Learning Spatially Embedded Discriminative Part Detectors for Scene Character Recognition

Yanna Wang*†, Cunzhao Shi*†, Baihua Xiao*†, Chunheng Wang*†
*The State Key Laboratory of Management and Control for Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing, China
†University of Chinese Academy of Sciences, Beijing, China

***Abstract*—Recognizing scene character is extremely challenging due to various interference factors such as character translation, blur and uneven illumination, etc. Considering that characters are composed of a series of parts and different parts attract diverse attentions when people observe a character, we should assign different importance to each part to recognize scene character. In this paper, we propose a discriminative character representation by aggregating the responses of the spatially embedded salient part detectors. Specifically, we first extract the convolution activations from the pre-trained convolutional neural network (CNN). These convolutional activations are considered as the local descriptors of the character parts. Then we learn a set of part detectors and pick the distinctive convolutional activations which respond to the salient parts. Moreover, to alleviate the effect of character translation, rotation and deformation, etc, we assign a response region for each part detector and search the maximal response in this region. Finally, we aggregate the maximal outputs of all the salient part detectors to represent character. The experiments on three datasets show the effectiveness of the proposed method for scene character recognition.**

***Keywords*-scene character recognition; part detectors; response region**

I. Introduction

With the popularity of mobile phones and surveillance cameras, scene text recognition becomes an important requirement for better understanding rich visual information in many real-life systems such as license plate recognition, image understanding and event retrieving, etc. Characters are the basic units of the text, and scene characters recognition has attracted increasing attentions in the computer vision community in recent years. However, scene character recognition is a challenging task since the characters always suffer from uneven illumination, background interferences, character translation, rotation and deformation, etc.

To tackle the existing challenges, a powerful representation is critical to scene character recognition. In this paper, we focus on scene character representation. There are two widely used feature representations for scene character recognition including hand-crafted features and neural network based features.

Most of hand-crafted feature based methods [1], [2], [3], [4], [5], [6], [7] used off-the-shelf HOG [8] like features for character recognition. They represented character image from two aspects including global-based representation and part-based representation. Generally, in order to obtain a global HOG feature, the input image is divided into several equally spaced square grids, then oriented gradient information is extracted from those predefined sub-regions. However, not all the sub-regions contain useful information. Some non-text regions may exist large gradient change and generate strong feature histogram values, which disturb the representation of scene character images. Moreover, these features are extracted from the pixel unites which do not contain more semantic information. In view of these drawbacks, existing character recognition systems using global hand-crafted features are considered unsatisfied and limit the overall system performance in unconstrained natural scene images.

Thus, some researchers [4], [9], [2], [10], [6] utilized part-based information instead of global information for scene character recognition. They aimed to learn useful character part representation to avoid the undesirable influence of background and obtained more semantic content. Shi et al. [4] introduced a part-based tree-structured model and Gao et al. [9] proposed a character representation named stroke bank. However, these two methods both need manually labeled character parts. Yao et al. [10] proposed the "strokelet" which was a multi-scale representation for character recognition. Li et al. [6] learned shareable character part features using filter banks. Compared with global-based representation, these part-based representations attain significant performances with intrinsic character part information. However, these methods need to predefine the parts with human annotation or obtain the parts by scanning the images with the sliding windows, then extract the hand-crafted features (e.g., HOG) to represent parts. They separate parts generation and feature extraction steps. Moreover, due to the insufficient discrimination ability of these hand-crafted features, the recognition accuracy is still not satisfactory.

Recently, a trend in the computer vision community has emerged towards deriving a global representation from neural network. Several work [11], [12], [13] utilized deep neural network model such as convolutional neural network (CNN) to recognize characters. These methods obtained a

Baihua Xiao* is corresponding author.

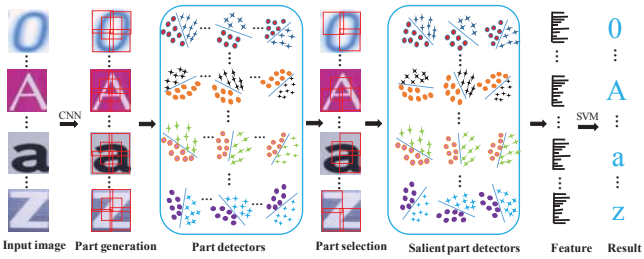

Figure 1. Flowchart of the proposed method for scene character recognition.

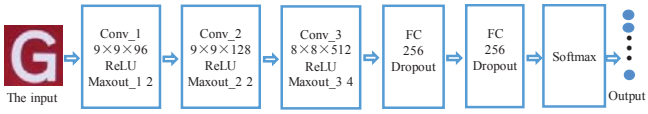

Figure 2. The architecture of our CNN model for scene character recognition. "Conv" represents the convolutional layer and "FC" represents the fully connected layer. The maxout operation is introduced in [17].

global representation from CNN model and achieved appealing improvement over hand-crafted feature based methods. However, the feature was extracted from the fully connected layers instead of the convolutional layers which conveyed more spatial structure information than fully connected layers.

Motivated by these analyses, considering that characters consist of a series of parts and different parts attract diverse attentions when people identify a character, we should assign different importance to each part to recognize a scene character. In this paper, we focus on learning discriminative part detectors for scene character recognition, which can endow the parts with different weights. Specifically, we first regard the convolutional kernels of CNN as a set of filters to extract the discriminative local descriptors for all the scene character classes, which is capable of representing the parts and learning the part features simultaneously. Further, we automatically learn the part detectors with different weights and pick the salient parts for each character class, which capture the important information of characters and decrease the effect of interference factors. Besides, in order to alleviate the influence of translation, rotation and deformation, etc, we embed the spatial location information into the part detectors. Finally, we generate a character representation by assembling all salient part confidences and use an SVM classifier for scene character recognition. We have conducted experiments on three datasets including three standard benchmarks ICDAR03 dataset [14], Chars74K dataset [15] and IIIT5K dataset [16]. The experimental results demonstrate the effectiveness of the proposed method for scene character recognition.

II. Proposed Method

In this section, we present the proposed method for scene character recognition. Our method mainly contains five steps: (1) generating the representation of character parts via CNN model; (2) learning the part detectors; (3) selecting the salient parts and corresponding detectors; (4) obtaining the feature representation; (5) getting the recognition result. The overall framework is given in Figure. 1.

A. Generating Part Representation

A large number of work [4], [2], [10], [6] have verified that using part information can improve scene character recognition performance. Besides, CNN has shown the powerful classification performance in many visual fields, which motivates us to obtain the character part feature with CNN.

CNN generates convolutional maps by a set of convolutional kernels, in which each spatial position is computed from a receptive field in the input image. Intuitively, each position of convolutional maps corresponds to a character part (i.e., a subregion of a character) and the layout of convolutional map reflects the spatial structure of these parts. To make full use of the spatial structure information of parts, in this paper, we use the features of convolutional layers rather than the fully connected layers.

Assume the output T from a convolutional layer is $H \times W \times D$ dimension, which includes a set of feature maps $\{M_n, n = 1, ..., D\}$. M_n with size $H \times W$ is the n-th feature map. The same position of the different convolutional maps reflects the identical part of the original character image. To enhance the part information, we extract the activation responses from all the convolutional maps at the same position to represent the corresponding part. As a result, each part is represented as a D-dimensional feature descriptor.

We train CNN for scene character recognition and the network architecture is shown in Figure. 2. We extract the activations of $conv\_2$ (after applying the ReLU) in our method. Assuming that the size of the input image is 24×24, the size of activation maps of $conv\_2$ is $8 \times 8 \times 128$.

B. Part Detectors

For the part descriptors obtained with CNN, we learn the part detectors. A good part detector has the capacity of distinguishing the special class from others. In this paper, for the input image, we first generate an initial global representation by concatenating all the convolutional descriptors. Then we learn the part detectors with SVM. The procedure is presented as follows:

(1) The p-th convolutional descriptor x_p of an input image is represented as:

$$x_p = [r_1^{i,j}, r_2^{i,j}, ..., r_{D-1}^{i,j}, r_D^{i,j}] \quad (1)$$

where $r_n^{i,j}$ denotes the convolutional response at the spatial position (i, j) of the n-th convolutional map.

Figure 3. Visualization of the spatial weight map of the convolutional layer.

(2) We represent the input image I by concatenating all convolutional descriptors in a fixed order:

$$f = [x_1, x_2, ..., x_{N-1}, x_N]^T \tag{2}$$

where N denotes the number of convolutional descriptors of an image and f is a $D \times N$-dimension vector. Then we employ L_2 normalization for the feature f.

(3) We conduct one-to-all strategy to learn a linear SVM classifier for each character class, which has the weights:

$$\omega = (\omega_1, \omega_2, ..., \omega_{N-1}, \omega_N) \tag{3}$$

where ω_i is a $C \times D$ matrix and C is the number of character categories. We regard the weights of SVM as the part detectors. Specifically, each row of ω is $N = H \times W$ part detectors belonging to the same class and each detector has D dimension. Moreover, each part detector corresponds to a special part position of character, and all the N detectors are assembled to distinguish certain class against others. Thus our detector not only has the local discriminative capacity against the corresponding position of other character classes, but also learns the global inter-class information. The parameters of SVM are learned on the training datasets.

C. Salient Part Detector Selection

Among all the parts, some come from background, which are invalid for character recognition. In Figure. 3, we exhibit the spatial weight map of the convolutional layer. The spatial weight map is generated by summing all the maps for each spatial position (i, j), i.e., $\sum_{n=1}^{D} M_n^{ij}$. Obviously, not all the convolutional responses reflect the character information. Moreover, each part exhibits different attention when recognizing character. People can identify the character by only using a portion of character parts. Besides, it will introduce the noise by using all the parts. Thus, we need to select the salient parts for each character class. Our main idea is to automatically mine the useful parts for each character class from the training datasets. The detailed process is described as follows.

(1) When we use SVM to classify with feature f, for the linear kernel case, the decision function is $\mathrm{sgn}(\omega f + b)$ and the final decision value is a linear combination $\omega f + b$. Inspired by this point, in order to obtain the parts which play greater roles for classifying the character image, for a positive training image and its label $l(l = 1, 2, ..., C)$, we calculate the score for each descriptor:

$$s_m^l = |\omega_m^l * x_m| \quad (m = 1, 2, ..., N) \tag{4}$$

where m denotes the index of part detector. We can see that the bigger value of s_m^l is, the more the corresponding descriptor contributes more to the final decision.

(2) We select the salient parts for characters. For each training image, we rank these scores s_m^l $(m = 1, 2, ..., N)$. We first select K salient parts with the top K highest scores for each image and record the positions in the convolutional map corresponding to the salient parts.

(3) We statistically learn the significant part information for the intra class. We compute the number of occurrences of salient parts within a class and select the top K most frequently appearing parts as the final salient parts. These positions of K salient parts are also recorded. This operation globally generates a histogram representation from the training dataset and alleviates the unreliable effect of individual character sample, which provides extra discriminative power to select the parts.

(4) Finally, we obtain the salient parts for each class and retain the SVM weights as the salient part detectors. The learned part set can be expressed as $\Omega = \{(Part_k, Pos_k, Weight_k)\}_{k=1}^{K}$, where $Part, Pos$ and $Weight$ are the discovered parts, corresponding positions and corresponding weights of SVM classifiers respectively.

We present two examples in Figure. 4 to show the response maps which are obtained by using salient part detectors to weight the corresponding convolutional descriptors on the special class against other classes. We use the salient part detectors of class "A" to detect characters "A", "0", "1" and "b" in the corresponding detector positions (see Figure. 4(a)). Obviously, the responses of "A" are significantly larger than other three characters. Similarly, the detectors of "B" operate on character "B" to obtain the response map which is much larger than those of "2", "C" and "W" (see Figure. 4(b)). The two examples show the discriminative capacity of the salient part detectors.

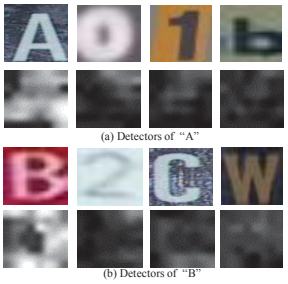

Figure 4. Two examples of salient part detectors operating on the corresponding class against other classes.

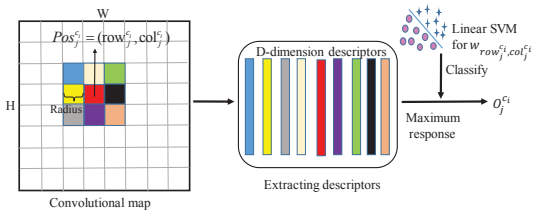

Figure 5. An example of obtaining the maximum response using a part detector.

D. Spatially Embedded Character Representation

Although we learn the salient parts for each class, it can not always precisely be adapted to each character due to character image translation, rotation and deformation, etc. Considering that characters are not the same as generic objects which randomly appear in the image, the location of the characters in the image changes within a certain range. Thus we propose a spatially embedded character representation by searching the maximal response around each salient part position with the corresponding part detector.

Specifically, for the character class c_i, at the j-th salient part position $(row_j^{c_i}, col_j^{c_i})$, we extract the convolutional descriptors from the surrounding coordinates $(row_j^{c_i} - Radius \sim row_j^{c_i} + Radius,\ col_j^{c_i} - Radius \sim col_j^{c_i} + Radius)$, where $Radius$ defines the search neighborhood size. Then we use the SVM weight corresponding to the part position $(row_j^{c_i},\ col_j^{c_i})$, i.e., $w_{row_j^{c_i},col_j^{c_i}}$ to score these extracted convolutional descriptors and conduct max-pooling to obtain the maximum response $O_j^{c_i}$, that is:

$$\begin{aligned} O_j^{c_i} &= \max_{p,q} r^{p,q} \cdot w_{row_j^{c_i},col_j^{c_i}} \\ s.t.\quad & 1 \le j \le K, \quad 1 \le Radius \le \min(H, W) \\ & row_j^{c_i} - Radius \le p \le row_j^{c_i} + Radius \\ & col_j^{c_i} - Radius \le q \le col_j^{c_i} + Radius \end{aligned} \tag{5}$$

where $r^{p,q} = [r_1^{p,q}, ..., r_D^{p,q}]$. We illustrate the process to obtain the maximum response $O_j^{c_i}$ in Figure. 5. We repeat the above step for each part in all the character classes and obtain the final feature $\mathcal{F}$ which has $K \times C$ dimensions as follows:

$$\mathcal{F} = [O_1^{c_1}, O_2^{c_1}, ..., O_K^{c_1}, ..., O_1^{c_C}, O_2^{c_C}, ..., O_K^{c_C}] \tag{6}$$

E. Character Recognition

We treat scene character recognition problem as a multi-class classification problem, e.g., English character recognition is a classification task of 62 classes. The character feature $\mathcal{F}$ is L2-normalized, and the final features of all training images are used to train multi-class SVM. For each character, the class label with the highest probability is assigned as the recognition result.

III. Experiment Results

A. Datasets

We first evaluate the proposed method on three standard English datasets including ICDAR03 [14] Chars74K [15] and IIIT5K [16]. They all contain 10 classes Arabic numbers and 52 classes English letters. ICDAR03 dataset contains 5897 training samples and 5337 test samples. For Chars74K dataset, similar to [18], we randomly select 30 images per class from which to generate 15 images for training and the rest for testing. IIIT5K dataset contains 9678 training samples and 15,269 test samples.

B. Implementation Details

The convolutional layers are initialized by the pre-trained CNN [13], and all the fully connected layers are initialized by using random weights with Gaussian distribution of 0 mean and 0.01 standard deviation. We adopt stochastic gradient descent (SGD) to fine-tune the CNN models for the four datasets using the gray images with the size of 24×24, respectively. An image is resized to 24×24 to pick the salient parts. We use MatConvNet to extract the convolutional maps and LIBLINEAR SVM to classify characters. Our method takes about 0.24s for each image on average with Inter i5 3.1GHz CPU.

C. Evaluation of the Numbers of Salient Detectors

We first evaluate the influence of the numbers (percent) of salient detectors, namely K for character recognition accuracy on ICDAR03 dataset by cross validation in the training process.

Figure. 6 shows the recognition accuracies when the percent of selected detectors per class ranges from 0.1 to 1 (i.e., the number of selected detectors varies from 6 to 64). As we can see, at the beginning the recognition accuracies improve with the increasing percent. When only selecting 50% detectors, the recognition accuracy has achieved about 82.1% which outperforms many existing methods. We obtain the best accuracy with the percent of 0.8. However, when the percent exceeds 0.8, the performances exhibit a weak drop which may due to introduced background parts. Thus in consideration of the computational time and the recognition accuracy, in this paper we choose the percent of salient detectors to be 0.8 for each class. The feature dimension is depend on the percent, for example, for English scene character the feature dimension is $0.8 \times 64 \times 62 = 3162$.

We also observe that the performances of our method vary slightly with the change of the percent parameters. When just using 10% detectors (i.e., 6 detectors), we obtain the acceptable performance of 80.3%. This phenomenon demonstrates that our salient part detector selection strategy further suppresses the background noise while retains the salient parts.

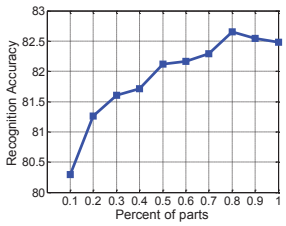

Figure 6. Character recognition results with different percents of detectors per class on ICDAR03 dataset.

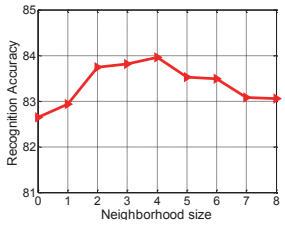

Figure 7. Character recognition results with different search neighborhood sizes on ICDAR03 dataset.

D. Evaluation of Search Neighborhood Size

We evaluate the influence of search neighborhood size for character recognition on ICDAR03 dataset by cross validation in the training process. We set the percent of salient detectors to be 0.8. Figure. 7 shows the recognition accuracies as the neighborhood size ranges from 0 to 8. We observe that with the increasing of neighborhood size, the recognition accuracies first improve. The best recognition accuracy is 84.0% when neighborhood size equals to 4. When the neighborhood size equals to 0, the proposed method achieves 82.7%. Compared with the best accuracy, the low recognition performance of size equaling to 0 lies in that the characters always have translation, rotation and deformation, and thus the learned part positions cannot accurately match with the test samples. The comparison shows that the proposed spatially embedded character representation can boost the recognition performance.

It is worth noting that when the neighborhood sizes are bigger than 4, the performance gradually decreases. This may due to the fact that large neighborhood can introduce noise. The experimental result demonstrates that searching the maximum response within a suitable neighborhood size can further avoid the undesirable effects of interference factors. We recommend the suitable neighborhood size roughly equaling to the half of the side length of convolutional map.

E. Comparison with Other Methods

We first compare the proposed method and the existing methods on English datasets, and the results are reported in Table I. Note that we directly use the suitable parameters (the percent of detectors equals to 0.8 and the neighborhood size equals to 4) of ICDAR03 dataset for Chars74K and IIIT5K dataset. As we can see, the proposed method shows better results than the existing methods on the three character datasets. The comparison shows that our method has stronger parameter adaptation for different datasets.

Table I
CHARACTER RECOGNITION RESULTS OF DIFFERENT METHODS ON ICDAR03, CHARS74K AND IIIT5K DATASETS (%).

Method	ICDAR03	Chars74K	IIIT5K
Multiple Kernel Learning [6]	-	55.3	-
HOG+SVM [8]	77.0	62.0	70.0
Sheshadri and Divvala [19]	70.5	69.7	-
Zhang et al. [7]	79.0	67.0	76.0
Co-HOG [5]	80.5	-	77.8
Coates et al. [20]	81.7	-	-
ConvCoHOG [5]	81.7	-	78.8
TSM [4] (49 Classes)	77.9	-	-
Lee et al. [21]	79.0	64.0	-
Stroke Bank [9]	79.8	65.9	-
SED [22]	82.7	67.5	-
DSEDR [23]	82.6	71.8	-
Liu and Lu [24] (49 Classes)	84.1	81.2	-
CNN_Softmax	81.5	74.4	78.8
Proposed method (62 Classes)	84.0	75.9	80.3
Proposed method (49 Classes)	**87.6**	**85.1**	**87.3**

Specifically, our method significantly outperforms global hand-crafted feature based methods [8], [5], [7] which use global HOG to represent the character image. The reason lies in that HOG captures the gradient information of the whole character image, which does not contain sufficient structure information and global feature tends to bring more noise. Besides, our method achieves superior performance over other part-based methods [4], [9], [21], [23], [22] which also use the hand-crafted feature to represent character. We automatically extract the parts from the images while [4], [9] require human to predefine character parts of each class. Moreover, the existing part-based methods extract part features from the original images, while we regard convolutional descriptors as part feature, which is more discriminative. We can obtain the part and learn feature at the same time. The comparison shows that the effectiveness of the proposed part generation strategy via CNN for character recognition.

Furthermore, the performance of the proposed method exceeds that of fully connected layer features (i.e., the CNN_Softmax) 2.5%, 1.5% and 1.5% on ICDAR03, Chars74K and IIIT5K datasets, respectively. We can conclude that our feature is more powerful than fully connected layer feature.

Note that [4] and [24] both merge the character classes that have similar structures, like 'C' and 'c', 'W' and 'w' into a new class, and finally have 49 classes to recognize. We also use 49 classes to recognize characters and the proposed method significantly outperforms [4] and [24].

IV. CONCLUSION

The proposed method automatically selects the salient part detectors for scene characters via the discriminative

descriptors of parts derived from CNN. Moreover, we embed the spatial region information into the character representation to boost the recognition. The proposed method is effective and computationally efficient. Experiments show that the proposed method achieves the superior performances on three datasets. Finally, there are several future study directions, for example, combining the multi-scale parts to generate a more powerful feature and learning more effective part detectors.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (NSFC) [grant numbers 61601462, 61531019 and 71621002].

REFERENCES

[1] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3538–3545.

[2] N. Lukas and M. Jiri, "Scene text localization and recognition with oriented stroke detection," in *ICCV*. IEEE, 2013, pp. 97–104.

[3] S. Tian, S. Lu, B. Su, and C. L. Tan, "Scene text recognition using co-occurrence of histogram of oriented gradients," in *ICDAR*. IEEE, 2013, pp. 912–916.

[4] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, and Z. Zhang, "Scene text recognition using part-based tree-structured character detection," in *CVPR*. IEEE, 2013, pp. 2961–2968.

[5] S. Tian, U. Bhattacharya, S. Lu, B. Su, Q. Wang, X. Wei, Y. Lu, and C. L. Tan, "Multilingual scene character recognition with co-occurrence of histogram of oriented gradients," *IEEE Transactions on Pattern Recognition*, vol. 51, pp. 125–134, 2016.

[6] Q. Li, T. Lu, P. Shivakumara, U. Pal, and C. L. Tan, "A new method based on bag of filters for character recognition in scene images by learning," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, 2015, pp. 391–395.

[7] Z. Zhang, Y. Xu, and C.-L. Liu, "Natural scene character recognition using robust pca and sparse representation," in *Document Analysis Systems (DAS), 2016 12th IAPR Workshop on*. IEEE, 2016, pp. 340–345.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 1. IEEE, 2005, pp. 886–893.

[9] S. Gao, C. Wang, B. Xiao, C. Shi, and Z. Zhang, "Stroke bank: a high-level representation for scene character recognition," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 2909–2913.

[10] C. Yao, X. Bai, B. Shi, and W. Liu, "Strokelets: A learned multi-scale representation for scene text recognition," in *CVPR*. IEEE, 2014, pp. 4042–4049.

[11] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 3304–3308.

[12] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "Photoocr: Reading text in uncontrolled conditions," in *ICCV*. IEEE, 2013, pp. 785–792.

[13] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *ECCV*. Springer, 2014, pp. 512–528.

[14] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "Icdar 2003 robust reading competitions," in *ICDAR*. IEEE, 2003, p. 682.

[15] T. E. de Campos, B. R. Babu, and M. Varma, "Character recognition in natural images," in *VISAPP*, 2009, pp. 273–280.

[16] A. Mishra, K. Alahari, and C. Jawahar, "Scene text recognition using higher order language priors," in *BMVC*, 2012.

[17] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, and Y. Bengio, "Maxout networks." *ICML (3)*, vol. 28, pp. 1319–1327, 2013.

[18] C. Yi, X. Yang, and Y. Tian, "Feature representations for scene text character recognition: A comparative study," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013, pp. 907–911.

[19] K. Sheshadri and S. K. Divvala, "Exemplar driven character recognition in the wild." in *BMVC*, 2012, pp. 1–10.

[20] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. Ng, "Text detection and character recognition in scene images with unsupervised feature learning," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 440–445.

[21] C.-Y. Lee, A. Bhardwaj, W. Di, V. Jagadeesh, and R. Piramuthu, "Region-based discriminative feature pooling for scene text recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 4050–4057.

[22] S. Gao, C. Wang, B. Xiao, C. Shi, W. Zhou, and Z. Zhang, "Learning co-occurrence strokes for scene character recognition based on spatiality embedded dictionary," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5956–5960.

[23] C.-Z. Shi, S. Gao, M.-T. Liu, C.-Z. Qi, C.-H. Wang, and B.-H. Xiao, "Stroke detector and structure based models for character recognition: A comparative study," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4952–4964, 2015.

[24] X. Liu and T. Lu, "Natural scene character recognition using markov random field," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, 2015, pp. 396–400.