A Novel Text Sample Selection Model for Scene Text Detection via Bootstrap Learning

Jun Kong[1,2*], Jinhua Sun [1], Min Jiang[1], Jian Hou[1]
[1] Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi, China, 214122
[e-mail: kongjun@jiangnan.edu.cn]
[2] College of Electrical Engineering, Xinjiang University
Urumqi, China, 830047
*Corresponding author: Jun Kong

Received April 8, 2018; revised August 4, 2018; accepted September 1, 2018; published February 28, 2019

Abstract

Text detection has been a popular research topic in the field of computer vision. It is difficult for prevalent text detection algorithms to avoid the dependence on datasets. To overcome this problem, we proposed a novel unsupervised text detection algorithm inspired by bootstrap learning. Firstly, the text candidate in a novel form of superpixel is proposed to improve the text recall rate by image segmentation. Secondly, we propose a unique text sample selection model (TSSM) to extract text samples from the current image and eliminate database dependency. Specifically, to improve the precision of samples, we combine maximally stable extremal regions (MSERs) and the saliency map to generate sample reference maps with a double threshold scheme. Finally, a multiple kernel boosting method is developed to generate a strong text classifier by combining multiple single kernel SVMs based on the samples selected from TSSM. Experimental results on standard datasets demonstrate that our text detection method is robust to complex backgrounds and multilingual text and shows stable performance on different standard datasets.

Keywords: Text detection, bootstrap learning, image segmentation, text sample selection model

This work was partially supported by the National Natural Science Foundation of China (61362030, 61201429), China Postdoctoral Science Foundation (2015M581720, 2016M600360), Jiangsu Postdoctoral Science Foundation (1601216C), Scientific and Technological Aid Program of Xinjiang (2017E0279).

http://doi.org/10.3837/tiis.2019.02.016 **ISSN : 1976-7277**

1. Introduction

In recent years, much attention has been placed on text detection technology due to the growing demand for applications. Text is one of the most important forms of communication in daily life. As a method of information exchange, it can be embedded into documents or scenes, which makes the documents and scenes more accessible and readable under complex backgrounds. According to the most of previous text detection algorithms [1, 2], text detection methods can be divided into two steps: text candidate extraction and text candidate classification. There are two typical limitations in those two steps, i.e., the low text recall rate and low generality.

Firstly, text candidates are extracted by utilizing the sequential filters to generate text candidates for prevalent text detection methods, which causes error accumulation and low text recall rate. Because of the sequential structure, the classification error will propagate continuously and the filtered text regions may never be retrieved. Those uncorrectable errors cause the low recall rate. Secondly, low generality is the main limitation of these algorithms that using labeled dataset to train a text classifier. It is difficult to extend the text algorithms to another dataset, because the tolerances of protocols are different in various datasets. Sometimes the superiority of the algorithm depends on the annotation precision of the dataset. In addition, most of the text localization algorithms can only handle a specific scenario due to the limitation of the training dataset, which requires manual annotation.

To solve those problems, we proposed a novel unsupervised text detection method inspired by bootstrap learning [3]. Bootstrap learning is a self-taught learning method similar to unsupervised learning. It represents the concept that extracting samples from current images instead of labeled datasets to make the algorithm unsupervised. Most of the existing methods extract text regions with the high confidence and lose the sight of the similar properties between the characters/words, which results in the low recall rate. In fact, the similar properties between cohesive characters, such as spatial location, size, color, and stroke width, provide more information than single character region. In order to get the high confidence, prevalent text detection methods ask large datasets for help. Bootstrap learning has achieved the state-of-art performance in the field of salient object detection [4]. In this paper, we propose a novel text detection method based on bootstrap learning to make full use of the similarity between characters and eliminate dependency on the dataset. Our algorithm can be divided into three steps: text candidate extraction, text sample selection, and text classifier training. In text candidate extraction stage, to avoid the low recall rate caused by sequential filters, superpixel is taken as the text candidates creatively. Based on similarities between feature spaces, such as color, texture, and intensity, superpixel groups the adjacent pixels into one region. Specifically, we develop the image segmentation method by utilizing simple linear iterative clustering (SLIC) and density-based spatial clustering of applications with noise (DBSCAN) to generate the superpixel candidates. In sample selection stage, we proposed a novel text sample selection model (TSSM) to extract text samples and the pending data (weak text samples) from current processing image instead of existing labeled datasets. Specifically, text samples and the pending data are extracted from sample reference maps by the double threshold scheme. The sample reference maps consist of strong text map and non-text map based on a combination of maximally stable extremal regions (MSERs) and saliency map. In classifier training stage, we generate a strong classifier by the method in [4, 5] that combines several single kernel with different features by a boosting algorithm.

Then we reclassify the pending data with the new trained strong classifier and fuse the result with the sample selection maps. **Fig. 1** shows the overall process of our proposed algorithm.

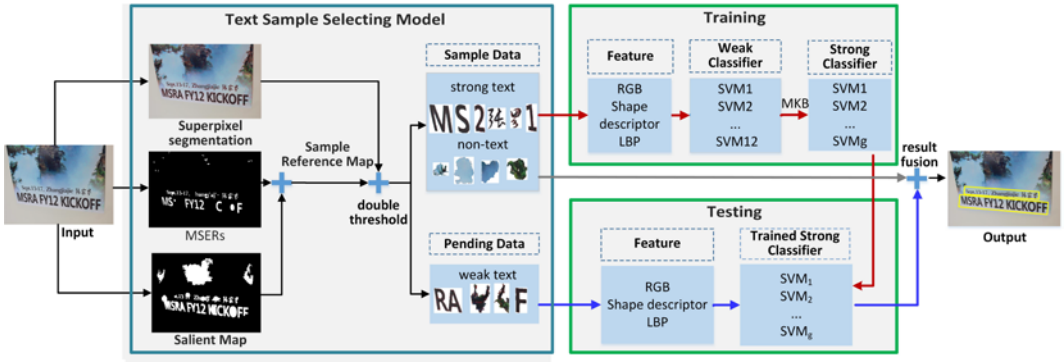


Fig. 1. Overall process of our text detection method via bootstrap learning

2. Related Work

In recent years, many text detection algorithms have been proposed. According to most previous text detection algorithms, such as [6-9], text detection methods can be divided into two main steps: text candidate extraction and text candidate classification.

In the first step, text candidate extraction is often conducted by the methods based on connected components (CCs) [10-15] or sliding window [16-20]. CCs based methods extract the regions with consistency of features or extreme areas. Stroke width transform (SWT) [10] and maximally stable extremal regions (MSERs) [13]are two representative techniques for CCs based methods. SWT offers edge map to obtain information about the text stroke in efficient way. The limitation of this method is that there are too many manual parameters. MSERs draw intensity stable regions as text candidates. Recently, several methods of MSERs refinement were proposed in [21, 22], which efficiently improve the precision for oriented text. The method in [23] refines and groups MSERs by geometric constraints. The lower recall rate and excessive number of manual parameters are the main disadvantages of this type of method. The sliding window based methods [16-19] detect the texts in a given scene image by shifting a window over all locations in multiple scales. But the exhaustive search leads to the unavoidable computation cost and quite a bit of false positives.

In the second step, many algorithms train a text classifier for further text classification with existing datasets. [24-26] adopt SVM and random forest as text classifiers. The raw pixel intensity is adopted with a pixel level text classifier SVM in [26]. The main difficulty of SVM classifier is the selection of the kernel function. Particularly, it is much more complex when the datasets contain thousands of diverse images with different properties.

In recent years, text detection work is gradually influenced by deep neural network. Those text detection algorithms can be divided into three categories. The first is based on the image segmentation [27]. The text area is extracted by the Text-Block FCN of segmentation, and then some post-processing methods are adopted to obtain the text bounding box. The second is based on the candidate box [28]. It uses the FCRN to detect text boundary boxes directly. The last is the mixed method [29], which adopts Faster-RCNN for multi-task learning and combines the methods of segmentation and boundary detection. They have achieved competitive performance in the field of text detection. However, overreliance on datasets is a more fatal flaw for these algorithms compared to traditional machine learning algorithms.

In this paper, we proposed a novel unsupervised text detection method inspired by bootstrap learning. Bootstrap learning represents the concept that extracting samples from the current image instead of labeled datasets to make the algorithm unsupervised. For get better samples in the current unlabeled image, we observe that text always appears on the salient object and sometimes the text can be regarded as the salient object in scene images. Hence, salient object detection is used to improve the precision of candidates extracted from MSERs. Specifically, the salient detection method promotes a holistic perspective, while MSERs pay more attention to the text details. Then, sample reference maps including strong text map and non-text map are generated by TSSM and inherit both of the advantages of MSERs and saliency map. Finally, we divide all the superpixels into three groups： strong text, non-text and weak text group by the double thresholds. The superpixels in strong text and non-text groups can be regarded as positive and negative samples.

Since the samples are selected using the TSSM with high confidence in the current image, the samples can be used to train an elegant classifier. To solve the problem of how to choose an optimal kernel for SVM, we adopt a multiple kernel boosting method with four different types of kernels, including linear, polynomial, RBF, and sigmoid, to generate the strong text classifier [4]. The multi-kernel method searches for the best kernel and the optimal combination of kernels and features for a specific detection project. The optimal weak classifier is selected at each iteration of boosting automatically. Therefore, the strong classifier is adaptive to the specific image. Our novel text detection algorithm is shown in the following sections.

3. Text Candidate Extraction

3.1 Image Segmentation

It is difficult for traditional candidate extraction methods to strike a balance between the amount and precision of the text candidates. In our work, superpixel is taken as our text candidates and the output of TSSM. It is grouped by using the similarities of adjacent pixels. Superpixel captures the structural characteristics of images and reduces the complexity of image processing effectively.

Typically, the SLIC is the most commonly used segmentation method to obtain the superpixels because of its efficiency. However, the main limitation of this segmentation method is that the cohesive regions are divided. In order to get the superpixels with characteristic consistency, the pixels that belong to the same character or word should be divided into the same superpixel. Therefore, we obtain the advanced superpixels by using DBSCAN after the image is segmented by SLIC. DBSCAN find the regions of any shapes without too much parameters in a convenient way.

As shown in **Fig. 2(a)** and **Fig. 2(c)**, images are segmented into $M0$ original superpixels by SLIC. These original superpixels distribute evenly in size and shape. The uniform background and the integrated character are divided into several superpixels, which reduces the feature discrimination of the superpixel. As shown in **Fig. 2(b)** and **Fig. 2(d)**, there are M advanced superpixels after clustering the original superpixels by DBSCAN. Those advanced superpixels are represented as $\{SP_i\}, i = 1, \ldots, M$. As we can see, the number of advanced superpixels M is much smaller than the number of original superpixels $M0$. Therefore, superpixels cover the whole images without information loss when the number of candidates is kept within a small range. This is the most contribution of the superpixel, which leads to a high recall rate and robust performance on diverse datasets.

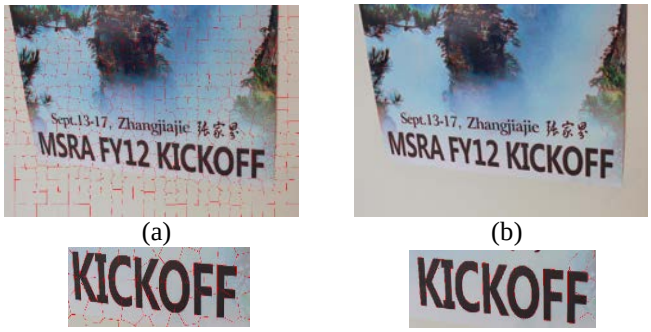


Fig. 2. (a) and (c) show the segmented original superpixels by SLIC. (b) and (d) show the clustered advanced superpixels by DBSCAN.

3.2 Image Features

In this paper, most of the features are generic while there are some differences between the sample selection stage and the classifier training stage.

In the sample selection stage, three feature descriptors, RGB, CLElab and local binary pattern (LBP) [30] are used to generate the saliency map. RGB feature descriptors provide complementary color features, which is lacking in the MSERs extracting process. Compared with the RGB color space, CLElab is a device-independent color system based on physiological characteristics. This means that it is a digital way to describe human visual induction. The LBP operator provides the texture information from an image, which is also very helpful for distinguishing text/non-text regions in the following steps. We employ LBP features within the 3×3 domain. In order to avoid the redundancy of binary modes, the statistical property is improved by employing the uniform pattern [30] that reduces the pixel value between 0 and 58. Each of the superpixels constructs an LBP histogram that is represented as $\{h_j\}$, $j = 1,2,\ldots,59$, where h_j is the value of the j-th bin in LBP pattern.

In the classifier training stage, we use the extent of each superpixels as a kind of shape descriptor feature to replace the CLElab, which is the ratio of pixels in the superpixel to pixels in the minimum bounding box. That is because the extent of each superpixel can reflect the shape uniqueness of the text. Note that all pixel-level features should be transformed into the local feature of the unit of the superpixels, because the image is segmented by the superpixels. In this paper, we obtain each superpixel feature value by calculating the average feature value of pixels in the corresponding superpixel. The use of local features makes the features more discriminatory, and the local feature is more stable by using the mean value to calculate the local feature.

4. Text Sample Selection

4.1 Text Sample Selection Model

In this paper, to obtain text samples, we propose the text sample selection model (TSSM). There are three main steps in our TSSM. In the first step, we generate sample reference map by MSER and saliency map. Then we calculate the proportions of text pixel and non-text pixels in a superpixel by the sample reference map. Finally, we divide all the superpixels into three groups：strong text, non-text and weak text group by the double thresholds. The superpixels in strong text and non-text group are regarded as positive and negative samples. The labeled superpixel samples are the output of TSSM.

4.1.1 MSERs

The maximally stable extremal regions (MSERs) is a local affine invariant feature proposed by [31]. An extremal region is a group of pixels that are connected and whose intensities are quite close.

$$q(i) = \frac{|Q_{i+\Delta} - Q_i|}{|Q_i|} \quad (1)$$

where i denotes the threshold value. $|Q|$ denotes the number of pixels in Q. Δ is the variance of intensity. $q(i)$ is the variance of the connected region.

The MSER algorithm looks for a series of thresholds, so the output of the detector is not a binary image but a series of nested regions. In order to obtain text regions with higher confidence, we employ a variety of geometric constraints to filter candidate MSERs. To achieve better performances, we also adopt the stroke width transform (SWT) for filtering [10]. As shown in **Fig. 3,** the strict geometric filter [21] of MSERs results in the lower text recall and higher precision. In this paper, the survival MSERs are not the final outputs but as the sample candidates, which is different from the classic methods. In order to improve the accuracy of sample, we combine MSERs with saliency map.

Fig. 3. Detected MSERs in different images

4.1.2 Saliency Map

As shown in **Fig. 4**, image salient detection is used to locate and extract the regions that typically attract attention in images.

The center-bias prior and the dark channel prior are combined to get a robust saliency map. Because the dark channels [32] are mostly produced by colored or dark object and shadows, the intensity value of an object should be relatively low .These characteristics are exactly what the objects have. For pixel p, the dark channel prior $S_d(p)$ is calculated by the following formula:

$$S_d(p) = 1 - min_{q \in b(p)}\left(min_{ch \in \{r,g,b\}} In^{ch}(q)\right) \quad (2)$$

where $b(p)$ represents the 5×5 image block with the center pixel p. $In^{ch}(q)$ is the color value of pixel q on the corresponding color channel ch. Note that all the color values are normalized into [0, 1]. $S_d(p)$ represents the possibility of $b(p)$ being a salient object.

Images are segmented into M final superpixels, $\{SP_i\}, i = 1, \ldots, M$. The pixels around the image boundary are viewed as backgrounds, $\{SP_j^b\}, j = 1, \ldots, M_b$, where M_b represents the number of background superpixels. We use the following formula to calculate the dark channel prior of each superpixel:

$$S_d(SP_i) = \frac{1}{|SP_i|}\sum_{p \in SP_i} S_d(p) \quad (3)$$

where $|SP_i|$ is the number of pixels in SP_i.

We use the following formula to calculate the coarse saliency value for each superpixel:

$$S_0(SP_i) = E(SP_i) \times S_d(SP_i) \times \sum_{K\in\{F_1,F_2,F_3\}}\left(\frac{1}{M_b}\sum_{j=1}^{M_b} d_K(SP_i, SP_j^b)\right) \quad (4)$$

where K is the space of the features including the LBP ($F1$) texture features, color feature RGB ($F2$) and CIELab ($F3$). $d_k(SP_i, SP_j^b)$ is the Euclidean distance between region SP_i and SP_j^b in all the feature space. It is worth mentioning that $d_k(SP_i, SP_j^b)$ and $E(SP_i)$ should be normalized into [0, 1]. $E(SP_i)$ shows the distance between the image center and the center of the SP_i, which is computed based on the center prior [33]. We assign a superpixel saliency value to each inner pixel to obtain a pixel level saliency map $salmap0$.

Fig. 4. Salient object detection

We use the graph cut method [34, 35] to filter images. To get the foreground map $salmap$, we use the max-flow [36] method to minimize the cut cost and predict the possibility of each pixel being foreground. $salmap$ is a binary final saliency map shown in **Fig. 5**.

(a) (b)

Fig. 5. (a) The original saliency map $salmap0$ without graph cut.
(b) The final saliency map $salmap$ with graph cut.

4.1.3 Sample Reference Maps

Based on the hypothesis that the text always appears on the saliency target and sometimes text is the saliency object, TSSM generates sample reference maps including strong text map and non-text map by the combination of MSERs and saliency map. As shown in **Fig. 6(a)**, MSERs pay more attention to the character details and provide amounts of reliable text regions. It can be seen that MSER algorithm discard some character areas, such as 'R', 'A', 'K', 'I' and 'F'. As shown in **Fig. 6(b)**, the salient object detection method finds almost all text regions in a holistic way. It also can be seen that a large number of non-text areas are detected. Obviously, the MSERs and the saliency map are complementary in text precision and recall rate. In order to find the connection between them, we use the following formula to combine:

$$Sm = mser \cap salmap \quad (5)$$

$$Nm = C_{\cup}(mser \cup salmap) \quad (6)$$

where, Sm is the matrix of the strong text map and Nm is the matrix of the non-text map. C represents the entire image. $mser$ and $salmap$ are the MSERs and the saliency map. As we can see from **Fig. 6(c)**, the regions in white are considered as text regions with high confidence and the regions in black represent non-text regions with a certain probability. Our strong text map is more accurate than MSERs after most of the non-text regions are filtered by Eq. (5). As shown in **Fig. 6(d)**, the regions in white are considered as non-text regions with high confidence and the regions in black represent text regions with a certain probability. The ground truth text regions, which have been grouped into non-text region in **Fig. 6(b),** are excluded in **Fig. 6(d)**, such as 'S', 'e', and 'p'. Therefore, our non-text map is more accurate than general saliency map in non-text detection.

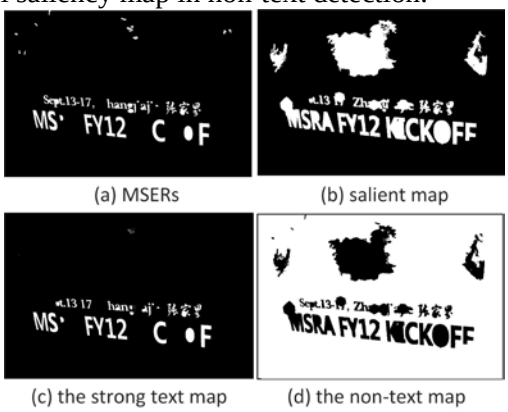


Fig. 6. The combination of MSERs and saliency map

4.1.4 Superpixel Sample Selection

Based on pixel-level sample reference maps, we propose a novel double threshold scheme to obtain the superpixel-level samples.

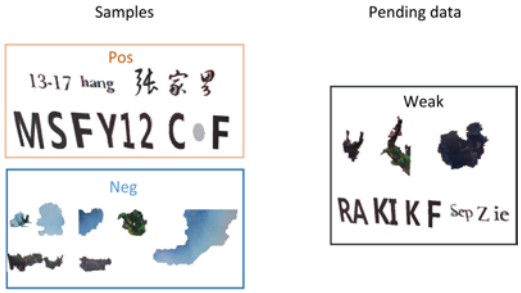


Fig. 7. The results of superpixel sample selection. (The superpixels in orange box are strong texts and the superpixels in blue box are non-texts. The superpixels in black box are weak texts.)

Generally, most traditional algorithms divide the text candidate into text and non-text groups directly. The classification is relatively rough, and it is easy to produce classification error. Therefore, TSSM adopts the double threshold scheme that divides all the superpixel candidates into strong text (labeled with '+1'), non-text (labeled with '-1'), and weak text. The classification is determined by probability of candidate being text region. The strong text and non-text superpixels are included in samples and the weak text superpixels are regarded as the pending data. Pending data allows the first classification a reasonable degree of tolerance. The double threshold scheme is shown as following:

$$l(SP_i) = \begin{cases} strong\ text\ , & P_1(SP_i) > thr_1 \\ non-text\ , & P_2(SP_i) > thr_2 \\ weak\ text\ , & otherwise \end{cases} \quad (7)$$

$$P_1(SP_i) = \frac{|SP_i^p|}{|SP_i|} \quad , \quad P_2(SP_i) = \frac{|SP_i^n|}{|SP_i|} \quad (8)$$

Where SP_i is the advanced superpixels introduced in Sec.3.1. $|SP_i^p|$ and $|SP_i^n|$ are the number of positive pixels and negative pixels in SP_i calculated by the text strong map and non-text map, respectively. $|SP_i|$ represents the number of pixels in SP_i. P_1 and P_2 represent the possibility of SP_i being positive and negative samples. The samples are represented as $\{r_i, l_i\}_{i=1}^{H}$, where r_i represents the *i*-th sample, l_i represents the label value of the sample, and H represents the number of samples. In particular, H^p represents the number of positive superpixels and H^n shows the number of negative superpixels. The selection of threshold values thr_1 and thr_2 are shown in section 7.2.2. After all the superpixels are labeled, all the samples are put into the multiple kernel boosting classifier for training. The weak text is reclassified by the strong classifier. Our purpose is to train a classifier to classify the weak text superpixels into the strong text superpixel or the non-text superpixel utilizing the similar properties between cohesive characters in the same image.

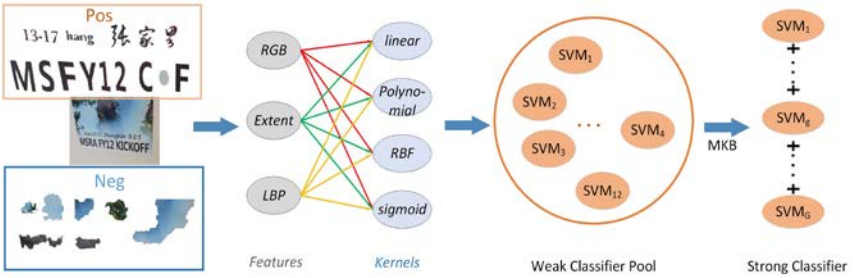


Fig. 8. The illustration of multiple kernel boosting process

5. Text Classifier Training

5.1 Text Classifier

In our algorithm, we extract text samples from current image and then classify the pending data into text regions or non-text regions. It's a classic binary classification problem, so we adopt SVM as classifier. The main difficulty of SVM classification is the selection of the kernel function. In particular, it is more complex when the datasets contain thousands of diverse images with different properties. We adopt multiple kernel boosting (MKB) method [4] with four kinds of different kernels including linear, polynomial, RBF, and sigmoid. SVM classifier with a single kernel and a single feature is used as a weak classifier, and the strong classifier is obtained by the iterative learning weak classifier with boosting algorithm.

MKB is an improved algorithm based on multiple kernel [37]. For arbitrary input pictures, we obtain the samples according to TSSM and then train the strong classifier. These SVM kernels $\{k_n\}_{n=1}^{N}$ are combined in the following manner:

$$\mathrm{k}(\mathrm{r}, r_i) = \sum_{n=1}^{N} \beta_n\, k_n(\mathrm{r}, r_i),\ \sum_{n=1}^{N} \beta_n = 1,\ \beta_n \epsilon R_+ \quad (9)$$

where β_n is the weight of the *n*-th kernel. N is the number of weak classifiers. As shown in **Fig. 8**, $N = N_f \times N_k, N_f = 3$ is the number of features and $N_k = 4$ is the number of kernels.

For distinguishing samples, the combination is changed as follows:

$$Y_r = \sum_{n=1}^{N} \beta_n \sum_{i=1}^{H} \alpha_i l_i \, k_n(\mathrm{r}, r_i) + \bar{b} \tag{10}$$

where α_i is the Lagrange multiplier and $\bar{b}$ is the bias in the standard SVM algorithm. The parameters$\{\alpha_i\}$, $\{\beta_n\}$ and $\bar{b}$ can be learned from a joint optimization process. Note that Eq. (10) is a very common SVM algorithm on multiple kernels. In this article, we replace the simple combination of the weak classifier with the boosting algorithm to optimize a strong classifier by adaptively changing parameters. Therefore, Eq. (10) can be rewritten as:

$$Y_r = \sum_{n=1}^{N} \beta_n ((\alpha^T) k_n(\mathrm{r}) + \bar{b}_n) \tag{11}$$

where $\alpha = [\alpha_1 l_1, \alpha_2 l_2, \ldots, \alpha_H l_H]^T$, $k_n(\mathrm{r}) = [k_n(\mathrm{r}, r_1), k_n(\mathrm{r}, r_2), \ldots, k_n(\mathrm{r}, r_H)]^T$, $\bar{b} = \sum_{n=1}^{N} \bar{b}_n$. By setting the decision function of a single-kernel SVM as $Z_n(r) = (\alpha^T) k_n(\mathrm{r}) + \bar{b}_n$, the parameters can be learned directly. Thus, Eq. (11) can be rewritten as:

$$Y_r = \sum_{g=1}^{G} \beta_g Z_g(r) \tag{12}$$

where G is used to record the number of iterations of the boosting algorithm. A single kernel SVM is considered as a weak classifier, while a strong classifier Y_r is a combination of each weak classifier according to the weight. After G iterations, we obtain a strong classifier Eq. (12) based on the samples selected by TSSM. Note that all of the samples are from current unlabeled image, which is the main idea of bootstrap learning. The illustration of multiple kernel boosting process is shown in **Fig. 8**.

Fig. 9. The image fusion process

Fig. 10. (a) Text candidates without grouping. (b) Text detection result with grouping.

5.2 Result Fusion

After the strong classifier is generated, all the weak text superpixels are put into this strong classifier and generate the preliminary result map shown in **Fig. 9(a)**. In fact, all superpixels can be put into the strong classifier in the meantime, which means that the previous samples are reclassified. This is less operationally complex, but it reduces the accuracy of classification to reclassify superpixels with such credible labels generated by TSSM. In order to present the image text location results as a whole, we need to integrate the preliminary output image of indeterminate weak text superpixels with the previous sample selection maps. We use the following formula to fuse:

$$Fm = min(\sim Nm, \; max(Sm, Pm)) \tag{13}$$

where Pm is the matrix of the preliminary output map. $\sim Nm$ is the anti-matrix of the non-text map. Fm is the fused image result map as shown in **Fig. 9(b)**. As we can see, the

fused image is a gray scale map with a number of background areas. As in the previous section 4.2, we also use the graph cut method to segment the text from the background as shown in **Fig. 9(c)**. As seen from **Fig. 9(c)**, there are still several false positives in the graph cut image. These false positives are discarded in the following text grouping step.

5.3 Text Grouping

Some of the credible text regions are already found in words or characters. In fact, the required form of output is different in different datasets. Usually, the form of output is either word-level or sentence-level. The MSRA-TD500 dataset takes a sentence-level text result of the evaluation, as the text in this dataset is mixed with English and Chinese and a sentence can provide much more reliable information than individual words in the Chinese reading.

Fortunately, one of the advantages of our algorithm is that it is easy to group text. In the first place, as many credible text regions as possible are found as shown in **Fig 10(a).** Second, because multiple types of features are extracted in the stage of saliency map generation and strong classifier training, we can use the same features for grouping directly. Minority false positives, which are isolated in the graph cut image, are removed before grouping. Text regions are grouped into one sentence by comparing the spatial locations and the feature properties. We adopt the minimum-area encasing rectangle [38] method to provide the compact bounding box as the output, which is different from previous works [11, 22, 39]. As shown in **Fig. 10(a)**, the outputs are at the word level without grouping, **Fig. 10(b)** is the grouping result at the sentence level.

6. Bootstrap Learning Algorithm of TSSM

An overview of our proposed method is summarized in **Algorithm 1.**

Algorithm 1

Input:
Current image
Output:
Estimated text location result map Fm.

Step1: superpixel segmentation
1: Generate $M0$ original superpixels by SLIC.
2: Generate M final superpixels SP_i by clustering the original superpixels using DBSCAN.
Step2: Text sample selection
1: Generate $mser$ map and $salmap$ by MSER and salient object detection method.
2: generate the text reference maps Sm and Nm via Eq. (5) and Eq. (6).
3: Compute the possibilities $P_1(SP_i)$ and $P_2(SP_i)$ of SP_i being positive and negative samples via Eq. (8) and compare them with thr_1 and thr_2 via Eq. (7), then get the positive superpixel samples H^p, the negative samples H^n and the pending data.
Step3: Text classifier training
1: Train the strong classifier Y_r with H^p and H^n by MKB method via Eq. (9) to Eq. (12)
2: Generate the preliminary output map Pm by inputing pending data into Y_r.
Step4: Get fused and grouped image result map Fm.

7. Experiment and Analysis

7.1 Experiment Result

7.1.1 Datasets

We conducted our experiments on three public datasets, namely, MSRA-TD500 [11], the oriented scene text dataset (OSTD) [15] and the ICDAR 2013 dataset [40].

MSRA-TD500 has a total of 500 pictures of natural scenes. There are 200 pictures for testing and 300 pictures for training. This database is oriented to multi-orientation and multilingual text, especially for Chinese and English mixed text. And most of the text is on the cards, which is consistent with our TSSM hypothesis that text always appears in the salient regions. This dataset includes the large variation in fronts, sizes and colors in the text, multilingual text with multi-orientation, and the complex and multiple backgrounds.

The OSTD is relatively small, with only 89 pictures in total. These pictures include indoor and outdoor scenes with multi-orientation text. In addition, the various changes in perspective, font, and style are another challenge for this dataset. For convenience of comparison, we use the same evaluation protocols as MSRA-TD500 to test on this dataset.

The ICDAR 2013 dataset is the most popular horizontal English text dataset. It contains 229 training images and 233 testing images. It was put forward in the ICDAR competitions in 2013. [40] introduces the evaluation method and there is an online system for evaluation.

7.1.2 Experiment Result

Unlike other datasets-based methods, our algorithm does not need to distinguish training or testing images. But in order to perform a fair comparison with other algorithms, the following data are from the images in the testing set. In addition, we compare them with some state-of-the-art algorithms based on those datasets. Our text detection algorithm results are shown in **Fig. 11**.

Fig. 11. The text detection result on above publicly available datasets: Our results are marked in yellow bounding

Table 1 shows the performance of our method on MSRA-TD500, which is evaluated based on the protocols from [11]. These include the overlap ratio and the angle between the estimated rectangle and the ground truth rectangle. Once the overlap ratio between them is greater than 0.5 and the angle is less than $\pi/8$, the estimated rectangle is regarded as the

correct detection. The final precision P and recall R are defined as: $P = |TP|/|E|$, $R = |TP|/|T|$, where TP is the set of true positive rectangles, E is the set of evaluation rectangles, and T is the set of ground truth rectangles. $|\cdot|$ represents the number of rectangles in the set. The F-score is defined as: F-score=$2P * R/(P + R)$. **Table 2** shows the performance and comparisons between our algorithm and other classic methods on OSTD. We adopt the same protocols as MSAR-TD500 to evaluate the performance on OSTD. In order to demonstrate the robustness of our algorithm, we also perform experiments on a horizontal text dataset ICDAR2013 shown in **Table 3**, which is evaluated by the protocols from [25] as well as an online system for evaluation.

Table 1. Performance comparison on MSRA-TD500.

Methods	P	R	F-score
[11]	0.63	0.63	0.60
[8]	0.71	0.62	0.66
[21]	0.68	0.83	0.75
[22]	0.81	0.63	0.71
[29]	0.77	0.70	0.74
[41]	0.72	0.79	0.75
[19]	0.85	0.78	0.81
ours	0.81	0.80	0.81

Table 2. Performance comparison on OSTD

Methods	P	R	F-score
[10]	0.37	0.32	0.32
[15]	0.56	0.64	0.55
[21]	0.67	0.79	0.73
[22]	0.69	0.79	0.74
[11]	0.77	0.73	0.76
ours	0.78	0.80	0.79

Table 3. Performance comparison on ICDAR 2013 dataset

Methods	P	R	F-score
[23]	0.85	0.63	0.72
[22]	0.84	0.65	0.73
[24]	0.88	0.65	0.74
[21]	0.72	0.78	0.77
[42]	0.86	0.70	0.77
[43]	0.89	0.70	0.78
[19]	0.85	0.76	0.80
[29]	0.92	0.81	0.86
[41]	0.87	0.84	0.86
ours	0.86	0.78	0.82

Because different datasets have various difficulties in different aspects, we chose an algorithm that performs well on a particular dataset to display that our algorithm have a robust performances on different datasets. And most of the methods are not open source. We cannot evaluate all the methods on all the 3 datasets. So in **Table 1** to **Table 3**, we list the best accuracy reported in the reference papers for a fair comparison. We can see from **Table 1** to **Table 3** that our algorithm is superior to most of the algorithms. Compared with the state-of-art traditional text detection algorithms [21, 22], which is mainly aim at dealing with Chinese and English mixed text, our algorithm has better performances on three datasets and greatly improves the R and F-score. Compared with the state-of-art methods [29, 41] based on convolutional neural networks, our algorithm have obvious advantages on the MSRA-TD500 dataset and the performances are quite comparable on the other datasets. Although, they have achieved competitive performances on ICDAR 2013 dataset, the overreliance on datasets is a more fatal flaw for these algorithms. Compared to other general methods [8, 10, 11, 15, 23], our algorithm has an obvious advantage in terms of text recall rate. We believe that this is because our algorithm has fewer hypotheses than other methods. The only hypothesis is that the text is a salient object or the text may appear on a saliency object. The experimental results show that our algorithm has quite stable and good performance on different datasets at the same time, while most of the other algorithms can only achieve good results on specific datasets. We believe that is the case because we extract training samples from current unlabeled image instead of existing datasets, which makes the training feature more distinguished.

7.2 Parameter Analysis

There are three main manual parameters of our algorithm: the number of superpixels M, double threshold values $thr1$ and $thr2$. They have great influences on the performance of the experiment and we will discuss them as following.

7.2.1 The number of superpixels M

The number of superpixels directly affects the image segmentation and candidate classification. Too many superpixels also cause the number of explosion and high computational costs. So the choice of M is very important. Therefore, we discussed the performance of text detection and the average processing of each image under different M to find the optimum value. As shown in **Fig. 12**, the experiment performance (blue lines) has an obvious increase when M is between 200 and 450. With the increase of M, the performance gradually and finally approaches to a steady state. We guess that may because DBSCAN cannot improve the performance of clustering any more when M reaches a certain level. The average processing time (green line) is in a stable and low state, when M is lower than 400. And the average processing time increases by geometric progression when M lager than 400. Therefore, M is taken as 400 in this paper under the consideration of performances and average processing time.

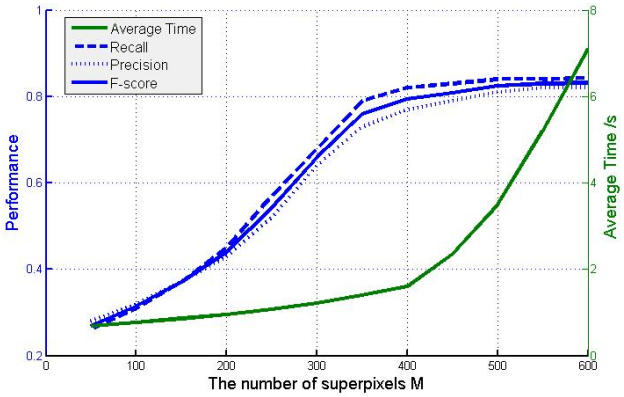
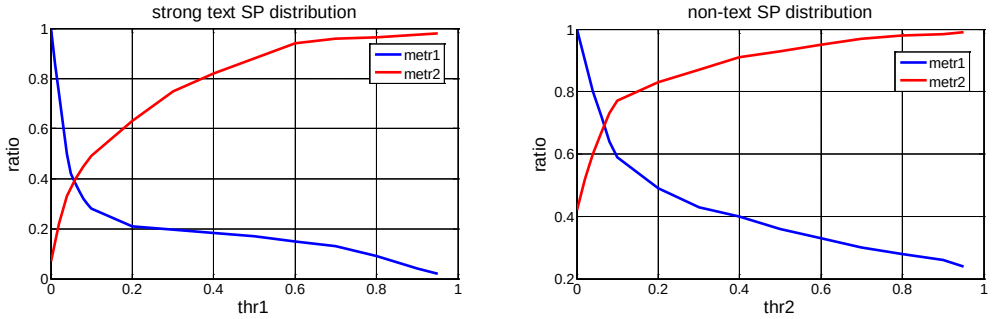


Fig. 12. The selection of M



(a) (b)

Fig. 13. Sample quality graphs

7.2.2 Double Threshold Value Selection

In order to examine the effect of thresholds on the quality of samples, we studied the distribution of samples under different thresholds. Two metrics are proposed to restrain the threshold value range jointly. These two criteria are defined by the following formulas:

$$metr1 = \frac{H^s}{M} \tag{14}$$

$$metr2 = \frac{H^s}{H^s_{gt}} \tag{15}$$

where H^s is the number of selected samples including H^p and H^n. H^p and H^n represents the number of positive and negative samples calculated by the strong text map and non-text map separately. H^s_{gt} is the ground truth number of samples including H^p_{gt} and H^n_{gt}, which represents the number of positive and negative samples. In the process of computing, H^s and H^s_{gt} must correspond to each other; i.e., they both belong to positive samples or negative samples. From the formulas above, we can observe that $metr1$ is the recall rate descriptor of samples and $metr2$ is the precision descriptor of samples. According to the two metrics, superpixel sample quality graphs including the strong text SP distribution and non-text SP distribution are plotted as **Fig. 13**.

As shown in **Fig. 13**, two metrics are in the opposite change direction with the growth of the threshold values. $metr1$ and $metr2$ shows the importance of the quantity and correct classification of samples. Therefore, in this paper, our thresholds $thr1$ and thr2 are selected to be 0.65 and 0.7 respectively.

8. Conclusion

In this paper, we proposed a novel robust scene text detection method based on TSSM inspired by bootstrap learning. TSSM selects samples from current image instead of the individual labeled dataset. Hence, our proposed text detection algorithm is significant for unsupervised detection projects. To improve the precision of samples, TSSM combines MSERs and saliency maps to generate sample reference maps, which provides the samples with high accuracy. In addition, the text candidate in a novel form of superpixel is proposed to improve the text recall rate and reduce the number of text candidates. Experimental results show that our algorithm obtain superior text recall and exhibit robust performance on different datasets. This is a completely new approach in the field of text detection, which is liberated from the existing datasets. It is of great value to apply bootstrap learning to text detection. In the future, we could extend the proposed method to an end-to-end system.

References

[1] Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: recent advances and future trends," *Frontiers of Computer Science*, vol. 10, pp. 19-36, February, 2016. Article (CrossRef Link)

[2] Q. Ye and D. Doermann, "Text detection and recognition in imagery: a survey," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, pp. 1480-1500, July, 2015. Article (CrossRef Link)

[3] J. Berry, I. Fasel, L. Fadiga, and D. Archangeli, "Training deep nets with imbalanced and unlabeled data," in *Proc. of 13th Annual Conference of the International Speech Communication Association 2012, INTERSPEECH 2012*, pp. 1754-1757, January, 2012. https://www.isca-speech.org/archive/interspeech_2012/i12_1756.html

[4] N. Tong, H. Lu, R. Xiang, and M. H. Yang, "Salient object detection via bootstrap learning," *Computer Vision and Pattern Recognition*, pp. 1884-1892, June, 2015. Article (CrossRef Link)

[5] F. Yang, H. Lu, and Y. W. Chen, "Human tracking by multiple kernel boosting with locality affinity constraints," in *Proc. of Computer Vision - ACCV 2010 - Asian Conference on Computer Vision, Queenstown, New Zealand, November 8-12, 2010, Revised Selected Papers*, pp. 39-50, November, 2010. Article (CrossRef Link)

[6] H. Cho, M. Sung, and B. Jun, "Canny text detector: fast and robust scene text localization algorithm," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3566-3573, June, 2016. Article (CrossRef Link)

[7] X. Bai, B. Shi, C. Zhang, X. Cai, and L. Qi, "Text/non-text image classification in the wild with convolutional neural networks," *Pattern Recognition*, vol. 66, pp. 437-446, June, 2017. Article (CrossRef Link)

[8] L. Kang, Y. Li, and D. Doermann, "Orientation robust text line detection in natural images," *Computer Vision and Pattern Recognition*, pp. 4034-4041, June, 2014. Article (CrossRef Link)

[9] V. K. Pham and G. S. Lee, "Robust text detection in natural scene images," in *Proc. of Australasian Joint Conference on Artificial Intelligence*, pp. 720-725, December, 2016. Article (CrossRef Link)

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

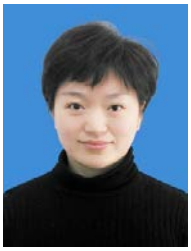
[11] Z. Tu, Y. Ma, W. Liu, X. Bai, and C. Yao, "Detecting texts of arbitrary orientations in natural images," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1083-1090, June, 2012. Article (CrossRef Link)

[12] W. 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, 2013. Article (CrossRef Link)
[13] H. Chen, S. S. Tsai, G. Schroth, and D. M. Chen, "Robust text detection in natural images with edge-enhanced Maximally Stable Extremal Regions," in *Proc. of IEEE International Conference on Image Processing*, pp. 2609-2612, September, 2011. Article (CrossRef Link)
[14] Y. Zheng, J. Liu, H. Liu, Q. Li, and G. Li, "Integrated method for text detection in natural scene images," *Ksii Transactions on Internet & Information Systems*, vol. 10, pp. 5583-5604, November, 2016. Article (CrossRef Link)
[15] C. Yi and Y. Tian, "Text string detection from natural scenes by structure-based partition and grouping," *IEEE Transactions on Image Processing*, vol. 20, pp. 2594-2605, March, 2011. Article (CrossRef Link)
[16] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 366-373, July, 2004. Article (CrossRef Link)
[17] L. Neumann and J. Matas, "Scene text localization and recognition with oriented stroke detection," in *Proc. of IEEE International Conference on Computer Vision IEEE Computer Society*, pp. 97-104, December, 2013. Article (CrossRef Link)
[18] S. M. Hanif and L. Prevost, "Text detection and localization in complex scene images using constrained adaboost algorithm," in *Proc. of International Conference on Document Analysis and Recognition*, pp. 1-5, July, 2009. Article (CrossRef Link)
[19] 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 2015 IEEE International Conference on Computer Vision* pp. 4651-4659, April, 2015. Article (CrossRef Link)
[20] J. Lee, J. S. Park, C. P. Hong, and Y. H. Seo, "Illumination-robust foreground extraction for text area detection in outdoor environment," *Ksii Transactions on Internet & Information Systems*, vol. 11, pp. 345-359, January, 2017. Article (CrossRef Link)
[21] X. C. Yin, X. Yin, K. Huang, and H. W. Hao, "Robust text detection in natural scene images," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 36, pp. 970-983, May, 2014. Article (CrossRef Link)
[22] X. C. Yin, W. Y. Pei, J. Zhang, and H. W. Hao, "Multi-orientation scene text detection with adaptive clustering," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, pp. 1930-1937, September, 2015. Article (CrossRef Link)
[23] 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, pp. 107-116, January, 2013. Article (CrossRef Link)
[24] 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, 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, 2011. Article (CrossRef Link)
[26] K. I. Kim, K. Jung, and J. H. Kim, "Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," *Pattern Analysis & Machine Intelligence IEEE Transactions on*, vol. 25, pp. 1631-1639, December, 2003. Article (CrossRef Link)
[27] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," *Computer Vision and Pattern Recognition*, pp. 4159-4167, June, 2016. Article (CrossRef Link)
[28] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," *Computer Vision and Pattern Recognition*, pp. 2315-2324, June, 2016. Article (CrossRef Link)

[29] W. He, X. Y. Zhang, F. Yin, and C. L. Liu, "Deep direct regression for multi-oriented scene text detection," in *Proc. of IEEE International Conference on Computer Vision*, pp. 745-753, October, 2017. Article (CrossRef Link)
[30] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 24, pp. 971-987, July, 2002. Article (CrossRef Link)
[31] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," in *Proc. of the British Machine Vision Conference 2002*, vol. 22, pp. 761-767, September, 2004. Article (CrossRef Link)
[32] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 33, pp. 2341-2353, December, 2011. Article (CrossRef Link)
[33] H. Jiang, J. Wang, Z. Yuan, T. Liu, N. Zheng, and S. Li, "Automatic salient object segmentation based on context and shape prior," in *Proc. of British Machine Vision Conference*, pp. 110.1-110.12, January, 2011. Article (CrossRef Link)
[34] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 23, pp. 1222-1239, November, 2001. Article (CrossRef Link)
[35] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?," in *Proc. of European Conference on Computer Vision*, pp. 65-81, April, 2002. Article (CrossRef Link)
[36] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 26, pp. 1124-1137, September, 2004. Article (CrossRef Link)
[37] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proc. of International Conference*, pp. 6-14, July, 2004. Article (CrossRef Link)
[38] H. Freeman and R. Shapira, "Determining the minimum-area encasing rectangle for an arbitrary closed curve," *Communications of the Acm*, vol. 18, pp. 409-413, July, 1975. Article (CrossRef Link)
[39] H. I. Koo and D. H. Kim, "Scene text detection via connected component clustering and nontext filtering," *IEEE Transactions on Image Processing*, vol. 22, pp. 2296-2305, June, 2013. Article (CrossRef Link)
[40] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. I. Bigorda, S. R. Mestre, *et al.*, "ICDAR 2013 robust reading competition," in *Proc. of International Conference on Document Analysis and Recognition*, pp. 1484-1493, August, 2013. Article (CrossRef Link)
[41] H. Turki, M. B. Halima, and A. M. Alimi, "Text detection based on MSER and CNN features," in *Proc. of Iapr International Conference on Document Analysis and Recognition*, pp. 949-954, January, 2018. Article (CrossRef Link)
[42] 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, April, 2014. Article (CrossRef Link)
[43] 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 & Recognition*, vol. 18, pp. 125-135, June, 2015. Article (CrossRef Link)

Jun Kong received the M.S. degree in pattern recognition and intelligent system from Institute of Intelligent Machines, Chinese Academy of Sciences in 2003, and PH.D. degree in electronic science and technology from Shanghai Institute of Technical Physics, Chinese Academy of Sciences in 2011. He joined Jiangnan University in 2004, where he is currently an associate professor and the assistant dean of the school of Internet of Things Engineering. He is the author of more than 30 journal papers and has co-published a book. Now he is presiding over ten research projects. His research interests include computer vision, image processing, target tracking and human action recognition. He is a member of CCF.

Jinhua Sun was born in 1993, and received her undergraduate diploma and bachelor's degree in 2016. She is now a graduate student of the school of Internet of Things Engineering at Jiangnan University. Her major is Computer Science and Technology, and her research interests include computer vision and scene text detection.



Min Jiang received the Ph.D. degree from the Institute of Plasma Physics, Chinese Academy of Sciences, China, in 2005. She is currently a Professor with the School of Internet of Things Engineering, Jiangnan University. Her primary research is in the area of machine learning and computer vision with broad applications, such as public surveillance, human–computer interaction, and biomechanics. She has received support from the National Natural Science Foundation of China, the Technology Research Project of the Ministry of Public Security of China. She is a member of IEEE.

Jian Hou was born in 1993, and received his undergraduate diploma and bachelor's degree in 2016. He is now a graduate student at the school of Internet of Things Engineering in Jiangnan University. His major is Electronic and Communication Engineering, and his research interests include computer vision and image processing.