A Fast and Accurate Fully Convolutional Network for End-to-end Handwritten Chinese Text Segmentation and Recognition

***Abstract*—Handwritten Chinese Text Recognition (HCTR) is a challenging problem due to its high complexity. Previous methods based on over-segmentation, hidden Markov model (HMM) or long short-term memory recurrent neural network (LSTM-RNN) have achieved great success in recognition results. However, all of them, including over-segmentation based methods, are incompetent in accurate segmentation of single character. To solve this problem, we propose a fast and accurate fully convolutional network for end-to-end segmentation and recognition of handwritten Chinese text. Experiments on CASIA-HWDB datasets and ICDAR 2013 competition dataset show that our method achieves a competitive performance on recognition and produces great character segmentation results. Moreover, our model reaches a real-time speed of 70 fps, which is fast enough for various applications.**

***Keywords*-handwritten Chinese text recognition; end-to-end segmentation and recognition; fully convolutional network**

I. Introduction

Handwritten Chinese Text Recognition (HCTR) is a challenging problem with various applications such as bank check and mail address recognition. Although lots of solutions have been proposed and great progress has been made across more than forty years, HCTR is still not well solved today, mainly because of the large number of Chinese character classes.

Previous methods [1]–[4] based on over-segmentation, also called explicit segmentation, firstly obtain candidate segmentation-recognition paths from consecutive over-segments and then execute path search algorithm by integrating classifier outputs, geometric context and linguistic context. Combined with neural network language model [4], over-segmentation based system has achieved impressive performance on HCTR. However, the touching and overlapping characters are difficult to handle when using over-segmentation based frameworks. Moreover, the whole system is delicately designed but not end-to-end trainable. Except systems using over-segmentation, Hidden Markov Model (HMM), which was firstly applied in speech recognition, can also be applied on recognizing handwritten Chinese text lines [5], [6]. Besides, there are some methods [7], [8] exploiting Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) and Connectionist Temporal Classification (CTC) [9] recently. But the drawback of algorithms using HMM or LSTM-RNN is that their results don't contain character segmentation information, only contain the sequence of predicted character classes.

Recent years, thanks to the powerful deep neural networks, tremendous advances have been achieved on detection task. Current detection methods can roughly divided into two categories: two stage and one stage methods. Two stage detection methods [10]–[14] follow a specific pipeline: (1) region proposal algorithm such as selective search and Region Proposal Network (RPN) [12]; (2) classification on proposed regions. Apart from two stage methods, one stage methods [15]–[18] detect objects by a single neural network, viewing the detection as a regression problem. These methods have been widely applied on scene text detection and recognition [19]–[22], but have not been used to segment characters in handwritten Chinese text lines.

Taking advantage of deep neural detection networks to solve the shortcomings of previous HCTR methods, we propose a novel fast and accurate end-to-end fully convolutional network for handwritten Chinese text segmentation and recognition , which can produce character segmentation and recognition results simultaneously. Our network follows the one stage detection strategy, solving the challenging HCTR problem from a new perspective. Compared with the over-segmentation based systems, our method solves the problem of segmentation difficulty to some extent. Unlike some HMM and LSTM-RNN based methods which only fulfill recognition task, both segmentation and recognition tasks are accomplished by our approach. Therefore, our method has more various applications such as erasing the sensitive or privacy information from the text line images accurately after analyzing the recognition results.

Adopting some specific designs regarding to HCTR, our method achieves competitive performance in both segmentation and recognition while reaches real-time speed, running at 70 fps averagely. As for recognition, our method achieves 89.61% AR, 90.62% CR without language model and 94.88% AR, 95.51% CR with language model on ICDAR 2013 handwriting recognition competition dataset [1]. And the character segmentation result is 93.44% F-measure on the test set of CASIA-HWDB2.0-2.2 [23].

II. Methodology

A. Overall Architecture

Recent years, the convolutional neural networks (CNN) [24] have succeeded in many domains like image classification, object detection and semantic segmentation. Our model adopts fully convolutional structure which is more

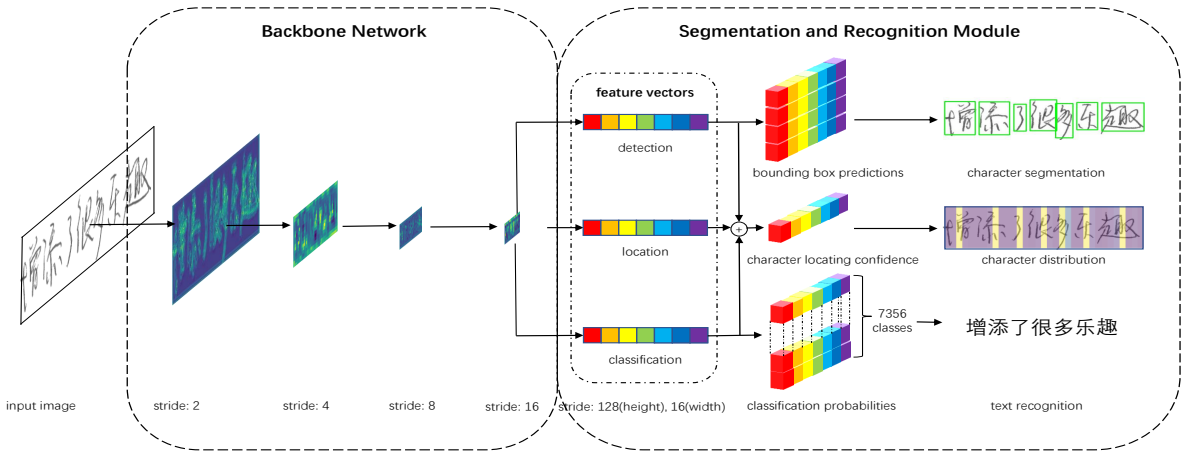

Figure 1. The overall architecture of proposed network: (1) after preprocessing, the handwritten Chinese text line image is input to the network with height of 128 pixels; (2) hierarchical feature maps are generated by fully convolutional backbone, having strides of {2, 4, 8, 16} pixels with respect to the input text line image respectively. (3) based on the final feature maps of backbone, Segmentation and Recognition Module (SRM), which consists of location, detection and classification branches, outputs both segmentation and recognition results simultaneously with the help of the character distribution.

flexible and efficient, in addition, more suitable for our task. Compared with western languages, it is very important to keep its aspect ratio when recognizing a Chinese character. Fig.2(a) shows that a Chinese character with different aspect ratios will change to another character or split into two characters. And for the recognition of text lines, it is not necessary to focus on the whole image. Only local area with suitable size is enough, as depicted in Fig.2(b). Because of the local connectivity characteristic of convolution layer, fully convolutional networks are able to process images with diverse sizes at adaptive speeds without changing their aspect ratios and make predictions only focusing on related local region. Therefore, compared with the structure with fully connected layers or long short-term memory, a fully convolutional architecture is more suitable for HCTR.

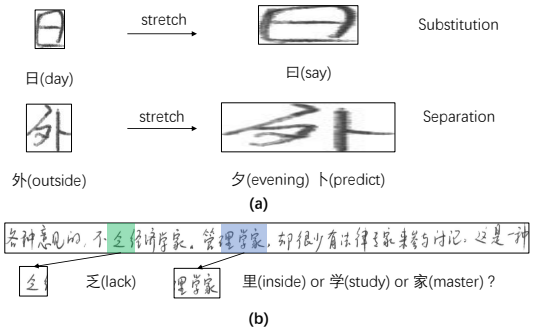

Figure 2. (a) Aspect ratio is a key factor for right recognition of Chinese characters. (b) From human perspective, a local region with suitable size should be focused on when recognizing a text line. Note that the green region is the receptive field which our model makes single prediction on while the blue one is a larger receptive field.

The overall architecture of our model is shown in Fig.1. After preprocessing, the sloped text lines are rectified to be horizontal and the heights of text lines are normalized to 128 pixels. Taking the preprocessed text lines as input, the backbone network containing several successive convolution layers and residual modules [25], produces hierarchical feature maps with strides of {2, 4, 8, 16} pixels with respect to the input text lines. Originating from the final feature maps of the backbone network, three branches are extended, aiming at detection, location and classification respectively. These three branches form the Segmentation and Recognition Module (SRM) together, which produces segmentation and recognition results at the same time.

B. Backbone Network

We propose a new backbone network to extract features from text line images for subsequent SRM. Our design is inspired by residual networks [25], which are widely used as backbone network and proven to have great performance for various tasks. The network consists of successive convolution layers with 3×3 kernel size and residual modules. Each convolution layer is followed by batch normalization and uses leaky ReLU as activation. Assuming the shape of input text line image is $128 \times W_i$, the backbone network and the corresponding output size are depicted in Table I, where each rectangle represents a block with residual module and the residual module with downsampling means before the adding operation, the input of block will be downsampled by a convolution layer with kernel size of 3 and stride of 2.

C. Segmentation and Recognition Module

Following the backbone network, the Segmentation and Recognition Module (SRM) shown in Fig.3 is applied on the extracted feature maps. The module consists of three

Table I
BACKBONE NETWORK

Type	Filters	Size/Stride	Downsample	Output
Convolution Convolution Residual	64 64	3x3/2 3x3/1	True	$64 \times \frac{W_i}{2}$
Convolution Convolution Residual	64 64	3x3/1 3x3/1	False	$64 \times \frac{W_i}{2}$
Convolution Convolution Residual	128 128	3x3/2 3x3/1	True	$32 \times \frac{W_i}{4}$
Convolution Convolution Residual	128 128	3x3/1 3x3/1	False	$32 \times \frac{W_i}{4}$
Convolution Convolution Residual	256 256	3x3/2 3x3/1	True	$16 \times \frac{W_i}{8}$
Convolution Convolution Residual	256 256	3x3/1 3x3/1	False	$16 \times \frac{W_i}{8}$
Convolution Convolution Residual	512 512	3x3/2 3x3/1	True	$8 \times \frac{W_i}{16}$
Convolution Convolution Residual	512 512	3x3/1 3x3/1	False	$8 \times \frac{W_i}{16}$

branches named location, detection and classification respectively. The location branch indicates where characters locate. The detection branch predicts the coordinates and shapes of bounding boxes. And the classification branch recognizes the character.

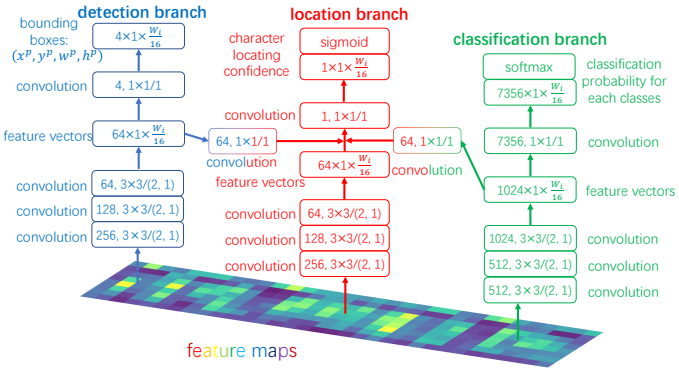

Figure 3. Design of segmentation and recognition module

Unlike object detection and recognition in natural scene, the characters in the text lines are distributed along a single dimension. Therefore, for each branch, after three 3×3 convolution layers with different strides on height and width, the height of feature maps is normalized to 1 pixel while the width stays unchanged, yielding multiple one dimensional feature vectors. Denoted as L_{fv}, the length of the feature vector is $\frac{W_i}{16}$. Thus, we apply L_{fv} consecutive horizontal grids on the text line image as shown in Fig.4 and make predictions for each grid. Different from the grid in Yolo [17], our grid has an aspect ratio of 0.125 when projected to original images, because of the different strides of convolution layer on height and width. It means that our grids distribute densely on horizontal dimension but there is only one grid on vertical dimension, which prevents the redundant predictions and computing.

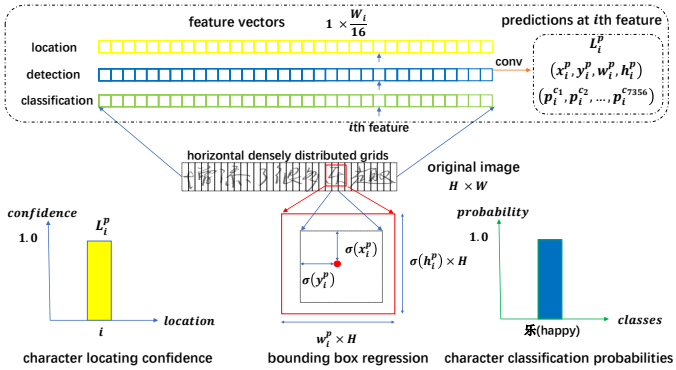

Figure 4. Prediction mechanism based on feature vectors

1) Location: It is essential to predict whether there is a character locating in the grid, which is a key factor determining the performance of the whole module. In the structure of SRM, although the three branches are responsible for different tasks, these three tasks are highly related. The detection and classification results can help determine the presence of characters. Therefore, feature vectors of other two branches undergo a 1×1 convolution layer and the outputs are added to feature vectors of location branch, forming refined feature vectors combining multiple context information. The visualization results in Fig.5(a) combine the information from detection and classification branches while the results in Fig.5(b) do not. Compared with Fig.5(b), the predicted locations in Fig.5(a) have higher recall and precision. The new feature vectors are input to one 1×1 convolution layer, resulting in a vector l^p of length L_{fv}. Applying sigmoid function to l^p, we get character location confidence vector L^p. Each element in L^p indicates the confidence of characters locating in the corresponding grid.

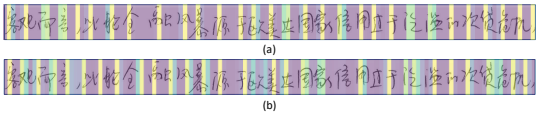

Figure 5. Combining three branches together results in more accurate character location

We adopt binary cross entropy loss on location branch and a character is viewed to locate in a grid if its central point is within the area of this grid. Because grids distribute more densely than characters, negative samples are much more than positive ones. So the losses from negative and positive samples are averaged respectively.

2) Detection: It is natural that both the heights and widths of characters are related to the heights of text line images, especially for Chinese character. Therefore, detection branch directly predicts the ratios of characters' heights and widths with respect to the height of text line image, which is simple and effective. As for the coordinates of central point, we predict the offset to the top-left point of grid like Yolo [17]. Assuming the detection results for the ith grid are $(x_i^p, y_i^p, w_i^p, h_i^p)$, the bounding box transferred to the original image is as follows:

$$x_b = \frac{i - 1 + sigmoid(x_i^p)}{L_{fv}} \times W \tag{1}$$

$$y_b = sigmoid(y_i^p) \times H \tag{2}$$

$$w_b = w_i^p \times H \tag{3}$$

$$h_b = sigmoid(h_i^p) \times H \tag{4}$$

where H, W are the height and width of the original image. Mean square error is used as loss function. Only the predictions on the grid with characters locating in are used to compute loss.

3) Classification: Owing to the high complexity of Chinese character classification (7356 classes), more feature vectors are generated in classification branch. Using a convolution layer with 7356 channels, a $7356 \times L_{fv}$ matrix is produced from the 1024 feature vectors. After doing softmax operation, the classification branch outputs the probabilities of 7356 classes for each grid. We adopt cross entropy loss to optimize the classification branch. And the same as detection branch, the losses on the grid without characters locating in are ignored.

D. Transcription

1) Without Language Model: Under the situation without language model, we combine the location confidence and classification probabilities together to be the confidence of detection results as Equation (5). Then the bounding boxes after Non Maximum Suppression (NMS) are sorted by the x coordinate. The recognition result is the label sequence of sorted bounding boxes and the bounding boxes become segmentation results.

$$Conf_i^{bbox} = 0.8 \times L_i^p + 0.2 \times \max(p_i^{c_1}, p_i^{c_2}, ..., p_i^{c_{7356}}) \tag{5}$$

2) With Language Model: We use back-off trigram language model. Viewing $1 - L_i^p$ as the probability of blank, we use beam search algorithm [26] for transcription.

III. Experiments

A. Datasets

We conduct experiments on CASIA-HWDB [23] datasets and ICDAR 2013 handwriting recognition competition dataset [1]. The detail information of both datasets is shown in Table II.

Table II
Detail information of CASIA-HWDB and ICDAR 2013

dataset	type	samples	bounding boxes
CASIA-HWDB1.0-1.2	character	3895135	-
CASIA-HWDB2.0-2.2	textline	52230	✓
ICDAR 2013	textline	3432	×

B. Data Preprocessing

As depicted in Fig.6, data preprocessing consists of four steps.

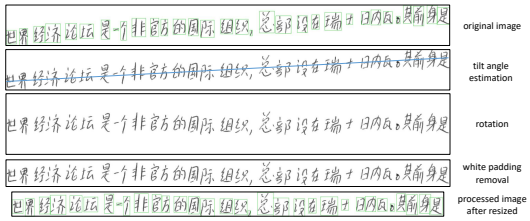

Figure 6. Data Preprocessing Pipeline

1) Tilt Angle Estimation: Given a text line image with oblique content, we use the coordinates of black points as data and fit the data by linear regression. Then we can get a line indicating the tilt angle of the content. Assuming the slope of the line is a, we can calculate the tilt angle θ. The minus is necessary because the coordinates along vertical dimension increase from top to bottom.

$$\theta = -arctan(a) \tag{6}$$

2) Rotation: To make the content of text line horizontal, the image should be rotated by θ clock-wise centered on the central point of text line. The size of image after rotation is:

$$h_r = h \times |\cos(\theta)| + w \times |\sin(\theta)| \tag{7}$$

$$w_r = w \times |\cos(\theta)| + h \times |\sin(\theta)| \tag{8}$$

where (h, w) is the original shape and (h_r, w_r) is the shape after rotation. The relationship between coordinates before rotation and after rotation is:

$$\begin{bmatrix} x_r - \frac{w_r}{2} \\ \frac{h_r}{2} - y_r \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} x - \frac{w}{2} \\ \frac{h}{2} - y \end{bmatrix} \tag{9}$$

where (x, y) is the coordinate before rotation and (x_r, y_r) is the coordinate after rotation. For each bounding box annotation (x_b, y_b, w_b, h_b), the central point's coordinate (x_b, y_b) is transformed to (x_{br}, y_{br}) using Equation (9) while h_b, w_b is transformed to h_{br}, w_{br} as Equation (7) and (8).

3) White Padding Removal: There are white paddings at top and bottom of rotated image which are useless for segmentation and recognition. We remove those paddings to highlight the area containing characters.

4) Resize: All the text line images would be resized after removing redundant white paddings in vertical dimension. The heights of images are normalized to 128 pixels while keeping aspect ratios.

C. Data Augmentation

Using the massive single character samples in CASIA-HWDB1.0-1.2, we synthesize extra 200 thousand text line samples. The number of characters in a sample has a normal distribution with a mean of 26 and a standard deviation of 7.5. The classes of characters are randomly selected. An example of synthesized data is shown in Fig.7

Figure 7. Example of synthesized data

D. Training Strategy

Although we add some random factors to the location of characters when synthesizing data, the spatial distribution of synthesized data is different from real data, which will mislead detection and location branches. Therefore, the synthesized data is only used to update parameters of classification branch.

E. Accuracy

1) Recognition: To prove the effectiveness of our model on handwritten Chinese text recognition, we train our model on 52230 samples from CASIA-HWDB2.0-2.2 and 200 thousand synthesized samples. As shown in Table III, among the results without language model, our method achieves the highest Accuracy Rates (AR) and competitive Correct Rates (CR) which is only 0.15% lower than the best results. Except the results marked by * which take advantage of neural network language model, our method also achieves the highest AR only using traditional back-off trigram language model. In addition, as shown in Table IV, we verified the effectiveness of three branches design of SRM. Single branch means producing the detection, location and classification results in one branch. Double branches mean merging detection and location branches. Triple branches refer to SRM without combining multiple features to predict location.

2) Segmentation: We compare our model with two popular detection methods, Faster R-CNN [12] and Yolo v3 [16], which are the representatives of two-stage and one-stage strategies. Different from our method, we only train them to recognize character and non-character while our method need to recognize 7357 classes, which are non-character and the 7356 character classes. The training data is CASIA-HWDB2.0-2.2 train set and the test data is CASIA-HWDB2.0-2.2 test set. Table V shows that there is not a huge gap between the performances of our method and detection methods. However, our model is much faster and smaller, as well as has abilities of not only segmentation but also recognition. Moreover, pretrained models are not required by our method. Examples of segmentation results are shown in Fig.8.

Table III
COMPARISON OF RECOGNITION RESULTS

Method	Without LM		With LM	
	AR	CR	AR	CR
HIT-2 [1]	–	–	86.73	88.76
Messina *et al.* [8]	83.50	–	89.40	–
Wu *et al.* [7]	86.64	87.43	90.38	–
Du *et al.* [6]	83.89	-	93.50	–
Wang *et al.* [3]	88.79	**90.67**	94.02	95.53
Wu *et al.* [4]	–	–	**96.20***	**96.32***
Our Method	**89.61**	90.52	**94.88**	95.51

Table IV
EFFECTIVENESS OF SRM

Structure	AR	CR
Single branch	63.08	63.10
Double branches	87.14	87.75
Triple branches	88.39	89.01
SRM	89.61	90.52

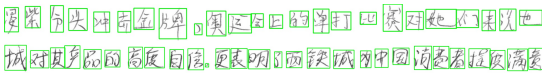

Figure 8. Examples of segmentation results

Table V
COMPARISON OF SEGMENTATION RESULTS (EVALUATED WITH IOU THRESHOLD OF 0.5)

Method	Precision	Recall	F-measure	Speed	Storage
Faster R-CNN [12]	0.9790	**0.9754**	0.9760	5fps	272M
Yolo v3 [16]	**0.9881**	0.9647	**0.9763**	15fps	246M
our method	0.9456	0.9236	0.9344	**70fps**	**125M**

F. Speed

Our model can realize real-time handwritten Chinese text segmentation and recognition, reaching an average speed of amazing 70 fps without language model and 1.16 fps with language model (sequentially run the model and decoding algorithm using language model) on GTX 1080ti GPU. Apart from the high average speed, speed of our model is adaptive for different input size due to the fully convolutional architecture. For short text line images, our model can be much faster than 70 fps.

IV. CONCLUSION

In this paper, we view the HCTR problem from a new perspective. Applying advanced detection methods to HCTR task, we propose a novel fast and accurate end-to-end fully

convolutional network for handwritten Chinese text segmentation and recognition. Our method achieves impressive performance on both segmentation and recognition and solve the shortcomings of previous methods, which shows the promising prospect of our method.

REFERENCES

[1] F. Yin, Q.-F. Wang, X.-Y. Zhang, and C.-L. Liu, "Icdar 2013 chinese handwriting recognition competition," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 1464–1470.

[2] Q.-F. Wang, F. Yin, and C.-L. Liu, "Handwritten chinese text recognition by integrating multiple contexts," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 8, pp. 1469–1481, 2012.

[3] S. Wang, L. Chen, L. Xu, W. Fan, J. Sun, and S. Naoi, "Deep knowledge training and heterogeneous cnn for handwritten chinese text recognition," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2016, pp. 84–89.

[4] Y.-C. Wu, F. Yin, and C.-L. Liu, "Improving handwritten chinese text recognition using neural network language models and convolutional neural network shape models," *Pattern Recognition*, vol. 65, pp. 251–264, 2017.

[5] T.-H. Su, T.-W. Zhang, D.-J. Guan, and H.-J. Huang, "Off-line recognition of realistic chinese handwriting using segmentation-free strategy," *Pattern Recognition*, vol. 42, no. 1, pp. 167–182, 2009.

[6] J. Du, Z.-R. Wang, J.-F. Zhai, and J.-S. Hu, "Deep neural network based hidden markov model for offline handwritten chinese text recognition," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 3428–3433.

[7] Y.-C. Wu, F. Yin, Z. Chen, and C.-L. Liu, "Handwritten chinese text recognition using separable multi-dimensional recurrent neural network," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 79–84.

[8] R. Messina and J. Louradour, "Segmentation-free handwritten chinese text recognition with lstm-rnn," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2015, pp. 171–175.

[9] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[11] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[13] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[16] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[18] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[19] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[20] M. Liao, B. Shi, and X. Bai, "Textboxes++: A single-shot oriented scene text detector," *IEEE transactions on image processing*, vol. 27, no. 8, pp. 3676–3690, 2018.

[21] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 67–83.

[22] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: an efficient and accurate scene text detector," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 5551–5560.

[23] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang, "Casia online and offline chinese handwriting databases," in *2011 International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 37–41.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[26] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International Conference on Machine Learning*, 2014, pp. 1764–1772.