A Fast Uyghur Text Detector for Complex Background Images

Chenggang Yan, Hongtao Xie, Jianjun Chen, Zhengjun Zha, Xinhong Hao, Yongdong Zhang, *Senior Member, IEEE*, and Qionghai Dai, *Senior Member, IEEE*

***Abstract*—Uyghur text localization in images with complex backgrounds is a challenging yet important task for many applications. Generally, Uyghur characters in images consist of strokes with uniform features, and they are distinct from backgrounds in color, intensity, and texture. Based on these differences, we propose a *FASTroke* keypoint extractor, which is fast and stroke-specific. Compared with the commonly used MSER detector, FASTroke produces less than twice the amount of components and recognizes at least 10% more characters. While the characters in a line usually have uniform features such as size, color, and stroke width, a component similarity based clustering is presented without component-level classification. It incurs no extra errors by incorporating a component-level classifier while the computing cost is drastically reduced. The experiments show that the proposed method can achieve the best performance on the UICBI-500 benchmark dataset.**

***Index Terms*—Uyghur text localization, FASTroke keypoint extractor, Uyghur sence image.**

I. Introduction

TEXT localization in images with complex background is crucial to many real-world vision tasks, such as optical character recognition (OCR), text image analysis and scene understanding [1]. While a lot of researches on localizing text in Chinese and English, the localization of Uyghur text has not been well studied. A few of approaches have been proposed for Uyghur text localization in images with complex background [2]. Despite encouraging progress by them, the efficiency and accuracy of Uyghur text localization is still unsatisfactory, hindering the deployment of these approaches in real-world applications. As the locations of text vary significantly across images, exhaustive search generates a huge amount of text location candidates and is time-consuming. Moreover, the candidates often contain a lot of non-texts, yielding a heavy burden of distinguishing texts from non-texts, which is challenging due to the cluttered background and variation of Uyghur text in font size, style and color etc.

The state-of-the-art works for text localization in images with complex background can be roughly categorized into two categories: the sliding window method and the connected component analysis method [3]. The sliding window method detects texts through shifting a window on multiple scales [4]. The exhaustive search through a sliding window often achieves a high recall however has heavy computational complexity. The connected component analysis method first detect candidates of text blocks and then learns a text/non-text classifier to localize text blocks. The approach has achieved the best performance on ICDAR 2011 [5]. However, it also suffers from high computational complexity, due to a large amount of text candidates with many non-text components that have been filtered out through an additional text/non-text classifier.

In this paper, we propose a new approach for Uyghur text localization in images with complex background. To extract components efficiently and accurately, we propose a FAST-like keypoint [6] which fires on text strokes. The components are then found by a flood-fill algorithm with the extracted keypoints. Finally, the components are clustered into lines based on component similarity and the non-text lines are filtered out by a line classifier. The main contributions of this paper are three-fold.

1) To discover stroke ending, cross and bend, the FASTroke keypoints as a stroke-specific extractor is proposed. These keypoints can be extracted efficiently with less non-text components.
2) We improve the text line construction to deal with nearly horizontal text lines, which can increase the robustness of localizing English text lines. Components at the same horizontal position are grouped together. For each group, components are clustered as line candidates based on component similarity.

Manuscript received September 26, 2017; revised January 10, 2018 and March 7, 2018; accepted April 20, 2018. Date of publication May 18, 2018; date of current version November 15, 2018. This work was supported in part by the National Nature Science Foundation of China (61525206, 61771468, 61622211, 61472392 and 61620106009), in part by the National Key Research and Development Program of China (2017YFC0820600), and in part by the Youth Innovation Promotion Association Chinese Academy of Sciences (2017209). The associate editor coordinating the review of this manuscript and approving it for publication was Dr.Yonggang Wen. *(Corresponding author: Hongtao Xie.)*

C. Yan is with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China, and also with the Institute of Information and Control, Hangzhou Dianzi University, Hangzhou 310018, China (e-mail: cgyan@hdu.edu.cn).

H. Xie, Z. Zha, and Y. Zhang are with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China (e-mail: htxie@ustc.edu.cn; zhazj@ustc.edu.cn; zhyd73@ustc.edu.cn).

J. Chen is with the National Engineering Laboratory for Information Security Technologies, Institute of Information Engineering, School of Cyber Security, Chinese Academy of Sciences, Beijing 100093, China (e-mail: chenjianjun@iie.ac.cn).

X. Hao is with the Science and Technology on Mechatronic Dynamic Control Laboratory, Beijing Institute of Technology, Beijing 100081, China (e-mail: haoxinhong@bit.edu.cn).

Q. Dai is with the , Department of AutomationTsinghua University, Beijing 100084, China (e-mail: qhdai@tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2018.2838320

1520-9210 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

3) A novel framework for Uyghur text localization in complex background is presented. It no longer needs the component-level classifier and thus can avoid incurring extra errors or computing overhead.

In addition, a new benchmark dataset UICBI-500 is recommended in Section IV-A. Experimental results on component extraction show that the FASTroke-based component extractor reduces twice of false-positive components and executes two times faster as compared to the commonly used maximally stable extremal region on intensity channel (I-MSER) [7]. The F-measure of the proposed method achieves 74.4%, which improves the state-of-the-art by 11.2% on UICBI-500 dataset. Furthermore, the fast Uyghur localization framework executes almost 15 times faster than the method in [8].

The remainder of this paper is organized as follows. In Section II, an overview of related works is presented. Section III presents the proposed Uyghur text localization method. Experimental evaluation and analysis are presented in Section IV, followed by conclusions in Section V.

II. Related Work

Text localization in complex background images is a challenging task. A lot of approaches have been proposed for text localization in recent years. One class of methods is sliding window classification [9]–[12], that shifts windows on multiple scales to search text region candidates. At the same time, whether the window is a text region or not, it can be distinguished with a classifier, such as AdaBoost [13], [14], random forest [15], support vector machine (SVM) [16] or convolutional neural network (CNN) [17]–[19]. These methods usually achieve a high recall. However, window sliding and text/non-text classification are time-consuming.

The other class of methods is connected component analysis [20]–[22]. In general, it extracts components with regional consistency features consisting of color, edge, stroke width, extremal region, and point etc. The related connected component analysis approaches with the above features are introduced as follows.

Color features: Ordinarily, text is composed of consistent color which differs from the color of background. Correspondingly, many approaches based on color features have been developed [23]–[25]. The components are extracted by pixel color similarity clustering which is not robust to multi-color and illumination. The preprocessing that uses a mean-shift algorithm to reduce color complex [26] can improve the robustness, but it brings extra computational cost. Generally, the color features are sensitive to the complex background. The color-based methods are usually time-consuming.

Edge features: This kind of methods is based on the fact that text exhibits a strong and symmetric gradient against its background [27]. In [28]–[30], edge features are used to discover text candidates and the pixels with large and symmetric gradient are extracted as text components. The edge features are more robust to color gradients and illumination. However, they are sensitive to image contrast and background with a strong and symmetric gradient. Moreover, the edge-based text extraction is complicated and time-consuming.

Stroke width transform: The stroke width transform (SWT) is proposed by Epshtein *et al.* [31], which can be regarded as an upgraded version of edge features. SWT detects strokes by finding pixel pairs with symmetric gradient, and then generates a stroke map. SWT is a stroke-specific feature and has turned out to be competitive for text localization in images with high resolution and complex background. However, the performance of SWT exclusively relies on the accuracy of edge detection. Moreover, SWT and component construction result in high computational cost.

Extremal region features: In recent years, the region-based methods have become the mainstream of connected component analysis, particularly, the extremal region (ER) or maximally stable extremal regions (MSER) [32]. The ERs or MSERs are extracted as candidate components in [7], [33], [34]. Although these methods are effective, there remain two limitations. On one hand, the region features are not text-specific. Hence, an extra text/non-text classifier is required, such as CNN [35], SVM [8], and random forest [2] etc. Yin *et al.* presented a MSER-tree pruning algorithm to remove false detections [5]. All the false detection suppressing approaches suffer from high computational cost, low speed and weak robustness. On the other hand, the extracted MSERs in the single channel contain a great deal of non-text and duplicate regions resulting in heavy computation of text/non-text region classification.

The MSER-based methods are not robust to image blur either. The multi-color-channel enhanced MSER is proposed to improve the robustness [8], [36]. While it improves *recall*, it further increases the processing time-cost. The number of non-text and duplicate components is almost increased by m times, where m is the number of channels.

Point features: The earlier works built on point features mainly focus text detection in videos [37], [38]. Generally, captions in video have a simple background, and thus a simple keypoint detector such as Harris [38] can be used to localize text. Then, text regions are formed by a series of morphology operations on the binary corner image. These methods are simple and efficient, however are sensitive to noise and become invalid for complex background images. In [39], a FASText detector was put forward to localize texts in complex background images and has achieved good performance on ICDAR2013 dataset. Michal *et al.* [39] detected keypoint in multi-scale images, and also produced many repeated detections which increase the computation cost of the subsequent steps.

Despite the success of the sliding window and connected component analysis methods on text localization in complex background images, they produce large amount of non-text candidates, which bring in heavy computation for candidate extraction and classification. In this paper, we present an effective Uyghur text localization method to lift those restrictions.

III. Uyghur Text Localization With FASTroke

The flowchart of the proposed approach is illustrated in Fig. 1. It consists of five stages: keypoint detection, component

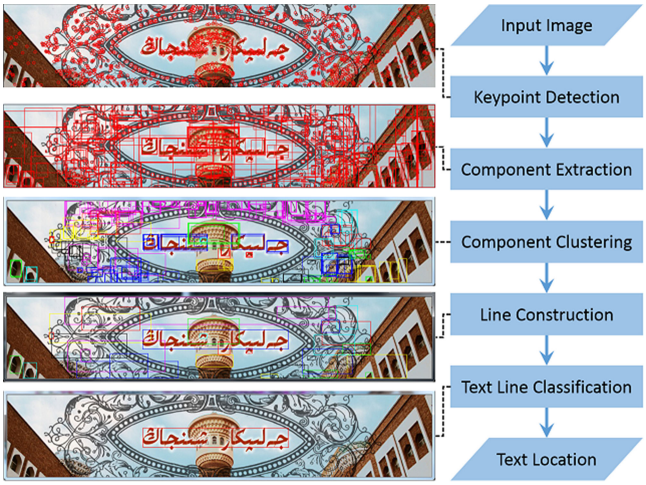


Fig. 1. The flowchart of the proposed method consisting of five steps. **Best view in color**.

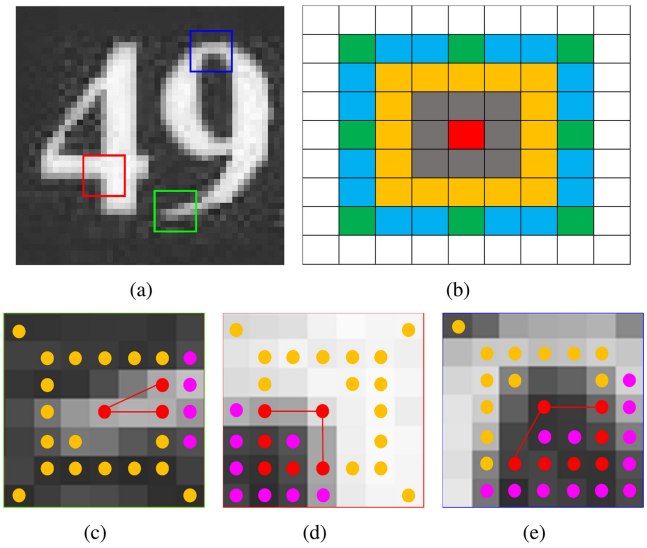


Fig. 2. (a) The three types of stroke features. (b) The detector template. (c) The acute-angle keypoint corresponds to **stroke ending**. (d) The right-angle keypoint corresponds to **stroke cross**. (e) The obtuse-angle keypoint corresponds to **stroke bend**.

extraction, component clustering, line construction and text line classification. We elaborate the approach in the following subsections. Subsection III-A presents the proposed FASTroke keypoint extractor and Subsection III-B describes the text component extraction method. As the component clustering is a part of line construction, we present it together with the text line construction and classification in Subsection III-C.

A. FASTroke Keypoint Detector

There are three types of stroke features in characters, which are stroke ending, cross and bend, as shown in Fig. 2(a). The most related keypoint of FAST [40] only extracts the stroke features which respond to stroke ending, such as the tail of "9". As a result, it overlooks the characters without stroke endings like "o" and "0". Besides, FAST is not a text-specific keypoint detector and produces numerous non-text candidates.

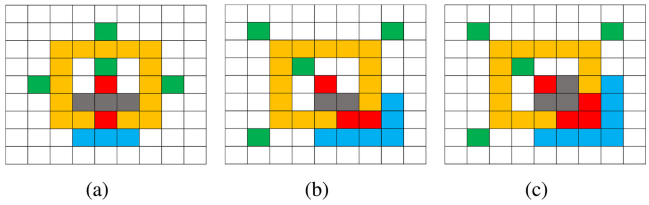


Fig. 3. The acute-angle keypoints.

Here, we propose a stroke-specific keypoint extractor – FASTroke, which can effectively extracts text candidates and generates less non-text candidates. The FASTroke keypoint detector defines three types of keypoints for the three stroke features. The acute-angle keypoint corresponds to stroke ending, such as the example in Fig. 2(c). The right-angle keypoint fires on stroke cross, as shown in Fig. 2(d). The obtuse-angle keypoint lies on stroke bend, as in Fig. 2(e). The details of FASTroke keypoint detection are described as follows.

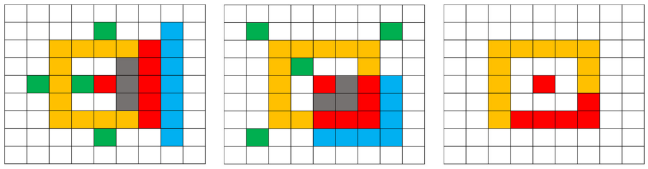
For each pixel p, we define a template to verify the kind of keypoint it belongs to. The detector template is a 7×7 rectangle, as shown in Fig. 2(b). The center pixel of the template is the keypoint candidate p marked as red. The pixels around p are grouped into 4 areas, with the inner and middle areas marked as gray and yellow respectively, and the outer areas and corners marked as blue and green respectively. Each pixel x in these areas is mapped to one of the three values: 0, 1 or 2. The mapping function is defined as:

$$L(p,x) = \begin{cases} 1 & \text{if} \quad I_p - I_x \quad \geq t \\ 0 & \text{if} \quad |I_x - I_p| \quad < t \\ 2 & \text{if} \quad I_x - I_p \quad \geq t \end{cases} \tag{1}$$

where I_p is the image intensity value of p and t is a margin parameter. The value 1 means that x is darker than p, value 0 means x is similar to p, and value 2 signifies that x is brighter than p.

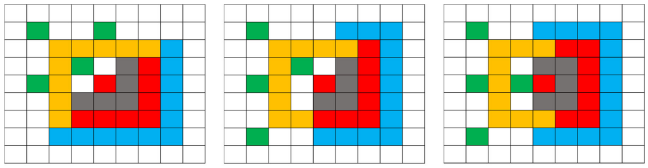
At first, pixels in the middle area are examined. The pixel p is a keypoint candidate if there exist such two contiguous sets of P_0 and P_1 (or P_0 and P_2) with $|P_0| < 8$. Then, according to the value of $|P_0|$, the pixel p is labeled as one of the three kinds of keypoint candidates. To further confirm that whether p is text relevant, we need to checked in the inner and outer area of the corresponding pixels. There are two rules related: the internal continuity rule and the external distinction rule. The former ensures the keypoint to be a stroke ending rather than an isolated point, while the latter checks the continuity of the background. The three types of FASTroke keypoint are described as follows.

In consideration of the acute-angle keypoints, we get the conclusion of $|P_0| \in \{1, 2, 3, 4\}$. Fig. 3 shows three cases of the acute-angle keypoints. In the case of Fig. 3(a), $|P_0| = 1$. The pixel of P_0 is marked as red. The internal continuity verification refers to the inner (gray) and outer (blue) areas. All of these pixels (c), in the inner and outer areas, do not satisfy



(a) (b) (c)

Fig. 4. The right-angle keypoints.



(a) (b) (c)

Fig. 5. The obtuse-angle keypoints.

$L(p, c) = 0$, and then the current point p will be discarded. In the external distinction verification, the green pixels are examined. If there exist any pixel (c) that fails to satisfy $L(p, c) = L(p, x)$, where x stands for a pixel in the yellow area, and then p is not an acute-angle keypoint.

The right-angle keypoints mainly hit the stroke cross, such as "T, 4, G", as presented in Fig. 4. In this case, $|P_0| = 5$, however it cannot always get a regular right-angle keypoint. Fig. 4(c) shows a non-right-angle keypoint. The regular right-angle keypoint merely incorporates 8 cases. Fig. 4(a) and (b) show two typical examples.

In the case of the obtuse-angle keypoint, we have $|P_0| \in \{6, 7\}$. Fig. 5 offers three obtuse-angle keypoint verification cases. Different from the acute-angle keypoint, the internal continuity verification is rather strict. In the gray area, there exist at least two contiguous pixels similar to the center. In other respects, the current point p is not the obtuse-angle keypoint.

Each of the three kinds of keypoints pass a simple non-maximum suppression performed on a 3×3 neighborhood. Only the keypoint with the highest response will be kept.

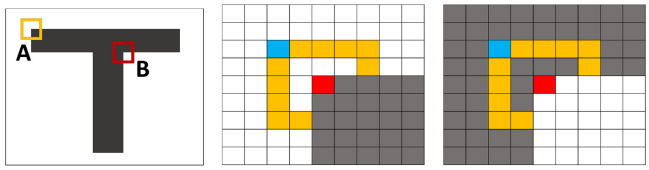
B. Text Component Extraction

The text component extraction is based on the assumption that the intensity of text component has internal consistency and external distinction. The FASTroke keypoint has discovered "seeds" of components, based on which these we can thus capture the whole component via the local consistency flood-fill. Sample results of the proposed component extraction method are illustrated in Fig. 6.

The acute-angle keypoint can be directly deemed as a component tail. Since the stroke is surrounded by P_1 or P_2, the pixel p must be a part of stroke, as shown in Fig. 3(a). In view of the peculiarity of the right-angle and obtuse-angle keypoints, the seed of flood-fill is slightly distinct from the acute-angle keypoint, because these keypoints may not hit the stroke, as the red keypoint in Fig. 7(a). To extract characters in this situation, we regard both the keypoint and the middle pixel of P_1 or P_2 as seeds, as the red and blue pixels shown in in Fig. 7(c).

(a) (b) (c) (d)

Fig. 6. The component candidate extraction process and results. (a) The original images. (b) The FASTroke keypoint detection results, where the acute-angle keypoints are marked as green, right-angle keypoints as red and obtuse-angle keypoints as blue. (c) The flood fill results. (d) The bounding boxes.



(a) (b) (c)

Fig. 7. (a) An example of the flood-fill seed selection. (b) The local detail of keypoint A. (c) The local detail of keypoint B.

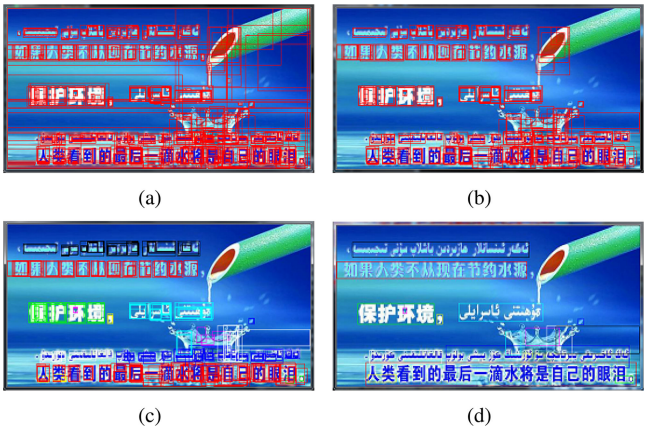
For the similarity threshold (θ) decision, there are two cases, which are that the stroke is brighter or darker than the background. In view of a bright keypoint, the similarity threshold θ_1 is above the extreme intensity value of pixels in P_1:

$$\theta_1 = \max(I_x) + 1 \mid x \in P_1. \tag{2}$$

Correspondingly, if given a dark keypoint, the similarity threshold θ_2 is below the extreme intensity value of pixels in P_2:

$$\theta_2 = \min(I_x) - 1 \mid x \in P_2. \tag{3}$$

The proposed component extraction is effective to discovery the majority of text candidates. Nevertheless, the loss of a few candidate is unavoidable, because sometimes the FASTroke is not able to find a threshold in a low contrast image. Some lost candidates will be retrieved in the text line construction stage, which will be explained in Section III-C.



(a) (b) (c) (d)

Fig. 8. The procedure of line construction. (a) The component extraction. (b) The heuristic rule filtration. (c) The similarity clustering. (d) The text line candidates.

C. Text Line Construction and Classification

a) Text Line Construction: The text line construction can be viewed as a component clustering process, which is based on the component similarity composed of size, color and location. It mainly contains two parts: the noise-reduction in view of heuristic rule and the component clustering. Fig. 8 shows the procedure of line construction.

Several heuristic rules are applied so as to remove the obvious non-text component. First, the components with too small or too large size are non-text, that is

$$\begin{cases} 10 \le w_c \le w_i/2, & \text{if } w_i > 100 \\ 10 \le h_c \le h_i/3, & \text{if } h_i > 100 \end{cases}, \tag{4}$$

where w_c represents the width of component, h_c is height, w_i is width of image and h_i is height. The aspect ratio (r) of a text component is in the range of $0.2 < r < 5$, $r = h_c/w_c$. These heuristic rules are beneficial to candidate reduction, and can partly suppress the false positives, which will be demonstrated in Section IV.

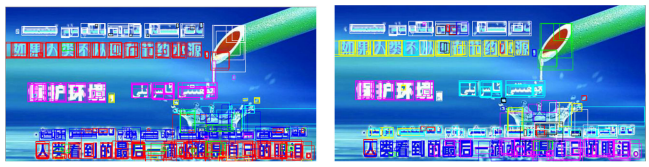
The component similarity clustering consists of two stages. In the first stage, components on the same horizontal line are split into a group. Components within each group are clustered, in accordance with component similarity as line candidates. The details are explained as follows.

The grouping stage mainly handles the horizontal texts. The extracted components (C) are divided into groups ($\mathcal{G}$) by Algorithm 1. The grouping result is presented in Fig. 9(a).

In the clustering stage, components are clustered into lines by component similarity incorporating size, color and location. Firstly, those components with too small or too large size in a group are separated, and then the average width of the rest components is viewed as a character width (w_c). The components satisfying the location and color similarity conditions are organized as a line. An example of clustering result is illustrated in Fig. 9(b). The details of clustering process are elaborated in Algorithm 2, where the $LocationDist(a, b)$ calculates the gap between a and b, and $ColorDist(a, b)$ represents a color difference ΔE in Lab color space [41].

Algorithm 1: The horizontal grouping algorithm

procedure HGROUPING (C, $\mathcal{G}$)
  2D-list $L_{copy}[|C|]$
  for $i = 1 \rightarrow |C|$ **do**
    for $j = 1 \rightarrow |C|$ **do**
      if $j \neq i$ & $C[j]$ is a repetition of $C[i]$ **then**
        $L_{copy}[i] \leftarrow C[j]$
        $L_{copy}[j] \leftarrow C[i]$
      end if
    end for
  end for
  while $C \neq \varnothing$ **do**
    $S \leftarrow C[0]$
    $C \leftarrow C - C[0]$
    for $i = 1 \rightarrow |S|$ **do**
      for $j = 1 \rightarrow |C|$ **do**
        if $C[j]_{y\text{-range}} \cap s_{y\text{-range}} \neq \varnothing, s \in S$ **then**
          $S \leftarrow C[i]$
          $C \leftarrow C - C[i]$
        end if
      end for
      $S \leftarrow S \cup L_{copy}[i]$
    end for
    $\mathcal{G} \leftarrow S$
  end while
end procedure



(a) (b)

Fig. 9. The results of component clustering. (a) The horizontal groups. (b) The final clusters.

(a) (b)

Fig. 10. The text line classification. (a) The text line candidates. (b) The final text lines.

b) Text Line Classification: Without the component-level text/non-text classification, the constructed line candidates inevitably contain some non-text lines. With the help of line construction, a line classifier is utilized to distinguish text/non-text lines. An example of text line classification is shown in Fig. 10. Since Uyghur text has abundant texture characteristics, the HoG feature can well represent Uyghur text. Moreover, the effective-

Algorithm 2: The component clustering algorithm

procedure CLUSTERING ($\mathcal{G}$, $\mathcal{L}$)
  for each $G \in \mathcal{G}$ **do**
    while $G \neq \varnothing$ **do**
      $L \leftarrow G[0]$
      $G \leftarrow G - 0$
      for each combination$\{a, b\}, a \in L, b \in G$ **do**
        $d_i \leftarrow LocationDist(a, b)$
        $d_c \leftarrow ColorDist(a, b)$
        if $d_i \leqslant w_c$ & $d_c \leqslant t_c$ **then**
          $L \leftarrow L + b$
          $G \leftarrow G - b$
        end if
        if $d_i > w_c$ & $d_c \leqslant t_c$ **then** ▷ Find loss
          get binary image (i) between a, b
          find components C in i
          $G \leftarrow G \cup C$
        end if
      end for
      $\mathcal{L} \leftarrow L$
    end while
  end for
end procedure

Fig. 11. Several examples of the expanded images in UICBI-500. The green rectangles are ground truth labels.

ness of HoG-SVM classifier has already been demonstrated in text/non-text line classification task [8].

IV. Experiments

In this section, we first introduce a new benchmark dataset and the evaluation protocol. Then, we report the qualitative evaluation of FASTroke keypoint detector. Afteraward, we elaborate the experimental results of the proposed approach and the performance comparison to multiple state-of-the-art methods on three public datasets. The system is implemented on a laptop with a 3.2 GHz 4-core CPU and OpenCV-3.0.

A. Dataset and Evaluation Protocols

1) UICBI-500 Dataset: This paper recommends the UICBI-500 dataset an expanded version of the UICBI-400 [42]. The new images consist of advertisement slogans, nature scene and born-digital images. The UICBI-500 dataset is more challenging than UICBI-400, and the challenges mainly stem from two aspects. First, the diversity of the texts in consideration of the fonts, sizes and colors is much more intricate. Second, the image background is more diverse and contains more patterns that are rather hard to differentiate from the text. The training set involves 300 images randomly selected from UICBI-500, and the remaining constitutes the test set.

2) Component Extraction Evaluation Protocol: To measure the performance of component extraction, we employ the evaluation protocol in [42]. The recall, precision and repetition are defined as follows.

$$\begin{cases} recall & = \frac{area(D \cap G)}{area(G)} \\ precision & = \frac{N_p}{N_t} \\ repetition & = \frac{N_r}{|D|} \end{cases}, \tag{5}$$

where D is the detection set and G is the ground-truth set. N_p is the number of positive detections satisfying the condition $\frac{area(D \cap G)}{area(D)} \geqslant 0.8$ and $h_{D_i}/h_{G_j} \geqslant 0.6$. It means that if most of D falls into G, this D is regarded as a positive detection. Besides, N_r is the number of repetition.

3) Text Localization Evaluation Protocol: The most popular evaluation protocol is proposed by Wolf *et al.* [43], which considers three matching cases between the ground-truth (G) and the detection (D): one-to-one, one-to-many and many-to-one. It computes the precision (p), recall (r) and f-measure, and remains an effective measurement to the text localization result. The p and r are defined as follows:

$$\begin{cases} r(G, D, t_r, t_p) = \frac{\sum_i Match_G(G_i, D, t_r, t_p)}{|G|} \\ p(G, D, t_r, t_p) = \frac{\sum_j Match_D(D_j, G, t_r, t_p)}{|D|} \\ f\text{-measure} = \frac{2pr}{p+r} \end{cases}, \tag{6}$$

where t_r is the constraint on area recall and $t_r = 0.8$. t_p is the constraint on area precision and $t_p = 0.4$. The $Match$ function definition is as below:

$$Match_X(X_k, Y, t_r, t_p) = \begin{cases} 1 & \text{if} \quad X_k \cap Y = 1 \\ 0 & \text{if} \quad X_k \cap Y = \varnothing \\ \frac{1}{1+\ln(m)} & \text{if} \quad X_k \cap Y = m \end{cases}, \tag{7}$$

where $X_k, (k \in [1,\ |X|])$ matches against m rectangles of Y in the one-to-many case and $m \in [2,\ |Y|]$.

B. Qualitative Evaluation of Keypoint Detectors

Fig. 12 illustrates sample results of the four keypoint detectors. The text in the solid bounding box is marked by all of these detectors. The text "PARIS" in the dashed bounding box is lost by FAST and Harris, as shown in Fig. 12(a) and (b). In Fig. 12(c), all texts are found, however, the number of extracted keypoints is almost 1.78 times of the FASTroke keypoints. These results have demonstrated that the FASTroke keypoint detector is effective to detect text and produces less candidates.

C. Experiments on ICDAR 2011 Dataset

The ICDAR 2011 Robust Reading Competition (Challenge1: Born Digital Images) database is widely used for benchmarking

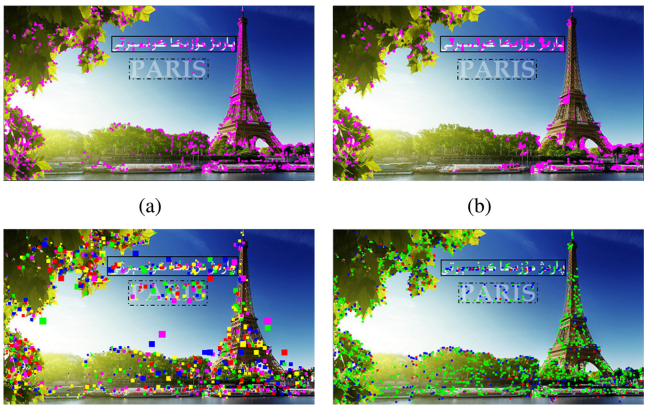


Fig. 12. (a) FAST extracts 1,098 keypoints. (b) Harris extracts 1,610 keypoints. (c) FASText extracts 2,875 keypoints. Different mark sizes represent the disparate scales of keypoints. (d) FASTroke extracts 1,615 keypoints.

TABLE I
THE EVALUATION RESULTS OF COMPONENT EXTRACTION ON ICDAR 2011

Method	*recall*	*precision*	*repetition*	$\bar{t}\ (ms)$	$\overline{\lvert D\rvert}$
RGB-MSER[8]	0.743	0.309	0.707	971.8	1524.7
HSV-ER[36]	0.751	0.278	0.562	566.1	1071.6
I-MSER[5]	0.677	0.316	0.540	320.7	503.4
FASTroke	0.874	0.378	0.106	168.4	363.9

TABLE II
THE EVALUATION RESULTS OF COMPONENT EXTRACTION WITH HEURISTIC RULE DENOISE ON ICDAR 2011

Method	*recall*	*precision*	*repetition*	$\bar{t}\ (ms)$	$\overline{\lvert D\rvert}$
RGB-MSER[8]	0.637	0.367	0.663	1295.3	1082.2
HSV-ER[36]	0.644	0.333	0.506	758.4	758.0
I-MSER[5]	0.570	0.374	0.488	413.2	358.1
FASTroke	0.710	0.386	0.123	170.5	190.7

text detection algorithms. It is divided into a testing set with 141 images and a training set with 410 images. These images primarily derive from Web and Emails.

1) Evaluation of Component Extraction: Table I shows the results of component extraction by multiple extractors on the ICDAR 2011 dataset. The results show that the proposed FASTroke detector possesses better efficiency and produces much fewer candidates. In particular, compared with the I-MSER, FASTroke obtains a higher $recall$ but the detected components are less than half that of I-MSER. The primary cause lies in that FASTroke partly benefits from the low $repetition$.

Table II shows the corresponding evaluation results of component extraction with heuristic rule denoise (HRD). As we can see, HRD reduces the candidates by near 1/3 times, though it implicates $recall$ that alleviates the computation of succeed process. From the results in Tables I and II,we can see that FASTroke is the most efficient extractor.

2) Evaluation of Text Localization: Text localization results on the ICDAR 2011 dataset are reported in Table III, where the methods in [18] and [19] are on the basis of deep learning.

TABLE III
THE EVALUATION RESULT ON ICDAR 2011

Method	*recall*	*precision*	f	$\bar{t}\ (s/image)$
Our	0.696	0.785	0.759	0.75
Yao *et al.*[44]	0.657	0.822	0.730	N/A
Yin *et al.*[5]	0.842	0.935	0.886	N/A
ER[36]	0.647	0.731	0.687	N/A
MSER[7]	0.525	0.689	0.596	N/A
CTPN[18]	0.790	0.890	0.840	0.14
He *et al.*[19]	0.740	0.910	0.820	0.50

TABLE IV
THE EVALUATION RESULT OF COMPONENT EXTRACTION ON ICDAR 2013

Method	*recall*	*precision*	*repetition*	$\bar{t}\ (ms)$	$\overline{\lvert D\rvert}$
RGB-MSER[8]	0.821	0.258	0.514	6488.0	6827.1
HSV-ER[36]	0.859	0.148	0.284	5992.3	8579.2
I-MSER[5]	0.753	0.273	0.406	2017.4	2153.4
FASTroke	0.877	0.217	0.0715	398.7	921.5

TABLE V
THE EVALUATION RESULTS OF COMPONENT EXTRACTION WITH HEURISTIC RULE DENOISE ON ICDAR 2013

Method	*recall*	*precision*	*repetition*	$\bar{t}\ (ms)$	$\overline{\lvert D\rvert}$
RGB-MSER[8]	0.789	0.304	0.540	11574.0	5139.2
HSV-ER[36]	0.826	0.183	0.308	11999.7	6245.3
I-MSER[5]	0.719	0.319	0.434	2850.6	1622.8
FASTroke	0.818	0.298	0.096	399.7	468.3

Unfortunately, the accuracy and speed of the proposed algorithm are not comparable to that of the deep learning methods. Our approach performs not that well on accuracy because this dataset contains many non-horizontal text lines excluded by text line classifier. The computation of deep learning methods normally is intricate. However, they use a lot of GPUs to accelerate and obtain a high speed.

D. Experiments on ICDAR 2013 Dataset

The ICDAR 2013 Robust Reading Competition (Challenge2: Focused Scene Text) database is broadly used for scene text detection. The images mainly come from nature scene, which quite differs from the ICDAR 2011 Challenge1. It comprises a testing set with 233 images and a training set with 229 images. We select both the ICDAR 2011 and the ICDAR 2013 as benchmarks, owing to the fact that both the born digital and focused scene are subsets of the complex background images.

1) Evaluation of Component Extraction: In Table IV, the component extraction results of several commonly used extractors are compared on the ICDAR 2003 dataset. Table V is the corresponding assessment results of component extraction with HRD, which demonstrates the effectiveness of heuristic rules on scene image dataset. Both two tables confirm the fact that FASTroke is faster than the rest and produces much fewer candidates. In particular, FASTroke compared with I-MSER, obtains a higher $recall$, but the detected components have reduced at least by half.

TABLE VI
THE EVALUATION RESULT ON ICDAR 2013

Method	*recall*	*precision*	*f*	$\bar{t}$ $(s/image)$
Our	0.704	0.815	0.755	0.97
FASText[39]	0.693	0.840	0.768	0.15
He *et.al*[19]	0.730	0.930	0.820	0.50
CTPN[18]	0.830	0.930	0.880	0.14
TDN[45]	0.740	0.830	0.780	0.61
Faster-RCN[46]	0.710	0.750	0.730	0.13

TABLE VII
THE EVALUATION RESULT OF COMPONENT EXTRACTION ON UICBI-500

Method	*recall*	*precision*	*repetition*	$\bar{t}$ (ms)	$\overline{\lvert D \rvert}$
RGB-MSER[8]	0.742	0.144	0.634	3402.2	5490.6
HSV-ER[36]	0.780	0.124	0.467	1867.8	4092.7
I-MSER[5]	0.556	0.140	0.503	965.5	1772.3
FASTroke	0.725	0.123	0.094	320.7	743.9

TABLE VIII
THE EVALUATION RESULTS OF COMPONENT EXTRACTION WITH HEURISTIC RULE DENOISE ON UICBI-500

Method	*recall*	*precision*	*repetition*	$\bar{t}$ (ms)	$\overline{\lvert D \rvert}$
RGB-MSER[8]	0.721	0.174	0.649	5089.0	4358.7
HSV-ER[36]	0.756	0.152	0.490	2860.8	3201.9
I-MSER[5]	0.535	0.166	0.522	1354.2	1413.1
FASTroke	0.692	0.185	0.127	322.8	443.5

TABLE IX
THE EVALUATION RESULT ON UICBI-400

Method	*recall*	*precision*	*f*	$\bar{t}$ $(s/image)$
Our	0.846	0.815	0.830	0.95
RGB-MSER[8]	0.888	0.776	0.828	15.43
MSER[5]	0.522	0.749	0.616	N/A
ER[36]	0.313	0.479	0.300	24.59
TDN[45]	0.793	0.885	0.834	0.61
Faster-RCN[46]	0.618	0.715	0.662	0.13

2) Evaluation of Text Localization: The localization results on the ICDAR 2013 dataset are listed in Table VI. In contrast to the Faster-RCN, our method achieves the improvement of the *precision* by 6.5%.

E. Experiments on UICBI-400 and UICBI-500 Datasets

1) Evaluation of Component Extraction: Table VII reports the comparison of component extraction on the UICBI-500 dataset. Table VIII gives the evaluation result of component extraction with heuristic rule denoise. In both tables, when it contrast to the commonly used I-MSER [7], the FASTroke based component extractor generates 2 times fewer components and runs 2 times faster.

2) Evaluation of Text Localization: The evaluation results on the UICBI-400 and the UICBI-500 datasets are summarized in Tables IX and X.

Our method achieves the highest f-measure on both datasets. Moreover, the time-cost of our approach remains the lowest and

TABLE X
THE EVALUATION RESULT ON UICBI-500

Method	*recall*	*precision*	*f*	$\bar{t}$ $(s/image)$
Our	**0.725**	**0.763**	**0.744**	**0.96**
RGB-MSER[8]	0.600	0.659	0.628	14.56
ERs[36]	0.379	0.523	0.439	25.73

Fig. 13. Successful results of several challenging images.

Fig. 14. Several failure cases.

the system runs over 15 times faster than [8]. The high *precision* benefits from three facts: 1) The FASTroke keypoint extractor is good at discovering text components. 2) The component similarity clustering can effectively form text lines. 3) The line classifier is accurate at text line verification. The low time-cost mainly relies on the less component candidates produced by FASTroke and the framework with no extra component classification cost.

Several successful text localization examples are displayed in Fig. 13, which demonstrate that our framework is effective to localize Uyghur text in complex images and is also robust to font-size, style and noise. However, there exist some failures which expose a few shortcomings, as shown in Fig. 14. The FASTroke keypoint detector is sensitive to image contrast and loses the text with low contrast. The line classifier sometimes is confused on the HoG representation of Uyghur text-like non-text lines, such as fence, tree branch and the like.

V. CONCLUSION AND FUTURE WORK

Due to the complex background and text diversity, fast text localization in images remains a challenging task. In this paper, we propose a novel approach which contains two effective

modules for Uyghur text localization in complex background images. One is the stroke-specific FASTroke keypoint detector that can spot the stroke ending, cross and bend effectively. The other is the component similarity clustering algorithm, which does not include component-level classification. Instead, the extracted components are directly constructed into lines according their similarities.

A new benchmark dataset UICBI-500 is constructed, which is more challenging than existing ones. The experiment results on UICBI-500 have two indications. First, the FASTroke keypoint detector has the best efficiency and produces the fewest candidates. Second, the proposed approach achieves the best performance compared to state-of-the-art methods and improves f-measure by 11.2 These results have demonstrated the effectiveness of FASTroke keypoint detector and component similarity clustering algorithm. More importantly, as the FASTroke keypoint detector extracts less components and the similarity clustering algorithm dose not need extra component classification, the speed of our approach is near real-time.

The proposed method is lightweight that it can be applied to portable devices and embedded system for image text detection. However, it also has two limits. One is that the proposed method simply manages horizontal direction text. The other is our technology cannot effectively deal with the multi-language dataset. The future work will focus on developing a multi-orientation and multi-language robust solution and building a text recognition system.

REFERENCES

[1] C. Yao, X. Bai, and W. Liu, "A unified framework for multioriented text detection and recognition," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4737–4749, Nov. 2014.

[2] S. Liu, H. Xie, C. Zhou, and Z. Mao, "Uyghur language text detection in complex background images using enhanced MSERs," in *Proc. Int. Conf. Multimedia Model.*, 2017, pp. 490–500.

[3] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 7, pp. 1480–1500, Jul. 2015.

[4] K. L. Bouman, G. Abdollahian, M. Boutin, and E. J. Delp, "A low complexity sign detection and text localization method for mobile applications," *IEEE Trans. Multimedia*, vol. 13, no. 5, pp. 922–934, Oct. 2011.

[5] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 970–983, May 2014.

[6] J. Chen, H. Xie, Y. Hu, and C. Yan, "Uyghur text localization with fast component detection," in *MultiMedia Modeling*. Cham, Switzerland: Springer International Publishing, 2018, pp. 565–577.

[7] L. Neumann and J. Matas, "Text localization in real-world images using efficiently pruned exhaustive search," in *Proc. Int. Conf. Document Anal. Recognit.*, 2011, pp. 687–691.

[8] J. Chen *et al.*, "Robust Uyghur text localization in complex background images," in *Proc. Pac. Rim Conf. Multimedia*, 2016, pp. 406–416.

[9] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 1457–1464.

[10] A. Coates *et al.*, "Text detection and character recognition in scene images with unsupervised feature learning," in *Proc. Int. Conf. Document Anal. Recognit.*, 2011, pp. 440–445.

[11] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "Photoocr: Reading text in uncontrolled conditions," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 785–792.

[12] S. Tian, *et al.*, "Text flow: A unified text detection system in natural scene images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4651–4659.

[13] J. J. Lee, P. H. Lee, S. W. Lee, A. Yuille, and C. Koch, "Adaboost for text detection in natural scene," in *Proc. Int. Conf. Document Anal. Recognit.*, 2011, pp. 429–434.

[14] S. M. Hanif and L. Prevost, "Text detection and localization in complex scene images using constrained adaboost algorithm," in *Proc. 10th Int. Conf. Document Anal. Recognit*, 2009, pp. 1–5.

[15] Y.-F. Pan, X. Hou, and C.-L. Liu, "A hybrid approach to detect and localize texts in natural scene images," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 800–813, Mar. 2011.

[16] 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," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1631–1639, Dec. 2003.

[17] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 1–20, 2016.

[18] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Proc. Eur. Conf. Comput. Vis*, 2016, pp. 56–72.

[19] T. He, W. Huang, Y. Qiao, and J. Yao, "Text-attentional convolutional neural network for scene text detection," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2529–2541, Jun. 2016.

[20] A. Zamberletti, L. Noce, and I. Gallo, "Text localization based on fast feature pyramids and multi-resolution maximally stable extremal regions," in *Proc. Asian Conf. Comput. Vis*, 2014, pp. 91–105.

[21] H. I. Koo and D. H. Kim, "Scene text detection via connected component clustering and nontext filtering," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2296–2305, Jun. 2013.

[22] X.-C. Yin, W.-Y. Pei, J. Zhang, and H.-W. Hao, "Multi-orientation scene text detection with adaptive clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1930–1937, Sep. 2015.

[23] C. Mancas-Thillou and B. Gosselin, "Spatial and color spaces combination for natural scene text extraction," in *Proc. IEEE Int. Conf. Image Process.*, 2006, pp. 985–988.

[24] C. Mancas-Thillou and B. Gosselin, "Color text extraction with selective metric-based clustering," *Comput. Vis. Image Understanding*, vol. 107, no. 12, pp. 97–107, 2007.

[25] C. Yi and Y. L. Tian, "Text string detection from natural scenes by structure-based partition and grouping," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2594–2605, Sep. 2011.

[26] N. Nikolaou and N. Papamarkos, "Color reduction for complex document images," *Int. J. Imag. Syst. Technol.*, vol. 19, no. 1, pp. 14–26, 2009.

[27] P. Shivakumara, W. Huang, and C. L. Tan, "Efficient video text detection using edge features," in *Proc. 19th Int. Conf. Pattern Recognit.*, 2008, pp. 1–4.

[28] J. Park *et al.*, "Automatic detection and recognition of korean text in outdoor signboard images," *Pattern Recognit. Lett.*, vol. 31, no. 12, pp. 1728–1739, 2010.

[29] R. Huang, P. Shivakumara, and S. Uchida, "Scene character detection by an edge-ray filter," in *Proc. 12th IEEE Int. Conf. Document Anal. Recognit.*, 2013, pp. 462–466.

[30] L. Wu, P. Shivakumara, T. Lu, and C. L. Tan, "A new technique for multi-oriented scene text line detection and tracking in video," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1137–1152, Aug. 2015.

[31] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*, 2010, pp. 2963–2970.

[32] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vis. Comput.*, vol. 22, no. 10, pp. 761–767, 2004.

[33] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Proc. Asian Conf. Comput. Vis*, 2010, pp. 770–783.

[34] M.-C. Sung, B. Jun, H. Cho, and D. Kim, "Scene text detection with robust character candidate extraction method," in *Proc. 13th IEEE Int. Conf. Document Anal. Recognit.*, 2015, pp. 426–430.

[35] W. Huang, Y. Qiao, and X. Tang, "Robust scene text detection with convolution neural network induced MSER trees," in *Proc. Eur. Conf. Comput. Vis*, 2014, pp. 497–511.

[36] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3538–3545.

[37] X. Huang and H. Ma, "Automatic detection and localization of natural scene text in video," in *Proc. 20th IEEE Int. Conf. Pattern Recognit.*, 2010, pp. 3216–3219.

[38] X. Zhao *et al.*, "Text from corners: A novel approach to detect text and caption in videos," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 790–799, Mar. 2011.

[39] M. Busta, L. Neumann, and J. Matas, "Fastext: Efficient unconstrained scene text detector," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1206–1214.
[40] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 105–119, Jan. 2010.
[41] W. S. Mokrzycki and M. Tatol, "Color difference delta e—A survey," *Faculty of Appl. Informat. Mathemat. Warsaw Univ. Life Sci.*, 2011.
[42] Y. Song *et al.*, "Robust and parallel Uyghur text localization in complex background images," *Mach. Vis. Appl.*, vol. 28, pp. 755–769, 2017.
[43] S. Fang *et al.*, "Detecting Uyghur text in complex background images with convolutional neural network," *Multimedia Tools Appl.*, vol. 76, no. 13, pp. 1–21, 2017.
[44] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

Chenggang Yan received the B.S. degree in computer science from Shandong University, Jinan, China, in 2008 and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2013. He is currently a Professor with Hanzhou Dianzi Univeristy, Hangzhou, China. Before that, he was an Assistant Research Fellow with Tsinghua University. His research interests include machine learning, image processing, computational biology, and computational photography. He has authored or co-authored more than 30 refereed journal and conference papers. As a co-author, he received got the Best Paper Awards in International Conference on Game Theory for Networks 2014, and SPIE/COS Photonics Asia Conference 9273 2014, and the Best Paper Candidate in International Conference on Multimedia and Expo 2011.

Hongtao Xie received the Ph.D. degree in computer application technology from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2012. He is currently a Research Professor with the School of Information Science and Technology, University of Science and Technology of China, Hefei, China. His research interests include multimedia content analysis and retrieval, similarity search, and parallel computing.

Jianjun Chen received the B.E. and Master's degrees in software engineering from Changsha University of Science and Technology, Changsha, China. He is currently working toward the Ph.D. degree at the School of Cyber Security, Institute of Information Engineering, University of Chinese Academy of Sciences, Beijing, China. His research interests include multimedia content analysis and retrieval and pattern recognition.

Zhengjun Zha received the B.E. and Ph.D. degrees from the University of Science and Technology of China, Hefei, China, in 2004 and 2009, respectively. He is currently a Full Professor with the School of Information Science and Technology, University of Science and Technology of China, the Vice Director of National Engineering Laboratory for Brain-Inspired Intelligence Technology and Application. He was a Researcher with the Hefei Institutes of Physical Science, Chinese Academy of Sciences, from 2013 to 2015, a Senior Research Fellow with the School of Computing, National University of Singapore (NUS), from 2011 to 2013, and a Research Fellow there from 2009 to 2010. He has authored or coauthired more than 100 papers in these areas with a series of publications on top journals and conferences. His research interests include multimedia analysis, retrieval and applications, as well as computer vision etc. Prof. Zha was the recipient of multiple paper awards from prestigious multimedia conferences, including the Best Paper Award and Best Student Paper Award in ACM Multimedia, etc.

Xinhong Hao received the B.S., M.S., and the Ph.D degrees in mechatronic engineering from Beiing Institute of Tecnology (BIT), Beijing, China, in 1996, 1999, and 2007, respectively. She is currently an Associate Professor with BIT. Her main research interests include signal processing and pattern recognition.

Yongdong Zhang (M'08–SM'13) received the Ph.D. degree in electronic engineering from Tianjin University, Tianjin, China, in 2002. He is currently a Professor with the School of Information Science and Technology, University of Science and Technology of China, Hefei, China. His current research interests include multimedia content analysis and understanding, multimedia content security, video encoding, and streaming media technology. He has authored more than 100 refereed journal and conference papers. Dr. Zhang was a recipient of the Best Paper Awards in PCM 2013, ICIMCS 2013, and ICME 2010, the Best Paper Candidate in ICME 2011. He is an Editorial Board Member of Multimedia Systems Journal and Neurocomputing.

Qionghai Dai (SM'05) received the B.S. degree in mathematics from Shanxi Normal University, Xian, China, in 1987, and the M.E. and Ph.D. degrees in computer science and automation from Northeastern University, Shenyang, China, in 1994 and 1996, respectively. He has been a Facculty Member with the Tsinghua University, Beijing, China, since 1997. He is currently a Cheung Kong Professor with Tsinghua University and is the Director of the Broadband Networks and Digital Media Laboratory. His current research interests include signal processing and computer vision and graphics.