Copyright ⓒ2016 KSII

Integrated Method for Text Detection in Natural Scene Images

Yang Zheng[1], Jie Liu[2], Heping Liu[1], Qing Li[1], Gen Li[2]

[1]School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, P. R. China

[e-mail: yzheng@xs.ustb.edu.cn; e-mail: liqing@ies.ustb.edu.cn]

[2]Institute of Automation, Chinese Academy of Sciences, Beijing 100190, P. R. China

*Corresponding author: Qing Li

Received May 7, 2016; revised September 9, 2016; accepted October 12, 2016; published November 30, 2016

Abstract

In this paper, we present a novel image operator to extract textual information in natural scene images. First, a powerful refiner called the Stroke Color Extension, which extends the widely used Stroke Width Transform by incorporating color information of strokes, is proposed to achieve significantly enhanced performance on intra-character connection and non-character removal. Second, a character classifier is trained by using gradient features. The classifier not only eliminates non-character components but also remains a large number of characters. Third, an effective extractor called the Character Color Transform combines color information of characters and geometry features. It is used to extract potential characters which are not correctly extracted in previous steps. Fourth, a Convolutional Neural Network model is used to verify text candidates, improving the performance of text detection. The proposed technique is tested on two public datasets, i.e., ICDAR2011 dataset and ICDAR2013 dataset. The experimental results show that our approach achieves state-of-the-art performance.

Keywords: Stroke Color Extension, character classifier, Character Color Transform, Convolutional Neural Network

1. Introduction

With the development of science and technology, the Internet has brought forth tremendous new products, services and demands. Text detection and localization in natural scene images has become a crucial task. Although some recent approaches [1]-[3] have been proposed in computer vision, some problems remain largely unsolved. The difficulties mainly relate to three aspects [4]: (1) Texts in natural scene images exhibit entirely different colors, scales and orientations; (2) Some backgrounds are very similar to texts, for example, bricks, fences, and grasses; (3) Some texts are destroyed by external factors, such as distortion, low contrast, noise, illumination and partial occlusion.

There are three mainstream algorithms for detecting and locating texts in current methods: sliding window-based methods [5]-[7], connected component (CC)-based methods [8]-[10] and hybrid methods [3]. Sliding window-based methods usually train discriminative models to detect text patches at multiple scales and group them into text regions. They are effective for clutter backgrounds and multilingual environments, but a small image patch often does not contain sufficient discrimination information. These methods are slow since the image has to be processed on multiple scales. Connected component-based methods always extract connected components as character candidates by using local properties. Stroke Width Transform (SWT) [8] and Maximally Stable Extremal Region (MSER) [9] are widely used as basic algorithms due to their efficiency and stability, but they perform poorly under severe conditions, such as blur, non-uniform illumination and disconnected strokes. The hybrid methods try to combine the advantages of sliding window-based and CC-based methods in order to achieve high robustness with low computation requirements.

Our goal is to develop a text localization system that combines the advantages of Stroke Width Transform, gradient features, color information and Convolutional Neural Network (CNN) model, while overcoming their inherent limitations. It makes the following major contributions:

(1) A new refiner called the Stroke Color Extension (SCE) that combines the original Stroke Width Transform (SWT) with the color information of strokes is proposed to extract characters. The SCE refiner effectively enhances intra-character connection while removing non-character component, leading to a significantly better performance than SWT algorithm.

(2) A character classifier is trained by using gradient features [11] to remove non-character components.

(3) A character extractor called the Character Color Transform (CCT), which combines the color information of characters with geometry features, is proposed to extract potential characters that are not extracted correctly in previous steps. The remaining connected components (CCs) are considered as character candidates.

(4) Based on the distances between adjacent characters and adjacent words, text lines are separated into words candidates, and then a Convolutional Neural Network (CNN) model is used to verify text candidates.

The remainder of this paper is organized as follows: In Sec.2, we briefly introduce some previous methods that are related to the proposed algorithm in this paper. In Sec.3, we describe the proposed algorithm in detail, including connected component extraction, connected component filtration, potential character extraction, the word generation and verification. In Sec.4, we describe the results of the proposed algorithm and discuss the comparison with other previous algorithms. Finally, conclusion remarks and future works are given in Sec.5.

2. Related Work

Over the past years, some research institutions and researchers expended a lot of effort and achieved a great progress on text detection in natural scene images. Zhang et al. [12] made a comprehensive survey on text detection from natural scene images. The purpose of them was to classify, review and analyze the existing algorithms, discuss the databases and performance measurement and point out future work. They classified the localization and detection methods into five types: edge-based, texture-based, connected component-based, stroke-based and the other methods. There are two representative approaches for natural scene text detection: SWT [8] and MSER [9]. We will focus on some works that are most relevant to the proposed algorithms. MSER was adopted by Neumann and Matas [9], [13], [14]. They assumed each component is an independent extremal region (ER) and used maximally stable extremal regions (MSERs) to extract possible characters. Although MSER achieved great progress in previous methods, too many non-character components were extracted. Researchers have proposed some methods to address these problems. L. Sun, Q. Huo [15], [16] used contrast extremal region (CER) [15] and color-enhanced CER [16] to extract ERs as character candidates. Weilin Huang [17] used MSER Tress to reduce the number of windows scanned and to enhance detection of low-quality texts and then used sliding-window with CNN to correctly separate the connections of multiple characters into components. Xucheng Yin [2] used the strategy of minimizing regularized variations to extract MSERs as character candidates and grouped them into text candidates by single-link clustering algorithm and then verified text candidates by a text classifier.

Epshtein [8] proposed a novel algorithm named Stroke Width Transform (SWT) to measure the distance of two gradient orientations between pairs of edge pixels and to group pixels with similar widths into a connected component as a character candidate. Weilin Huang [10] proposed a Stroke Feature Transform (SFT) filter based on SWT and trained two classifiers based on novel Text Covariance Descriptors (TCDs) to detect text. The SWT algorithm is heavily used in several recent studies and also serves as the foundation for the

SCE refiner proposed in this work. However, a weakness of SWT lies in its inability to handle some characters whose edges cannot be detected correctly. To resolve this problem, a SCE refiner that combines the SWT algorithm with color information is proposed in this paper.

Recently, there is a new development trend of employing deep convolutional neural networks [18]–[20] for text detection and localization in natural scene images. Some novel algorithms [17], [21], [22] used deep convolutional neural networks to achieve superior performance over conventional methods [8], [23], [24]. In our work, we also use convolutional neural networks to better eliminate non-text regions produced in the word generation stage, while maintaining relatively high precision.

3. Proposed Approach

In this section, we describe the proposed method in detail. The proposed text detection method is organized four subsections: (1) connected component extraction; (2) connected component filtration; (3) potential character extraction; (4) word generation and verification. The flowchart of the proposed method is presented in **Fig. 1** and each subsection will be described as follows:

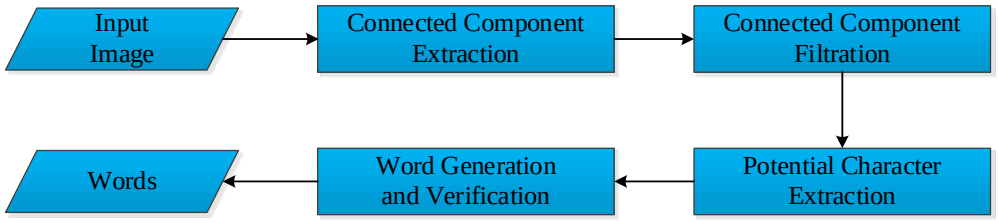

Fig. 1. Flowchart of the proposed method.

3.1 Connected Component Extraction

The introduced Stroke Width Transform (SWT) [8] has been shown to be effective for text detection. It detects stroke pixels by shooting a pixel ray along the gradient orientate ***dp*** from an edge pixel ***p*** to the opposite edge pixel ***q***. If the gradient orientations of the pair of edge pixels satisfy the specified threshold, each element of the output image corresponding to the pixel along the segment **[*p, q*]** is assigned the same stroke width, which is the distance between the pair of edge pixels. Otherwise, the ray is discarded as invalid. The process of Stroke Width Transform is presented in **Fig. 2**.

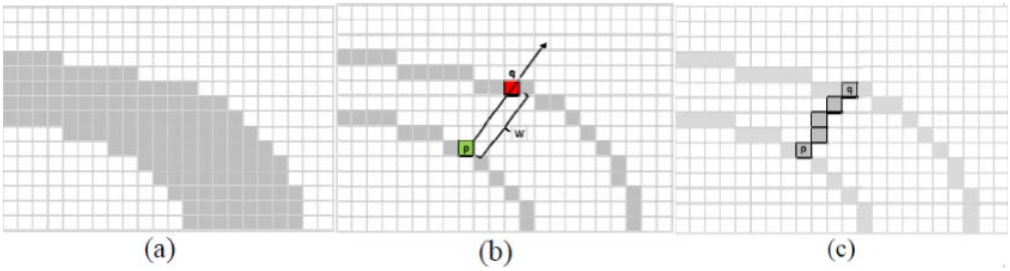

Fig. 2. (a) A typical stroke. (b) A ray from pixel ***p*** to pixel ***q***. (c) The found width of the stroke

In this way, pixels with similar strokes are grouped into a connected component as a character candidate. Although the algorithm operates on the edge image and reduces the effect of background, it needs a high-level technique for edge detection. In real cases, a large number of edge pixels are not detected correctly under severe conditions, for example, non-uniform illumination, distortion and so on. These drawbacks cause two major problems: (1) a low recall of character detection. Some characters are separated into connected components, because pairs of edge pixels are not detected correctly; (2) a low precision of character detection. Some pixels that are not edges of stroke are grouped into a connected component, because the gradient orientations of the pair of edge pixels satisfy the specified threshold.

These problems are critical to system performance, because the proposed method relies on the output of SWT. Color information always varies smoothly in a character, so the neighbourhood coherency constraint is used to extend the output of SWT to resolve these problems. We refer to this new method as Stroke Color Extension (SCE). The flowchart of connected component extraction is presented in **Fig. 3** and the algorithm of SCE is described below in detail:

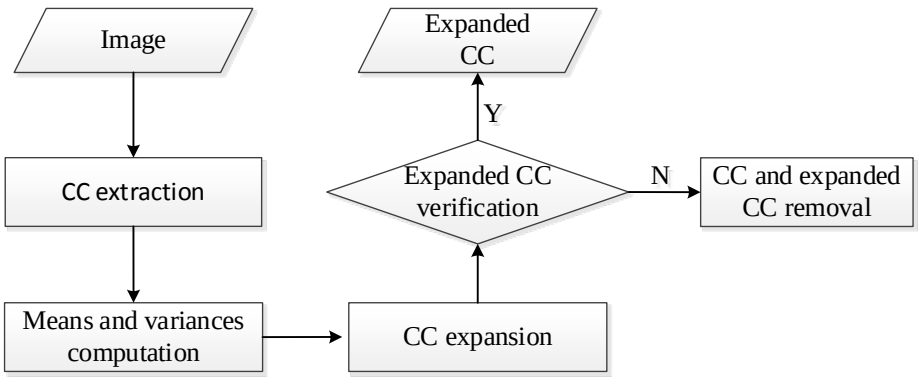

Fig. 3. The flowchart of Stroke Color Extension.

(1) Connected components are extracted by the SWT algorithm.

(2) In each connected component, the pixel values are sorted in ascending order in RGB channels to compute the median. Considering the effect of noise, a half number of total pixels whose pixel values surround the medians are selected to compute the mean and variance in each channel. The relationship between pixel values, means and variances are shown in the following. Note that edge pixels are the junctions between background and stroke, so the pixel values are not very representative. To prevent the influence of edge pixels, they are eliminated in this process.

$$\begin{cases} R_{mean} - kR_{\sigma^2} \le R \le R_{mean} + kR_{\sigma^2} \\ G_{mean} - kG_{\sigma^2} \le G \le G_{mean} + kG_{\sigma^2} \\ B_{mean} - kB_{\sigma^2} \le B \le B_{mean} + kB_{\sigma^2} \end{cases} \tag{1}$$

In RGB channels, R_{mean}, G_{mean}, B_{mean} represent the means of pixel values and R_{σ^2}, G_{σ^2}, B_{σ^2} represent the variances of pixel values. The R, G, B represent the values of current pixel in RGB channels. In each channel, the value of current pixel should satisfy a restricted range. The minimum threshold of range is the difference between the mean and k times variance. The maximum threshold of range is the sum between the mean and k times variance. The parameter k is set to 3 by experiment, which is learned on the training set of ICDAR2013. $3\sigma^2$ satisfies the requirement of pixel values and has a better adaptability.

(3) According to the mean and variance in each channel, other pixels are detected to enlarge the connected component (CC). If at least two RGB channels of current pixel that surrounds the CC satisfy the condition (Equation 1), it will be input into the CC. We will detect other pixels until no pixel satisfies the condition, and the final CC is called the expanded CC.

(4) If the original CC is a part of character, the expanded CC may be a complete character. If the original CC is non-character component, it may be expanded greatly. To verify the reliability of the expanded CC, a ratio of pixel number between expanded and original CCs is adopted. If the ratio is not exceed 5.0, the expanded CC replaces the original CC as a character candidate.

Compared to the original SWT approach, SCE refines some missing rays caused by missing edge information to prevent intra-component separations. Furthermore, it help us to reduce the number of incorrect connections substantially. Due to the stroke color constraint, the stroke pixels have better uniformity and stronger distinction from background pixels. Hence, the SCE refiner effectively ensures the integrity of characters and prunes incorrectly-connected stroke components, as shown in **Fig. 4**. Overall SCE refiner leads to more accurate component extraction than SWT.

The examples of connected component extraction are presented in **Fig. 4**. Input images are shown in the leftmost column. In the middle column, the examples show that the SWT method generates some connected components, including characters, parts of characters and non-character components. After using the SCE refiner, parts of characters are refined completely and non-character components are pruned correctly, as shown as in the rightmost column.

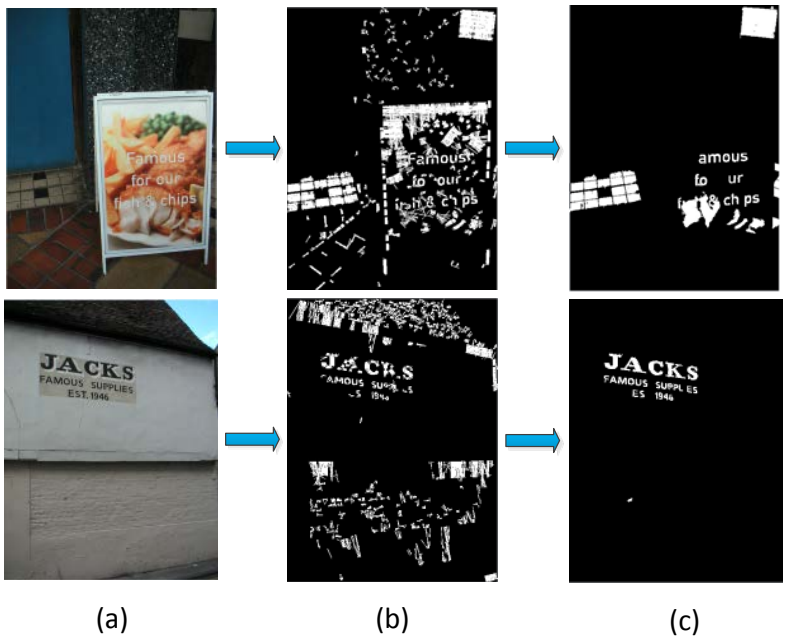

Fig. 4. Procedure of connected component extraction. (a) Original image. (b) Connected components extracted by SWT algorithm. (c) Connected components extracted by SCE refiner.

3.2 Connected Component Filtration

Connected component filtration involves two steps: the realization of character normalization and the extraction of character feature. We briefly describe it, which mainly follows the previous work in [11].

(1) The realization of character normalization.

Bilinear interpolation is used to transform the input image into the normalized image, and the size of normalized plane is set to 35×35.

(2) The extraction of character feature.

First, a blurring mask is convolved with the normalized image. Usually, a Gaussian mask is used as following. Parameters x and y represent the distance between the current point and the corresponding point. σ determines the width of the Gaussian filter. According to the definition of Gaussian filter, the weight decreases as the distance increases.

$$h(x,y)=\frac{1}{2\pi\sigma^2}\exp(-\frac{x^2+y^2}{2\sigma^2}) \tag{2}$$

Second, the Sobel operator is used to compute the x/y components of gradient. If a gradient direction lies between two standard direction, it is decomposed into two components in the two standard directions, as shown in **Fig. 5**. So the gradient image is decomposed into eight directions corresponding to eight feature planes.

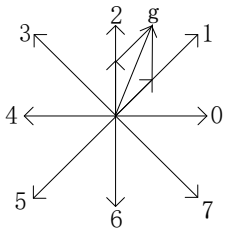

Fig. 5. Decomposition of gradient direction.

Third, each feature plane is partitioned into uniform zones, and the intensity of each zone is accumulated as a measurement. The size of normal image is set to 35×35 and the size of uniform zone is set to 7×7, so the uniform zones equals to 25 in a feature plane. There are eight feature planes, so the dimension of feature equals to 200.

To train the character classifier that identifies characters and non-character components, the C_SVC is used as the type of classifier and the Radial Basis Function (RBF) is used as the type of kernel. The training samples are collected from the training set of ICDAR 2013. The examples of connected component filtration are presented in **Fig. 6**. In the middle column, some incorrectly-connected stroke components and other outliers are extracted. They influence the performance the character detection. In the rightmost column, the examples show that the character classifier retains a lot of characters and prunes some non-character components. It is obvious that the character classifier improves the precision of character detection.

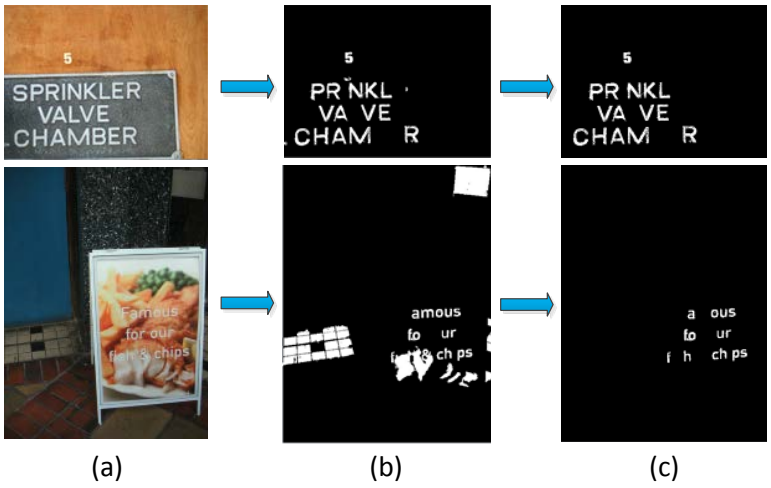

Fig. 6. Procedure of connected component filtration. (a) Original image. (b) Connected components extracted, including character components and non-character components. (c) Connected components classified by classifier.

3.3 Potential Character Extraction

A method combines color information with geometric features is proposed to extract potential characters. We refer to this new method as Character Color Transform (CCT). The flowchart of CCT is presented in **Fig. 7** and the related work of CCT as follows in detail:

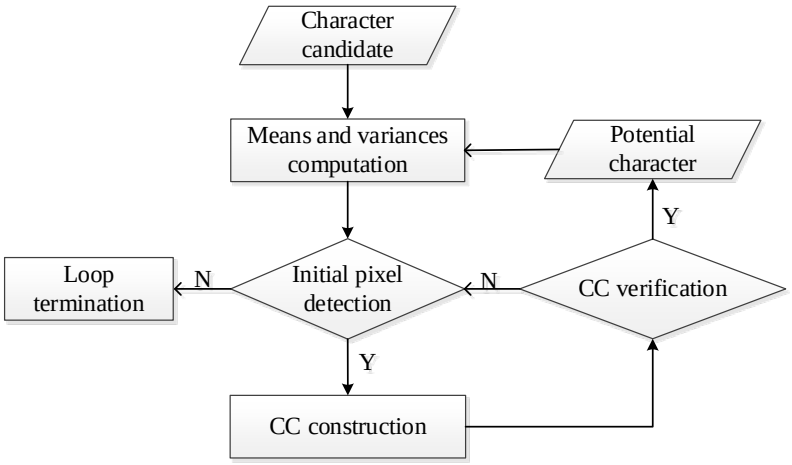

Fig. 7. The flowchart of Character Color Transform

(1) Some characters have similar properties in a line, e.g., color values, width, height, and so on, so these flexible properties are used to group characters into text lines. There are two steps as follows:

First, characters are sorted in the ascending order of x-coordinates of character centers.

Second, at the beginning of the first character, we search for other characters based on similar properties in the right direction. If a character satisfies the following conditions, we will search for another character based on this character. By that analogy, the relevant characters are grouped into the same line and annotated. Then, we repeat the above process to group other unannotated characters into new lines from left to right.

The similar properties of two characters are following: similar width (the width ratio must not exceed 5.0); similar height (the height ratio must not exceed 1.5); a reasonable horizontal distance (the horizontal distance must not exceed eight times the width of the wider one); a reasonable vertical distance (the vertical distance of two character centers must not exceed 0.6 times the height of the higher one).

(2) The potential characters always appear in three different regions in a text line: the left region of the first character, the region between adjacent characters and the right region of the last character. According to the color consistency of adjacent characters, we will combine color information with geometry features to extract potential characters.

To extract potential characters in the left region of first character, there are three steps:

First, the first character is considered as a basic character, and the means and variances of basic character in RGB channels are computed (the process is the same as the second step in section 3.1). We search for a pixel from bottom to up to the left at a distance of three pixels. If the means, variances and pixel values satisfy the condition (equation 1), the pixel will be considered as the initial pixel and put into a set named pixelsSet; otherwise, continue to move three pixels distance to the left until no initial pixel is detected.

Second, if the pixel values of two channels of pixel surrounding the pixelsSet satisfy equation 1 and the pixel value of another channel satisfies $Color_{mean} - kColor_{\sigma^2} - 5 \le Color \le Color_{mean} + kColor_{\sigma^2} + 5$, this pixel will be pushed into the pixelsSet. We repeat the above process to detect other pixels until no pixel satisfies the conditions. The potential character candidate is made up of the pixels of pixelsSet.

Third, the potential character candidate is compared with the basic character by means of geometry rules. The rules are presented as follows: the number of pixels in the pixelsSet should exceed 15; the width ratio of two characters must not exceed 8; the height ratio must not exceed 2.5; the vertical distance of two characters must not exceed 0.5 times the height of the higher one. If the potential character candidate satisfies these rules, it will be the potential character and be considered as the basic character, then we will continue to extract other potential characters based on the basic character. Otherwise, the potential character candidate is discarded and we will move three pixels distance to left to search the new initial pixel.

To extract potential characters in the right region of the last character, the steps are very similar to those above. The only difference is that we will extract potential characters from

left to right. If the distance between adjacent characters exceed 0.5 times the width of the wider one, we will extract potential characters from left to right, otherwise, there is no potential character between them.

Based on the consistency of adjacent characters, the characters of text line have uniform color information and similar geometry features. The CCT extractor is used to extract potential characters, which increases the number of detected characters substantially. The examples of potential character extraction are presented in **Fig. 8**. In the middle column, there is a text line "HAMBLION TRANSPORT" (bottom example), but only four characters ('H', 'A', 'B', 'A') are detected as character candidates in previous steps. When we use the CC extractor, other characters of the same text line are extracted correctly, as shown as in the rightmost column (bottom example).

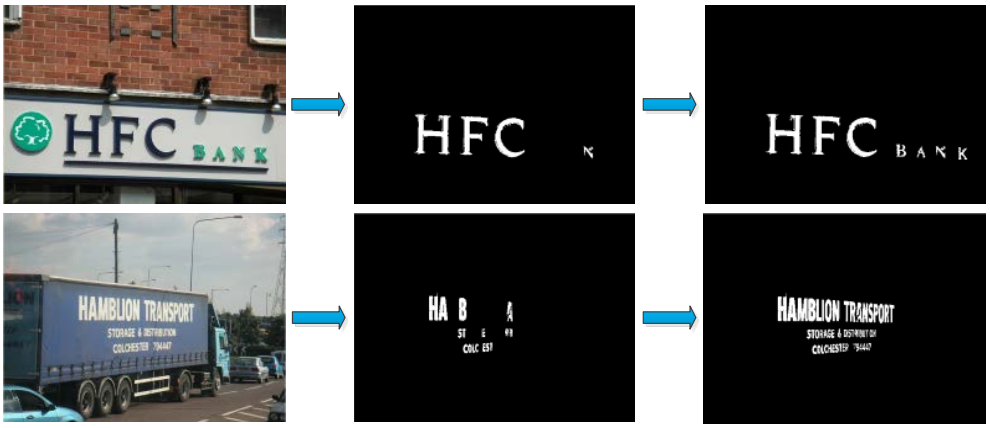

Fig. 8. Procedure of potential character extraction. (a) Original image. (b) Character candidates. (c) Potential characters extracted by CCT extractor.

3.4 Word Generation and Verification

The remaining components are considered as character candidates, including connected components that are extracted by SCE and potential characters that are extracted by CCT. To separate text lines into words, we compute a histogram of horizontal distances between adjacent characters and estimate the distance threshold that separates intra-word characters from inter-word characters. The threshold is computed to group characters into words and equals to two times the median of distances. Two characters are grouped into a word if the distance between them <= (computed threshold or 5).

Some non-character components are removed by the proposed algorithm in previous steps, but there are still a few non-character components, for example, fence, brick and circle. These components are grouped into non-text regions to influence the performance of text detection. A CNN model [22] is used to remove them, which increases the precision of text

detection. The output of model includes 37 categories (26 letters, 10 digits, 1 background), and it achieves 98.2% text/non-text precision on the training set of the ICDAR2003.

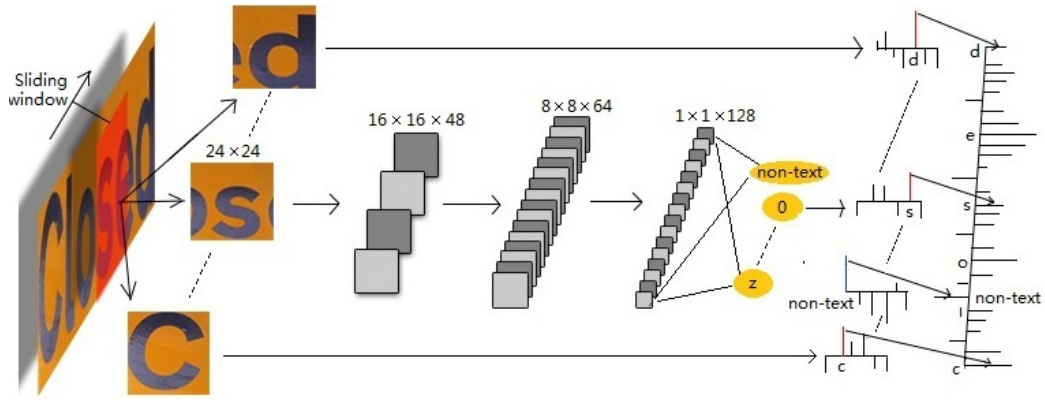

Fig. 9. Procedure of word verification based on CNN model.

For a word candidate, an input patch is obtained by a sliding window and the size is set to 24×24. The height of sliding window is set to H_w that equals to the word height, and the width of sliding window is set to $W_s = k \times H_w$. The step of sliding window is set to $S_w = 0.1 \times H_w$. Because the output of model includes 37 categories, the category which corresponds to the largest score is considered as the category of window patch. All the highest scores of sliding windows construct a CNN response histogram H. If the window patch is considered as background, the histogram value is set to $H(\mathrm{w}) = -H(\mathrm{w})$. The process in detail is illustrated in **Fig. 9**, and the evaluation scheme is illustrated as follows:

(1) If the category which corresponds to the maximum score of H is a character, the word candidate is a text.

(2) The sum of text probabilities is set to $sum(H+)$, and the sum of non-text probabilities is set to $sum(H-)$. If $sum(H+) > sum(H-)$, the word candidate is a text.

(3) If there is only one scanning window and the output is identified as letters ‘O’, ‘0’, ‘I’, ‘L’, the word candidate is non-text. (It is unusual that a single ‘O’, ‘0’, ‘I’, ‘L’ appears in an image.)

(4) Otherwise, it is non-text.

To detect different sizes of characters, we repeat this process three times on k = [0:5; 1; 1:5]. The word candidate will be consider as text if it satisfies the rules at least once.

4. Experiments and Results

The proposed algorithm is evaluated on the ICDAR datasets: ICDAR2011 [25] and ICDAR2013 [26], which are widely cited standard benchmark for scene text detection, and follows the standard evaluation protocol in this field. The performance of character detection and text detection are depicted in detail in the following subsections to prove the effectiveness of our method.

The dataset used in ICDAR2011 is inherited from the benchmark used in the previous ICDAR competitions. It has undergone extension and modification, since there are some problems with the previous dataset, for example, imprecise bounding boxes and inconsistent definitions. It includes 299 training images and 255 testing images.

The ICDAR2013 dataset is a subset of ICDAR 2011. Several images that duplicated over training and testing sets are removed and a small number of the ground truth annotations are revised. It includes 229 training images and 233 testing images.

4.1 Character Detection Performance

To demonstrate the effectiveness of the proposed algorithm, we compare each module with respect to character extraction ability. The ability is measured by using the character detection eratio on the testing set of the ICDAR2013 dataset. We chose it because it provides 233 color coded images corresponding to the images of the testing set in the Task of Text Segmentation. Each color marks a character and white is background in an image, so the connected components are extracted as characters by using color information. It includes 233 images and 5818 characters.

To make fair and direct comparison possible, we use the following definition of character detection:

$$R(character) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{|G_i|}\max_{k=1}^{|D_i|} m(G_i^{(j)}, D_i^{(k)})}{\sum_{i=1}^{N}|G_i|} \tag{3}$$

$$m(G_i^{(j)}, D_i^{(k)}) = \begin{cases} 1 & \frac{|G_i^{(j)} \cap D_i^{(k)}|}{|G_i^{(j)}|} \geq 0.8 \ \ and \\ & \frac{|G_i^{(j)} \cap D_i^{(k)}|}{|D_i^{(k)}|} \geq 0.8 \\ 0 & otherwise \end{cases} \tag{4}$$

$R(character)$ is the ratio between true characters and all true characters that should be

detected. Where $G_i^{(j)}$ and $D_i^{(k)}$ are the jth ground truth rectangle and kth detection rectangle in image i, $m(G_i^{(j)}, D_i^{(k)})$ is the match score between the jth ground truth rectangle and kth detection rectangle. If the ratio between intersection area and ground truth rectangle is not less than 0.8, and the ratio between intersection area and detection truth rectangle is not less than 0.8, the match score is set to 1. It means that the character is detected correctly by our algorithm, otherwise, it is set to 0.

Table 1. Comparing SCE refiner with SWT algorithm.

Method	Character number	Non-character number	Detection rate	Proposal number
SWT	2756	119361	2756/5818=0.474	524
SCE	3597	18292	3597/5818=0.618	94

Table 2. Performance of character classifier

		Number	Classification rate
Character	Correct classification	3176	3176/3578=0.888
	Incorrect classification	402	
Non-character	Correct classification	13777	13777/14708=0.937
	Incorrect classification	931	

Table 3. Result of potential character extraction

	Character number	Non-character number
Classification result	3176	931
Potential character	955	1031
Character candidate	4131	1962

To prove the effectiveness of the SCE refiner, as shown in **Table 1**, 2756 character components and 119361 non-character components are detected by the SWT algorithm, and 3597 character components and 18292 non-character components are detected by SCE algorithm. SWT detects 47.4% of the characters, and the average number of detected components per image is 524. SCE detects 61.8% of the characters and only produces 94 connected components on average for each image. Compared to the SWT algorithm, SCE algorithm refines more than 841 characters to improve the recall of character detection and removes 101069 non-character components to improve the precision of character detection.

Some connected components are extracted by the SWT algorithm on the training set of the ICDAR2013 dataset, including character components and non-character components. Then, we collect 2165 character components as positive samples and 2389 non-character

components as negative samples manually to train the character classifier. In **Table 2**, 3176 character components and 13777 non-character components are correctly classified by the classifier, and 402 character components and 931 non-character components are wrongly classified by the classifier. We can see that 88.8 % of the character components are retained and 93.7% of the non-character components are removed. These components will be adopted in the next step to extract potential characters, so the complete character is very necessary even if a little characters are wrongly removed. The result shows that the classifier effectively reduces the number of non-character components.

The remaining components are used to extract potential characters, including 3176 character components and 931 non-character components. As shown in **Table 3**, 955 potential characters are correctly extracted by CCT extractor. 4131 characters are eventually extracted by our algorithm. It means that 71.0% of the characters are correctly detected from 233 testing images. Some non-character components are extracted by mistake and can be divided into the following two categories: (1) The detected rectangle is matched to multiple ground truth rectangles. These components cannot influence the performance of text detection. (2) Some components always appear in a few images, such as fence, leaf. They are pruned easily in subsequent steps.

4.2 Text Detection Performance

We follow the standard evaluation protocol to compare our method with other methods for scene text detection, including the top performers on the ICDAR competitions. There are three important metrics in performance assessment: precision, recall and F-measure. Precision measures the ratio between true positives and all detections, while recall measures the ratio between true positives and all true texts that should be detected. F-measure is a harmonic mean of precision and recall, which is the single indicator of algorithm performance.

The evaluation method used in ICDAR 2011 was originally proposed by Wolf et al. [27]. The protocol of Wolf et al. presents recall and precision based on some constraints on detection quality and considers three matching cases: one-to-one, one-to-many and many-to-one. One-to-one matching means that a single ground truth bounding box is matched to a single detected bounding box. One-to-many matching means that a single ground truth bounding box is matched to multiple detected bounding boxes. Many-to-one matching means that multiple ground truth bounding boxes are matched to a single detected bounding box. The definition of recall, precision, and F-measure are following:

$$precision(G,D,t_r,t_p)=\frac{\sum_j Match_D(D_j,G,t_r,t_p)}{|D|} \quad (5)$$

$$recall(G,D,t_r,t_p)=\frac{\sum_i Match_G(G_i,D,t_r,t_p)}{|G|} \quad (6)$$

$$F_{measure}=2\times\frac{precision\times recall}{precision+recall} \quad (7)$$

Where G and D represent ground truth rectangle set and detection rectangle set. $t_r\in[0,1]$ is the constraint on area recall and $t_p\in[0,1]$ is the constraint on area precision. The standard values of them are set to 0.8 and 0.4, respectively. $Match_D$ and $Match_G$ are the scores which take different types of matching cases. The evaluation protocol for ICDAR2013 is similar with that of ICDAR2011, expect for a number of heuristics cues. For more details, please refer to [26].

After word construction, 949 word candidates and 151 non-texts are generated. As shown in **Table 4**, 926 word candidates and 118 non-texts are correctly classified by the CNN classifier, 23 word candidates and 33 non-texts are wrongly classified by the CNN classifier. We can see that 78.2% of the non-texts are removed, while 97.6% word candidates are still retained. The CNN classifier improves the precision of text detection, although it reduces the recall of text detection.

Table 4. Performance of text verification based on CNN model

		Number	classification Rate
Text	Correct classification	926	926/949=0.976
	Incorrect classification	23	
Non-text	Correct classification	118	118/151=0.782
	Incorrect classification	33	

Table 5. Performance of different algorithms evaluated on the ICDAR2011 dataset (%)

Algorithm	Precision	Recall	F-measure
Proposed	89.91	73.86	81.10
Tian et al. [28]	86.24	76.17	80.89
Zhang et al. [29]	84.00	76.00	80.00
Huang et al. [17]	88.00	71.00	78.00
Zamberletti et al. [30]	86.00	70.00	77.00
Yin et al. [2]	86.29	68.26	76.22
Neumann and Matas [1]	85.40	67.50	75.40
Yao et al. [31]	82.20	65.70	73.00
Kim et al. [25]	82.98	62.47	71.28

The performance of the proposed approach as well as other methods on ICDAR2011 dataset are described in **Table 5**, our method achieves the precision, recall and F-measure of 89.91%, 73.86%, and 81.10%, respectively. We estimate the running time of our method with the testing set of ICDAR 2011 dataset. The speed of Neumann and Matas' method [14] (including text localization and recognition) is 1.8s per image on a standard PC. The average processing time of Tian et al. [28] is 1.4s per image. The average runtime of Zhang et al. [29] is 30s per image and the speed of Shi et al.'s method [32] on a PC with a 2.33GHZ processor is 1.5s per image. In comparison, the total running time for the proposed method is 1.6s per image. The connected component extraction takes the most time in our method. It consumes about 0.9s for one image. The time for the connected component extraction is increased due to the SWT algorithm is applied both bright text on dark background and dark text on bright background.

Table 6. Performance of different algorithms evaluated on the ICDAR2013 dataset (%)

Algorithm	Precision	Recall	F-measure
Proposed	89.63	73.67	80.87
Tian et al. [28]	85.15	75.89	80.25
Zhang et al. [29]	88.00	74.00	80.00
Lu et al. [33]	89.22	69.58	78.19
Zamberletti et.al [30]	85.61	70.01	77.03
Yin et.al [2]	88.47	66.45	75.89
Neumann and Matas [14]	87.51	64.84	74.49
Text Detector CASIA [32]	85.00	63.00	72.00

On ICDAR 2013 dataset, our method achieves the precision, recall and F-measure of 89.63%, 73.67% and 80.87%, respectively. As shown in **Table 6**, the winning algorithm in the ICDAR Robust Reading Competition 2013 reports a F-measure of 75.89%, while our approach obtains 80.87%. As on ICDAR 2011 dataset, the proposed approach also achieves state-of-the-art performance.

As shown in **Table 5** and **Table 6**, the proposed approach obtains similar results on ICDAR2011 dataset and ICDAR2013 dataset. It outperforms state-of the-art techniques clearly in precision, recall, and F-measure. It confirms that the effectiveness of our algorithm, especially its advantage in handling various challenging scenarios.

Fig. 10 shows some successful examples. Some words are detected correctly in some variations. As we observe, the proposed technique is robust against severe conditions, such as stroke variation, no strong edge, and partially occluded **Fig. 11** shows some failure examples. It cannot deal with some challenging cases (for example, dot matrix fonts, low resolution), because our algorithm is one of connected component-based methods.

Fig. 10. Successful cases of the proposed method

Fig. 11. Failure cases of the proposed method

The good performance is largely due to the overall structure of our proposed technique. The scene text detection approach follows four sequential steps comprising connected component extraction, connected component filtration, potential character extraction, word generation and verification. The errors, which occur in the current step, will be resolved in the next step. The non-character components extracted in the first step can be removed by the character classifier in the second step. The characters that are not extracted in the first step are correctly extracted in the third step. In the same way, the ones that are not wrongly classified in the second step also can be extracted. In addition, the CNN employed in the stage of word verification gives a high performance of text detection.

5. Conclusion

A novel algorithm is proposed to detect text in natural scene images. After carefully analyzing the results of all the test images, our algorithm achieves better performance than others due to the following factors: (1) A SCE refiner, which is extended from the original SWT approach by considering stroke color, is used for character detection. It refines characters and rejects non-character components in order to improve the recall and precision of character detection. (2) A character classifier is trained by using gradient features to

identify connected components, leading to improved precision of character detection, even if a few characters are wrongly classified. (3) A CCT extractor can integrate character color with geometry features. It is proposed to extract potential characters, which improves the number of detected characters. (4) A CNN model is used to verify word candidates. Some non-text regions are pruned correctly and words are retained as many as possible. The experimental results demonstrate that our algorithm outperforms other competing methods on publicly available datasets.

We empirically analyze several main limitations of our technology for further research, which are also open issues in the literature. First, some multilingual texts have quite different characteristics from English texts. So, how to detect multilingual texts simultaneously is slightly more challenging. Second, how to handle texts of varying orientations is also a challenge, especially for similar multiple text lines with a seriously skewed distortion. Third, how to detect some highly blurred texts under complex background needs to be further investigated. Moreover, we should extend the proposed method to an end-to-end text recognition system.

References

[1] L. Neumann and J. Matas, "On combining multiple segmentations in scene text recognition," in *Proc. of 12th International Conference on Document Analysis and Recognition*, pp.523-527, August, 2013. Article (CrossRef Link)

[2] 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, September, 2014. Article (CrossRef Link)

[3] 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, March, 2011. Article (CrossRef Link)

[4] C. Yao, X. Zhang, X. Bai, W. Liu, Y. Ma and Z. Tu, "Rotation-invariant features for multi-oriented text detection in natural images," *PLOS One*, vol.8, no.8, August, 2013. Article (CrossRef Link)

[5] X. Chen and A.L. Yuille, "Detecting and reading text in natural scenes," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp.366-373, June 27 - July 2, 2004. Article (CrossRef Link)

[6] K. Wang and S. Belongie, "Word spotting in the wild," in *Proc. of 11th European Conference on Computer Vision*, pp.591-604, September 5-11, 2010. Article (CrossRef Link)

[7] L. Neumann and J. Matas. "Scene text localization and recognition with oriented stroke detection," in *Proc. of IEEE International Conference on Computer Vision*, pp.97-104, December 1-8, 2013. Article (CrossRef Link)

[8] B. Epshtein, E. Ofek and Y.Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp.2963-2970, June 13-18, 2010. Article (CrossRef Link)

[9] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Proc. of Asian Conference on Computer Vision*, pp.770-783, November 8-12, 2010. Article (CrossRef Link)

[10] W.-L. Huang, Z. Lin, J. Yang and J. Wang, "Text localization in natural images using stroke feature transform and text covariance descriptors," in *Proc. of IEEE International Conference on Computer Vision*, pp.1241-1248, December 1-8, 2013. Article (CrossRef Link)

[11] C.-L. Liu, K. Nakashima, H. Sako and H. Fujisawa, "Handwritten digit recognition: investigation of normalization and feature extraction techniques," *Pattern Recognition*, vol.37, no.2, pp.265-279, February, 2004. Article (CrossRef Link)

[12] H. Zhang, K. Zhao, Y.-Z. Song and J. Guo, "Text extraction from natural scene image: A survey," *Neurocomputing*, vol.122, no.51, pp.310–323, December, 2013. Article (CrossRef Link)

[13] L. Neumann and J. Matas, "Text localization in real-world images using efficiently pruned exhaustive search," in *Proc. of International Conference on Document Analysis and Recognition*, pp.687-691, September 18-21, 2011. Article (CrossRef Link)

[14] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp.3538-3545, June 16-21, 2012. Article (CrossRef Link)

[15] L. Sun and Q. Huo, "A component-tree based method for user-intention guided text extraction," in *Proc. of 21th International Conference on Pattern Recognition*, pp.633-636, November 11-15, 2012. Article (CrossRef Link)

[16] L. Sun and Q. Huo, "An improved component tree based approach to user intention guided text extraction from natural scene images," in *Proc. of 12th International Conference on Document Analysis and Recognition*, pp.383-387, August 25-28, 2013. Article (CrossRef Link)

[17] W.-L. Huang, Y. Qiao and X.-O. Tang, "Robust scene text detection with convolution neural network induced mser trees," in *Proc. of 13th European Conference on Computer Vision*, pp.497-511, September 6-12, 2014. Article (CrossRef Link)

[18] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. of Advances in Neural Information Processing Systems*, pp.1097-1105, December 3-8, 2012. Article (CrossRef Link)

[19] Y.-L. Cun, B.-E. Boser, J.-S. Denker, D. Henderson, R.-E. Howard and et al, "Handwritten digit recognition with a back-propagation network," in *Proc. of Advances in Neural Information Processing Systems*, pp.396-404, November 26-29, 1990. Article (CrossRef Link)

[20] C.- Y. Lee, S. Xie, P. Gallagher, Z. Zhang and Z. Tu, "Deeply-supervised nets," in *Proc. of 18th International Conference on Artificial Intelligence and Statistics*, pp.562-570, May 9-12, 2015. Article (CrossRef Link)

[21] T. Wang, D. J. Wu, A. Coates and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proc. of 21th International Conference on Pattern Recognition*, pp.3304-3308, November 11-15, 2012. Article (CrossRef Link)

[22] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *Proc. of 13th European Conference on Computer Vision*, pp.512-528, September 6-12, 2014. Article (CrossRef Link)

[23] C. Yi and Y. Tian, "Text string detection from natural scenes by structure based partition and grouping," *IEEE Transactions on Image Processing*, vol.20, no.9, pp.2594-2605, September, 2011. Article (CrossRef Link)

[24] C. Yao, X. Bai, W. Liu, Y. Ma and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp.1083-1090, June 16-21, 2012. Article (CrossRef Link)

[25] A. Shahab, F. Shafait and A. Dengel, "Icdar 2011 robust reading competition challenge 2: reading text in scene images," in *Proc. of International Conference on Document Analysis and Recognition*, pp.1491-1496, September 18-21, 2011. Article (CrossRef Link)

[26] D. Karatzas, F. Shafait, S. Uchida and M. Iwamura, "Icdar 2013 robust reading competition," in *Proc. of 12th International Conference on Document Analysis and Recognition*, pp.1484-1493, August 25-28, 2013. Article (CrossRef Link)

[27] 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, September, 2006. Article (CrossRef Link)

[28] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu and C. L. Tan, "Text flow: a unified text detection system in natural scene images," in *Proc. of IEEE International Conference on Computer Vision*, pp.4651-4659, December 7-13, 2015. Article (CrossRef Link)

[29] Z. Zhang, W. Shen, C. Yao and X. Bai, "Symmetry-based text line detection in natural scenes," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp.2558-2567, June 7-12, 2015. Article (CrossRef Link)

[30] A. Zamberletti, L. Noce and I. Gallo, "Text localization based on fast feature pyramids and multi-resolution maximally stable extremal regions," in *Proc. of Asian Conference on Computer Vision*, pp.91-105, November 1-5, 2015. Article (CrossRef Link)

[31] C. Yao, X. Bai and W. Liu, "A unified framework for multioriented text detection and recognition," *IEEE Transactions on Image Processing*, vol.23, no.11, pp.4737-4749, November, 2014. Article (CrossRef Link)

[32] 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, January, 2013. Article (CrossRef Link)

[33] S. Lu, T. Chen, S. Tian, J.-H. Lim and C.-L. Tan, "Scene text extraction based on edges and support vector regression," *International Journal on Document Analysis and Recognition*, vol.18, no 2, pp.125-135, June, 2015. Article (CrossRef Link)

Yang Zheng received the BS degree from LinYi university in 2009 and received the ME degree from Guilin University of Electronic Technology in 2012. He is currently pursuing a PhD at the University of Science and Technology Beijing. His current research interests include pattern recognition, image processing, and video analysis.

Jie Liu is now a research associate at Institute of Automation, Chinese Academy of Sciences(CASIA). He received the PhD degree in pattern recognition and intelligent systems from CASIA. His research interests include pattern recognition, deep learning, image processing and especially the applications to scence text detection and recognition.

Heping Liu is a professor and PhD supervisor in the School of Automation and Electrical Engineering, University of Science and Technology, Beijing. He received his BE degree from the University of Science and Technology Liaoning of China, Anshan, in 1977, and his ME and PhD degrees from Nagoya Institute of Technology, Japan, in 1988 and 1992, respectively. His major research interests include robust control and adaptive control.

Qing Li is a professor and PhD supervisor in the School of Automation and Electrical Engineering, University of Science and Technology, Beijing. He received his B.E., M.E. and PhD degrees from University of Science and Technology Beijing in 1993, 1996 and 2000, respectively. His major research interests include Intelligent Control Theory, Artificial intelligence and so on.

Gen Li received the BS degree from Huazhong University of Science and Technology in 2014. He is currently pursuing a master degree at Institute of Automation, Chinese Academy of Sciences. His current research insterests include pattern recognition, machine learning, deep learning, and scene text detection and recognition.