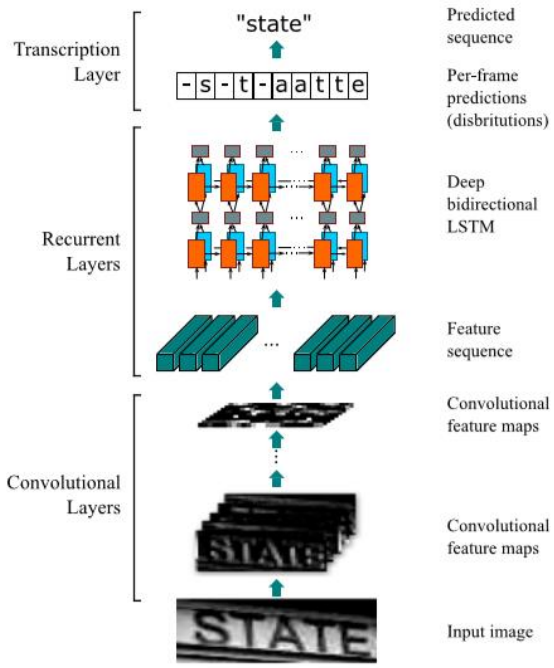


Figure 10 The CRNN Structure

Network Configuration Summary

Type	Configuration
Transcription	-
Bidirectional-LSTM	#hidden units:256
Bidirectional-LSTM	#hidden units:256
Map-to-Sequence	-
Convolution	#maps:512,k2*2,s1*1,p0*0,bn
MaxPooling	k2*2,s1*2
Convolution	#maps:512,k3*3,s1*1,p1*1
Convolution	#maps:512,k2*2,s1*1,p1*1,bn
MaxPooling	k2*2,s2*2
Convolution	#maps:128,k3*3,s1*1,p1*1
MaxPooling	k2*2,s2*2
Convolution	#maps:64,k3*3,s1*1,p1*1
Input	W*32 gray-scale image

*The first row is the top layer. 'k', 's', 'p' stand for kernel, stride and padding sizes respectively. For example, "k3*3" represents 3*3 kernel size. "bn" stands for batch normalization.*

Figure 11 Before the set of CNN is changed

Figure 12 After the set of CNN is changed

The network structure of CRNN is shown in Figure 10, and CRNN is mainly composed of three parts

1) The Convolutional Neural Network

The convolutional neural network functions as extracting a feature map from text images to transfer it to RNN (Recurrent Neural Network) for training. It can use multiple universal models to extract a feature map, such as VGG.

Researchers make a key modification of the setting of the convolutional neural network, as shown in Figure 11. The author changed two Max Pooling layers from 2x2 into 1x2, meaning that the height has been halved four times, and the width only twice, when a feature map is extracted. Since input text images are always slender, as shown in Figure 12, such a slight modification can extract the feature map more accurately, conductive to the following training.

2) Recurrent Neural Network (RNN)

Before being transferred into RNN for training, the feature map from the convolutional neural network still needs some modifications to meet the feature vector sequence requirement of RNN, as shown in Figure 13, where every feature vector is generated from left to right above the feature map and contains features of 512-dimension.

Then, the yielded feature vector is transferred into RNN for training. Every feature vector in a feature sequence is a very time step of RNN. Since ordinary RNN may face gradient disappearance problem and fail to get context information, CRNN uses a two-way LSTM(Long-Short Term Memory), as shown in Figure 14.

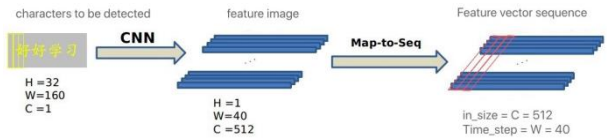


Figure 13 RNN Feature Vector

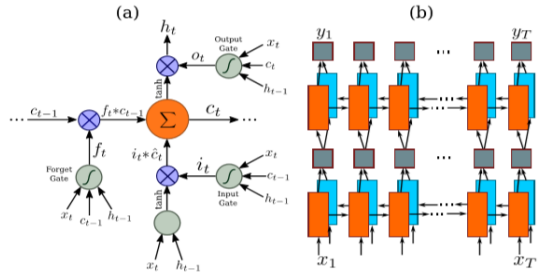


Figure 14 Bidirectional LSTM

3) Connectionist Temporal Classification Loss (CTC loss)

In the former RNN, which outputs a probability of every character corresponding to each feature sequence, how to transcribe the probability into a corresponding character is the very difficulty of end-to-end text recognition. Referring to CTC Loss proposed in [7], CRNN is able to model the

text recognition into a sequence recognition task, realizing the end-to-end training.

CTC Loss mainly functions to combine different sequence recognition results. CTC applies a sequence combination system. As for a text image in Figure. 15, our expected recognition result is “ab”. However, if the sequence combination system is not designed, the result will be “aaabbb”, obviously not the result we want. It may be available for a simple example, such as to delete repeated characters, but it will not work when facing words that contain repeated characters themselves, like “book”. Thus, a sequence combination system is needed to be designed to address this problem.

CTC combines sequences by adding a bland label in character library. We use a “-” to mark it. As for the example in Figure 15, the recognition result will be “a---b”, and the possible sequence of book will be “bbooo-ookk”, which is called as coding. The method of decoding is, firstly, to delete repeated characters, for instance, the recognition sequence of book after deletion would be “bo-ok”, secondly, to delete the bland label, i.e. “-”, the final result would be “book”.

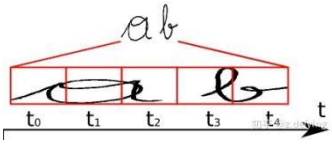


Figure 15 CTC LOSS

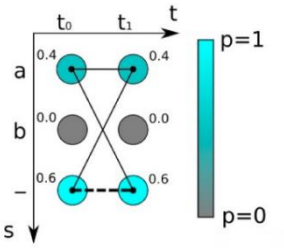


Figure 16 Calculation in RNN

In addition to the sequence combination system, another important contribution of CTC is CTC Loss which is also the key to realize the text recognition to end-to-end training. As for the simple example of ab in *Figure*. 15, the calculation of RNN refers to Figure 16, where “aa”, “a-“, and “-a” may be recognized as “a”, as the black route in Figure 16 shows. As for the expected output sequence, we make $x = (x^1, x^2, \ldots x^T)$, wherein T is the sequence length, then, Loss can be calculated:

$$p(l|x) = \sum_{\pi \in B(l)} p(\pi|x)$$

π is one of the routes through which the recognition result is x, so the Loss function means to get a sum of probabilities of all routes through which the recognition result is an expected text, then our training target can be clearly set as maximizing the value of the function.

In virtue of CTC Loss, the text recognition can be designed as an end-to-end network to simplify the processing and to increase the accuracy of recognition.

In [7], we mentioned that CRNN is pioneering, however, as OCR technology research progresses, its range expands from documents to OCR under a natural scene, where a text is often unconventional and can be rotated towards various angles, even can be curved, so the result must be unsatisfactory if the text is directly sent to CRNN for recognition. If the text is only be rotated towards a certain angle, CRNN can also be used to recognize after the detection result is rotated towards the horizontal direction; while for a curved text, there is no universal method available, in that how to rotate or curve different texts totally depends on designers’ idea.

In [8], researchers proposed a special network – the RARE Network to recognize the foresaid curved and rotated text. The network can not only recognize the text, but also adjust it. The whole idea is to adjust first and to recognize later. As shown in Figure 17, the curved text is adjusted to a normal one in horizontal direction, then it is transferred to a recognition network.

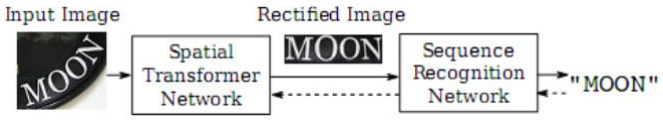


Figure 17 The Function of the Recognition Network

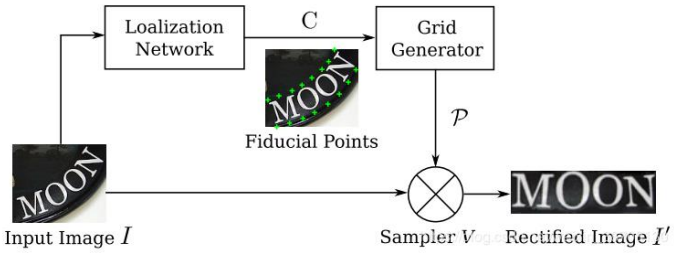


Figure 18 The Whole Structure of STN

4) Adjustment – Spatial Transformer Network(STN)

The overall structure of STN refers to Figure 18. First of all, a set of datum point C are got from the input image I by a location network. Based on the datum point C, a grid generator forms. Finally, the image will be adjusted after being sent into the sampling grid P.

In the location network, the quantity of datum point is marked as K, then the coordinate set should be $C = [c1, c2, \ldots ck]$. Every coordinate ci is composed of x and y coordinates. The network applies CNN to achieve a regression. The output quantity of the last FC Layer would be 2k and be normalized to (-1, 1) by tanh function. Since the normalized coordinate is also applied to the coordinate of datum point, the range remains the same. The most significant feature of the network is Ground Truth which requires no mark of datum points. The network is supervised

by partial networks of STN, and the principle of back-propagation is implemented to transfer a gradient into the location network for upgrading and learning.

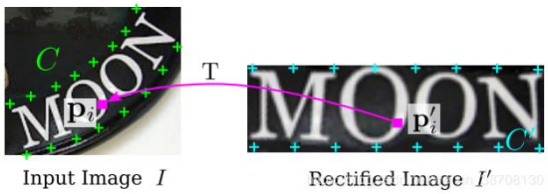


Figure 19 The Mesh Generator

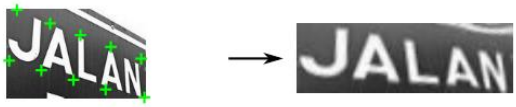


Figure 20 Result from the Sampler

The mesh generator is used to adjust a bent frame to its normal shape with the help of the TPS transformation mentioned in [11]. This comes from the principle where that a bent steel pate could be adjusted to its normal shape with the help of some fixed control points, which is shown in Figure 19. Therefore, the core of the TPS transformation is to calculate the transformation matrix based on the previous datum points, in order to get the datum point of a standard text.

Finally the simpler uses the rectified datum point to implement linear interpolation on samples from the previous datum points. Figure 20 is the generated picture.

5) *Recognition — Selective Refinement Network (SRN)*

After the text imagine is rectified, we will recognize the text using tools such as CRNN. We will make a model of text recognition, a task that is viewed as sequence recognition. CNN and RNN are used respectively to encode the text and calculate the probability. RARE uses a 7-layer CNN and a bidirectional LSTM for text recognition. The Loss function is similar to CRNN which uses CTC Loss.

B. ASTER: An Attentional Scene Text Recognizer with Flexible Rectification

Similar to RARE mentioned in [8], the ASTER network mentioned in [9] has a rectification network and a recognition network. Compared with RARE, ASTER is more accurate and faster. Its author has applied for a commercial patent for the network. This proves its great accuracy and speed, and shows that the network has commercial value and can be launched in the market.

The ASTER network has two parts:

1) The Rectification Network

The rectification network of ASTER shares the same principle with that of RARE. As shown in Figure 21, a group of datum points are generated through the location network. A transformation matrix T is obtained through the TPS transformation. At last, the sampler is used to implement linear interpolation to create the outcome of a picture.

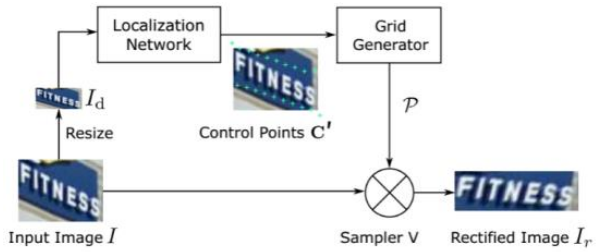


Figure 21 The Attention Mechanism

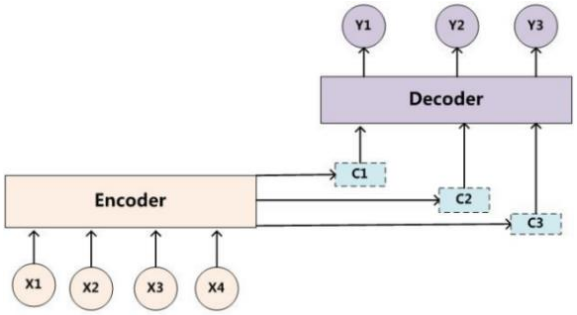


Figure 22 The Rectified Network of ASTER

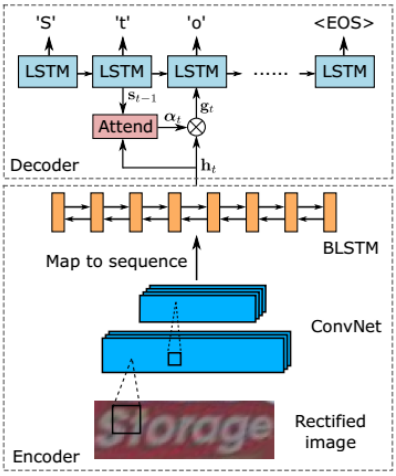


Figure 23 The Design of the Recognition Network

However, the rectification network of ASTER is different from STN of RARE. There are some nuances.

1) Before the text image from text detection input into the network, the ASTER network resizes the image to a small image and locate it according to the datum point. At last, the imagine is resized to its original size. The purpose is to largely reduce the network parameters and thus the detection requires less computation, which plays an important role in the commercialization of ASTER.

2) STN in RARE uses tanh to narrow the value domain when locating the last FC layer in the network, in comparison with ASTER that directly conducts clipping instead of using tanh. This helps accelerate network convergence as well as preventing sampling points from falling outside the domain. Given that the author has applied for a patent for the network, this essay does not elaborate how the clipping methods work.

2) The Recognition Network

The ASTER network is a good example of another mainstream text recognition. The recognition part of ASTER uses the Attention mechanism rather than CTC. The

Attention mechanism is shown in Figure 22. It is a common recognition technique that comes from voice recognition. The Encoder-Decoder structure, as shown in Figure 23, is a coding and decoding process of network output. During the process, x1, x2, etc. are of the same weight. In fact, not every input is equally important, which is similar to our daily conversation where not all words count. According to this, the Encoder-Decoder structure featuring the Attention mechanism is introduced to voice recognition.

According to Encoder-Decoder, the recognition network could be designed as Figure 23. The feature extraction of the text image by CNN and its sequence conversion by RNN can be regraded as Encoder. The Decoder section is composed of the LSTM and the Attention mechanism.

IV. Table Detection

A. Table Detection using Deep Learning

Prior to this, the most effective method to recognize tables without deep learning was through a series of corrosion and expansion operations, in order to highlight the frame. There are two obvious flaws of this method: 1. It is impossible to separate the content of the table from the table itself. 2. It is unable to accurately recognize tables that do not have borders on all four sides or tables with various border styles.

Given the diverse styles of tables, traditional detection techniques are infeasible, so in [10] the author introduced Faster RCNN mentioned in [1] to detect tables. The key aspect of the paper is how to distinguish the table from other textual content, since in a document, images, text and forms, are the same to a computer. Faster RCNN is used to detect objects because in a natural scenario, an object is more prominent in contrast to a table in a document. To apply Faster RCNN to table detection, the table must be prominent.

The author mainly used distance transformation to process the image, and then sent it to Faster RCNN for training and recognition. The steps of this processing algorithm are shown in the following picture.

```
procedure IMAGE TRANSFORMATION(I)
  b ← EuclideanDistanceTransform(I)
  g ← LinearDistanceTransform(I)
  r ← MaxDistanceTransform(I)
  P ← ChannelMerge(b,g,r)
  return P
```

M	L=1√5^{-1}		L=10√5^{-1}		L=100√5^{-1}	
Gev	5π	3π	5π	3π	5π	3π
300	0.596	0.694	0.800	0.867	0.923	0.950
400	0.450	0.546	0.671	0.762	0.851	0.902
500	0.340	0.423	0.543	0..642	0.758	0.833
600	0.258	0.327	0.431	0.526	0.650	0.743
700	0.151	0.231	0.318	0.386	0.500	0.599
800	0.137	0.172	0.232	0.292	0.388	0.478
900	<0.1	0.131	0.174	0.220	0.294	0.370
1000	<0.1	<0.1	0.132	0.164	0.223	0.282
1100	<0.1	<0.1	<0.1	0.123	0.165	0.208
1200	<0.1	<0.1	<0.1	<0.1	0.122	0.149

(a)

(b)

Figure 24 Comparison between the photo unprocessed and processed

Bow Running Time

TrainSet	Work	50(s)	100(s)	500(s)	1000(s)	2000(s)
100car 100pod 100cyl	Train Peod	0.09 9.58	0.16 9.86	0.47 10.2	1.36 10.6	3.06 11.7
200car 200pod 100cyl	Train peod	0.20 9.58	0.29 9.82	1.22 9.73	2.39 10.2	5.7 11.6
500car 500pod 100cyl	Train peod	0.50 9.27	0.57 9.68	4.00 10.4	8.11 11.3	17.8 18.4
1000car 2000pod 100cyl	Train peod	1.37 9.38	2.45 9.70	11.6 11.1	21.9 12.6	46.7 15.2
4000car 1000pod 100cyl	Train peod	6.24 9.83	10.3 9.68	34.5 10.9	90.2 13.2	170.5 17.1

TrainSet	Work	50(s)	100(s)	500(s)	1000(s)	2000(s)
100car 100pod 100cyl	All Bal	58.0% 56.0%	55.8% 55.8%	63.7% 63.7%	67.8% 67.8%	73.3% 73.3%
200car 200pod 100cyl	All Bal	53.4% 66.3%	56.3% 67.2%	56.2% 75.3%	54.3% 76.4%	53.6% 77.4%
500car 500pod 100cyl	All Bal	48.8% 68.7%	47.6% 71.4%	50.8% 73.8%	50.0% 75.0%	51.3% 74.7%
1000car 2000pod 100cyl	All Bal	48.6% 72.8%	48.9% 72.4%	51.4% 75.4%	51.3% 75.4%	52.9% 78.9%
4000car 1000pod 100cyl	All	33.3%	40.6%	38.3%	39.3%	39.9%

TrainSet	Work	OVO(S)	OVA(S)	K-class(S)
100car 100pod 100cyl	Train predict	3.03 11.7	4.24 11.8	1.13 0.48
200car	Train	5.71	10.1	0.55

200pod 100cyl	predict	12.0	12.3	0.33
500car 500pod 100cyl	Train predict	17.8 13.4	31.8 15.1	NA NA
1000car 2000pod 100cyl	Train predict	46.7 15.2	91.6 19.2	NA NA
4000car 1000pod 100cyl	Train predict	176.5 17.1	324.9 22.1	NA NA

Figure 25 Result of RCNN（Tables）

The author performed the European distance transformation, the block distance transformation and the maximum distance transformation for the three color channels B, G and R respectively, and finally merged the three processed color channels to obtain the processed image.

The result is shown in Figure 24. Figure 24 (a) is the unprocessed image. Figure 24 (b) is the processed image. It can be seen that the table content is merged and thus it is more prominent compared to the background of the unprocessed image. This is one of the most critical points of the paper, and has led researchers to pay attention to the application of deep learning in form detection.

The result, shown in Figure 25, shows that it is a right decision to introduce Faster RCNN to table detection.

V. Conclusion

In recent years, deep learning has been increasingly popular and it has begun to be applied to various areas ranging from facial recognition to target detection, OCR included. Researchers continuously apply target detection networks of different types such as Faster and RCNN to the OCR area, in order to detect texts. Compared with traditional text recognition, text recognition embedded by deep learning is more accurate with stable noise immunity and robustness. It is able to resist influences such as changes in backgrounds. The end-to-end network of deep learning is more accurate when used in the area of text detection, compared with traditional image processing and cutting, and character classification enabled by machine learning. Moreover, as research progresses, researchers have shifted their focus from everyday document processing to more complex text detection and text recognition in natural scenes, and developed excellent structures such as CTPN and CRNN.

Apart from text detection and text recognition, table recognition has also gained attention from researchers. The diverse styles of tables and their internal structures make it no less difficult to recognize them than those recognized by OCR in natural scenes. Therefore, there have been numerous table recognition techniques in conferences such as ICDAR (International conference on Document Analysis and Recognition). Among them, some use traditional image processing, some use improved table recognition techniques on networks such as Faster RNN and YOLO. Some papers even began to create an end-to-end network to for table detection and recognition.

References

[1] Platt, J. "Fast Training of Support Vector Machines using Sequential Minimal Optimization. " 2000.

[2] Ren S, He K , Girshick R , et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. 2015.

[3] Tian, Zhi, et al. "Detecting Text in Natural Image with Connectionist Text Proposal Network." (2016).

[4] Zhou, Xinyu, et al. "EAST: An Efficient and Accurate Scene Text Detector." (2017).

[5] Baoguang Shi, Xiang Bai, and Serge Belongie. "Detecting Oriented Text in Natural Images by Linking Segments." 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) IEEE, 2017.

[6] Graves, A., Fernández, S., & Gomez, F. (2006). Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks.

[7] Shi, Baoguang, X. Bai, and C. 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 & Machine Intelligence 39.11(2015):2298-2304.

[8] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai. Robust scene text recognition with automatic rectification. In CVPR, pages 4168– 4176, 2016

[9] Shi, Baoguang & Yang, Mingkun & Wang, Xinggang & Lyu, Pengyuan & Yao, Cong & Bai, Xiang. (2018). ASTER: An Attentional Scene Text Recognizer with Flexible Rectification. IEEE Transactions on Pattern Analysis and Machine Intelligence. PP. 1-1. 10.1109/TPAMI.2018.2848939.

[10] Gilani, Azka, et al. "Table Detection Using Deep Learning." ICDAR IEEE Computer Society, 2017.

[11] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. IEEE Trans. Pattern Anal. Mach. Intell., 11(6):567–585, 1989.