2nd International Conference on Automatic Control and Information Engineering (ICACIE 2017)

Natural Scene Text Detection Based on Multi-Channel FASText

Guo Chenfeng
School of Printing and Packaging
Wuhan University
Wuhan, China
guochenfeng@whu.edu.cn

Liu Juhua*
School of Printing and Packaging
Wuhan University
Wuhan, China
liujuhua@whu.edu.cn

***Abstract*—In view of the complexity of background and the variety of text in natural scene images, a multi-channel FASText based text detection method for natural scene images is proposed. To detect more texts as much as possible, the character candidates are extracted by proposed multi-channel FASText algorithm from the R, G and B component image respectively. Then, texture features of the character candidates are extracted to train a random forest character classifier and the non-characters are eliminated. At last, the character regions are merged into text regions according to the color distance feature and geometric adjacency feature. The proposed approach on ICDAR 2013 dataset achieves 76.76%, 80.17%, and 78.43% in recall rate, precision rate and f-score respectively. Compared to other state-of-the-art methods, both the recall rate and f-score are improved. Experimental result shows that the proposed method is effective to natural scene text detection.**

Keywords—multi-channel; FASText; natural scene; text detection

I. INTRODUCTION

Natural scene images usually contain a lot of high-level semantic information, such as road signs, traffic signs, building names and so on, which is one of the key clues to describe and understand the content of image scene. Text detection is a prerequisite for ensuring the reliability of image content analysis applications. However, texts in natural scene images usually vary greatly in size, color, background and resolution, and can be influenced by shelter, illumination, angle and so on. These factors make text detection difficult in natural scene images.

Existing natural scene text detection methods can be divided into two groups: sliding window based methods and connected component based methods. The methods based on sliding window usually make use of a text classifier to check the cell windows of various scales and angles [1]. The cell windows that are identified as text are merged to get the text regions. Commonly used classification features include underlying features of artificially designed (color, gradient, texture, structure) and deep feature of CNN. The detection process of such methods is relatively simple. Nevertheless, these methods are slow because of the number of windows that needs to be classified is large. The connected component based methods extract character candidates by aggregating image pixels with similar brightness, stroke width or other attributes (MSERs (Maximally Stable Extremal Regions) [2] [3] [4], SWT (Stroke Width Transform) [5], FASText [6]). Then the character candidates are classified and merged. These methods have been increasingly exploited because it is fast and effective.

Character candidate extraction is one of the most important steps of connect component based text detection methods in natural scene. MSERs is a widely used character candidate extraction algorithm, which is fast and efficient. However, character candidates extracted by the MSERs algorithm contain a large number of repetitive nested regions and the number of it is very large [2], which increases the difficulty of the subsequent steps. Busta and Neumann [6] proposed FASText algorithm, which detects the stroke key points to locate the possible characters in image. Then individual characters are segmented from the background by taking stroke key points as seed. The FASText algorithm is faster and produces less character candidates than MSERs. However, it extracts character candidates from the grayscale image and does not take color information into account, which leads to the missing of parts of the texts.

In order to improve the robustness of natural scene text detection, a multi-channel FASText based natural scene text detection method is proposed. A multi-channel FASText algorithm is designed to extract character candidates from the R, G and B component images, so as to minimize the loss of characters as much as possible; secondly, in order to remove the non-character regions precisely and ensure the accuracy of text detection, we trained a random forest character classifier [7] that uses Mean Local Binary Pattern(MLBP) [8] and Histogram of Oriented Gradient(HOG) [9] features to classify each character candidate; lastly, the character regions are grouped into text regions according to the color distance feature and geometric adjacency feature.

II. NATURAL SCENE TEXT DETECTION BASED ON MULTI-CHANNEL FASTEXT

The proposed approach includes following procedures: character candidate extraction based on multi-channel FASText, character candidate classification and text construction. Fig. 1 and Fig. 2 show the workflow and the step-by-step effect of the proposed natural scene text detection method respectively.

This work was supported by National Natural Science Foundation of China (61601335), Hubei Provincial Natural Science Foundation (2016CFB157), The National Key Technology R&D Program (2013BAH03B01), Project supported by the Key Laboratory of Satellite Surveying and Mapping Application Center, NASG (KLSMTA-2016-04)

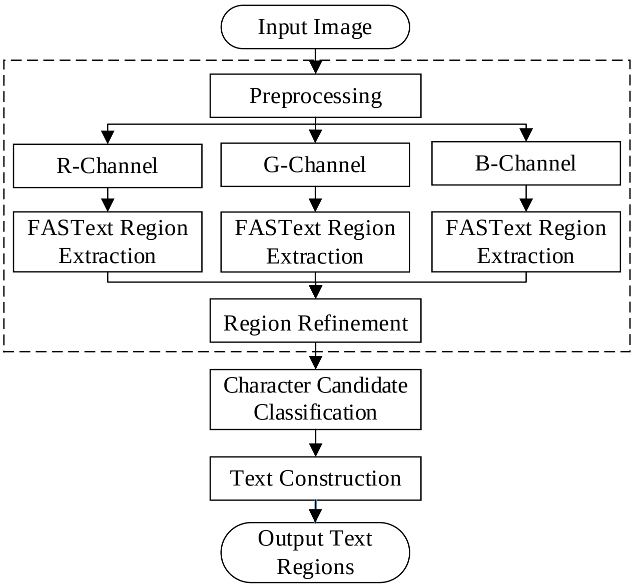
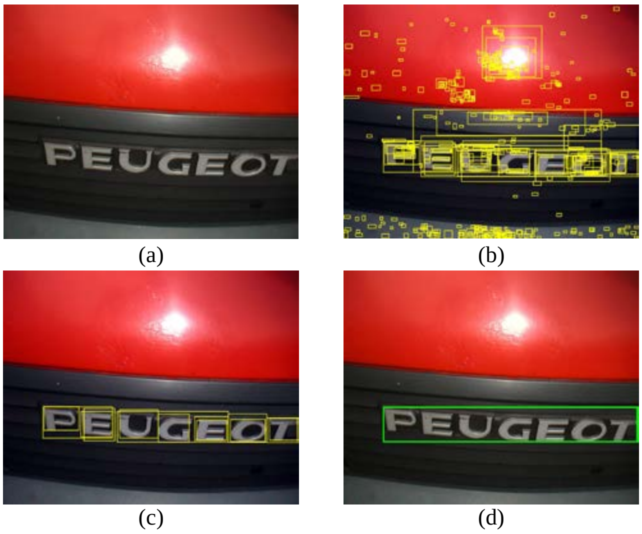

Input Image
Preprocessing
R-Channel
G-Channel
B-Channel
FASText Region Extraction
FASText Region Extraction
FASText Region Extraction
Region Refinement
Character Candidate Classification
Text Construction
Output Text Regions

Fig. 1 Workflow of proposed method



(a) (b)

(c) (d)

Fig. 2 The step-by-step effect of proposed text detection method

(a)original image, (b)character candidate extraction, (c)character candidate classification (d) text construction

A. *Character candidate extraction based on Multi-channel FASText*

Due to the unsatisfactory imaging environment or equipment, the natural scene images often suffer from the problems like tone level compression, low contrast and so on, which may lead to the loss of character. Therefore, it is necessary to preprocess the natural scene image before extracting the character candidates [3]. In order to improve the image quality, we enhance the image contrast by stretching image histogram.

Based on the observation that character is formed of strokes and the pixels in character regions are with similar brightness, the FASText algorithm firstly detects stroke keypoints by comparing the brightness relationship between the 12 pixels on the circumference of the circular cell window with a radius of 3 and the center pixel to locate the possible positions of the text in the image. Then the stroke keypoints are taken as the seed by flood-fill algorithm [6] to segment individual character from background.

The traditional FASText algorithm uses the brightness information of grayscale image to extract character candidates, which discards the color information, leading to the missing of characters and the reduction on recall. One of the key steps to improve the performance of text detection methods is to extract as much characters as possible. In order to reduce the loss of characters and take image color information into account, we extracted FASText character candidates from the R, G and B component images respectively. The character extraction results of the different color channels can be complementary to each other, namely, the characters that are not detected in a certain color channel can be detected in other color channels (see Fig. 3 (b)).

Although extracting character candidates from three channels can detect much more true characters, it leads to more repetitive regions at the same time. In order to reduce the computational complexity of subsequent steps and optimize the character extraction result, the repetitive regions and parts of the background regions are removed as much as possible by region refinement.

The analysis of character feature shows that the distribution of character geometric information is regular. A region whose width or height is too large or too small can't be a character. In addition, the same character may be extracted multiple times in different color channels, resulting in lots of repetitive regions. These repetitive regions are overlapped in the spatial distribution, and the color of these regions are very similar. Following the above observations, the process of region refinement is as follows: firstly, set thresholds for width, height, and aspect ratio of the character candidates, so as to remove the obvious non-character regions; then, remove the repetitive regions according to the following rule: when two character candidates meet the formula (1) and (2), the one with larger area is reserved.

- The area of the overlapping region is large enough

$$min\left(\frac{|A|\cap|B|}{|A|},\frac{|A|\cap|B|}{|B|}\right)>0.7 \qquad (1)$$

Where A and B denote two different character candidates respectively, and $|*|$ denotes the area of the bounding box of a character candidate.

- The color distance is within a certain threshold

$$dis(C_A,C_B)<5 \qquad (2)$$

$$dis(C_A,C_B)=\sqrt{\sum_{k=1}^{3}\left(g(k)_A-g(k)_B\right)^2} \qquad (3)$$

Where $dis(*)$ denotes the color distance, and $g(k)$ denotes the grayscale mean of the color channel *k* of a character candidate.

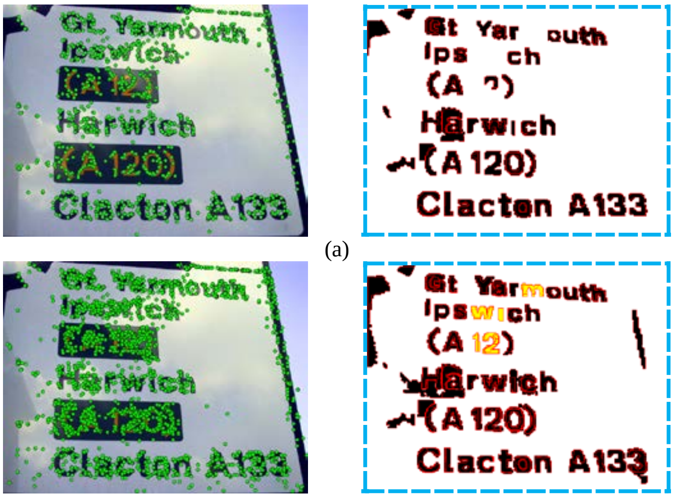


Fig. 3 Comparison of character candidate extraction result of the original FASText and proposed multi-channel FASText algorithm

(a) traditional FASText algorithm, (b) multi-channel FASText algorithm

Fig. 3 compares the character candidate extraction result of traditional FASText algorithm and the proposed multi-channel FASText algorithm. In (a) and (b), the first column represents the result of the stroke keypoint detection; the second column represents the result of character candidate extraction. The white, black and red regions represent the backgrounds, characters and the boundary of regions respectively. The yellow regions in (b) identify the characters that can't be extracted by traditional FASText algorithm. As can be seen from the figure, the multi-channel FASText algorithm can extract more characters compared with the traditional FASText algorithm.

B. *Character candidate classification*

In natural scene image, the texture information regularity of text regions is higher than backgrounds. We combined the MLBP and HOG features to train a random forest character classifier which is used to verify the characters. The HOG feature captures edge and contour information by counting the histogram of gradient direction distribution of the local image region. Considering that there are many background objects such as windows and strokes that have similar contour property as text, additional MLBP feature is combined to achieve better text and background discrimination effect. The MLBP feature has anti-noise ability, and is suitable for natural scene image.

In the training phase, we firstly prepare character and background samples from the training set images. When extracting features, the samples are transformed into grayscale image and scaled to size of 32×32 pixels. In order to reduce the feature dimension while achieving a good feature reproduction effect, we used the Uniform Pattern MLBP [8]. The cell size is set to 16×16 pixels, and 59-dimensional MLBP features are extracted from each cell. A sample is represented by a group of 3×3 cells, on which a 531-dimensional MLBP eigenvector $\boldsymbol{Hp}$ is extracted. For the HOG feature, the method proposed by [9] is used in this paper. The cell size is set to 4×4 pixels, and 31-dimensional HOG features are extracted from each cell. A sample is represented by a group of 8×8 cells, on which a 1984-dimensional HOG eigenvector $\boldsymbol{Hg}$ is extracted. Then the MLBP and HOG features are spliced into a 2515-dimensional eigenvector $\boldsymbol{H}=[\boldsymbol{Hp},\boldsymbol{Hg}]$. During the training process, we randomly select samples from the original set with replacement to form a new training set to training a decision tree. Each training set has the same samples number as the original set. At each node of a trained tree, we used a random subset of the features to find the best split.

In the testing phase, the MLBP and HOG features of character candidates are extracted in the same way as the training phase. Then the eigenvector is fed into the character classifier and the candidate regions classified as background will be eliminated. Fig. 2 (b) and (c) shows the character classification result, from which we can see that character classification effectively eliminates the backgrounds.

C. *Text construction*

The texts obtained by the character candidate classification exist in the form of discrete characters. The next step is to construct discrete characters into text regions. Texts in natural scene images are usually approximately horizontal, and the adjacent characters are similar in size, distance and color. Therefore, we used the geometric adjacency and the color distance features to construct the text.

At the beginning of construction, assume that the first input character as the first text region. Next, for each new input character, formula (4)-(7) determine whether it can be merged with a character in an existing text region: if there exists a combination relation, the new input character belongs to an existing text region and add the new input character to the region; otherwise, a new text region is created to hold the character. After the merging of all characters, the bounding box of the existing text regions is computed and regarded as the text detection result.

- The longitudinal distance of bounding box

$$abs\left(P_{A_y}-P_{B_y}\right)<0.8\times\frac{H_A+H_B}{2} \tag{4}$$

- The transverse distance of bounding box

$$abs(P_{A_x}-P_{B_x})<1.68\times\max(W_A,W_B) \tag{5}$$

- The height threshold bounding box

$$\frac{H_A}{H_B}\in[0.41,2.41] \tag{6}$$

- The color distance of character regions

$$dis\left(C_A,C_B\right)<35 \tag{7}$$

Where (P_{A_X}, P_{A_Y}), (P_{B_X}, P_{B_Y}) denote the center of the bounding box of the character *A* and *B* respectively, and H_*, W_* denote the height and width of the bounding box of the character respectively.

III. Experiment

The proposed method is evaluated on the most cited ICDAR 2013 Robust Reading Competition Dataset [10], which contains 233 test set images and 229 training set images. We generated 35,713 character samples and 35,845 background samples from the training set to train the random forest character classifier. The oob (out-of-bag) error of the classifier is 6.13%.

We conducted two level experiments to compare the proposed method with other state-of-the-art methods: (1) the character candidate extraction result, (2) the text detection result. All of the experiments were carried out on a laptop (Intel (R) Core i7 4712 MQ 4-core CPU 2.3 GHz / 8 GB RAM, C++). For an image with a size of 480×640 pixels, the average processing time is 0.98s.

A. *Character Candidate Extraction Performance*

We used the character-level recall rate (Rc) [6] to evaluate the character candidate extraction performance. A character is considered as extracted if it meets the following constraint requirements:

$$\frac{|D| \cap |G|}{|D|} > 0.7 \tag{8}$$

$$\frac{|D| \cap |G|}{|G|} > 0.7 \tag{9}$$

Where $|D|$ denotes the area of the bounding box of the extracted character region, $|G|$ denotes the area of the bounding box of the ground truth character region. The higher the character-level recall rate, the more characters are extracted.

First, we compared the influence of different color space on the proposed multi-channel character extraction algorithm (see Tab. I). The change of the Rc value in different color spaces fluctuates up and down at 0.10%, which indicates that the color space has slight impact on the ability of the algorithm to extract characters. In particular, the minimum number of character candidates can be obtained in RGB color space, which is 60.77% of the average of other color spaces. In the process of character candidate extraction, with the help of region refinement, the number of character candidates is reduced by 33.99% (from 633824 to 418409), which ensures the efficiency of subsequent steps.

Second, we compared the proposed character extraction algorithm with other character extraction algorithms (see Tab. Ⅱ). MSERs-1C and MSERs-3C respectively represent the use of the MSERs algorithm to extract character candidates on grayscale and three color component images of R, G, and B. FASText-1C and FASText-3C respectively represent the use of the traditional FASText algorithm and proposed multi-channel FASText algorithm to extract character candidates. Compared with FASText-1C, the Rc value is improved by 3.20% by the multi-channel FASText algorithm, which shows that the proposed algorithm can extract more characters. Compared with MSERs-3C, the Rc value is slightly higher, and the number of character candidates is only 39.81% of the former, which indicates that the multi-channel FASText algorithm performs better at character candidate extraction.

TABLE I. COMPARISON OF CHARACTER CANDIDATE EXTRACTION IN DIFFERENT COLOR SPACE

Color Space	**Region Number Ratio**	*Rc* **(%)**
RGB	1.00	91.86
Lab	1.73	91.91
Luv	1.60	91.87
YCrCb	1.62	91.77

TABLE II. COMPARISON OF THE RESULTS OF DIFFERENT CHARACTER CANDIDATE EXTRACTION ALGORITHMS

Algorithm	**Region Number Ratio**	*Rc* **(%)**
MSERs-1C	0.81	88.90
MSERs-3C	2.51	91.83
FASText-1C	0.45	88.66
FASText-3C	1.00	91.86

TABLE III. COMPARISON OF PROPOSED METHOD WITH OTHER METHODS

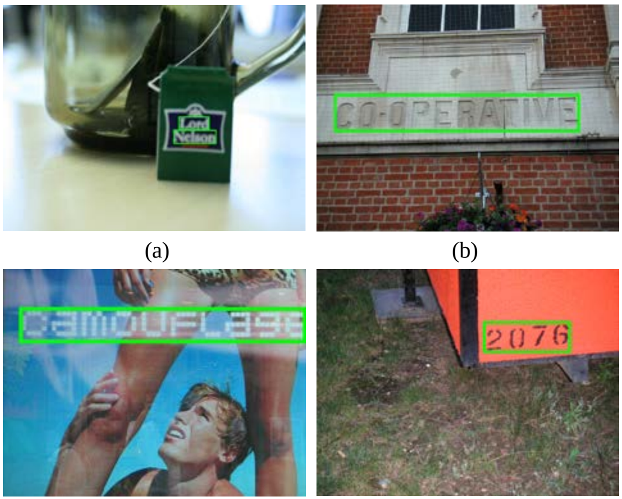
Algorithm	***R*** **(%)**	***P*** **(%)**	***F*** **(%)**
Proposed Method	**76.76**	80.17	**78.43**
iwrr2014 [5]	70.01	85.61	77.03
FASText based [6]	69.30	84.00	76.80
ERs based [4]	71.30	82.10	76.30
MSER based [3]	68.72	85.39	76.15
USTB_TexStar [2]	66.45	**88.47**	75.89
CASIA NLPR [10]	68.24	78.89	73.18
I2R_NUS_FAR [10]	69.00	75.08	71.91
I2R_NUS [10]	66.17	72.54	69.21

B. *Text Detection Performance*

The text-level detection result is evaluated by three commonly used metrics in natural scene text detection: recall (R), precision (P) and f-score (F) [10].

The performance comparison of proposed method with other methods on ICDAR 2013 dataset are shown in Tab. Ⅲ. The recall, precision and f-score of the proposed method achieve 76.76%, 80.17% and 78.43% respectively. Both recall and f-score are improved compared with other methods. In this paper, we extracted character candidates from the three color components of R, G and B of the image, which takes the image color information into account. So more text regions can be detected and the recall is much higher. In the meantime, multi-channel causes that many background regions are extracted as character candidates, which increases the difficulty of character classification and makes the precision lower than some other methods relatively. In summary, the proposed method obtains the highest f-score, indicating that the method can get the best text detection performance.

Fig. 4 illustrates several text detection examples of the proposed method on the ICDAR 2013 dataset, where the green wireframe marked regions are the detected text regions. The proposed method worked well under various challenging conditions, such as blur text (Fig. 4 (a)), low contrast (Fig. 4 (b)), dot matrix fronts (Fig. 4 (c)) and broken strokes (Fig. 4 (e)), which indicates the robustness and effectiveness of the method.



(a) (b)

(c) (d)

Fig. 4 Examples of text detection result of proposed method

IV. Conclusion

In this paper, we proposed a multi-channel FASText based natural scene text detection method. Compared with the character candidate extraction algorithms based on MSERs or traditional FASText, the proposed multi-channel FASText algorithm can extract more characters. In combination with the subsequent multi-feature fusion classification step, the accuracy of text detection is guaranteed. Compared to other state-of-the-art methods, the proposed method achieves higher recall and f-score in ICDAR 2013 dataset, which demonstrates the effectiveness of the proposed method. However, the proposed method can't successfully detects the text that the color is uneven or parts of it integrate with the background, which are worthy of further exploring.

References

[1] Tian Z, Huang W, He T, et al. Detecting text in natural image with connectionist text proposal network[C]//European Conference on Computer Vision. Springer International Publishing, 2016: 56-72.

[2] Yin X C, Yin X, Huang K, et al. Robust text detection in natural scene images[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(5): 970-983.

[3] Liu Liu J, Su H, Yi Y, et al. Robust text detection via multi-degree of sharpening and blurring[J]. Signal Processing, 2016, 124(C):259-265.

[4] Neumann L, Matas J. Real-time lexicon-free scene text localization and recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(9): 1872-1885.

[5] Huang W, Lin Z, Yang J, et al. Text localization in natural images using stroke feature transform and text covariance descriptors[C]// IEEE International Conference on Computer Vision. IEEE, 2013:1241-1248.

[6] Busta M, Neumann L, Matas J. FASText: efficient unconstrained scene text detector[C]//Proceedings of the IEEE International Conference on Computer Vision. 2015: 1206-1214.

[7] Breiman L. Random forests[J]. Machine Learning, 2001, 45(1): 5-32.

[8] Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(7): 971-987.

[9] Felzenszwalb P F, Girshick R B, McAllester D, et al. Object detection with discriminatively trained part-based models[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(9): 1627-1645.

[10] Karatzas D, Shafait F, Uchida S, et al. ICDAR 2013 robust reading competition[C]//Document Analysis and Recognition (ICDAR), 2013 12th International Conference on. IEEE, 2013: 1484-1493.