Scene Text Detection with Novel Superpixel Based Character Candidate Extraction

Cong Wang[1,2], Fei Yin[1], Cheng-Lin Liu[1,2]
[1]National Laboratory of Pattern Recognition, Institute of Automation of Chinese Academy of Sciences
95 Zhongguancun East Road, Beijing 100190, P.R. China
[2]University of Chinese Academy of Sciences, Beijing, P.R. China
Email: {cong.wang, fyin, liucl}@nlpr.ia.ac.cn

***Abstract*—Maximally stable extremal region (MSER) is popularly used for candidate character candidate extraction in scene text detection. Its requirement of maximum stability hinders high performance on images of high variability. In this paper, we propose a novel character candidate extraction method based on superpixel segmentation and hierarchical clustering. The proposed superpixel segmentation algorithm for scene text image takes advantage of the color consistency of characters and fuses color and edge information. Based on superpixel segmentation, character candidates are extracted by single-link clustering. To improve the accuracy of non-text candidate filtering, we use a deep convolutional neural networks (DCNN) classifier and double threshold strategy for classification. Experimental results on public datasets demonstrate that the proposed superpixel based method performs better than MSER in character candidate extraction, and the proposed system achieves competitive performance compared to state-of-the-art methods.**

I. Introduction

Text in scene images usually conveys rich and precise high-level semantic information. Detecting scene text is important for numerous potential applications such as scene understanding, image and video retrieval, and automotive assistance. Consequently, many computer vision communities and document analysis communities pay much attention to text detection in scene images. However, scene text detection is still a challenging and unsolved problem. The major challenges stem from the following factors. First, in scene text image, the background can be very complex and there are some region components such as signs, bricks, and grass, which are difficult to distinguish from text. Second, scene text has high variation in character color, font, size, orientation, and languages. Furthermore, the poor quality of scene text image pose hard challenges for the text detection because of uneven lighting, blurring, perspective distortion, low contrast, low resolution and occlusion etc.

A variety of works for text detection have been published in recent years. Existing works can be roughly divided into three categories: sliding-window based methods, connected component based methods and hybrid methods. The sliding-window based methods detect text information by sliding a multi-scale sub-window through all possible location of an image, and design a text/non-text classifier to eliminate noisy windows [1]–[5]. Sliding window based methods can have high recall and detect character with multiple components as a whole. However, the major limitation is high computational cost due to exhaustive search.

On the other hand, connected component based methods first extract character candidates from an scene image, and then refine the candidates to suppress non-text candidates. They separate text and non-text information at pixel-level by running a fast low-level detector. The retained pixels with similar properties are then grouped together to construct possible text candidates. In the category, two representative methods are Stroke Width Transform (SWT) [6] and Extremal Region (ER) [7] based methods. Extending from the original SWT, Huang et. al [8] proposed the Stroke Feature Transform (SFT), which incorporates color cues of text pixels and leads to significantly enhanced performance on inter-component separation and intra-component connection. Maximally Stable Extremal Region (MSER) [9] belongs to ER and is widely adopted to extract character candidates for scene text detection in [10], [11]. Despite the success of MSER based methods in recent years, several open problems still need to be addressed. First, MSER based methods are difficult to obtain the high text detection performance due to its requirement for maximum stability. Second, some text objects in images are not ERs whose pixels have either higher or low intensity than its outer boundary pixels, and cannot be extracted by MSER based methods directly. While the second problem is an intrinsic limitation of ER-based methods, the first problem has been addressed by some researchers. Sun et.al [12] propose generalized Color-Enhanced Contrasting Extremal Region (CER), and He et.al [13] propose Contrast-Enhancement Maximally Stable Extremal Regions (CE-MSERs). In [14], [15], the efficient and effective ER tree pruning methods are proposed.

Furthermore, hybrid methods combine sliding window based methods and connected component based methods [16], [17]. They first detect text regions by sliding window base method and then extract connected components for word or text-line generation.

In this paper, we propose a novel method for scene text detection which belongs to connected component based method. We observe that almost characters in scene text images are color consistency. Taking advantage of the propriety, we propose a novel character candidate extraction method that is based on superpixel segmentation and hierarchical clustering. The proposed superpixel segmentation algorithm for scene text

image fuses color and edge information. And then we extract character candidates based on single-link clustering algorithm. The proposed character candidate extraction method is capable of extracting some characters that do not satisfy the definition of ER and performs better than MSER on public databases.

A powerful text/non-text classifier or component filter is critical for scene text detection. Many methods focus on developing hand-crafted features and adopting efficient classifiers, such as AdaBoost, SVM and Random Forests. Recently, by leveraging the advances of deep learning, some powerful deep models have been developed to filter components in [1], [13]. These deep models all are trained with the samples which are cropped from the input image and warped as the same size. Because of the pre-processing of samples, the loss of contextual information and unwanted geometric distortion are inevitable. Then the detection performance can be compromised.

In this paper, we adopt a deep convolutional neural networks (DCNN) to classify text/non-text components, which is different from the model adopted in [1], [13]. The classifier we adopt applies to input image of arbitrary size but not fixed size. Because of using the contextual information and avoiding unwanted geometric distortion, the adopted classifier is more robust against text-like components, such as bricks, windows or leaves.

The rest of paper is organized as follows. Section II gives an overview of the detection method. Section III describes the proposed character candidate extraction method. Section IV addresses character candidate filtering and Section V addresses text grouping. Section VI presents our experimental results and Section VII offers concluding remarks.

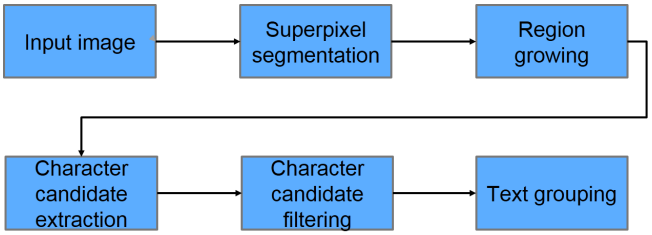

Fig. 1. Flowchart of our scene text detection system.

II. SYSTEM OVERVIEW

The flowchart of our scene text detection system is shown in Fig.1. Firstly, we segment the input image into superpixels and then refine the superpixels through region growing. Character candidates are further extracted based on hierarchical clustering algorithm. Secondly, we use a deep convolutional neural network and double threshold strategy to filter character candidates. In the end, we obtain the final localization result by text grouping. The details of the proposed scene text detection system are described in following sections.

III. CHARACTER CANDIDATE EXTRACTION

A. Superpixel segmentation

First of all, we segment the input scene text image into surpixels. One of the representative and efficient algorithm for superpixel segmentation is SEEDS [18]. SEEDS algorithm starts from an initial superpixel partitioning and continuously refines the superpixels by modifying the boundaries. However, the defined energy function is not suitable to over-segment scene text image. Due to the drawback of SEEDS algorithm, we propose a new superpixel segmentation algorithm for scene text image which is simple to use and understand. The entire algorithm is summarized in Algorithm 1. We initialize the superpixel segmentation by a regular grid. Each pixel in the input image has an initial superpixel label. For each pixel i on the boundary of superpixel map, pixel j is in its eight-neighborhood. We compute the distance between the pixel i and pixel j using the distance metric function we design. If the neighboring pixel which has the minimum distance with the center pixel i is j, we change the superpixel label of pixel i to the superpixel label of pixel j. We repeat the process until no pixel changes its superpixel label or maximum iteration is achieved. Finally, a post-processing step enforces the very small superpixels to be merged into the neighboring superpixel based on minimum color distance. Some examples of the proposed superpixel segmentaton are shown in the first row of Fig.2.

Algorithm 1: Superpixel segmentation algorithm

Input:
Initialize the superpixel segmentation by a regular grid;
Initialize the superpixel label of pixel i as $label(i)$;
Iteration $t = 1$ and Maximum iteration T;
Initialize variable $change = 1$;
Output: the result of superpixel segmentation

while $t \leq T$ *and* $change = 1$ **do**
  $change = 0$
  for *each pixel* i **do**
    if *pixel* i *on the boundary of superpixels* **then**
      Compute the distance $D(i,j), j \in N_i$

$$D(i,j) = \lambda D_c(i,j) + (1-\lambda) D_e(i,j) + \eta D_s(i,j)$$

      Obtain the most similar neighboring pixel k

$$k = \arg\min_j D(i,j)$$

      if $label(i) \neq label(k)$ **then**
        $label(i) = label(k)$
        $change = 1$
      end
    end
  end
  $t = t + 1$
end

A proper distance metric function is the key to the proposed superpixel segmentation algorithm. The distance metric function we design to compute the distance between center pixel and its neighborhood fuses the color and edge information. For pixel i, its corresponding superpixel label is denoted as

Fig. 2. Some examples of surperpixel segmentation and region growing. The first row shows the results of the proposed superpixel segmentation algorithm. The second row shows the results of region growing.

s_i, and pixel j belongs to its neighboring pixels ($j \in N_i$).

Color distance Color information is important low-level feature for describing regions, especially character regions in scene text image. The color distance $D_c(i,j)$ is defined as follows:

$$D_c(i,j) = \|C_i - C_{s_j}\|_2, \tag{1}$$

where C_i is the color of pixel i and C_{s_j} is the mean color of the superpixel s_j that the pixel j belongs to. The color information is measured in RGB color space.

Edge cost Edge information is also important low-level feature for describing regions. When segmenting the image into superpixels, we incline to update the superpixel label of center pixel with the superpixel label of the neighboring pixel with weakest magnitude. The edge cost $D_e(i,j)$ is defined as follows:

$$D_e(i,j) = -minM_k, \text{ if } k \in N_i, s_k \neq s_j, \tag{2}$$

where M_k denote the magnitude of the pixel k.

Combining the color distance with edge cost, the distance function we design for superpixel segmentation $D(i,j)$ is defined as

$$D(i,j) = \lambda D_c(i,j) + (1-\lambda)D_e(i,j) + \eta D_s(i,j), \tag{3}$$

$$D_s(i,j) = 1/((1 + exp(-1/n_{s_j}))), \tag{4}$$

where $D_s(i,j)$ denotes the size of superpixel s_j. D_s can be considered as a regularity term. It encourages the pixel i is merged into the neighboring larger superpixel. In experiment, parameters λ and η are set to 0.8 and 0.05 respectively.

B. Region growing

The input image is over-segmented by superpixel and there are too many superpixels for background. We make region growing to reduce the number of superpixels. The standard flood-fill algorithm is adopted to grow regions. Some examples of region growing are shown in the second row of Fig.2. The distance function $d_g(s_i, s_j)$ between neighbouring superpixel pair (s_i, s_j) is defined as follows:

$$d_g(s_i,s_j) = \rho_1 d_c(s_i,s_j) + \rho_2 d_e(s_i,s_j), \tag{5}$$

where $d_c(s_i,s_j)$ denotes the Euclidean distance of mean color between superpixel pair (s_i,s_j) in LUV color space and $d_e(s_i,s_j)$ denotes the mean magnitude within the common border of the neighboring superpixel pair (s_i,s_j). In addition, the threshold to terminate region growing is set to 0.01 experientially. Note that in order to obtain a good segmentation result for low contrast scene image, the threshold is set strictly.

C. Hierarchical clustering

It is possible that some characters in a scene text image are still segmented into multiple parts after region growing. Therefore, we adopt a greedy merging process to obtain character candidates as complete as possible based on single-link clustering algorithm. Besides collecting the superpixel regions after region growing as character candidates, we further collect new character candidates at each merging step.

We first compute the distance for each pair of adjacent superpixels. This distance function $d_h(s_i,s_j)$ of a superpixel pair (s_i,s_j) is defined as

$$d_h(s_i,s_j) = d_c(s_i,s_j), \tag{6}$$

where $d_c(s_i,s_j)$ denotes the Euclidean distance of mean color between superpixel pair (s_i,s_j) in LUV color space.

Once the distance of all superpixel pairs are computed, the hierarchical clustering algorithm merges the most similar superpixel pair into a new larger superpixel. Following the merging of superpixels, the distance scores are updated accordingly. The threshold to terminate hierarchical clustering is set to 0.06 experientially.

After hierarchical clustering, many character candidates can be obtained. We intuitively remove candidates that are too large or too small in size to reduce the number of non-character candidates.

IV. CHARACTER CANDIDATE FILTERING

For all character candidates, we need to retain text and filter non-text as possible. To improve the accuracy of non-text candidate filtering, we use a deep convolutional neural networks (DCNN) classifier and double threshold strategy for text/non-text classification.

A. DCNN for text/non-text classification

The architecture of DCNN we adopt is shown in Fig.3, which is inspired by Fast R-CNN [19]. However, there are still two differences between them.

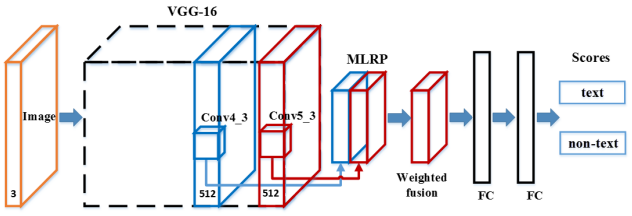

Fig. 3. Architecture of DCNN for text/non-text classification.

1) Loss: Generally, object proposals Fast R-CNN classifies are provided by selective search algorithm. the true object proposals may obviously deviate from ground truth sometimes. And then it is benificial and neccessiray to adopt multi-task loss, including classification loss and bounding box regression loss. However, because the true character candidates extracted by the proposed method in this paper generally have a high IoU overlap with charcter-level ground truth, we find that bounding box regression task is not neccessiray and we only use the classification task with softmax loss.

2) Multi-level ROI Pooling (MLRP): Fast R-CNN simply applies ROI pooling over the last convolutional layer (conv5_3) in the VGG16 model. However, due to high variation of character in size, it is not enough to only apply ROI pooling over conv5_3. We adopt MLRP over conv4_3 and conv5_3 feature map of VGG16 network and obtain two $512 \times H \times W$ ROI pooling layers (both H and W are set to 7 in practice). And then we concatenate the two ROI pooling layers and weighted fusion them with $512 \times 1 \times 1$ convolutional layer.

In the training phase of DCNN, the layers that share with VGG16 model are initialized by a pre-trained VGG16 model for ImageNet classification [20], and all the weights of the new layers are initialized with a zero mean and a standard deviation of 0.01 Gaussian distribution. The base learning rate is 0.001 and is divided by 10 for each 20K mini-batch until convergence. A momentum and weight decay we use are set to 0.9 and 0.0005 respectively. In addition, in order to boost the detection accuracy, we use multiple rescaled versions of input image in test phase. The length of short side of input image are rescaled into 600, 800 and 1000 respectively.

B. Double threshold strategy

Compared with the evaluation criterion of general object detection, the evaluation criterion of text detection is more strict, which needs higher IoU overlap between detected bounding box and ground truth. Therefore, we adopt double threshold strategy similarly to [15]. In the experiment, the high confidence threshold is set as 0.96 to increase the precision of text detection, and the low confidence threshold is set as 0.01 to increase the recall of text detection. Based on the confidence scores the above-described text/non-text classifier outputs, all candidates are classified into three classes: strong character, weak character and non-character. The weak characters can be true character or non-character. We start from each strong character R_s and track its neighboring character candidate classified as weak character R_w. When each R_w satisfies the similar text properties against R_s, we change the status of R_w to R_s and investigate its neighborhood recursively. The bounding box of R_s and R_w are denoted as $[xmin_s, ymin_s, xmax_s, ymax_s]$ and $[xmin_w, ymin_w, xmax_w, ymax_w]$ respectively. The height of R_s and R_w are denoted as h_{R_s} and h_{R_w} respectively. And the mean color of R_s and R_w are denoted as C_{R_s} and C_{R_w} respectively. The text properties we use are as follows:

The height ratio

$$min(h_{R_s}, h_{R_w})/max(h_{R_s}, h_{R_w}) < 0.6, \quad (7)$$

The horizontal interval

$$\frac{max(xmin_s, xmin_w) - min(xmax_s, xmax_w)}{min(h_{R_s}, h_{R_w})} < 3.0, \quad (8)$$

The vertical overlap

$$\frac{min(ymax_s, ymax_w) - max(ymin_s, ymin_w)}{max(h_{R_s}, h_{R_w})} > 0.4, \quad (9)$$

Color difference in Luv color space

$$\|C_{R_s} - C_{R_w}\|_2 < 0.03. \quad (10)$$

C. Post-processing

For all surviving character candidates after classification, we adopt non-maximum suppression (NMS) with an IoU overlap threshold of 0.7 to suppress low-scoring character candidates. In addition, even after NMS processing, multiple inner or outer candidates may still exist for one character instance. To address this problem, we further filter the inner bounding boxes of each character candidate according to coordinate position.

V. TEXT GROUPING

After text/non-text classification by the deep convolutional neural network and double threshold strategy, a large number of non-text components have been eliminated and only a small number of text components are retained to construct text-lines. We can group two neighboring text components into a pair if they have similar geometric properties described in equation (7), (8), (9). Then, the pairs are merged sequentially if they include a same text component. A text-line is constructed until no pairs can be merged further.

VI. EXPERIMENTAL RESULTS

In this section, we first briefly introduce two public datasets and some details about experimental setting. Then, we quantitatively evaluate the performance of the proposed character candidate extraction method, adopted text/non-text classifier and final text detection.

A. Datasets and Experimental Setting

1) Datasets: Our proposed scene text detection system is evaluated on two widely used scene text datasets: ICDAR 2011 and ICDAR 2013. The ICDAR 2011 dataset includes 229 training images and 255 test images, and the ICDAR 2013 dataset includes 229 and 233 images for training and test, respectively.

Fig. 4. Some examples of character touching on ICDAR 2011. The first row shows the character-level recall results under one-to-one matches. And the second row shows the actual character-level recall results under one-to-one and many-to-one matches.

TABLE I
CHARACTER-LEVEL RECALL RATE OF CHARACTER CANDIDATE EXTRACTION ON ICDAR 2011 TEST SET

Method	No. of candidates	Recall(%)
All ERs	1,729,833	89.6
MSERs	39,762	53.9
Sung et al. [14] (Gray)	75,124	86.0
Sung et al. [14] (Gray+Cr+Cb)	93,357	87.7
Cho et al. [15]	629,932	95.1
SCCs (only one-to-one matches)	423,441	87.9
SCCs (one-to-one and many-to-one)	423,441	95.6

2) Experimental setting: When training text/non-text classifier, our all training images are generated from the training set of ICDAR 2013. Then to increase the number of training samples and their diversities, we further make data augmentation by flipping these training images and obtain 458 training images in total. In addition, we consider a character candidate which has an IoU overlap greater than 0.7 with a character-level ground truth as a positive candidate, and a character candidate which has an IoU overlap lower than 0.3 with any character-level ground truth as a negative candidate. Note that we ignore the candidates which have IoU overlap with a character-level ground truth in a range $[0.3, 0.7]$.

B. Evaluation of Character Candidate Extraction

A character candidate which has an IoU overlap greater than 0.5 with a character-level ground truth is considered as a correct candidate in [15]. The criterion is one-to-one matches. The proposed superpixel based character candidates extraction method (referred as SCCs) is clustering-based method and it can cluster multiple touching or very close characters in a word into single superpixel. Therefore, the recall rate of SCCs is decreased only under one-to-one matches. In order to obtain the actual character-level recall rate of SCCs, we also take into account many-to-one matches. If one touching character candidate D_j in a word-level ground truth G_k matches a set S_m which consists of some character-level ground truth G_k^i under many-to-one matches, the following three conditions are satisfied at the same time:

$$\begin{aligned} &Area(G_k \cap D_j)/Area(D_j) > 0.8, \\ &Area(G_k^i \cap D_j)/Area(G_k^i) > 0.8, \forall i \in S_m, \\ &Area(B_{S_m} \cap D_j)/Area(B_{S_m} \cup D_j) > 0.5, \end{aligned} \tag{11}$$

where G_k^i denotes the ith character-level ground truth in a word-level ground truth G_k and B_{S_m} denotes the bounding box of the set S_m.

Table I shows the evaluation result of character-level recall rate between the proposed SCCs and ER based methods on ICDAR 2011. The recall rate of SCCs outperforms MSER and [14]. In addition, SCCs achieves a comparable recall rate and has fewer candidates compared with [15] when we adopt both one-to-one and many-to-one matches. Some examples of character touching are given in Fig.4. We implement SCCs using Matlab and Mex programming. Our testing environment is a PC running MS Windows 7 64bit version with Intel Core i7 CPU of 3.4GHz. On ICDAR 2011 test set, the average processing time of superpixel segmentation, region growing and hierarchical clustering are 0.578 seconds, 0.197 seconds and 0.023 seconds respectively.

C. Evaluation of Text/Non-text Classifier

Evaluation of text/non-text classifier on ICDAR 2011 test set is shown in Table II. The deep convolutional neural network model we adopt (referred as Text-RCNN) increase the character-level recall rate compared with the previous CNN models [1] [13]. In the experiment, the confidence threshold is set as 0.9. Note that we do not adopt double threshold strategy for fair comparison.

TABLE II
EVALUATION OF TEXT/NON-TEXT CLASSIFIER ON ICDAR 2011 TEST SET

Method		Recall	Precision	F-measure
CE-MSERs [13]	CNN [1]	0.71	0.85	0.77
	Text-CNN [13]	0.74	0.91	0.82
SCCs	Text-RCNN	0.78	0.87	0.82

D. Evaluation of Detection Performance

We evaluate the detection result of the proposed scene text detection system on ICDAR 2013 dataset. We follow the standard evaluation protocol of ICDAR 2013 by using the DetEval tool offerd by the authors of [21] . The performance of the proposed system in terms of Recall, Precision and F-measure is compared in Table III. As the table shows, experimental result of our method achieves comparable performance compared with other representative methods, especially connected component based methods. Some examples of detection results on ICDAR 2013 test set are given in Fig.5.

VII. Conclusion

In this paper, we propose a novel character candidate extraction method based on superpixel segmentation and hierarchical clustering. The proposed superpixel segmentation algorithm

Fig. 5. Some examples of detection results on ICDAR 2013.

TABLE III
TEXT DETECTION RESULT ON ICDAR 2013 TEST SET

Method	Recall(%)	Precision(%)	F-measure(%)
Yin et al. [10]	69.28	88.80	77.83
Zhang et al. [2]	74	88	80
Tian et al. [5]	75.89	85.15	80.25
Sung et al. [14]	74.23	88.65	80.80
Cho et al. [15]	78.45	86.26	82.17
He et al. [13]	76.29	**92.69**	83.69
Our method	**81.86**	87.43	**84.56**

for scene text image takes advantage of the color consistency of characters and fuses color and edge information. Based on superpixel segmentation, character candidates are extracted by single-link clustering. To improve the accuracy of non-text candidate filtering, we use a deep convolutional neural network classifier and double threshold strategy for classification. Experimental results on public datasets demonstrate that the proposed superpixel based character candidate extraction method performs better than MSER, and the proposed system achieves competitive performance compared to state-of-the-art methods. Our future work aims at non-Latin (like Chinese) text detection, where many characters consist of multiple connected characters, and need special care for candidate character extraction and filtering.

ACKNOWLEDGMENT

This work has been supported by the National Natural Science Foundation of China (NSFC) Grants 61411136002 and 61633021.

REFERENCES

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

[2] Z. Zhang, W. Shen, C. Yao, and X. Bai, "Symmetry-based text line detection in natural scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2558–2567.

[3] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2004, pp. 366–373.

[4] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *International Journal of Computer Vision*, vol. 116, no. 1, pp. 1–20, 2016.

[5] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, and C. Lim Tan, "Text flow: A unified text detection system in natural scene images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4651–4659.

[6] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2963–2970.

[7] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.

[8] W. Huang, Z. Lin, J. Yang, and J. Wang, "Text localization in natural images using stroke feature transform and text covariance descriptors," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1241–1248.

[9] D. Nistér and H. Stewénius, "Linear time maximally stable extremal regions," in *European Conference on Computer Vision*, 2008, pp. 183–196.

[10] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 970–983, 2014.

[11] W. Huang, Y. Qiao, and X. Tang, "Robust scene text detection with convolution neural network induced mser trees," in *European Conference on Computer Vision*, 2014, pp. 497–511.

[12] L. Sun, Q. Huo, W. Jia, and K. Chen, "Robust text detection in natural scene images by generalized color-enhanced contrasting extremal region and neural networks," in *International Conference on Pattern Recognition*, 2014, pp. 2715–2720.

[13] T. He, W. Huang, Y. Qiao, and J. Yao, "Text-attentional convolutional neural network for scene text detection," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2529–2541, 2016.

[14] M.-C. Sung, B. Jun, H. Cho, and D. Kim, "Scene text detection with robust character candidate extraction method," in *International Conference on Document Analysis and Recognition*, 2015, pp. 426–430.

[15] H. Cho, M. Sung, and B. Jun, "Canny text detector: Fast and robust scene text localization algorithm," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3566–3573.

[16] Y.-F. Pan, X. Hou, and C.-L. Liu, "A hybrid approach to detect and localize texts in natural scene images," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 800–813, 2011.

[17] S. Zhu and R. Zanibbi, "A text detection system for natural scenes with convolutional feature learning and cascaded classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 625–632.

[18] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool, "Seeds: Superpixels extracted via energy-driven sampling," in *European Conference on Computer Vision*, 2012, pp. 13–26.

[19] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[21] C. Wolf and J.-M. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," *International Journal of Document Analysis and Recognition*, vol. 8, no. 4, pp. 280–296, 2006.