Implementation of Text Detection and Recognition in Natural Scenes

Written by
Zhang Yixi(23020201153824),
Sheng Manjin(23020201153798),
Liang Qiuyuan(23020201153773)
Xiamen University
Xiamen, China 361005

Abstract

Automated detection and recognition of texts in natural scenes have been a research challenge for years, largely due to the arbitrary variation of text appearances in perspective distortion, text line curvature, text styles and different types of imaging artifacts. The recent deep networks are capable of learning robust representations with respect to imaging artifacts and text style changes. This paper leverages CTPN and CRNN for text detection and recognition, repectively. The CTPN detects a text line in a sequence of fine-scale text proposals directly in convolutional feature maps which allows it to explore rich context information of image. CRNN integrates feature extraction, sequence modeling and transcription into a unified framework for text recognition. It is an end-to-end trainable, in contrast to most of the existing algorithms whose components are separately trained and tuned. By combining these two deep learning networks together, our text recogntion and detection system meets the requirements of most secene text and recognition tasks and achieves superior performance with high-accuracy testing results.

1 Introduction

Texts in scene image contain high-level important semantic information, which is help to analyzing and understanding the corresponding environment. With the rapid popularization of smart phones and mobile computing devices, images with text data are acquired more conveniently and efficiently. Therefore, scene text recognition (STR) has become active research topic in computer vision, and its related applications are including image retrieval, automatic navigation and human-computer interaction, etc. (Karaoglu et al. 2017; Yin et al. 2014).

Text detection and recognition are two fundamental tasks for STR, as Figure 1 shows. Text detection aims to determine the position of text from input image, and the position is often represented by a bounding box. Generally, the shape of target bounding box may be rectangle, oriented rectangle or quadrilateral. More precisely, parameters (x,y,w,h), (x,y,w,h,θ) and ($x_1,y_1,x_2,y_2,x_3,y_3,x_4,y_4$) can be used to denotes horizontal, rotated and arbitrary quadrilateral bounding box respectively. 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 most cases, 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. Considering a small lexicon, word spotting offers an effective strategy for realizing end-to-end recognition(Ye and Doermann 2015).

Different from traditional Optical Character Recognition that transcribes characters or words from scanned documents, scene text recognition is quite difficult due to a wide variety of factors, such as variability of font and color, distortion, occlusion, low resolution, cluttered background, and the like. Based on the techniques it uses, STR can be roughly divided into two types: traditional methods and methods based on deep learning.

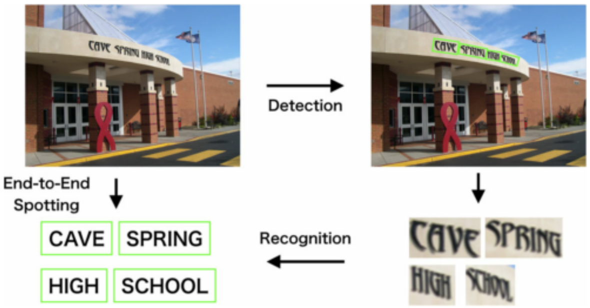

Figure 1: Schematic diagram of scene text detection and recognition.

In early research, hand-crafted features were used for text recognition, such as histogram of oriented gradients descriptors, connected components, and stroke width transform. The traditional methods framework can mainly be divided into five steps: text positioning, text verification, text detection, text segmentation and text recognition.

Recently, deep learning has been widely used in semantic segmentation and general object detection, and achieved great success. Accordingly, related methods are also being adopted in the field of text detection and text recognition. For text detection, semantic segmentation based detectors first extract text blocks from the segmentation map generated by fully convolutional network (FCN). After that, bounding boxes of text are obtained by complex post-processing. Similar to text detection, scene text recognition also experiences the transition from traditional means using handcrafted features to deep learning era which can br roughly classified into three categories: character classification based, word classification based and sequence based methods.

Also, Text detection and recognition are usually combined to implement text spotting, rather than being treated as separate tasks which is called as end-to-end deep learning detection frameworks. In a unified system, the recognizer not only produces recognition outputs but also regularizes text detection with its semantic-level awareness(Liao, Shi, and Bai 2018).

For our work, in text detection stage, we use a novel Connectionist Text Proposal Network (CTPN) that directly localizes text sequences in convolutional layers, avoiding further post-processing by an additional costly CNN detection model(Tian et al. 2016b). This overcomes a number of main limitations raised by previous bottom-up approaches building on character detection. For text recognition stage, we leverages a neural network model whose network architecture is specifically designed for recognizing sequence-like objects in images. The neural network model is named as Convolutional Recurrent Neural Network (CRNN)(Shi, Bai, and Yao 2015), since it is a combination of DCNN and RNN. CRNN can be directly learned from sequence labels (for instance, words), requiring no detailed annotations (for instance, characters). More details and the architecture of CTPN and CRNN are described in Section 3 and the evaluation of experiments is shown in Section 4.

2 Related Work

2.1 Scene Text Recognition

Existing scene text recognition work can be broadly grouped into two categories. One category adopts a bottom-up approach that first detects and recognizes individualcharacters. The other category takes a top-down approach that recognizes words or text lines directly without explicit detection and recognition of individual characters.

Most traditional scene text recognition systems follow a bottom-up approach that first detects and recognizes individual characters by using certain hand-crafted features and then links up the recognized characters into words or text lines using dynamic programming and language models. Different scene character detection and recognition methods have been reported by using sliding window(Wang, Babenko, and Belongie 2011), connected components(Neumann and Matas 2012), extremal regions(Neumann and Matas 2016), Hough voting(Bai, Yao, and Liu 2016), co-occurrence histograms(Tian et al. 2016a), etc., but most of them are constrained by the representation capacity of the hand-crafted features. With the advances of deep learning in recent years, various CNN architectures and frameworks have been designed for scene character recognition. For example, (Bissacco et al. 2013) adopts a fully connected network to recognize characters, (Wang et al. 2012) uses CNNs for feature extraction. On the other hand, these deep network based methods require localization of individual characters which is resource-hungry and also prone to errors due to complex image background and heavy touching between adjacent characters.

To address the character localization issues, various top-down methods have been proposed which recognize an entire word or text line directly without detecting and recognizing individual characters. One approach is to treat a word as a unique object class and convert the scene text recognition into an image classification problem(Jaderberg et al. 2016). In addition, recurrent neural networks (RNNs) have been widely explored which encode a word or text line as a feature sequence and perform recognition without character segmentation. For example, (Su and Lu 2017) extract histogram of oriented gradient features across a text sequence and use RNNs to convert them into a feature sequence. (Bušta, Neumann, and Matas 2017; Shi, Bai, and Yao 2017) propose end-to-end systems that use RNNs for visual feature representation and CTC for sequence prediction. In recent years, visual attention has been incorporated which improves recognition by detecting more discriminative and informative image regions. For example, (Lee and Osindero 2016) learns broader contextual information and uses an attention based decoder for sequence generation. (Cheng et al. 2017) proposes a focus mechanism to eliminate attention drift to improve the scene text recognition performance. (He et al. 2018) designs a novel character attention mechanism for end-to-end scene text spotting.

2.2 Recognition of Distorted Scene Texts

The state-of-the-art combining RNNs and attention has achieved great success while dealing with horizontal or slightly distorted texts in scenes. On the other hand, most existing methods still face various problems while dealing with many scene texts that suffer from either perspective distortions or text line curvatures or both.

Prior works dealing with perspective distortions and text line curvatures are limited but this problem has attracted increasing attention in recent years. The very early works (Lu, Chen, and Ko 2006) correct perspective distortions in document texts as captured by digital cameras for better recognition. These early systems achieve limited successes as they use hand-crafted features and also require character-level information. The recent works (Shi et al. 2019) also take an image rectification approach but explore spatial transformer networks for scene text distortion correction. Similarly, (Bartz, Yang, and Meinel 2018; Liu, Chen, and Wong 2018) integrate the rectification and recognition into the same network. These recent systems exploit deep convolutional networks for rectification and RNNs for recognition, which have shown very promising recognition performance.

Note some attempt has been reported in recent years which handles scene text perspectives and curvature distortions by managing deep network features. For example, (Cheng et al. 2018) describes an arbitrary orientation network that extracts scene text features in four directions to deal with scene text distortions.

3 Proposed Method

We adapt the two-staged methods for our work, Specifically, we use CTPN for text detection and CRNN for text recognition. The description of the network process as shown in Figure 2.

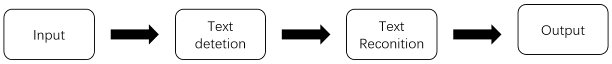

Figure 2: The description of the network processing.

3.1 CTPN

Connectionist Text Proposal Network(CTPN) is a text detection algorithm proposed in ECCV 2016. CTPN, combined with CNN and LSTM depth network, can effectively detect the horizontal distribution of text in complex scenes. CTPN model mainly includes three parts: convolution layer, Bi LSTM layer and full connection layer. The three key contributions of CTPN are follows:

First, casting the problem of text detection into localizing a sequence of fine-scale text proposals. It develops an anchor regression mechanism that jointly predicts vertical location and text/non-text score of each text proposal, resulting in an excellent localization accuracy. This departs from the RPN prediction of a whole object, which is difficult to provide a satisfied localization accuracy.

Second, proposing an in-network recurrence mechanism that elegantly connects sequential text proposals in the convolutional feature maps. This connection allows our detector to explore meaningful context information of text line, making it powerful to detect extremely challenging text reliably.

Third, both methods are integrated seamlessly to meet the nature of text sequence, resulting in a unified end-to-end trainable model. It is able to handle multi-scale and multilingual text in a single process, avoiding further post filtering or refinement.

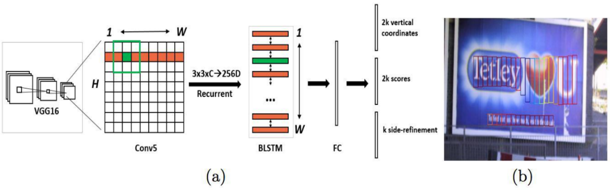

Figure 3: Architecture of the Connectionist Text Proposal Network (CTPN)

3.1.1 Detecting Text in Fine-scale Proposals

Text detection is different from object detection. Text detection does not have an obvious closed boundary, and it is also a sequence. There may be no clear distinction between multi-level components such as stroke, character, word, text line and text.

As can be seen from the above Figure 4(Left), it is difficult to accurately predict the level of word detection by RPN, because each character in the word is separated, and the head and tail of the text cannot be well distinguished. Therefore, the algorithm proposes a vertical anchor mechanism, which can simultaneously predict the text / non text score and the position of Y axis of each proposal. Relative predicted vertical coordinates (v) with respect to the bounding box location of an anchor as:

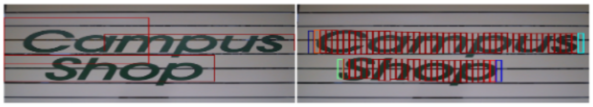

Figure 4: Left: RPN proposals. Right: Fine-scale text proposals.

$$v_c = \left(c_y - c_y^a / h_a\right), v_h = log\left(h/h^a\right) \tag{1}$$

$$v_c^* = \left(c_y^* - c_y^a / h_a\right), v_h^* = log\left(h^*/h^a\right) \tag{2}$$

where $v = \{v_c, v_h\}$ and $v_* = \{v_c^*, v_h^*\}$ are the relative predicted coordinates and ground truth coordinates, respectively. c_y^a and h^a are the center (y-axis) and height of the anchor box, which can be pre-computed from an input image. c_y and h are the predictedy-axis coordinates in the input image, while c_y^* and h^* are the ground truth coordinates.

3.1.2 Recurrent Connectionist Text Proposals

To improve localization accuracy, it split a text line into a sequence of fine-scale text proposals, and predict each of them separately. Obviously, it is not robust to regard each isolated proposal independently.

Due to the importance of context information for the detection task, this model uses bidirectional LSTM, each LSTM has 128 hidden layers. After adding RNN, the whole detection will be more robust.

3.1.3 Side-Refinement

The fine-scale text proposals are detected accurately and reliably by our CTPN. Text line construction is straightforward by connecting continuous text proposals whose text/non-text score is > 0.7. Text lines are constructed as follow. First, we define a paired neighbour (B_j) for a proposal B_j as $B_j \to B_i$, when (i) B_j is the nearest horizontal distance to B_i, and (ii) this distance is less than 50 pixels, and (iii) their vertical overlap is > 0.7. Second, two proposals are grouped into a pair, if $B_j \to B_i$ and $B_i \to B_j$. Then a text line is constructed by sequentially connecting the pairs having a same proposal.

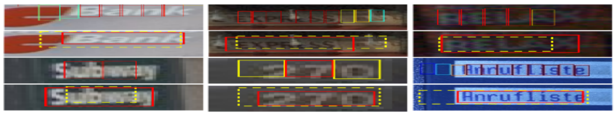

Figure 5: CTPN detection with (red box) and without (yellow dashed box) theside-refinement. Color of fine-scale proposal box indicate a text / non-text score.

When the two horizontal proposals are not covered by the text line of ground truth, the predicted position will be inaccurate. The above problems have little impact on object detection, but it can not be ignored in text detection, especially in small text detection. Therefore, side refinement is proposed to solve this problem. This method can accurately estimate the offset of each anchor / proposal in the left and right horizontal directions. The offset is calculated as follows:

$$o = \left(x_{side} - c_x^a\right)/w^a, o^* = \left(x^*_{side} - c_x^a\right)/w^a \qquad (3)$$

where x_{side} is the predicted x-coordinate of the nearest horizontal side (e.g. left or right side) to current anchor. x^*_{side} the ground truth (GT) side coordinate in x-axis, which is pre-computed from the GT bounding box and anchor location. c^a_x is the center of anchor inx-axis.w^a is the width of anchor, which is fixed,w^a= 16.

3.2 CRNN

After the processing of text detection by CTPN, a series of sequence-like objects in images can be attained. The next step is to focus on recognizing the text on the image that have been located. Convolutional Recurrent Neural Network (CRNN) is uesd to accomplish it as the architecture of CRNN is simple and it can achieves better performance on scene texts.

The network architecture of CRNN, as shown in Figure 6, consists of three components, including the convolutional layers, the recurrent layers, and a transcription layer.

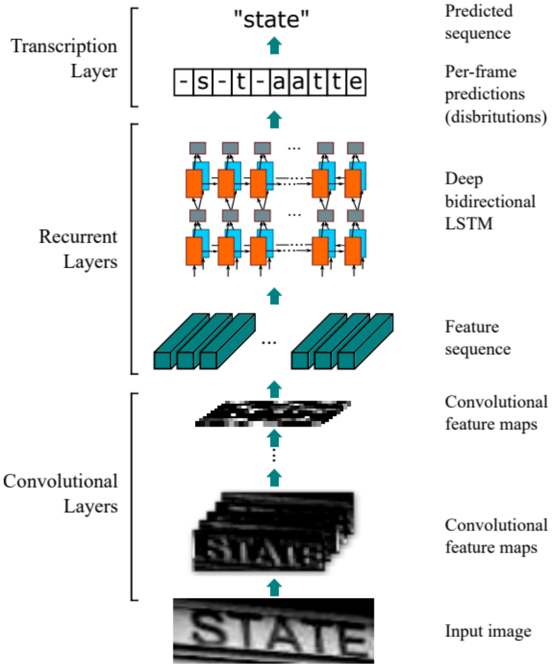

Figure 6: The network architecture of CRNN.

The convolutional layers, which extract a feature sequence from the input image, is constructed by taking the convolutional and max-pooling layers from a standard CNN model. After feeding the images into the network, a sequence of feature vectors is extracted.

In the recurrent layers, CRNN predicts a label distribution for each frame in the feature sequence. Specially, Long-Short Term Memory(LSTM) is used as the RNN unite, as it can capture long-range dependencies. As contexts from both directions are useful to each other, CRNN combines two LSTMs into a bidirectional LSTM and then stack multiple bidirectional LSTMs. The deep structure allows higher level of abstractions than a shallow one, and has achieved significant performance improvements in the task of speech recognition.

Transcription is the process of converting the per-frame predictions made by RNN into a label sequence. The algorithm adopts Connectionist Temporal Classification (CTC) layer which typically used in STR as a prediction module. CTC can maximize the likelihood of an output sequence by efficiently summing over all possible input-output sequence alignments, and allow the classifier to be trained without any prior alignment between the input and target sequences.

Denote the training dataset by $X = \{I_i, l_i\}_i$, where I_i is the training image and I_i is the ground truth label sequence. The objective is to minimize the negative log-likelihood of conditional probability of ground truth:

$$O = -\sum_{I_i, l_i \in X} \log p(l_i|y_i) \qquad (4)$$

where y_i is the sequence produced by the recurrent and convolutional layers from I_i.

4 Experiments

In this section, we will introduce the details of our experimental settings and the experimental results.

4.1 CTPN

The CTPN can be trained end-to-end by using the standard back-propagation and stochastic gradient descent (SGD). Training samples are the anchors, whose locations can be pre computed in input image, so that the training labels of each anchor can be computed from corresponding GT box.

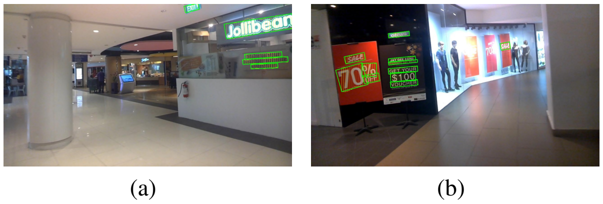
(a) (b)

Figure 7: Some examples on icdar2015.

Training data: The ICDAR 2015 (Incidental Scene Text - Challenge 4) includes 1,500 images which were collected by using the Google Glass. The training set has 1,000 images, and the remained 500 images are used for test. This dataset is more challenging than previous ones by including arbitrary orientation, very small-scale and low resolution text.

Implementation Details: We explore the very deep VGG16 model pre-trained on the ImageNet data. We set epoc is 50 and train model on gpu 0 with learning rate 0.01

and with flip data augmentation. The NVIDIA GPUs is ask for least 2GB memory.Our model was implemented in Pytorch framework.

Experimental Results: The results of training 1000 images with icdar2015 on 500 test sets were: recall: 40.58%; precision: 61.17%; hmean: 48.79%.

4.2 CRNN

We conducted experiments on CRNN, it can be trained in a end-to-end way.

Datasets We use the Synthetic Chinese String Dataset as the training data. The dataset is made by randomly changing in font, size, grayscale, blur, perspective, stretch, etc. in the chinese corpus containing news and text. The dictionary contains a total of 5,990 characters in Chinese characters, punctuation, English, and numbers. There are 3.6 million images and their corresponding ground truth words in the dataset, we divide it into training and validation sets by 9:1, and about 60,000 test sets were tested. Even though the CRNN model is purely trained with synthetic chinese text data, it works well on real images from the scene text recogition.

Implementation Details All the images are resized to 280$\times$32 during training and testing. The network configuration we use in our experiments is summarized in Table 1. In the training processing, we use the Adam optimizer with β_1=0.5, β_2=0.999 and the learning rate as 10^{-4}. We set the epoch 1000.

Table 1: Network configuration summary. The first row is the top layer.'k', 's' and 'p' stand for kernel size, stride and padding size respectively.

Type	**Configurations**
Transcription	
Bidirectional-LSTM	#hidden units:512
Bidirectional-LSTM	#hidden units:512
BatchNormalization	-
Convolution	#maps:512, k:3 × 3, s:1, p:1
MaxPooling	Window:2 × 2, s:2
Convolution	#maps:512, k:3 × 3, s:1, p:1
BatchNormalization	-
Convolution	#maps:512, k:3 × 3, s:1, p:1
MaxPooling	Window:2 × 2, s:2
Convolution	#maps:256, k:3 × 3, s:1, p:1
BatchNormalization	-
Convolution	#maps:256, k:3 × 3, s:1, p:1
MaxPooling	Window:2 × 2, s:2
Convolution	#maps:128, k:3 × 3, s:1, p:1
MaxPooling	Window:2 × 2, s:2
Convolution	#maps:64, k:3 × 3, s:1, p:1
Input	280x32 gray-scale image

Experimental Result The Accuracy of training 1000 epoch with Synthetic Chinese String Dataset on testing set were: 91.63%. The visual results of some test date as show in Figure 8, we can see that the model can correctly recognize the text on the text image in most cases.

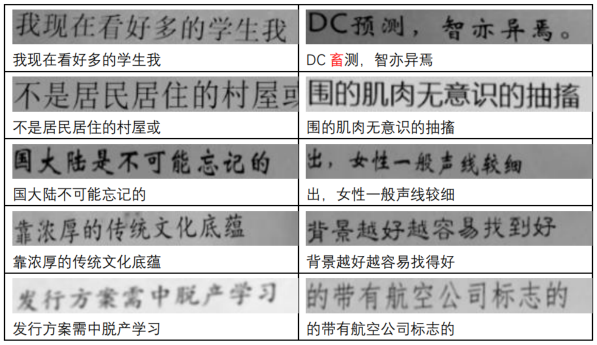

Figure 8: Some examples on test dataset.

4.3 Text Detection and Recoginition System

We looked for several scene text images on the Internet, the results of text detection and recognition on the network were shown in Figure 9.

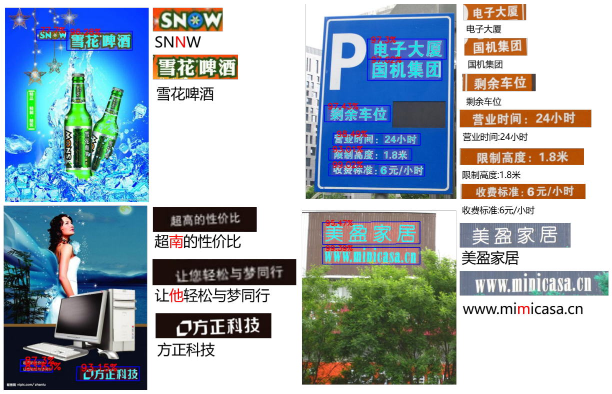

Figure 9: The results of the text detection and recognition on the images searching on the Internet.

5 Conclusion

In this paper, we implementate the text detection and recognition in natural scenes by adapting CTPN for text detection and CRNN for text reconition.

CTPN is an efficient text detector that is end-to-end trainable. The CTPN, combined with CNN and LSTM deep network, can effectively detect the horizontally distributed text in complex scenes. It is efficient by doing experiments on ICDAR 2015, with 0.14s / image running time. CRNN integrates the advantages of both Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) which make CRNN an excellent approach for image-based sequence recognition. The experiments on the dataset demonstrate that CRNN achieves superior or highly competitive performance. With CTPN and CRNN, we can get a system for text detection and recognition on the scene text images and achieve a remarkable result. Combining the two algorithms, the task of character recognition in natural scene is successfully realized.

References

Bai, X.; Yao, C.; and Liu, W. 2016. Strokelets: A Learned Multi-Scale Mid-Level Representation for Scene Text Recognition. *IEEE Transactions on Image Processing* 25(6): 2789–2802. doi:10.1109/TIP.2016.2555080.

Bartz, C.; Yang, H.; and Meinel, C. 2018. SEE: Towards Semi-Supervised End-to-End Scene Text Recognition. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16270.

Bissacco, A.; Cummins, M.; Netzer, Y.; and Neven, H. 2013. PhotoOCR: Reading Text in Uncontrolled Conditions. ICCV '13, 785–792. USA: IEEE Computer Society. ISBN 9781479928408. doi:10.1109/ICCV.2013.102. URL https://doi.org/10.1109/ICCV.2013.102.

Bušta, M.; Neumann, L.; and Matas, J. 2017. Deep TextSpotter: An End-to-End Trainable Scene Text Localization and Recognition Framework. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2223–2231. doi:10.1109/ICCV.2017.242.

Cheng, Z.; Bai, F.; Xu, Y.; Zheng, G.; Pu, S.; and Zhou, S. 2017. Focusing Attention: Towards Accurate Text Recognition in Natural Images. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 5086–5094. Los Alamitos, CA, USA: IEEE Computer Society. ISSN 2380-7504. doi:10.1109/ICCV.2017.543. URL https://doi.ieeecomputersociety.org/10.1109/ICCV.2017.543.

Cheng, Z.; Xu, Y.; Bai, F.; Niu, Y.; Pu, S.; and Zhou, S. 2018. AON: Towards Arbitrarily-Oriented Text Recognition.

He, T.; Tian, Z.; Huang, W.; Shen, C.; Qiao, Y.; and Sun, C. 2018. An end-to-end TextSpotter with Explicit Alignment and Attention.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2016. Reading Text in the Wild with Convolutional Neural Networks 116(1): 1–20. ISSN 0920-5691. doi:10.1007/s11263-015-0823-z. URL https://doi.org/10.1007/s11263-015-0823-z.

Karaoglu, S.; Tao, R.; Gevers, T.; and Smeulders, A. W. M. 2017. Words Matter: Scene Text for Image Classification and Retrieval. *IEEE Transactions on Multimedia* 19(5): 1063–1076. doi:10.1109/TMM.2016.2638622.

Lee, C.; and Osindero, S. 2016. Recursive Recurrent Nets with Attention Modeling for OCR in the Wild. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2231–2239. Los Alamitos, CA, USA: IEEE Computer Society. ISSN 1063-6919. doi:10.1109/CVPR.2016.245. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.245.

Liao, M.; Shi, B.; and Bai, X. 2018. TextBoxes++: A Single-Shot Oriented Scene Text Detector. *IEEE Transactions on Image Processing* 27(8): 3676–3690. doi:10.1109/TIP.2018.2825107.

Liu, W.; Chen, C.; and Wong, K.-Y. 2018. Char-Net: A Character-Aware Neural Network for Distorted Scene Text Recognition. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16327.

Lu, S.; Chen, B. M.; and Ko, C. 2006. A partition approach for the restoration of camera images of planar and curled document. *Image and Vision Computing* 24(8): 837 – 848. ISSN 0262-8856. doi:https://doi.org/10.1016/j.imavis.2006.02.008. URL http://www.sciencedirect.com/science/article/pii/S0262885606000904.

Neumann, L.; and Matas, J. 2012. Real-time scene text localization and recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3538–3545. doi:10.1109/CVPR.2012.6248097.

Neumann, L.; and Matas, J. 2016. Real-Time Lexicon-Free Scene Text Localization and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38(9): 1872–1885. doi:10.1109/TPAMI.2015.2496234.

Shi, B.; Bai, X.; and Yao, C. 2015. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition.

Shi, B.; Bai, X.; and Yao, C. 2017. 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* 39(11): 2298–2304. doi:10.1109/TPAMI.2016.2646371.

Shi, B.; Yang, M.; Wang, X.; Lyu, P.; Yao, C.; and Bai, X. 2019. ASTER: An Attentional Scene Text Recognizer with Flexible Rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41(9): 2035–2048. doi:10.1109/TPAMI.2018.2848939.

Su, B.; and Lu, S. 2017. Accurate recognition of words in scenes without character segmentation using recurrent neural network. *Pattern Recognition* 63: 397 – 405. ISSN 0031-3203. doi:https://doi.org/10.1016/j.patcog.2016.10.016. URL http://www.sciencedirect.com/science/article/pii/S0031320316303314.

Tian, S.; Bhattacharya, U.; Lu, S.; Su, B.; Wang, Q.; Wei, X.; Lu, Y.; and Tan, C. L. 2016a. Multilingual Scene Character Recognition with Co-Occurrence of Histogram of Oriented Gradients 51(C): 125–134. ISSN 0031-3203. doi:10.1016/j.patcog.2015.07.009. URL https://doi.org/10.1016/j.patcog.2015.07.009.

Tian, Z.; Huang, W.; He, T.; He, P.; and Qiao, Y. 2016b. Detecting Text in Natural Image with Connectionist Text Proposal Network. In Leibe, B.; Matas, J.; Sebe, N.; and Welling, M., eds., *Computer Vision – ECCV 2016*, 56–72. Cham: Springer International Publishing. ISBN 978-3-319-46484-8.

Wang, K.; Babenko, B.; and Belongie, S. 2011. End-to-End Scene Text Recognition. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, 1457–1464. USA: IEEE Computer Society. ISBN 9781457711015. doi:10.1109/ICCV.2011.6126402. URL https://doi.org/10.1109/ICCV.2011.6126402.

Wang, T.; Wu, D. J.; Coates, A.; and Ng, A. Y. 2012. End-to-end text recognition with convolutional neural networks. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, 3304–3308.