Learning Text Component Features via Convolutional Neural Networks for Scene Text Detection

Wafa Khlif*†, Nibal Nayef, Jean-Christophe Burie, Jean-Marc Ogier
*L3I Lab, University of La Rochelle
La Rochelle, France
Email:{wafa.khlif, nnayef, jcburie, jmogier}@univ-lr.fr

Adel Alimi
†REGIM Lab, ENIS
University of Sfax, Tunisia
Email:{wafa.khlif, adel.alimi}.regim@usf.tn

***Abstract*—Reading the text embedded in natural scene images is essential to many applications. In this paper, we propose a method for detecting text in scene images based on multi-level connected component (CC) analysis and learning text component features via convolutional neural networks (CNN), followed by a graph-based grouping of overlapping text boxes. The multi-level CC analysis allows the extraction of redundant text and non-text components at multiple binarization levels to minimize the loss of any potential text candidates. The features of the resulting raw text/non-text components of different granularity levels are learned via a CNN. Those two modules eliminate the need for complex ad-hoc preprocessing steps for finding initial candidates, and the need for hand-designed features to classify such candidates into text or non-text. The components classified as text at different granularity levels, are grouped in a graph based on the overlap of their extended bounding boxes, then, the connected graph components are retained. This eliminates redundant text components and forms words or textlines. When evaluated on the "Robust Reading Competition" dataset for natural scene images, our method achieved better detection results compared to state-of-the-art methods. In addition to its efficacy, our method can be easily adapted to detect multi-oriented or multi-lingual text as it operates at low level initial components, and it does not require such components to be characters.**

***Keywords*—Scene text detection; CNN; multi-level binarization; multi-level connected components; graph-based grouping**

I. Introduction

Text appears everywhere in our natural surrounding environments such as in traffic signs, license plates, advertisement billboards, business cards, building signs, labels on posted parcels and on name plates. The textual content in these images is a valuable source of information and useful for many applications such as interactive tourists' guidance, data mining and providing text accessibility for visually impaired people whether reading such text is a necessity for their everyday life or simply for navigating or enjoying the world around them.

Although it bears similarity to OCR problems in traditional document images, text detection in scene images is much more challenging due to, on one hand, complex layout with variable backgrounds and the high variations in text color, font, size and orientation, and on the other hand, lighting/shadow/occlusion problems introduced by acquisition conditions. New challenges also emerge in scene images of modern cities such as detecting multi-lingual text.

Most text detection systems are mainly composed of three stages. Firstly, finding character/word candidates or regions of interest. This could be done at pixel, interest point or zone levels. Usually, this stage is the most challenging and involves many complicated preprocessing steps. Secondly, the filtering stage(s) where initial candidates are classified as text or non-text components. Some methods use hand-designed features and multiple filtering steps within this stage. Finally, the grouping stage, in which text components are grouped into characters, words or textlines. Grouping methods are typically not adapted to multi-oriented or multi-lingual text.

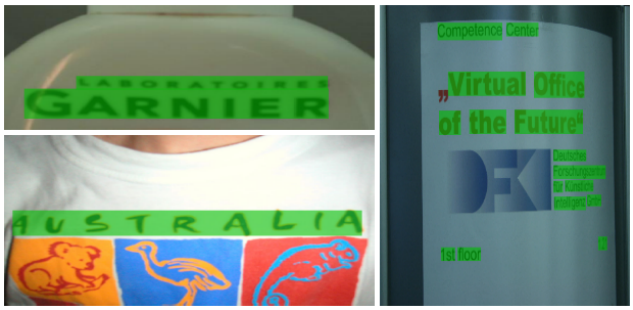

Fig. 1. Examples of successful text detection results by the proposed method. Correctly detected text zones are shown in green bounding boxes.

In order to overcome the mentioned challenges in the three text detection stages, we propose a novel method for text detection in natural scene images. In the first stage, finding initial candidates is performed by multi-level connected component extraction. This module handles complex background and variations in text scale/color by multiple binarizations. The module extracts redundant components of text/non-text at different granularity (text components could be parts of characters, characters, parts of words etc. and could be found multiple times). Our technique at this stage minimizes both the preprocessing steps, and the possibility of loosing potential text components before the next stages.

The features of the extracted raw components are then learned via a CNN in the second stage. The CNN is trained as a binary classifier to discriminate text from non-text components. For the third stage, we propose a general grouping method which could be easily adapted to multi-oriented text. The classified text components are first aggregated to form meaningful higher level components via linkage-based clustering (for example, broken parts of the same character would be grouped in the same cluster). Then, a graph is formed based on overlapping criteria of the components bounding boxes.

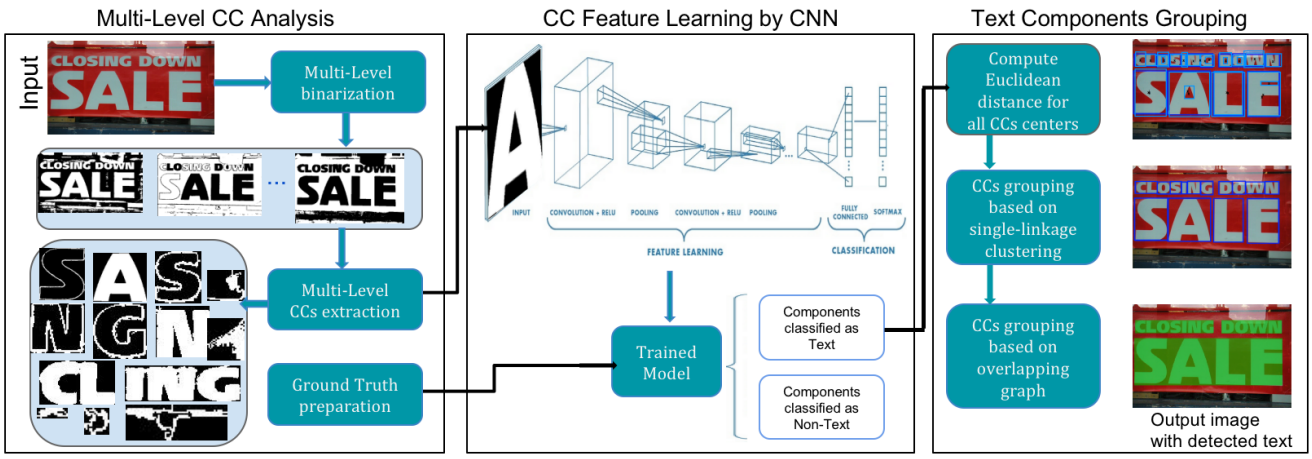

Fig. 2. Block diagram of the proposed method with its three modules. First, Multi-level Connected Component (CC) Analysis based on multi-level binarization. Second, Feature Learning with the Convolution Neural Network from the raw multi-level CCs. Third, grouping of text components based on linkage clustering and overlapping graph.

The connected graph components form text words. These grouping steps do not pose assumptions related to the text script/language.

Figure 1 shows example results of our proposed method, all the text within the image has been accurately detected. Such precise localization of the detected text allows the subsequent recognition steps to be applied successfully. The remaining sections of this paper are organized as follows: in Section II, we review text detection methods which use techniques similar to our work. Section III discusses the details of our proposed method. The experimental evaluation and analysis of results are presented in Section IV.

II. Related Work

Existing scene text detection methods can be categorized into two approaches: the traditional/classical approach based on hand-crafted features and the deep learning approach where features of text components are learned automatically. Both rely on a first stage of finding initial text candidates or regions of interest, as opposed to the holistic approach via fully convolutional networks. An extensive review of state-of-the-art in scene text detection can be found in the survey of Ye and Doermann [1].

In the first two approaches, identifying initial candidates is done using sliding windows [2], [3], connected components [4], [5] or MSER interest points [6], [7]. We review here the methods that use similar techniques to our method, in particular, connected components for finding initial candidates, and deep learning via CNN for text classification.

Epshtein et al. [5] extracted connected components from the image based on Stroke Width Transform (SWT). The CC extraction process is applied on both the image and its complement. While Huang et al. [8] used the color information in addition to the SWT. Liu et al. [9] applied a multi-scale adaptive local thresholding operator to generate two complementary binary images. They extracted connected components from both images. For the classification step, they all used a set of rules based on hand-crafted features.

Huang et al. [10] used both sliding windows and MSER for identifying regions of interest. MSER is used to reduce the number of scanned windows. Then, they applied CNN to classify the extracted regions. In another work of Wang et al. [3], sliding windows are run over high resolution input images to obtain a set of candidate textlines. The candidates are then classified to text and non-text using a deep CNN.

Zhang et al. [11] extracted character candidates using MSER. For classification, a CNN is used as a discriminative codebook to compute a bank of responses for each candidate. Then, an SVM classifier is used to decide the final prediction as text or non-text image region.

Zhu et al. [12] used convolutional k-means where simple k-means clustering is used to learn feature banks. Then, they used confidence-rated AdaBoost to classify patches as foreground (text) and background (non-text). A final step of CC extraction from characters candidates using color and edge features is applied to improve the word segmentation and change the output to word level.

Our method proposes a multi-level connected component extraction technique to prevent missing any possible candidate text components. Discriminative features are learned from the raw components via a CNN. A graph-based grouping technique is proposed to aggregate the classified text components into words. Our method does not make assumptions about the text orientation or script, hence, it could be generalized to detect multi-lingual and/or multi-oriented text.

III. The Proposed Method

Our proposed method with its 3-fold contributions aims at creating a segmentation-free and accurate scene text detection system. At each of the three stages of such a system, we propose a novel technique that overcomes the challenges faced by state-of-the-art methods. Figure 2 shows the architecture of our proposed method. The first module handles multi-level connected component extraction where a scene image is fed to this module as input. Multiple binarizations are applied on the input image and its complement before extracting the connected components from each binary image. This ensures

the extraction of low-contrast, light-on-dark and low resolution components. The resulting text and non-text components could be broken and/or extracted multiple times.

The second module is a classification module composed of a CNN that learns powerful features of the raw text and non-text components in the training phase. In the test phase, the trained CNN model classifies the components extracted in module 1 into text or non-text. The third module aims at creating the final output as text words from the components classified as text in module 2. We propose a grouping method in this module that starts by linkage-based clustering to group broken and/or redundant components of the same character/group of characters into the same cluster. Then, text words are formed by finding the connected components of a graph whose edges represent the amount of overlap among the bounding boxes of text components. The details of each module of the method are described in the following subsections.

A. Multi-Level Connected Components Analysis

Working at the connected component (CC) level to find initial text candidates is preferred to the pixel-level or interest point level – which are noise-sensitive and slow –, and to the sliding window level which cannot be easily adapted to multiple scales and orientations among other problems. However, CC extraction relies on the lossy binarization step, and may result in broken text components. We propose a multi-level CC extraction that overcomes those challenges.

Scene images may contain light text on dark background, linked or broken characters due to low resolution or low contrast text due to the complex background or lighting problems. A single binarization step whether adaptive or global cannot separate the foreground components from the background. We employ multi-level binarization to increase the probability of finding all possible text candidates. At this step, we do not filter out any binarized components. Figure 3 shows three different binarized versions of two example images. Each binarization reveals different components of the image. Combining the components from all the binary images yields redundant components, but it would minimize the number of lost components. In our multi-level binarization, we perform three different types of binarization one or more times with different thresholds. Each binarization is considered as a filter, and the choice of the binarization techniques and the number of filters is performed experimentally. The chosen combination of the binarized images allows us to retrieve the totality of the regions of interest in the image.

The first binarization method is inspired by Chen et al. [13], and is based on the idea that text strokes in the image have complete contours. As pixels on the contours have higher contrast than adjacent pixels, we compute the gradient magnitude for each pixel in each of the RGB channels to compute the local contrast, and generate an image with the largest value of the gradient magnitude. The image is then segmented into two parts: smooth regions (pixels with low contrast) and non-smooth regions (pixels with high contrast). Non-smooth regions are all considered as text regions. The smooth regions which fill the non-smooth regions are also extracted. A final binarized image is generated by merging the extracted smooth regions with the non-smooth image. This binarization has the ability to find low resolution components.

The second binarization is based on local adaptive thresholding. From the histogram of the image we select an individual thresholding for each pixel based on the range of intensity values in its local neighboring pixels. This operation is repeated until it converges. The threshold is then found by separating the histogram of intensity values into two classes. The third binarization is computed from the HSV space where the second binarization technique is applied on the complement of the Hue channel. These two binarizations find low contrast components and light-on-dark components.

Fig. 3. Multi-level binarization results of two test images. From left-to-right images are shown followed by their binarization results for: smooth/non-smooth binarization, adaptive thresholding binarization on the original image and on the complement of Hue channel. Note that some text components appear in only one of the binary image.

After computing the multiple binarizations, CCs are extracted from all the binarized images. The extracted text and non-text components could be broken and/or extracted multiple times. This increases the probability of finding all text components, hence minimizing the loss of any possible text candidates at this early stage of the text detection system. The redundancy in the extracted components is dealt with in the next modules.

Figures 4 and 5 show examples of the CCs extracted from the multi-level binarized images. Those collected candidates are of different fonts, sizes and different backgrounds. They can be parts of characters, multiple attached characters or random non-text components (as in Figure 4) or exact text characters (as in Figure 5).

The overall process of multi-level CC extraction is not based on any assumption about the orientation of the text or its script. The extracted components are not assumed to be of any specific shape or size or to be connected as characters. This allows our method – up to this stage – to be easily adaptable to detect multi-oriented and/or multi-lingual text.

Fig. 4. Samples of extracted connected components from different binarizations of different images. The components are of different sizes, orientations and shapes, and of variable type: letters, group of letters and varying non-text components.

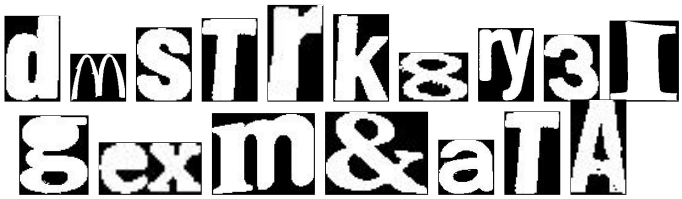

Fig. 5. Samples of extracted connected components from different binarizations. Those are the most frequent type of text components.

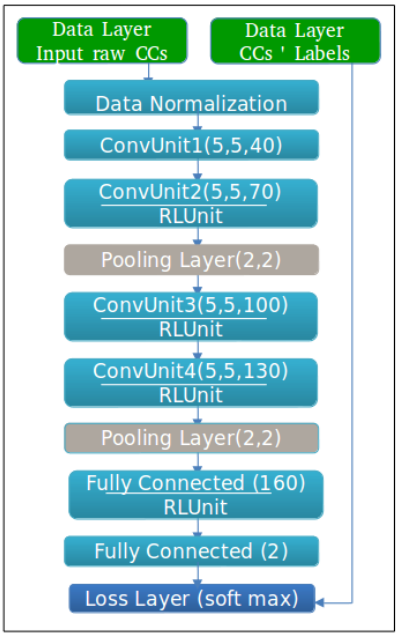

Fig. 6. Structure of the CNN classifier network. The ConvUnit(w,h,n) represents a convolution layer of n features with wxh kernel size, connected to a ReLUnit layer and pooling layer with kernels of size 2x2. Followed by two fully connected layers of 160 and 2 outputs respectively.

B. Learning Text Component Features via a CNN Classifier

The previous multi-level CC extraction module, results in thousands of raw text and non-text components with variable content and size characteristics. The enclosing boxes of these components are fed as separate input images to this classification module. To compute the likelihood of a component being text or not, the deep features of the components are learned using a CNN classifier.

The CNN architecture is shown in Figure 6. Our network is composed of a data layer, 4 convolutional layers, 2 pooling layers, 2 fully connected layers and a loss layer. As followed in best practices of building deep CNN classifiers, our convolutional layers are each followed by a rectified linear unit layer (except the first one), and every two consecutive convolution layers are followed by a pooling layer to reduce the size of the features dimension. The two fully connected layers generate the final 1-D feature vector for our binary classification problem. The standard *softmax* function is used in our loss layer. All input component images are normalized in the data layer. As our input candidate components are relatively small, we opted to use small kernels. The final fully connected layer has only two output connections because we have two classes: text and non-text.

Feature learning in CNN goes through two phases. In the training phase, we feed the network with the training data and their corresponding ground truth (the generation of ground truth is explained below). Training data corresponds to all the bounding boxes of the extracted multi-level connected components represented as separate images.

Through the training process, different feature maps are generated from text or the non-text component images at the different network layers. The trained model is hence built to be used in the test phase for classifying text versus non-text components. In the test phase, the same steps of extracting the multi-level connected components are followed to generate input test component images. Each image is classified as a text or a non-text component.

Our training/test samples are images of connected components. However, in the datasets of scene text detection, the ground truth is at the level of words or text-lines. Hence, we prepare a ground truth at the connected component level as follows. Based on the word bounding box in the ground truth of the dataset, we label a connected component as text if it overlaps with text bounding box in the ground truth with a ratio higher than 0.8, otherwise, it is labeled as a non-text components. This high overlap ratio yields accurate ground truth labeling of our candidate connected components.

C. Graph-based Grouping of Text Components

For the third module of our method, we propose a general grouping method which takes as input all the components labeled as text in a test image by the classifier in the previous module. The grouping method consists of two main steps. First, linkage-based clustering which aggregates the redundant and broken text components of the same character(s) into the same clusters. In the second step, a graph is formed based on overlapping criteria of the components bounding boxes of each cluster. The connected graph components form text words.

In the first step, a euclidean distance matrix is computed between the centers of each two text components. Then, a dendrogram is created by a single-linkage hierarchical clustering. The text component clusters are built in a bottom-up fashion, where at each step, a pair of text components are grouped into the same cluster if they are closest to each other according to the distance matrix. Clusters are formed by merging smaller clusters, and this pair-wise merging process is repeated until no pairs can be further merged. This grouping

step forces broken components of the same character(s) to be grouped in one cluster, as well as the redundant versions of the same text component. A bounding box is created for the text components within each cluster. These boxes represent the input text candidates for the next grouping step.

The second step builds a graph where the bounding boxes (text candidates) are the nodes. To create the edges, each two boxes are processed at a time as follows. The overlap between the extended bounding boxes of each two text candidates is computed. Two candidates (nodes) are linked by an edge if their overlap is higher than a threshold that is adaptive to the scale of the boxes. The adjacent nodes in this graph represent parts of the same word in the cases of successful grouping. Finally, the connected components of this graph are extracted as the detected text words. Figure 7 shows the described grouping steps applied on an image.

Fig. 7. Left: original image with all the text components. Middle: output of the first grouping step: the resulting boxes are mostly letters (or few merged letters merged). Right: final grouping output: text grouped at word level.

The advantages of our grouping method could be shown through its ability to find very small text components which may be lost in the preceding modules. For example, the dots or the small letters (or parts of letters) would be included in the final word box in the grouping module. By extending the size of the bounding box of a connected component with respect to its original size, the small components will be recovered if it has neighboring text component.

Our grouping method has also advantages over rule-based methods which require many hand-tuned parameters. Our method is based on hierarchical clustering of text components centers and it is adaptive to scale of text components when computing the overlapping graph. Moreover, the described grouping steps do not pose assumptions related to the text script or orientation.

This module concludes our proposed text detection method. Figure 8 shows examples of successful detection results of our method on the ICDAR2013 RRC dataset scene images [14].

IV. Experimental Evaluation

We have implemented our proposed methods in a text detection system. The system is evaluated on the standard public dataset of the ICDAR2013 Robust Reading Competition Challenge2: Focused Scene Text [14]. This dataset contains 462 images in total, split as 229 training images containing 848 words and 233 test images. We used the same split of the train and test used in the competition setting. As for the evaluation of the system, we use the standard recall, precision and f-measure metrics proposed in the RRC competition [14] and used in most scene text detection works. A detected bounding box is considered as a match if it overlaps a ground truth bounding box by more than 50%. The experiments and results discussion are detailed in the following subsections.

Fig. 8. Examples of successful text detection results on 6 test images of the RRC dataset [14]. The detected text is shown in green bounding boxes. The detected regions are mostly precise and cover a word or a textline.

A. Implementation Details

To learn the features from the multi-level raw connected components extracted from the images, we used the CNN network in Figure 6 which shows the size of feature maps and kernels of the different layers. It is trained by stochastic gradient descent with back-propagation and a maximum number of iterations of 10^4. After the multi-level CCs extraction, we resize the input CC images to 96x96 pixels. The CNN solver parameters are as follows. Weight decay is $5x10^{-4}$ and momentum is 0.9. The learning rate policy is *fixed* and the base learning rate is 10^{-3}. The experiments have been conducted using Caffe.

B. Results and Analysis

The number of inputs to the CNN network – the second module of our method – depends on the number of filters used for binarization in the multi-level CC extraction module. These inputs are the text and non-text extracted component images. In a first experiment, we would like to show the effectiveness of the multi-level CC extraction in finding candidate text components. Table I shows the number of the extracted text and non-text components of the ICDAR2013 dataset, while varying the number of binarization filters. The table also shows the effect of multi-level CC extraction on the text classification accuracy computed by the trained CNN network.

TABLE I. The extracted text and non-text connected components of the RRC dataset [14] using multiple binarizations, and the resulting CNN classification accuracy.

Number of filters	Type	Text	Non-Text	Total	Accuracy
One binarization	Train	4920	4160	9080	-
	Test	4359	8100	12459	78.29%
Two binarizations	Train	5980	4553	10533	-
	Test	4465	8600	13065	87.97%
Three binarizations	Train	6221	8519	14740	-
	Test	5820	12289	18109	**96.81%**

Table I shows that using multiple binarizations significantly improves text classification accuracy. The shown numbers of extracted components are from both the training and the test set, while the classification accuracy is shown for the test set. The third experiment using three binarizations presents the best

results as we get the majority of the possible candidates in the image. The problem of redundant components does not affect the ability of the CNN network to learn, and the resulting redundant text components are dealt with in the grouping step. Our method aims to retain all possible text components, and it is able to successfully classify parts-of-character components.

Secondly, we show our text detection results compared to state-of-the-art text detection methods including some recently published in 2017. Table II shows the text detection results of our method applied on the RRC dataset[14]. Our method outperforms state-of-the-art method by an F-score of 85.94%. Figures 1 and 8 show qualitative results of our method. The detected regions are in most of the cases precise and cover a word or a textline if the space between the words of the textline is very close to the spaces between characters in a word in the same line.

TABLE II. TEXT DETECTION RESULTS OF THE PROPOSED METHOD COMPARED TO STATE-OF-THE-ART METHODS ON THE ICDAR2013 RRC DATASET [14]

Method	Recall(%)	Precision(%)	F-measure(%)
Our method	82.28	89.94	**85.94%**
Zhu & Uchida.[15]	84.00	83.00	84.00%
Zhang et al.[16]	88.00	78.00	83.00%
He et al. [17]	73.00	93.00	82.00%
Faster R-CNN.[18]	75.00	86.00	80.00%
R-FCN [19]	76.00	90.00	83.00%

However, our method may fail in some cases as shown in Figure 9. For example, some logos are mis-classified as text. This is due to the similarity between some logos and text characters. In other cases, parts of a word are missed due to bad lighting conditions or very low resolution.

Fig. 9. Examples of failure cases including: strong highlights, transparent text and very small text. Red boxes show missed text and green boxes show correctly detected text.

V. CONCLUSIONS AND FUTURE DIRECTIONS

This paper has presented a novel system for scene text detection that combines multi-level connected component extraction with CNN-based feature learning to classify text and non-text components, followed by graph-based grouping of text components.

The multi-level CC extraction works on low level text components without a specific orientation, and minimizes the loss of any possible text candidates. This allows the scene text detection system to be easily adapted to multi-lingual and/or multi-oriented text detection. The CNN classification network learns discriminative features of the extracted text and not-text component images. This strong classifier is able to handle text components of different fonts, sizes and orientations, in addition to non-text components of different shapes. The graph-based grouping handles both the redundancy and the broken parts of text components.

Overall, these 3-fold contributions have led to a powerful scene text detection system that performs better than state-of-the-art systems. For future work we would like to extend our system to detecting multi-lingual text, and to investigate the technique of bounding box regression as a grouping method to handle arbitrarily oriented text.

VI. ACKNOWLEDGEMENT

The research leading to these results is funded by the Agence Nationale de la Recherche (ANR) in France under the AUDINM project and The Ministry of Higher Education and Scientific Research of Tunisia under the grant agreement number LR11ES48.

REFERENCES

[1] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *PAMI*, vol. 37, no. 7, pp. 1480–1500, 2015.

[2] J.-J. Lee, P.-H. Lee, S.-W. Lee, A. Yuille, and C. Koch, "Adaboost for text detection in natural scene," in *ICDAR*, 2011, pp. 429–434.

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

[4] C. Yi and Y. Tian, "Text extraction from scene images by character appearance and structure modeling," *Computer Vision and Image Understanding*, vol. 117, no. 2, pp. 182–194, 2013.

[5] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *CVPR*, 2010, pp. 2963–2970.

[6] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *CVPR*, 2012, pp. 3538–3545.

[7] L. Gomez and D. Karatzas, "A fast hierarchical method for multi-script and arbitrary oriented scene text extraction," *IJDAR*, vol. 19, no. 4, pp. 335–349, 2016.

[8] W. Huang, Z. Lin, J. Yang, and J. Wang, "Text localization in natural images using stroke feature transform and text covariance descriptors," in *ICCV*, 2013, pp. 1241–1248.

[9] X. Liu, K. Lu, and W. Wang, "Effectively localize text in natural scene images," in *ICPR*, 2012, pp. 1197–1200.

[10] W. Huang, Y. Qiao, and X. Tang, "Robust scene text detection with convolution neural network induced mser trees," in *ECCV*, 2014, pp. 497–511.

[11] C. Zhang, C. Yao, B. Shi, and X. Bai, "Automatic discrimination of text and non-text natural images," in *ICDAR*, 2015, pp. 886–890.

[12] S. Zhu and R. Zanibbi, "A text detection system for natural scenes with convolutional feature learning and cascaded classification," in *CVPR*, 2016, pp. 625–632.

[13] K. Chen, F. Yin, A. Hussain, and C.-L. Liu, "Efficient text localization in born-digital images by local contrast-based segmentation," in *ICDAR*, 2015, pp. 291–295.

[14] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazàn, and L. P. de las Heras, "Icdar 2013 robust reading competition," in *ICDAR*, 2013.

[15] A. Zhu and S. Uchida, "Scene text relocation with guidance," in *ICDAR*, 2017.

[16] W. S. C. Y. W. L. X. B. Zheng Zhang, Chengquan Zhang, "Multi-oriented text detection with fully convolutional networks," in *arXiv:1604.04018*, 2016.

[17] 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.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *IEEE Neural Information Processing Systems*, 2015.

[19] 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 29*, 2016.