Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 431608, 7 pages
http://dx.doi.org/10.1155/2015/431608

Research Article

Multilingual Text Detection with Nonlinear Neural Network

Lin Li,[1,2] Shengsheng Yu,[2] Luo Zhong,[1] and Xiaozhen Li[2]

[1]*School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China*
[2]*School of Computer Science, Huazhong University of Science and Technology, Wuhan 430070, China*

Correspondence should be addressed to Lin Li; lilin.wzy@gmail.com

Received 11 July 2015; Accepted 2 September 2015

Academic Editor: Xinguang Zhang

Multilingual text detection in natural scenes is still a challenging task in computer vision. In this paper, we apply an unsupervised learning algorithm to learn language-independent stroke feature and combine unsupervised stroke feature learning and automatically multilayer feature extraction to improve the representational power of text feature. We also develop a novel nonlinear network based on traditional Convolutional Neural Network that is able to detect multilingual text regions in the images. The proposed method is evaluated on standard benchmarks and multilingual dataset and demonstrates improvement over the previous work.

1. Introduction

Texts within images contain rich semantic information, which can be very beneficial for visual information retrieval and image understanding. Driven by a variety of real-world applications, scene text detection and recognition have become active research topics in computer vision. To efficiently read text from photography, the majority of methods follow the intuitive two-step process: text detection followed by text recognition [1]. To a large extent, the performance of text detection affects the accuracy of text recognition. Extracting textual information from natural scenes is a critical prerequisite for further text recognition and other image analysis tasks.

Text detection has been considered in many studies and considerable progress has been achieved in recent years [2–14]. However, most of the text detection methods have focused on English; few investigations have been done on the problem of the multilingual text detection. In our daily lives, multilingual texts coexist everywhere; many environments contain two or more scripts text in a single image and, for example, product tags, street signs, license plates, billboards, and guide information. More and more applications need to achieve text detection regardless of language type.

For different languages, the characters take many different forms and have inconsistent heights, strokes, and writing format. There are thousands of languages in the world, and the representative and universal features of multilingual text are still unknown. In addition, text embedded in images can be in variation of font style and size, different alignment and orientation, unknown colors, and varying lighting condition. Due to these factors, multilingual text detection in natural scenes is a challenging and important problem.

Our study is focused on learning the general stroke feature representations and detecting text from image even in a multiscript environment. Unlike traditional methods, which mainly relied on the combination of a number of hand-engineered features, we aim to test the feasibility of proposing a common text detector only using automatically learning text feature, by improving discriminative clustering algorithm, to obtain language-independent stroke features. The learned stroke features incorporating with nonlinear neural network provide an alternative way to effectively increase the character representational power. To use deep learning text feature, we are able to use simple nonmaximal suppression to locate text.

In the following, we first reviewed the recent published literature followed by the proposed multilingual text detection method from Section 3 to Section 4. In Section 5, the experimental evaluation is presented. The paper is concluded in Section 6.

2. Related Work

Existing methods proposed for text detection in natural scenes can be broadly categorized into two groups: connected component methods and sliding window methods.

Connected component methods separate text and non-text information at pixel-level, group text pixels to construct character candidates from images by connected component analysis. Epshtein et al. [2] leveraged the idea of the recovery of stroke width and proposed using the CCs in a stroke width transformed image. Yao et al. [3] extract regions in the Stroke Width Transform (SWT) domain. Neumann and Matas [4] posed the character detection problem as an efficient sequential selection from the set of Extremal Regions (ERs). Chen et al. [5] determined the stroke width using the distance transform of edge-enhanced Maximally Stable Extremal Regions (MSER). Using MSERs as CCs representing characters has become the focus of several recent works [6–9].

Sliding window-based methods, also known as region-based methods, scan a sliding subwindow through the image to search for possible texts and then use machine learning techniques to locate text. Wang et al. [10], extending their previous work [11], have built an end-to-end scene text recognition system based on a sliding window character classifier using Random Ferns. Wang et al. [12] use multi-layer neural networks for text detection. Jaderberg et al. [13] achieve state-of-the-art performance by implementing sliding window detection as a byproduct of the Convolutional Neural Network (CNN).

In the task of multilingual text detection, previous studies are mostly sliding window-based method. In [14, 16], authors have proposed similar methods using hand-engineered features to describe the text. Subwindow scanned on different scales and positions on the image pyramid in order to classify texts in images. Therefore, to achieve text detection which is invariance to language type, the feature representation is very important. However, little research attempted to apply deep learning to learn multilingual text feature. CNN is a special kind of neural network, and its deep nonlinear network structure shows the strong ability of learning discriminative features of datasets from observation samples. Therefore, we alternatively investigate the problem of multilingual text detection based on the framework of CNN.

3. Stroke Feature Learning

According to the study of linguistics, the basic feature of text is stroke, such as the Latin alphabet and Chinese basic stokes. And different languages share the same characteristics in appearance. Inspired by these ideas, it is possible that language-independent stroke features can be designed.

In order to cope with multilingual scenes, we seek to learn a bank of universal low-level stroke features directly from raw images. The learning stroke features should be able to capture the essential substructures of strokes. At the same time, they are of the most representative and discriminative stroke features. Many unsupervised learning algorithms can be used for learning the hidden data prototypes from dataset,

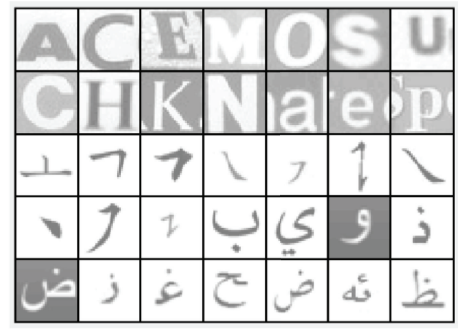

Figure 1: Training samples for stroke feature learning.

such as K-means clustering and sparse coding. The goal of sparse coding is to construct a dictionary D and minimize the error in reconstruction $\min_{D,s} \sum_i \|Ds^{(i)} - x^{(i)}\|_2^2 + \lambda\|s^{(i)}\|_1$, so that a data vector $x^{(i)} \in \mathbb{R}^n$ $(i = 1, \ldots, m)$ can be mapped to a code vector $s^{(i)}$. For every $s^{(i)}$, sparse coding algorithm is required to repeatedly solve a convex optimization problem. When applied to large scale image data, the optimization problem during the sparse coding procedure is very expensive. Relatively speaking, the optimal $s^{(i)}$ in classic K-means algorithm is simply as follows:

$$s_j^{(i)} = \begin{cases} D^{(j)\top} x^{(i)} & \text{if } j = \underset{j}{\operatorname{argmax}} \left| D^{(j)\top} x^{(i)} \right| \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

In addition, K-means has been identified as a fast and effective method to learn feature from images by computer vision researchers. Therefore, we improve the variant K-means clustering method proposed by Coates et al. [18] and use it to learn stroke feature representations, since it learns representative stroke features from large collections while much faster.

In particular, we first collect a set of training images, which are 32×32 gray scale images extracted from ICDAR 2003, ICDAR 2011, and ICDAR 2013 dataset, multilingual dataset, and Google. It contains a character in the middle of each image. Characters in training images include 26 uppercase letters, 26 lowercase letters, 10 digits, 20 Chinese basic strokes, and 28 Arabic alphabets. Some training images used for stroke feature learning are illustrated in Figure 1. We randomly extract m 8×8 pixel patches from images. Before training the cropped patches, we apply contrast normalized preprocessing for each patch. In order to avoid generating many highly correlated stroke features, ZAC whitening is used for the patches to yield vectors $x^{(i)} \in \mathbb{R}^{64}, i \in \{1, \ldots, m\}$.

Because K-means algorithm is highly dependent on the initialization process, the different initial guess of centroids affects the clustering result seriously. In order to lead to desirous clustering result, we propose a novel initialization method to choose suitable initial stroke features. We introduce the dispersion metric in the local information of data, guaranteeing the selection of initial centroids from the local spatial data-intensive region and the centroids apart from

Input: the patches x ($x^{(i)} \in \mathbb{R}^{64}, i = 1, \ldots, m$)
Output: initial features $F_0 \in \mathbb{R}^{64 \times n_1}$
(1) $C \leftarrow \emptyset$
(2) construct w_{ij} based on (2)
(3) computer dispersion metric $d^{(i)} = \sum_{j=1}^{m} w_{ij}$ and threshold $q = \text{median}(d)$
(4) **for** all data in x
(5) **if** data $x^{(j)}$ with dispersion metric $d^{(j)} > q$
(6) $C \leftarrow C \cup x^{(j)}$
(7) **end if**
(8) **end for**
(9) $F_0^{(1)} = x^{(i)}$, $x^{(i)}$ is random selected from C
(10) $F_0^{(2)} = \underset{x^{(j)}}{\text{argmax}} \left\| x^{(j)} - F_0^{(1)} \right\|, \quad \forall x^{(j)} \in C$
(11) set $k = 2$
(12) **repeat**
(13) $k = k + 1$
(14) $F_0^{(k)} = \underset{x^{(j)}}{\text{argmax}} \left\{ \min \left\| x^{(j)} - F_0^{(t)} \right\|, \ \forall x^{(j)} \in C, \ t = 1, \ldots, k-1 \right\}$
(15) **until** $k = n_1$

Algorithm 1: Stroke feature initialization method.

each other with a certain distance. Our initialization framework includes three steps: (1) estimating local dispersion metric for each set of data, (2) selecting the data which have higher metric than a threshold as candidates for initial features, and (3) determining initial stroke features from candidates. The implements of the proposed initialization method are as follows. We firstly construct an adjacency graph and Gram matrix; Gram matrix is computed according to the following:

$$w_{ij} = \begin{cases} 1 & \text{if } \left\| x^{(i)} - x^{(j)} \right\| < \varepsilon \\ 0 & \text{otherwise} \end{cases}$$
$$\varepsilon = \frac{\sum_{i=1}^{m} \sum_{j=1}^{m} \text{dis}\left(x^{(i)}, x^{(j)}\right)}{m \times (m-1)} \times 0.5, \tag{2}$$

where $\text{dis}(x^{(i)}, x^{(j)})$ is the distance of the patches $x^{(i)}$ and $x^{(j)}$. Secondly, we introduce the dispersion metric d with m components whose entries are given by $d^{(i)} = \sum_{j=1}^{m} w_{ij}$ ($i = 1, \ldots, m$). Set a threshold $q = \text{median}(d)$; if the value of $d^{(i)}$ associated with the data $x^{(i)}$ is larger than threshold; $x^{(i)}$ is marked as a candidate of initial features. Then, we use an algorithm similar to [19] to select initial stroke features from candidates. Because we use the stroke features as the first layer convolution kernels of our proposed CNN, n_1 is the number of first layer convolutional filters. The detailed steps of the proposed initialization method are presented in Algorithm 1.

After initializing F_0, we learn stroke features $F \in \mathbb{R}^{64 \times n_1}$ according to the following:

$$\min_{F, s^{(i)}} \sum_{i} \left\| F s^{(i)} - x^{(i)} \right\|^2. \tag{3}$$

For all $j \in \{1, \ldots, n_1\}$, compute inner products $F^{(j)\top} x^{(i)}$. Set the value of k which equals the value of j which maximizes the inner products. If $j = k$, then $s_j^{(i)} = F^{(j)\top} x^{(i)}$; or else $s_j^{(i)} = 0$. Then, fix $s^{(i)}$, minimizing (3) to obtain F. The optimization is done by alternating minimization over F and $s^{(i)}$. The full stroke feature learning algorithm with K-means is summarized in Algorithm 2.

For general clustering algorithm, the number of clusters is known in advance or set by prior knowledge. In our method, the learned stroke features incorporate with Convolutional Neural Network classifier for text detection. Therefore, we further study how to choose the appropriate number of features to achieve the highest text/no text classification accuracy. In order to analyze the impact of learned stroke feature number, we learned four stroke feature sets with different number. Given that the first layer convolution kernels of our CNNs have $n_1 = 96, 128, 256$ and 320, we train detector with different stroke feature sets. Evaluate the performance of the detection model at the subset of ICDAR 2003 test images. As shown in Figure 2, the F-measure increases as n_1 gets larger. Once n_1 equals 256, the recall is at maximum value, and about 80% of detected text matches ground truth. While n_1 is greater than 256, F-measure is not increased and even slightly reduced. Based on our detailed analysis, in our method, we select $n_1 = 256$.

4. Multilingual Text Detection

The idea of our text detection is to design "feature learning" pipeline that can lead to representative text features and use these features for detecting multilingual text. Two main components in this pipeline are as follows: (1) use the unsupervised clustering algorithm to generate a set of stroke features F; (2) build a hierarchy network and combine it with stroke features F to learn a high-level text feature. The first component has been described in detail in Section 3. How to

Input: m 8×8 input patches $x^{(i)} \in \mathbb{R}^{64}$
Output: learning stroke features $F \in \mathbb{R}^{64 \times n_1}$
Procedure:
(1) Normalize input

$$x^{(i)} = \frac{x^{(i)} - \text{mean}\left(x^{(i)}\right)}{\sqrt{10 + var\left(x^{(i)}\right)}}$$

(2) ZAC whiten input

$$VDV^{\top} = \text{cov}(x)$$
$$x^{(i)} = V(D + 0.1 \times I)^{-1/2} V^{\top} x^{(i)}$$

(3) Initialize F, follow the steps in Algorithm 1
(4) Repeat
Set $s_k^{(i)} = F^{(j)\top} x^{(i)}$ for $k = \text{argmax}_j |F^{(j)\top} x^{(i)}|$
Set $s_j^{(i)} = 0$ for all other $j \neq k$
Fix $s^{(i)}$, $\min_{F, s^{(i)}} \sum_i \|Fs^{(i)} - x^{(i)}\|^2$ s.t. $\|s^{(i)}\|_1 = \|s^{(i)}\|_\infty$ and $\|F^{(j)}\|_2 = 1$
Until convergence or reach iteration limitation

Algorithm 2: Stroke feature learning algorithm.

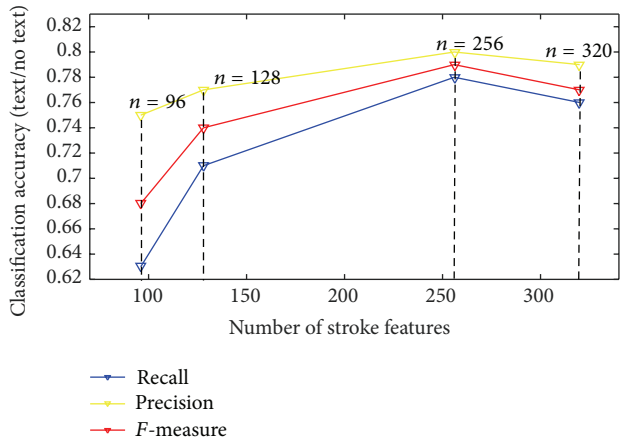

Figure 2: The accuracy analysis on different stroke feature number.

build and train the multilayer neural network is presented in Section 4.

By making several technical changes over traditional Convolutional Neural Network architectures [20], we develop a novel classifier for multilingual text detection. We have two major improvements: (1) different from traditional method that convolution kernel of CNN is randomly generated, we select the unsupervised learning stroke features F as the first layer convolution kernels of our network; (2) the intermediate features obtained from the learning process, which function in the second layers convolution kernels, can be used to more efficiently extract text features.

Our network has two convolutional layers with n_1 and n_2 filters, respectively. We fix the filters in the first convolution layer which are stroke features learned in Section 3; so low-level filters are $F \in \mathbb{R}^{64 \times n_1}$ and $n_1 = 256$. We build a set of labeled training datasets; all training images are 32×32 fixed-sized images (8877 positive, 9000 negative). Starting from the first layer, given an input image, the input is a grayscale cropped training image; that is, $z_0 = x$. The input convolves with 256 filters of size 8×8, resulting in a map of size 25×25 (to avoid boundary effects) and 256 channels. The first convolutional layer output z_1 is a new feature map computed by a nonlinear response function $z_1 = \max\{0, |F^{\top} x| - \alpha\}$, where $\alpha = 0.5$. Convolutional layers can be intertwined with pooling layers that simplify system parameters by statistical aggregation of features. We average pool over the first convolutional layer response map to reduce the size down to 5×5. The sequence continues by another convolutional and pooling layers, resulting in feature maps with 256 channels and size of 2×2; this size is the same as the dimension of the second layer convolutional filters. The second layer outputs are fully connected to the classification layer. The SVM classifier is used as a binary classifier that aims to estimate whether a 32×32 image contains text. We train the network using stochastic gradient descent and back-propagation. Classification error function includes loss term and regularization term. Loss term is a squared hinge loss and the norm used in the penalization is L2. We also use dropout in the second convolutional layer to help prevent over fitting. The structure of the proposed neural network is presented in Figure 3.

After our network has been trained, the detection process starts from a large, raw pixel input image and leverages the convolutional structure of the CNN to process the entire image. We slide a 32×32 pixels' window across an input image and put these sliding windows to the learned classifier. Use the intermediate hidden layers as features to classify text/no text and generate text bounding boxes. We set 12 different scales in our detection method. At a certain scale s, the input image's scale changes; the sliding window scans through the scaled image. At each point (x, y), if windows contain single centered characters, produce positive detector response $R_s[x, y]$. In each row r of the scaled image, check whether there are $R_s[x, y] > 0$. If there exists positive detector response, then form a line-level bounding box L_s^r with

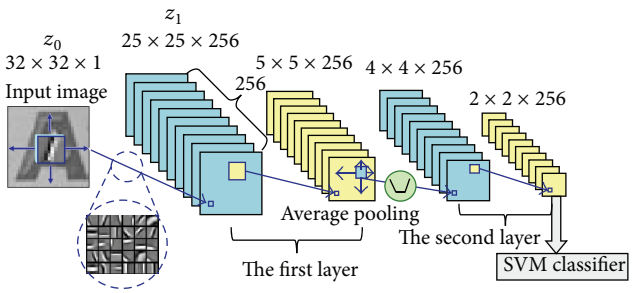

Figure 3: The proposed network for multilingual text detection.

the same height as the sliding window at scale s. And $\max(x)$ and $\min(x)$ are defined as the left and right boundaries of L_s^r. At each scale, the input image is resizing and a set of candidate text bounding boxes are generated independently. The above procedure was repeated 12 times and yields groups of possibly overlapping bounding boxes. We then apply nonmaximal suppression (NMS) to score each box and remove all boxes that overlaps by more than 50% with lower score and obtain the final text bounding boxes L.

5. Experiments

5.1. Dataset. To evaluate the effectiveness and robustness of the proposed text detection algorithm, we have conducted experiments on standard benchmarks, including the challenging datasets ICDAR 2003 [21], MSRA-TD500 [3], and KAIST [17].

The ICDAR 2003 Robust Reading and Text Locating database is a widely used public dataset for scene text detection algorithm. The database contains 258 training images and 251 testing images. It contains the path to each image and text bounding box annotations for each image.

MSRA-TD500 dataset contains images with text in English and Chinese. The dataset contains 500 images in total, with varying resolutions from 1296 × 864 to 1920 × 1280. These images are taken from indoor (office and mall) and outdoor (street) scenes using a packet camera.

KAIST provides a scene text dataset consisting of 3000 images of indoor and outdoor scenes. Word and character bounding boxes are provided as well as segmentation maps of characters. Texts in KAIST images are English, Korean, and a mixture of English and Korean.

We also created a new multilingual dataset that is composed of three representative languages: English, Chinese, and Arabic. These three languages stand for three types of writing systems: English standing for alphabet, Chinese standing for ideograph, and Arabic standing for abjad. Each group corresponding to the one language contains 80 images.

To learn the stroke feature, train samples include 5980 English text samples, 800 Chinese text samples, and 1100 Arabic text samples. Then, 3000 nontext samples are extracted from 200 negative images using bootstrap method. All these samples are normalized to 32 × 32, which is consistent with the detected window.

Table 1: Performance of the proposed method.

Language	Total image	Precision	Recall	F-measure
English	800	0.76	0.88	0.8
Korean	500	0.73	0.84	0.78
Chinese	200	0.68	0.96	0.79
Arabic	200	0.70	0.72	0.66

Figure 4: Text detection samples on different language images.

5.2. Results. The proposed algorithm is implemented using Intel Core i5 processor at 2.9 GHz 8 GB RAM and MATLAB R2014b.

To validate the performance of our proposed algorithm, we use the definitions in ICDAR 2003 competition [21] for text detection precision, recall, and F-measure calculation. Therefore, $P = \sum_{r_e \in E} m(r_e, T)/|E|$ and $R = \sum_{r_t \in T} m(r_t, E)/|T|$, where $m(r, R)$ is the best match for a rectangle r in a set of rectangles R, E, and T which are our estimated rectangles and the ground truth rectangles, respectively. We adopt the F-measure to combine the precision and recall figures into a single measure of quality, $F = 1/(\alpha/P + \alpha/R)$, where $\alpha = 0.5$.

Experiments are carried out on a set of images containing text in four different languages, namely, English, Chinese, Arabic, and Korean. English text images are selected from ICDAR 2003, ICDAR 2011, and ICDAR 2013, Korean images from KAIST, some Chinese images from MSRA-TD500 and the other from multilingual dataset, and Arabic images from multilingual dataset and Google. The results of these evaluations are summarized in Table 1. As can be seen from Table 1, F-measures on different language are close to each other, except Arabic, because the Arabic special nature of continuous writing style makes the recall of this script lower. The experiment result indicates that our method is not tuned to any particular language and performs approximately equally good on all the scripts.

Figure 4 shows some texts successful detection by our system on images containing different language text. Although the texts contained in training samples are only in English, Chinese, and Arabic, our method can detect the text not only in three representative languages, but also in a number of other languages, such as French, German, Korean, and Japanese. This shows that our method has some robustness.

Table 2: Performance comparison on different benchmarks.

Method	Dataset	Precision	Recall	F-measure
Pan [15]	ICDAR 2003	0.645	0.659	0.652
Zhou [16]	ICDAR 2003	0.37	0.79	0.53
Our method	ICDAR 2003	0.45	**0.80**	0.57
Lee [17]	KAIST	0.69	0.60	0.64
Our method	KAIST	0.59	**0.79**	**0.67**

Figure 5: Text detection samples on images containing two different languages text.

We also picked methods proposed by Zhou et al. [16], Pan et al. [15], and Lee et al. [17] for further consideration. These algorithms have good results on the standard benchmarks and use different approaches to detect text. The performance comparison analysis can be seen in Table 2. Our method has achieved high recall at different benchmarks. It also reflects the representative of our learned feature is strong, which can successfully detect all the information associated with the text in images.

But our test results are not good, with $P/R/F$-means of 0.30/0.32/0.31 on the MSRA-TD500 dataset. Shi et al. [6] have achieved the state-of-the-art text detection performance with 0.52/0.53/0.5 on the same dataset. The main reason is that the MSRA-TD500 dataset is created for the purpose of study of multiorientation text detection, which has a lot of images containing no horizontal text lines. But our method gives the text bounding boxes based on the horizontal direction.

Figure 5 shows some other test samples. The results reflect our method is efficient on the circumstance that a single image contains two or more different languages texts and numbers. The bottom row in Figure 5 shows some fail samples; some of these problems are miss detection for part of Arabic text, because Arabic words mostly are linked by continuous line. In this case, use of the stroke feature to detect text is not sufficient. Stroke width of the implementation is essential for such languages as Arabic. There are other problems caused by the interference terms which have the appearance similar to the text.

6. Conclusion

The aim of the study is to propose a multilingual text detection method. Traditional methods in this area mainly rely on large amounts of hand-engineered features or prior knowledge. Our work is distinct in two ways: (1) we use primitive stroke feature learned by unsupervised learning algorithm as network convolutional kernels; (2) we leverage the trained multilayer neural network to learn high-level abstract text features used for detector. Experiments on the public benchmark and multilingual dataset show our method is able to localize text regions of different scripts in natural scene images. The experiment results demonstrate the robustness of the proposed method.

From the failed samples in the experiments, we analyze the limitations of our technology for further improvement. On the one hand, some languages have continuous writing style, like Arabic; automatically learning features are not enough for detection; the connected components analysis will be added into our method to improve the precision of final results. On the other hand, multiorientation text problem will be considered.

Conflict of Interests

The authors declared that they have no conflict of interests regarding this work.

References

[1] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 2, pp. II366–II373, July 2004.

[2] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 2963–2970, June 2010.

[3] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1083–1090, June 2012.

[4] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 3538–3545, Providence, RI, USA, June 2012.

[5] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod, "Robust text detection in natural images with edge-enhanced maximally stable extremal regions," in *Proceedings of the 18th IEEE International Conference on Image Processing (ICIP '11)*, pp. 2609–2612, September 2011.

[6] C. Shi, C. Wang, B. Xiao, Y. Zhang, and S. Gao, "Scene text detection using graph model built upon maximally stable extremal regions," *Pattern Recognition Letters*, vol. 34, no. 2, pp. 107–116, 2013.

[7] A. Shahab, F. Shafait, and A. Dengel, "ICDAR 2011 robust reading competition challenge 2: reading text in scene images," in *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR '11)*, pp. 1491–1496, Beijing, China, September 2011.

[8] 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.

[9] L. Neumann and J. Matas, "Scene text localization and recognition with oriented stroke detection," in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 97–104, December 2013.

[10] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 1457–1464, IEEE, Barcelona, Spain, November 2011.

[11] K. Wang and S. Belongie, "Word spotting in the wild," in *Proceedings of the 11th European Conference on Computer Vision (ECCV '10)*, pp. 591–604, 2010.

[12] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR '12)*, pp. 3304–3308, November 2012.

[13] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR '14)*, 2014.

[14] A. Raza, I. Siddiqi, C. Djeddi, and A. Ennaji, "Multilingual artificial text detection using a cascade of transforms," in *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR '13)*, pp. 309–313, Washington, DC, USA, August 2013.

[15] 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.

[16] G. Zhou, Y. Liu, Q. Meng, and Y. Zhang, "Detecting multilingual text in natural scene," in *Proceedings of the 1st International Symposium on Access Spaces (ISAS '11)*, pp. 116–120, IEEE, Yokohama, Japan, June 2011.

[17] S. Lee, M. S. Cho, K. Jung, and J. H. Kim, "Scene text extraction with edge constraint and text collinearity," in *Proceedings of the 20th International Conference on Pattern Recognition (ICPR '10)*, pp. 3983–3986, August 2010.

[18] A. Coates, B. Carpenter, C. Case et al., "Text detection and character recognition in scene images with unsupervised feature learning," in *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR '11)*, pp. 440–445, September 2011.

[19] M. E. Celebi and H. A. Kingravi, "Deterministic initialization of the k-means algorithm using hierarchical clustering," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 26, no. 7, Article ID 1250018, 2012.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[21] S. Lucas, P. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robust reading competitions," in *Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR '03)*, pp. 682–687, Edinburgh, UK, August 2003.