Tracking Based Multi-Orientation Scene Text Detection: A Unified Framework With Dynamic Programming

Chun Yang, Xu-Cheng Yin, *Senior Member, IEEE*, Wei-Yi Pei, Shu Tian, Ze-Yu Zuo, Chao Zhu, and Junchi Yan

***Abstract*— There are a variety of grand challenges for multi-orientation text detection in scene videos, where the typical issues include skew distortion, low contrast, and arbitrary motion. Most conventional video text detection methods using individual frames have limited performance. In this paper, we propose a novel tracking based multi-orientation scene text detection method using multiple frames within a unified framework via dynamic programming. First, a multi-information fusion-based multi-orientation text detection method in each frame is proposed to extensively locate possible character candidates and extract text regions with multiple channels and scales. Second, an optimal tracking trajectory is learned and linked globally over consecutive frames by dynamic programming to finally refine the detection results with all detection, recognition, and prediction information. Moreover, the effectiveness of our proposed system is evaluated with the state-of-the-art performances on several public data sets of multi-orientation scene text images and videos, including MSRA-TD500, USTB-SV1K, and ICDAR 2015 Scene Videos.**

***Index Terms*— Scene text detection, tracking-based text detection, multi-orientation scene text, dynamic programming.**

I. Introduction

THE EXPLOSIVE growth of smart phones and online social media has led to the accumulation of large amounts of visual data, in particular, the massive and increasing collections of scene videos. Here, scene text (signage-text) is widely used as visual indicators for navigation and notification, and text detection and recognition from scene videos is one key factor for a variety of practical applications with reading in the wild [1], [2], such as assisting for visually impaired people [3]–[8], real-time translation [9]–[12], user navigation [13], traffic monitoring [14], [15], driving assistance systems [16], [17] and autonomous mobile robots [18], [19].

There are a variety of grand challenges for text detection in scene videos (see samples in Fig.1), where the typical issues are as follows. Firstly, in most cases of reading text in the wild, the captured scene text always exhibits multiple orientations and perspective distortions [20] (skew distortion). Secondly, compared with documents and scene images, scene text in video is always blurred and video frames usually have a lower resolution (low contrast). Thirdly, scene text in video sometimes moves in complex non-linear ways when the camera is zooming in or out, or rotating (arbitrary motion).

In the literature, for detecting scene text with skew distortion, some researchers have used specific grouping strategies [22]–[25] or designed robust image/text features [26]–[29]. To deal with low contrast, some researchers proposed deblurring models enhancing the edge intensity [30], or region based methods by using text/non-text classifiers to search possible text regions over windows [31], [32]. However, most of the above methods are only based on one channel (gray) and one scale (original size), which result in the missing of some important characters because of skew distortion and low contrast. In the cases of text with arbitrary motion, text detection using individual frames is failed because of introducing a variety of incorrect and noisy candidates. Consequently, only a very few researchers designed specific text detection techniques by utilizing spatial and temporal information and tracking text using multiple frames [12], [13], [33], [34], however, these methods do not uniformly integrate detection, prediction, tracking and their interactions.

In this paper, to address the problem of missing characters in scene images and frames due to skew distortion and low contrast, we propose a robust and precise multi-orientation text detection method which can extensively locate possible characters with multi-channel and multi-scale information fusion. In this method, an adaptive multi-channel character grouping method is proposed to robustly extract all possible character candidates, and an effective hybrid filter with Convolution Neural Networks, AdaBoost and Bayesian classifiers is designed to precisely verify the extracted text regions. Furthermore, to address the problem of incorrect and noisy candidates because of low contrast and arbitrary motion, we propose a novel tracking based scene text detection system, which combines detection, tracking and recognition by learn-

Manuscript received June 11, 2016; revised October 13, 2016, January 23, 2017, and March 12, 2017; accepted April 10, 2017. Date of publication April 18, 2017; date of current version May 9, 2017. The work was supported by the National Natural Science Foundation of China under Grant 61473036. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Peter Tay. *(Corresponding author: Xu-Cheng Yin.)*

C. Yang, W.-Y. Pei, S. Tian, and C. Zhu are with the Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China.

X.-C. Yin is with the Department of Computer Science and Technology, and also with the Beijing Key Laboratory of Materials Science Knowledge Engineering, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China (e-mail: xuchengyin@ustb.edu.cn).

Z.-Y. Zuo is with weibo.com, Beijing 100193, China.

J. Yan is with the Department of Computer Science and Technology, East China Normal University, Shanghai 200062, China, and also with IBM China Research Center, Shanghai 201203, China.

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/TIP.2017.2695104

ing globally in a unified integration framework via dynamic programming. In our system, the proposed multi-orientation text detection method is firstly performed locally in individual frames. The multi-strategy tracking techniques, e.g., tracking-by-detection, spatial-temporal context learning and template matching, are then used to predict the candidate text position in consecutive frames. Next, the tracking trajectories are linked by all detection, recognition and prediction information using a tracking network (graph), where the vertex weights are derived from both detection and recognition confidences, and the edge weights are based on the similarities between the current text block and the predicted ones. Thereafter, an optimal trajectory is learned globally in this network via dynamic programming. With this trajectory, the final detection and tracking results are simultaneously and immediately obtained. Moreover, our proposed system is verified on a variety of public scene text video databases, including MSRA-TD500 [22], USTB-SV1K [25] and ICDAR 2015 Scene Videos [21]. Experimental results show that our approach outperforms the state-of-the-art methods on all datasets. Specifically, our proposed method won the first place of "Video Text Detection" in the ICDAR 2015 Robust Reading Competition [21].

The main contributions of this work can be summarized as follows:

- Different from most conventional video text detection methods using individual frames, we develop a novel tracking based scene text detection approach using multiple frames, which combines detection, tracking and recognition by learning globally in a unified integration framework. Though it integrates several off-the-shell modules, text detection, tracking and recognition, our method provides a unified framework via dynamic programming for combining a variety of local and global information from detection, prediction and recognition with tracking techniques. To our best knowledge, this unified framework is proposed for text detection in scene videos for the first time in the literature.
- Different from most conventional multi-orientation scene text detection methods using one single channel or scale, we propose a robust multi-orientation scene text detection method with multi-information fusion and powerful text filtering, which extensively locate character candidates and detect text regions using multiple channels and multiple scales.
- Different from some recent tracking based video text detection methods using rule-based techniques, in this unified framework, we propose a dynamic programming based method to globally search the optimal results from the tracking trajectories with a weighted graph. These trajectories are globally and adaptively linked by all detection, recognition and prediction using a tracking graph.

The rest of this paper is organized as follows.[1] Related work is described in Section II. In Section III, we present the system overview of our proposed tracking based multi-orientation scene text detection system. Section IV and V describe the two major parts of the system, i.e., multi-orientation scene text detection with multi-information fusion and tracking based scene text detection methods in details, respectively. Comparative experiments are demonstrated in Section VI. Final remarks are presented in Section VII.

[1]Parts of this work previously appeared in [35]. Here, we present for the first time the entire system of tracking based scene text detection by extending text detection algorithms, including important implementation details, and adding related experimental results.

Fig. 1. Sample frames from ICDAR'15 dataset [21], where scene text always exhibits skew distortion, low contrast and arbitrary motion.

II. Related Work

There are numerous methods for scene text detection in images and videos, where multi-orientation scene text detection and tracking based scene text detection are two major topics of our system. In this section, we will review the related methods focusing on these two topics. Some other scene text detection methods can be referred to the recent survey papers [1], [2].

A. Multi-Orientation Scene Text Detection

One fundamental difficulty for detecting multi-orientation text is that the text line alignment (the axis-oriented assumption of a text line) feature can no longer be used to regularize the text construction, while most current clustering- or rule-based methods often rely on such information for character grouping and line construction [36]–[39] because the bottom alignment is the key and most stable feature for text lines [39]. Another main challenge is that in arbitrary orientations, it is complicated to train character and text classifiers, which results in the missing of some important characters.

To address the above problems, researchers have proposed a few non-horizontal scene text detection methods. Existing methods for multi-orientation scene text detection can roughly be categorized into two groups: grouping oriented methods [22]–[25] and feature oriented methods [26]–[29]. Grouping oriented methods use specific grouping techniques to locate and extract multi-orientation characters and text regions. Yao et al. proposed a multi-orientation scene text detection system by bottom-up grouping and top-bottom pruning techniques however with empirical rules [22], where pixels are first grouped into connected components (character candidates) using Stroke Width Transform (SWT) [40], character candidates are then linked to chains (text regions) by clustering, and text regions are finally verified by the discriminative classifiers. Later, they extended this work to

an end-to-end multi-orientation scene text recognition system [23]. Kang et al. treated orientation text line detection as a graph partitioning problem [24], where weak hypothesises are proposed by coarsely grouping Maximally Stable Extremal Regions (MSERs) [41] based on their spatial alignment and appearance consistency, and higher-order correlation clustering is used to partition the MSERs into text line candidates. Yin et al. designed a heuristic search and grouping strategy (forward-backward algorithm) to locate text lines, compute the line orientations and convert multi-orientation text lines into horizontal lines, before using their robust scene text detection technology [39], [42]. Recently, Yin et al. proposed a coarse-to-fine grouping method to sequentially group and detect multi-orientation scene text lines: morphology-based grouping, orientation-based grouping, and projection-based grouping all via adaptive hierarchical clustering [25].

Feature oriented methods involve designing orientation-invariant or robust image/text features to identify characters, words and lines. Shivakumara et al. combined edge and gradient features for multi-orientation scene text detection in video frames, and designed gradient vector flow based features to identify the dominant edge pixels of text components [26]. Risnumawan et al. introduced a robust arbitrary text detection system by identifying text pixel candidates with orientation-invariant features, e.g., mutual direction symmetry, mutual magnitude symmetry, and gradient vector symmetry [27]. Recently, Liang et al. proposed multi-spectral fusion strategies for both image enhancement and text pixel candidate detection of multi-orientation scene text detection in video [28], where convolving Laplacian with Wavelet sub-bands at different levels in the frequency domain is conducted for enhancing low resolution text pixels, and results from different sub-bands are fused for detecting candidate text pixels. Khare et al. designed histogram oriented moments features for text detection in video, which are invariant to rotation, scaling, and font variations [29]. By focusing on low-level character detection and character/text-line filtering, Huang et al. proposed a scene text detection method using stroke feature transform and text covariance descriptors [43]. Similarly, Li et al. designed "characterness" (an indicator of scene text) for character detection and filtering [44]. Recently, He et al. proposed text-attentional convolutional neural networks with multi-task learning by extracting text-related regions and features from the (text) image components [45].

All these methods are only based on one channel (gray) and one scale (original size), which always results in the missing of quite a few important characters and then text regions. To overcome the limitation of text detection only in one channel, multi-orientation text detection in different channels and scales should be exploited. However, only very few research efforts have been conducted in the literature. For example, Neumann and Matas presented an end-to-end real-time scene text localization and recognition system, where characters are detected and combined from multiple channels using knowledge-based rules [46]. Wang et al. designed a multi-channels scene text detection method [47], where connected component segmentation and character extraction are first conducted in each channel, and then the text component results in each channel are merged and grouped into words. Unfortunately, a fair part of words may be forced to separate by the fusion process in their method. In this paper, we propose a multi-information fusion based character grouping method to adaptively select and merge MSER components in different channels and scales. The merging process is performed on the connected component-level. Hence, almost every text region can be robustly extracted based on a variety of structured information with multiple channels or scales.

B. Tracking Based Scene Text Detection

Conventional methods for scene text detection in video mainly focus on detecting text in each individual frame or in some key frames. However, these methods cannot obtain high detection accuracy due to skew distortion, low contrast and arbitrary motion. It is worth noting that a key characteristic of video text is temporal redundancy. Thus, a variety of text tracking methods have been proposed for scene text detection in video [1], [2] i.e., text tracking is introduced in the detection process to reduce false alarms and improve the accuracy of detection. These strategies are called as tracking based text detection methods. In general, these methods can be categorized into temporal-spatial information based methods and fusion based methods. The former methods directly use temporal or spatial information to remove noises. The latter methods focus on merging detection and tracking results.

For temporal-spatial based methods, the temporal and spatial information is directly used to reduce false alarms in video text extraction, such as the duration of the text, i.e., the interval between the starting frame and the ending frame of the same text. With text tracking techniques, the text trajectory is constructed. This trajectory is utilized to decide whether the tracked text is effective [5]–[7]. The text trajectory is accepted as valid only if eventually its length exceeds a given threshold value; otherwise, it is regarded as noise and is discarded. For example, a region that has been detected or tracked should be selected in at least 3 individual frames [5]–[7].

For fusion based methods, integrating detection text results in frames with those previously tracked and the output is used to enable false positive suppression [13], [33]. In [13], the position, size and color histogram information of a detected and tracked text region are computed, and matching is performed by the Hungarian algorithm. Tracked text regions that are matched or have a positive score are selected, and the remaining regions are discarded. In [33], the tracked text line is updated by newly detected MSERs, which thus regenerates the tracking process. Recently, Zuo et al. [49] proposed a multi-strategy tracking based text detection method in scene videos, where tracking-by-detection, spatial-temporal context learning, and linear prediction are all performed to predict the candidate text location sequentially and the best matching text

[2]The goal of text tracking is to continuously determine the location of text across multiple dynamic video frames. Text tracking is useful for verification, integration, enhancement and speedup in video text detection and recognition e.g., tracking based detection and tracking based recognition. In recent years, a variety of text tracking methods have been investigated in the literature. More related details about text tracking can be referred to [1], [48].

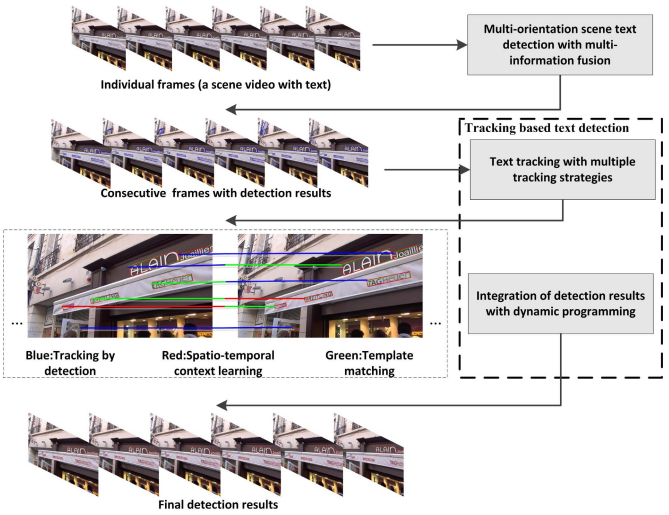


Fig. 2. Flowchart of our framework for tracking based multi-orientation scene text detection which includes text detection (in individual frames), text tracking, and integration of detection results. And the last two steps consist of the main part of tracking based text detection.

block from the candidates is adaptively selected using a rule-based method.

Most of the above tracking based methods utilize a specific tracking technique to trivially track text and heuristically combine text detection results. These methods still have limited performance because of grand challenges (e.g., skew distortion, low contrast and arbitrary motion) for scene text detection in video. In this paper, we propose a novel tracking based text detection system in scene videos, which combines detection and tracking uniformly by learning and searching globally with dynamic programming. Here, tracking trajectories are linked with all detection, recognition and prediction information with a tracking graph.

III. Framework Overview

The flowchart of our proposed framework for tracking based multi-orientation scene text detection is shown in Fig.2, which includes three major steps, i.e., text detection with multi-information fusion, text tracking with multiple tracking strategies, and integration of detection results with dynamic programming. The last two steps are also the key components of tracking based text detection (Section V).

- ***Text Detection with Multi-Information Fusion***: Firstly, in individual frames, multi-orientation scene text detection with multi-information fusion is conducted and possible text region candidates are localized and extracted extensively and locally. Here, a multi-channel character grouping method extracts all possible character candidates, an AdaBoost classifier and a map verifier identify these candidates as characters or non-characters, single clustering with distance metric learning thereafter groups characters into text regions, and an hybrid filter with Convolution Neural Networks, AdaBoost and Bayesian classifiers verifies the extracted text regions. More details are presented in Section IV.
- ***Text Tracking with Multiple Tracking Strategies***: Secondly, in consecutive frames, a variety of tracking approaches [49] (e.g., tracking-by-detection, spatial-temporal context learning and template matching) are performed to predict the candidate text position in next frames. A multi-strategy tracking approach including tracking by detection, Spatio-Temporal Context Learning(STCL) [50] and template matching is utilized to search and fuse different detection results. Here, the tracking by detection method makes use of detection outputs to initialize new trajectories and amend tracking outputs. To improve the detection recall, the template matching and STCL are used to predict more possible text region positions, where template matching is able to tackle text blur challenges, but it cannot handle multi-scale challenges. Complementary to template matching, STCL is applied to deal with multi-scale text. More details are presented in Section V-A.
- ***Integration of Detection Results with Dynamic Programming***: Finally, all detection and tracking results in consecutive frames are combined with dynamic programming to obtain the preferred results in an optimal tracking trajectory by globally and uniformly integrating detection, tracking, recognition and their interactions. Here, the tracking trajectories are linked with all detection, recognition and prediction information using a tracking network (graph), where the vertex weights are derived from both detection and recognition confidences, and the edge weights are based on the similarities between the current text block and the predicted ones. An optimal trajectory is learned globally in this network with a dynamic programming algorithm. More details are presented in Section V-B.

IV. Multi-Orientation Scene Text Detection With Multi-Information Fusion

Our multi-orientation text detection method extensively locates possible character candidates and extracts text regions with multiple channels and scales. Following the generic procedure of state-of-the-art scene text detection approaches [25], [36], [39], [42], our method includes four major components as follows (Some empirical results are shown in Fig.3):

- ***Character Candidates Extraction***: Character candidates are first extracted from MSERs in multiple color channels and scales. Details are presented in Section IV-A.
- ***Character Candidates Verification and Combination***: Character candidates are verified by a character classifier and a map verifier in each channel (scale), where the map verifier is derived from character candidates grouping hierarchical clustering. Noise character candidates are removed, and character components with high confidences in different channels (scales) are then selected and combined. Details are presented in Section IV-B.
- ***Text Region Candidates Construction***: Same to [25], character candidates (selected and combined from the previous step: character candidates verification and combination) are then grouped into text candidates from

Fig. 3. Sample results (from left to right columns) for text detection with multi-information fusion for (1) the original image, (2) character candidates extraction with MSERs in Gray, Green and Y_B channels (in the first row), (3) character candidates verification and combination, (4) text region candidates construction [25], and (5) the final detection results (after text region candidates filtering) (in the second row).

different channels (scales) by a coarse-to-fine character grouping step: morphology-based grouping, orientation-based grouping, and projection-based grouping all via adaptive hierarchical clustering. Details can be referred to [25].

- ***Text Region Candidates Filtering***: A hybrid filter with Convolution Neural Networks (CNN), AdaBoost and Bayesian classifiers is designed to finally filter text region candidates, i.e., to determine whether a text region candidate is text or not. The Bayesian filter and the AdaBoost classifier come from [39] and [51] respectively. Details are presented in Section IV-C.

Note that the text detection method with multi-information fusion by sequentially integrating several major steps (algorithms) will generally decrease the robustness and adaptation of the algorithms. As a result, there are many general techniques for such methods with sequent steps for text detection in the literature. Specifically, in our proposed approach, several techniques are designed and utilized for improving robustness. Here, character candidates verification and combination with a character classifier and a map verifier, and text region candidates filtering with a powerful text filter are utilized in our system to improve the robustness of the whole text detection method.

A. Character Candidates Extraction

MSER [41] is a technique for image segmentation in complex images, and also an effective way to extract connected components for character candidates in scene images. However, it often fails to extract character candidates with one (color) channel in scene images and videos when the text is designed with different colors, or with low contrast to the background, or affected by strong light. Fortunately, in most cases, almost all character shapes can be detected in one certain channel and one certain scale (not just the gray channel with the original size used in most conventional methods). For example, in Fig.3, MSERs in Gray, Green and Y_B channels correspond to different character candidates, and these character candidates can be combined to group text regions. Consequently, we design a text detection method with multi-information fusion from different channels.

In our work, an image is first converted into several sub images with different channels and scales. Here, gray, red, green, blue, YUV, et al. channels and different scales can be conveniently employed. Empirically, gray, green, and Y_B, and 1 and 2-time scales are found and effective in our system. Text in the gray channel is easily detected in common cases; red and yellow text in the green channel can be localized; and non-text with varied-illumination can be easily identified. Then, the MSERs algorithm with a minimizing regularized variations pruning strategy [39] is used to extract connected components (CCs) in each channel and scale.

B. Character Candidates Verification and Combination

First, an AdaBoost classifier is used to determine whether an MSER component is a character [39]. The confidence of a component is defined as follows.

$$\text{conf}(comp_i^{ch}) = \begin{cases} (0.5, 1] & \text{character} \\ [0, 0.5] & \text{non-character} \end{cases} \tag{1}$$

where $comp_i^{ch}$ is the i^{th} component in channel ch. A component is determined as a character when the confidence is greater than 0.5; otherwise it is a non-character. The higher the confidence is, the greater the probability for the component being a character will be.

Second, an MSER component (after being verified by the AdaBoost classifier) is again verified by considering the relations of adjacent character candidates (a map verifier). The coarse-to-fine strategy sequentially groups multi-orientation text region candidates [25], where the hierarchical clustering approach with distance metric learning is used to extract components $CCs^{ch} = \{comp_i^{ch} | i = 1 \cdots n\}$ in each channel (scale). The components with similar features are grouped in the same cluster, and the effectiveness of one component can be represented as the confidence of its cluster,

$$\begin{aligned} \text{eff}(comp_i^{cluster_j}) &= \text{conf}(cluster_j^{ch}) \\ &= \frac{\sum \text{conf}(comp_k^{cluster\_j})}{||cluster_j^{ch}||} \end{aligned} \tag{2}$$

where $cluster_j^{ch}$ is a cluster in channel (or scale) ch, $comp_i^{cluster\_j}$ is a component belonging to $cluster_j^{ch}$, and $||cluster_j^{ch}||$ is the number of components in $cluster_j^{ch}$.

Specifically, the difference features between two character components u and v are defined as a vector (for the clustering process),

$$\begin{aligned} f(u, v) = \{&d_w(u, v), d_h(u, v), d_{interval}(u, v), \\ &d_{stroke}(u, v), d_{color}(u, v)\} \end{aligned} \tag{3}$$

where the difference of measures like width, height, interval, stroke width and color are defined as follows,

$$d_w(u, v) = \frac{\max(w_u, w_v) - \min(w_u, w_v)}{\max(w_u, w_v)} \tag{4}$$

$$d_h(u, v) = \frac{\max(h_u, h_v) - \min(h_u, h_v)}{\max(h_u, h_v)} \tag{5}$$

$$d_{interval}(u, v) = \frac{\sqrt{(x_u - x_v)^2 + (y_u - y_v)^2}}{r_u + r_v} \tag{6}$$

$$d_{stroke}(u,v) = \frac{|sw_u - sw_v|}{\max(sw_u, sw_v)} \tag{7}$$

$$d_{color}(u,v) = \max(|c1_u - c1_v|, |c2_u - c2_v|, |c3_u - c3_v|) \tag{8}$$

where (x_u, y_u) is the centroid of u's circumcircle, r_u is the radius of its circumcircle, sw_u is the mean value of the stroke width of u, h_u and w_u is the height and width of the bounding rectangle of u, and $c1_u, c2_u, c3_u$ is the average values in three color channels of u. Here, the stroke width of a character candidate is computed by averaging over its pixels' stroke widths (see [51]). Therefore, the weight vector of the difference features f(u,v) is $w = \{w_w, w_h, w_{interval}, w_{stroke}, w_{color}\}$ which is automatically learned by a metric learning framework [25], and the distance is converted as,

$$d(u, v : w) = w^T f(u, v) \tag{9}$$

Note that these features (width, height, interval and stroke width from Equation 3 to Equation 8) are directly calculated from the character candidates after character candidates extraction in each channel with each scale.

Then, the fusion strategy is simply to select character candidates which have high effectiveness (in Equation (2)) from different channels (scales). As descried at the beginning of this section, these selected and combined character candidates are then grouped into text region candidates by a coarse-to-fine grouping technique.

As shown in Fig.3, the components extracted in the Y_B channel remedy well for the ones in the gray channel. Similarly, the missing characters in one channel can be found in other channels. Thus the MSERs components extracted by the multi-channel fusion method can result in a better performance for scene text extraction than using one single channel.

Note that two character components may be conflicted in some cases, i.e., there will be two or more text region candidates extracted from different channels for one real text region. But, only one should be reserved. Here, if the overlap of these text components is more than 80% on pixels, the one with greater confidence will be reserved in our method; otherwise, the text component is processed in a general procedure as shown in in Fig.4.

C. Text Region Candidates Filtering

As mentioned before, more character candidates will be extracted with the multi-information fusion method. At the same time, the number of negative candidates (non-character components being regarded as characters) will also increase inevitably. Consequently, the number of negative text regions generated by these negative candidates will also increase. To eliminate these negative regions, we design a powerful hybrid filter with CCs-based filters and a region-based filter one after another to filter text with a high efficient style. The CCs-based filters include a Bayesian classifier for computing the post-probability of text and non-text [39] and an AdaBoost classifier for identifying text regions [51], which are used to deal with the text region candidates with the uniform distribution of characters and a noise-free situation. Specifically, the CCs-based filters can precisely filter non-text region candidates

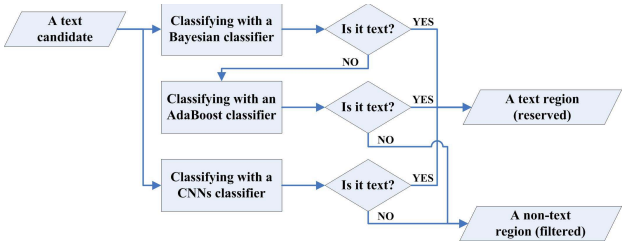


Fig. 4. Combination framework of text region candidates filtering, where the Bayesian classifier and the AdaBoost classifier address text region candidates with uniform characters and clear backgrounds, and the CNN classifier focuses on text region candidates with varied characters and complex backgrounds.

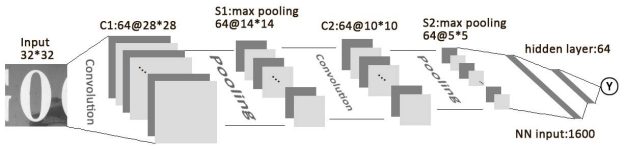


Fig. 5. Architecture of the Convolution Neural Networks (CNN) with two convolutional layers, two max pooling layers and a full connection layer.

in normal cases where characters have similar shapes and arrangements with clear backgrounds. In contrast, the region-based filter (a CNN classifier) focuses on the text region candidates with varied characters and complex backgrounds (see samples in Fig. 7).

As shown in Fig.4, the candidate text will be reserved if two conditions are satisfied. One is that the confidence (output) of the Bayesian filter is greater than 0.9 or the category of the AdaBoost classifier is verified to be the text. The other is that the confidence (output) of the CNN based filter is greater than 0.4. The structures of the Bayesian filter and the AdaBoost classifier are the same as [39]. The CNN model is used to determine whether an image patch (with sliding window) is a character. Similar to AlexNet CNNs [52] used in scene text detection and recognition [53], [54], the network of the CNN filter consists of two convolutional layers and two max pooling layers (see Fig. 5). Here, a square sliding window is sequentially scanned and classified from the text region candidate by the CNN filter to determine whether this text region candidate is text. Each window is resized to 32×32. The confidence (output) of the text region candidate is computed as follows,

$$\text{conf}_{CNN}(Candidate) = \frac{||W\{w_i \geq \theta\}||}{||W||} \tag{10}$$

where W is the set of all sliding windows in a text region candidate, w_i the output of the CNN classifier for the i_{th} window, $||W||$ is the number of windows, and $||W\{w_i \geq \theta\}||$ is the number of windows which are determined to be positive by the CNN classifier. θ is an empirical threshold, and is set as 0.75 in our system.

V. Tracking Based Scene Text Detection With Dynamic Programming

The framework for tracking based scene text detection with dynamic programming is shown in Fig.6, the pipeline of which

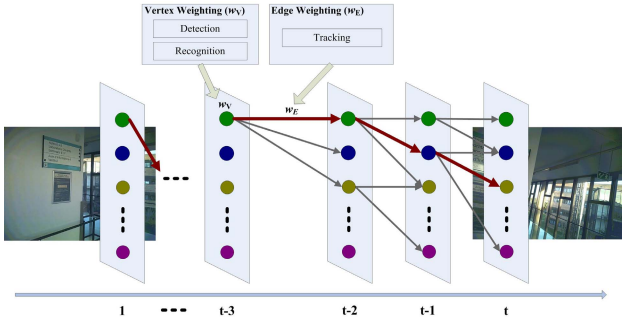


Fig. 6. The framework for tracking based scene text detection with dynamic programming, where w_V and w_E are vertices and edges' weights respectively of the tracking network (a weighted graph), and the RED line is an assumed optimal path by searching globally.

includes three major components. First, in individual frames scene text detection and recognition are conducted and possible text region candidates are localized and extracted extensively and locally, where multi-orientation scene text detection with multi-information fusion described above (Section IV) and a CNNs-based word recognition technique [54] are used. Second, in consecutive frames, a variety of tracking approaches are performed to predict the candidate text position in the next frame. Finally, all detection and tracking results in consecutive frames are combined with dynamic programming to obtain the preferred results in an optimal tracking trajectory by globally and uniformly integrating detection, tracking, recognition and their interactions.

This dynamic programming based method can also improve the system's robustness by globally searching and selecting the best matching from the candidate text regions. In conventional integration systems of sequent steps, if heuristic rules are used to combine different components and to locally select results, they may bring error accumulation to some extent. In our proposed method, in order to avoiding error accumulation, an optimal tracking trajectory is learned globally via dynamic programming for combining all detection, recognition and prediction information. Consequently, the method will select the optimal results (paths) globally, and the possible paths with error accumulation will be discarded in most cases.

Specifically, at the beginning, text detection and recognition are performed in individual frames. Then, there are three major steps in our tracking based scene text detection algorithm (shown in Algorithm 1). First, a text region is selected if it is not updated by any optimal trajectory before, i.e., the text region is initially derived from text detection. Second, this region is tracked in a sequence of frames by a multi-strategy tracking technique, i.e., tracking by detection, spatial-temporal context learning (STCL) [55] and template matching. Here, there are three tracking (prediction) positions in the next frame. A weighted graph is then constructed, where weights are from detection, prediction and recognition confidences. Consequently, text trajectories of this region are created dynamically. Third, an optimal trajectory is globally searched by dynamic programming in this weighted graph, as exemplified in Fig.6. Along this optimal trajectory, all related detection results for the target text region are updated. The above three steps

Algorithm 1 Tracking Based Scene Text Detection

Input: a video sequence with t frames $\{F_1, F_2, ..., F_t\}$
Output: updated detection results ($\{D_1^*, D_2^*, ..., D_t^*\}$) with optimal trajectories
Procedure:
detect scene text in individual frames $\{D_1, D_2, ..., D_t\}$
recognize text in individual frames $\{R_1, R_2, ..., R_t\}$
repeat
Stage 1: select a detected text region ($d_j^i \in D_i$) in frame $F_i (1 \leq i < t)$, if this detection result is not updated by any optimal trajectory
Stage 2: track this region in the next frames with tracking by detection, spatial-temporal context learning and template matching, and construct a weighted graph for this text region (see Section V-A).
Stage 3: search an optimal trajectory in the weighted graph with dynamic programming, and update all related detection results along this optimal trajectory (see Section V-B)
until all detection results (text regions) are updated by the optimal trajectories

are iteratively conducted until all detection results (from text detection) are updated by the optimal trajectories. Finally, text detection results can be selected and obtained from these updated results.

Note that several empirical rules can be used for speeding up. For example, for a new trajectory, the tracking by detection method is utilized to verify whether it is valid or not. The trajectory (called as the valid trajectory) is valid if it is matched in the first four consecutive frames; otherwise, it will be regarded as noise and discarded. Upon the processing of frame i (F_i), the trajectory is eliminated if there is no matching text region by a DCT-based hash algorithm [56] and the similarity of the color histogram. We also empirically prune the weighted graph for each detected text region. If the confidence from tracking by detection, STCL or template matching is below a threshold, the corresponding edge will be removed. Moreover, to reduce space and time complexity, as well as the time lag for on-line processing, a video sequence can be divided into sub sequences, and our method can be easily conducted on each sub sequence.

In the following, we describe the two major components of this tracking based text detection method: text tracking with multiple tracking strategies (Section V-A), and integration of detection results with dynamic programming (Section V-B).

A. Text Tracking

Text tracking is to determine the time that text regions appear and disappear, and the location of text continuously in dynamic video frames. Its main aim is to reduce the false alarms and to improve the detection accuracy in the dynamic scenes. Three tracking algorithms are used in our multi-strategy tracking method, namely, tracking by detection, STCL and template matching.

TABLE I
EXPERIMENTAL RESULTS ON MSRA-TD500

Method	Recall	Precision	f-score
Strategy I-1 (baseline)	0.60	0.82	0.69
Strategy I-2	0.52	**0.95**	0.67
Strategy I-3	**0.65**	0.75	0.70
Strategy I-4 (proposed)	0.58	**0.95**	**0.72**
He et al. [45] (2016)	0.61	0.76	0.69
Yin et al. [25] (2015)	0.63	0.81	0.71
Yin et al. [39] (2014)	0.61	0.71	0.66
Yao et al. [23] (2014)	0.62	0.64	0.61
Yao et al. [22] (2012)	0.63	0.63	0.60
Kang et al. [24] (2014)	0.62	0.71	0.66

The tracking by detection method makes use of detection results and is crucial for text trajectory creation and initialization. In the process of creating a text trajectory, the tracking-by-detection method associates detection results in consecutive frames to initialize new trajectories. To reduce the false alarms, we assume that the text is regarded as true text if it is detected in four consecutive frames. In the process of tracking the valid trajectory, the tracking-by-detection method is used to match the tracked outputs in the previous frame and detect outputs in the current frame. The key step in tracking-by-detection is the Hungarian algorithm which maps text regions in successive frames. Two types of features are used to calculate the similarity of text regions. The first type is the position information. We assume that if blocks belong to the same text, the difference in position is a normal distribution with zero mean, i.e.,

$$p_{pos}(t,d) = \frac{1}{\sqrt{2\pi}\sigma_p} exp(-\frac{d_{pos}(t,d)^2}{2\sigma_p^2}) \tag{11}$$

where t and d are text regions, $d_{pos}(t,d)$ is the Euclidean distance between the centroid of t and d, and σ_p is the distance variance. The other one is the color histogram information. Similarity, we assume that the difference of the same text is a normal distribution with zero mean, i.e.,

$$p_{col}(t,d) = \frac{1}{\sqrt{2\pi}\sigma_c} exp(-\frac{dHist(H_t,H_d)^2}{2\sigma_c^2}) \tag{12}$$

where t and d are text blocks, H_t and H_d are color histograms of t and d, $dHist(H_t,H_d)$ is the Bhattacharyya distance of H_t and H_d, and σ_c is the variance of $dHist(H_t,H_d)$. The final input of the Hungarian algorithm is the product of these two probabilities, i.e.,

$$p(t,d) = p_{pos}(t,d) \cdot p_{col}(t,d). \tag{13}$$

The STCL method is used to predict the position in the current frame of the detected or tracked text region in the previous frame in the valid trajectory. STCL uses a Bayesian framework to model the spatio-temporal relationships between the object interested and its surrounding regions. Thus, the tracking process is to compute a confidence map and obtain the best target location by maximizing an object location likelihood function [49].

Moreover, instead of linear prediction which is not suitable for arbitrary motion, a normalized correlation coefficient based template matching method is utilized to predict the position in the current frame of the detected or tracked text region in the previous frame in the valid trajectory. The normalized correlation coefficient is designed as

$$R(x,y) = \frac{\sum_{x',y'}(T'(x',y') \cdot I'(x+x',y+y'))}{\sqrt{\sum_{x',y'} T'(x',y')^2 \cdot \sum_{x',y'} I'(x+x',y+y')^2}} \tag{14}$$

where $R(x,y)$ is the estimated value of a point (x,y) in a result image R, I' and T' are the image and the sliding template respectively, and x' and y' are the positions of the sliding template T' in the image I'. At last, the location of a maximum value in R is the centroid of the prediction location. In our system, because the normalized correlation coefficient based template matching always returns a position even when the matched block does not contain text, a strategy of the combination of the similarity of the color histogram and a DCT-based hash algorithm [56] is proposed to determine whether the target text region and the predictive text region are similar or not. The trajectory will be eliminated if there is no matching in the tracking process.

Here, tracking-by-detection makes use of detection outputs to initialize new trajectories and amend tracking outputs, and usually has high accuracy. Template matching is used to address text blur and low contrast challenges, but fails to handle multi-scale challenges. STCL is applied to solve the problem of multi-scale text (with arbitrary motion). Both template matching and STCL can generally improve the recall performance. As a result, our multi-strategy tracking approach and the dynamic programming algorithm can combine advantages of these three tracking techniques and obtain good accuracy with fair recall.

Fig. 7. Experimental samples of our proposed methods on MSRA-TD500, where results of the baseline method, the proposed method without the hybrid filter, the proposed method with Bayesian and AdaBoost filters, and the entire proposed method (with all Bayesian, AdaBoost and CNN filters) are shown in the first, second, third and forth columns, respectively.

B. Integration of Detection Results

Instead of the rule-based technique in our previous method [49], we have developed a dynamic programming based method (an iterative improvement over [49]) to globally search and select the best matching from the candidate text regions for the target text. Moreover, in the process of multi-strategy tracking, each detected and tracked text region in the valid trajectory will be regarded as a node and added into the

TABLE II
EXPERIMENTAL RESULTS ON USTB-SV1K

Method	Recall	Precision	f-score
Strategy I-1 (baseline)	0.4970	0.4790	0.4878
Strategy I-2	0.4672	**0.5380**	0.5001
Strategy I-3	**0.5188**	0.4703	0.4934
Strategy I-4 (proposed)	0.4880	0.5369	**0.5112**
Yin et al. [25] (2015)	0.4541	0.4985	0.4753
Yin et al. [39] (2014)	0.4518	0.4500	0.4509
Yao et al. [23] (2014)	0.4405	0.4580	0.4491

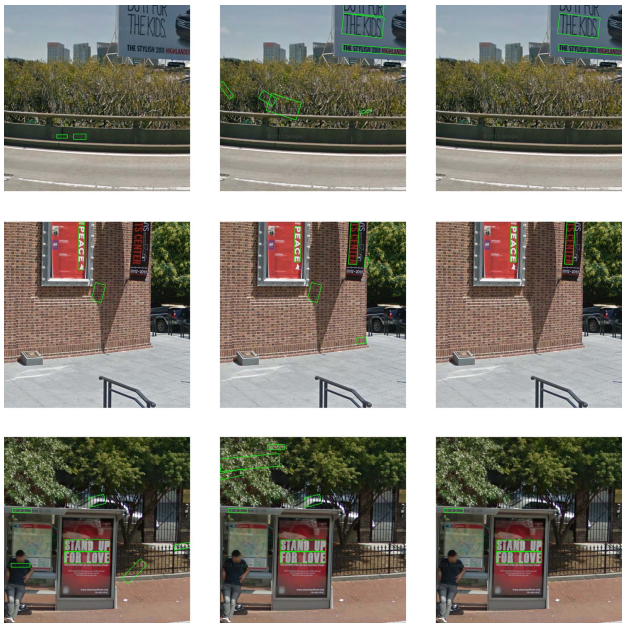


Fig. 8. Experimental samples of our method on USTB-SV1K, where results of the baseline method, the proposed method without the hybrid filter and the entire proposed method are shown in the first, second and third columns respectively.

network, and all the text regions corresponding to one target text region in consecutive frames consist of the nodes of the dynamic network (see Fig.6).

Specifically, this tracking network (a weighted graph) is first composed of detected and recognized text region candidates (vertices/nodes), predicted ones (vertices/nodes), and tracking trajectories (edges). In this graph, the vertex weights are derived from both detection and recognition confidences, and the edge weights are computed with the similarities between the current text block and its predicted ones. A dynamic programming algorithm is then performed on the weighted graph, and an optimal tracking trajectory is learned globally. Here, the Dijkstra algorithm for the shortest path problem in a weighted graph is employed. Afterwards, detection and tracking results in each frame are easily and directly obtained with this optimal trajectory.

Node weights and edge weights are two important parts for building the weighted graph. In our work, the combination of the detection confidence score and the recognition respond score [3] are used to compute the weight of the node,

$$W_V(n) = \alpha N_D(n) + (1-\alpha) N_R(n) \tag{15}$$

[3]In our system, a CNNs-based word recognition technique is used [54]. The recognition respond scores are directly derived from the outputs of the classifier.

TABLE III
PERFORMANCE OF THE CNN FILTER ON THE ICDAR2013 SET

	Positive samples		Negative samples	
	correct/incorrect	accuracy	correct/incorrect	accuracy
Train Set	30114/731	97.63%	30869/539	98.28%
Test Set	4622/608	88.38%	33984/586	98.31%

where $W_V(n)$ is the sum score of the node n, $N_D(n)$ and $N_R(n)$ are the detection confidence score and the recognition respond score of node n respectively, α is a weight coefficient between 0 and 1. The similarity of the color histogram combined with the edit distance is utilized to compute an edge weight between two nodes in the consecutive frames with

$$W_E(n_1, n_2) = \lambda P_{\text{color}}(n_1, n_2) + (1-\lambda) P_{\text{edist}}(n_1, n_2) \tag{16}$$

where n_1 and n_2 are two nodes in the consecutive frames, $W_E(n_1, n_2)$ is the sum similarity between two nodes, $P_{\text{color}}(n_1, n_2)$ and $P_{\text{edist}}(n_1, n_2)$ are the similarity of color histogram and edit distance of two nodes respectively, and λ is a weight coefficient between 0 and 1. The score of each node is the maximum score from the start node to the current node, i.e.,

$$\max Score(n_i^j) = \begin{cases} \max Score(n_{i-1}^k) + W_V(n_i^j) \\ \quad + W_E(n_{i-1}^k, n_i^j), & i > 1 \\ n_j^1, & i = 1 \end{cases} \tag{17}$$

where $\max Score(n_i^j)$ is the score of the j^{th} node in the i^{th} (video) frame (see Fig.6).

In the process of searching with dynamic programming, each text region including a detected or tracked text region (predicted text region) in the previous frame will be predicted in the current frame. If the total number of the predicted text regions for the same text region are more than 6 in a frame, we will select the text region with a higher score. Here the overlap between the selected text region and the other regions should be less than 95%. The text region with less similarity of the color histogram will be discarded when the overlap between the text regions is more than 95%. Finally, a node will be created with $W_V = 0$ when the trajectories are not matched in the tracking process. Thus, the nodes with the maximum score are determined as the text positions in the frames. After the optimal trajectory is determined, the final detection results are correspondingly selected on this trajectory.

VI. EXPERIMENTS

In this section, first we compare our approach with several recent state-of-the-art methods for multi-orientation text detection in scene images on several public datasets. Then, the experiments for scene text detection in scene videos are mainly conducted, compared and analyzed.

A. Experiments With Multi-Orientation Scene Text Detection

For the task of multi-orientation text detection in scene images, we choose two public multi-orientation scene text

TABLE IV
COMPARATIVE RESULTS FOR TEXT DETECTION IN SCENE VIDEOS (%)

Video	Minetto et al. [13]			Strategy II-1 (without tracking)			Strategy II-2 (Zuo et al. [49])			Strategy II-3 (proposed)		
	Precision	Recall	f-score	Precision	Recall	f-score	Precision	Recall	f-score	Precision	Recall	f-score
V1	0.55	0.80	0.63	0.73	0.64	0.68	0.82	0.62	0.71	0.82	0.70	0.76
V2	0.57	0.74	0.64	0.82	0.49	0.61	0.90	0.80	0.85	0.92	0.81	0.86
V3	0.60	0.53	0.56	0.66	0.55	0.60	0.75	0.60	0.67	0.73	0.64	0.68
V4	0.73	0.70	0.71	0.71	0.65	0.68	0.83	0.77	0.80	0.88	0.82	0.85
V5	0.60	0.70	0.63	0.66	0.59	0.62	0.88	0.62	0.72	0.89	0.87	0.88
average	0.61	0.69	0.63	0.68	0.62	0.64	0.84	0.68	0.75	0.85	0.77	**0.81**

datasets, MSRA-TD500 [22][4] and USTB-SV1K [25],[5] and follow the evaluation protocol in Yao's et al's work [22].

The MSRA-TD500 dataset is a multi-orientation dataset with 500 images where 300 images is for training and the rest is for testing. We compare our multi-channel fusion method with six state-of-the-art methods: two of Yin et al.'s methods [25], [39], two of Yao et al.'s methods [22], [23], Kang et al.'s method [24], and He et al.'s method [45]. For our proposed methods, we conduct the experiments with four settings, (1) Strategy I-1, the baseline method (with only gray channel and without the hybrid text region filter), (2) Strategy I-2, the baseline method with the hybrid filter, (3) Strategy I-3, the proposed method without the hybrid filter,[6] and (4) Strategy I-4, the proposed method. As shown in Table I, our proposed method without the hybrid filter achieves the highest recall (0.64) because the multi-information fusion strategy can extensively locate and extract character candidates in scene images. At the same time, more non-character components will be extracted and noisy text region candidates will also be constructed. In our proposed method, text region candidates filtering is performed for increasing largely the precision. As a result, our proposed method achieves the highest precision (0.92), and also the best (overview) performance ($f_{\text{measure}} = 0.72$).

Some typical challenging samples are shown in Fig.7. First, text in yellow with the red background has low contrast in gray scale (the third row). Thanks to the multi-information fusion strategy, text lines are localized correctly. In addition, text-like candidates, such as windows or grasses, are often regarded as text regions. The hybrid filter solves this problem with high accuracy. Specifically, the baseline method has challenges to detect text in scene images (shown in the first column) if the text is under complex backgrounds (the first row) and with low contrast (the second and third rows). On the contrary, the multi-information fusion strategy (results shown in the second column) can effectively detect these kinds of text. At the same time, it brings more noises inevitably. As a result, our powerful hybrid filter can pick up text region candidates with high accuracy (shown in the third and forth columns). In summary, our proposed method, multi-orientation scene text detection with multi-information fusion, performs much better than all the other methods.

[4]The MSRA-TD500 dataset is available at http://www.iapr-tc11.org/mediawiki/index.php/MSRA_Text_Detection_500_Database_(MSRA-TD500).

[5]The USTB-SV1K dataset is available at http://prir.ustb.edu.cn/TexStar/MOMV-text-detection/.

[6]Our proposed method without the hybrid filter means that the last step (text region candidates filtering in Section IV-C) of our multi-orientation scene text detection method is removed.

TABLE V
EXPERIMENTAL RESULTS (%) ON THE DATASET OF ICDAR 2015 ROBUST READING COMPETITION CHALLENGE 3 (TEXT DETECTION IN SCENE VIDEOS), WHERE "DEEP2TEXT I", "USTB-TEXVIDEO", "AJOU", "USTB-TEXVIDEO-II", "STRADVISION" AND "RTST-LUCASKANADE-2" ARE ALL PARTITION METHODS IN THE COMPETITION [21], AND "DEEP2TEXT I" IS THE WINNING METHOD OF OUR PARTITION TEAM

Method	MOTP	MOTA	ATA
Proposed method (Deep2Text I)	71.01	40.77	**45.18**
USTB-TexVideo	71.33	49.33	41.31
AJOU	73.25	53.45	38.77
USTB-TexVideo-II-2	72.47	50.38	35.71
StradVision	70.82	47.58	32.12
USTB-TexVideo-II-1	69.51	19.69	30.15
RTST-LucasKanade-2	64.44	-20.28	0.34

USTB-SV1K is a large practical challenging multi-orientation scene text dataset, which includes $1,000$ street view images of six typical USA cities directly crawled from Google Street View. These images have low resolution and are blurred artificially to some degree. Hence, it is a more challenging dataset for text detection. Here, similar to the settings on the MSRA-TD500 dataset, we compared our systems (i.e., Strategy I-1, the baseline method (with only gray channel and without the hybrid filter), Strategy I-2, the baseline method with the hybrid filter, Strategy I-3, the proposed method without the hybrid filter, and Strategy I-4, the proposed method) with some recent advanced methods. Similarly, the proposed method without filters achieves the highest recall because of the multi-information fusion strategy for extensively extracting character candidates. The entire system (with both multi-information fusion and the hybrid filter) achieves the best f-score (0.5112) performance.

In street view images, low resolution is a great challenge for multi-orientation text detection. The small text usually has low resolution, but it can be detected in a larger scale. As shown in Fig.8, some small text regions can be detected by our multi-information (multi-channel and multi-scale) fusion strategy. However, a part of images are severely distorted in USTB-SV1K. The hybrid filter in our method is sensitive to (severely) perspectively distorted text because the training set includes very few of this kind of samples. Hence, the proposed method is sensitive to perspectively distorted text, which is also a near future work in our research. Summarily, the same as MSRA-TD500, the experiments on USTB-SV1k verify that the fusion strategy contributes to improve recall and the CNN filter improves the precision effectively.

Fig. 9. Text tracking and detection results for our approach on sample videos from [13], where the blue numbers are the IDs of the tracked text regions.

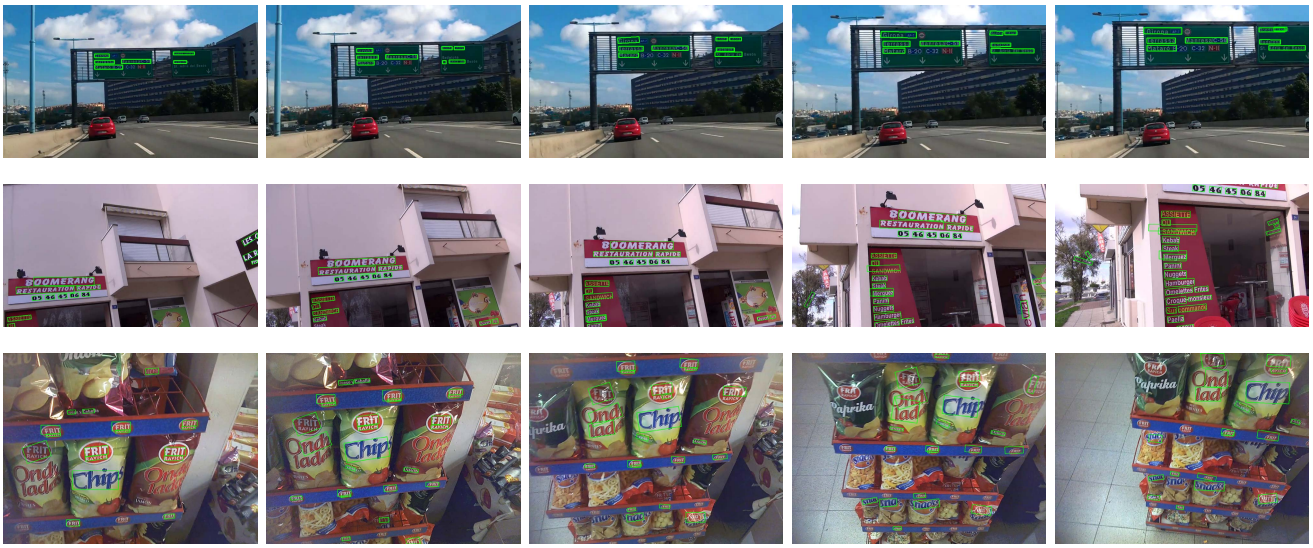


Fig. 10. Text tracking and detection results for our approach on sample videos (three groups of consecutive frames) from ICDAR 2015 Robust Reading Competition dataset [21].

Here, we also present the training procedure and some empirical results for the CNN filter. In the training process, first, a majority of the positive samples (90%) are generated by embedding characters to blank images (without text) automatically, and the rest of positive samples (10%) are extracted from the training set (ground truth, in gray scale) of ICDAR2013 Robust Reading Competition Challenge 2 Task 2. Next, the negative samples are non-text MSER components which have similar features to the positive samples. Each sample extends the bounding box to square according to the longer edge and adds a margin with 0.1 times of the edge. Summarily, the training set with $30,845$ positive and $31,408$ negative samples are finally constructed. The empirical performance of this model is shown in Table III, where the testing set of ICDAR2013 Robust Reading Competition Challenge 2 Task 2 is used.

B. Experiments With Scene Text Detection in Video

To evaluate our tracking based detection approach, a public video dataset with a variety of scene videos is used in our experiments.[7] In these scene videos, some text regions have affected by nature noise, distortions, blurring, and (sometimes) occlusions. We compare our approach (with different settings) with the methods in [13] and [49]. Here, we use the well-known metrics the precision, recall and f-score defined in [57].

Table IV presents the tracking based scene text detection performance of Minetto et al.'s method [13], Strategy II-1, the proposed method without text tracking (using individual frames), Strategy II-2, video text detection using multiple frames by rule-based techniques (the method in [49]), and Strategy II-3, the whole proposed method respectively. As seen from the table, compared with the performance of text detection in [13], the average performance of text detection in [49] has an increase of f-score by 12%. What's more, the performance of tracking based text detection method proposed in this paper is 6% higher than the method in [49]. In other words, the proposed multi-strategy tracking method is effective and robust to reduce false alarms and improve the accuracy of text detection. By the way, compared with the method without text tracking, our proposed tracking based text detection method improves the performance largely (averagely about 15% with f-score), which also verifies that tracking is a very important step in video text detection. Fig. 9 shows some scene text tracking results by our method on this public dataset.

Moreover, we also perform the experiments to evaluate our method on the recent challenging dataset of ICDAR 2015 Robust Reading Competition Challenge 3 (Text Detection and Recognition in Scene Videos). [8] This dataset includes a training set of 25 videos (13450 frames in total) and a test set of 24 videos (14374 frames in total), which are collected by the organizers in different countries, including the text in different languages. The video sequences correspond to 7 high level tasks in indoors and outdoors scenarios. Moreover, 4 different cameras are used for capturing different sequences. The text is always with low contrast and arbitrary motion.

[7] http://www.liv.ic.unicamp.br/ minetto/datasets/text/VIDEOS/

[8] http://rrc.cvc.uab.es/?ch=3&com=introduction

We use the ICDAR'15 Robust Reading Competition Platform to evaluate our proposed approach, and comparative results are shown in Table V, where "ATA" is the official metric of the ICDAR 2015 Robust Reading Competition Challenge [21]. Our proposed approach achieves the best performance for video text detection on this competition dataset, by improving about 4% with ATA compared with the second place.

ATA (Average Tracking Accuracy) provides an object tracking measure that penalizes fragments both in temporal and spatial dimensions. MOTA (Multiple Object Tracking Accuracy) is another metric to evaluate the performance of object tracking, which penalizes more on false positives and mismatches. MOTP (Multiple Object Tracking Precision) is used to evaluate the detection performance without the explicit penalization on the temporal errors. The result that our method achieves the highest score of ATA means that our method retrieves more tracks than others. The lower scores of MOTA show that because of severely perspective and aligned distortions, the number of false positives of our method is relatively large. Similar to the discussions in the experiments with the USTB-SV1K dataset, it is a near future topic for improving the robustness of text region candidates filtering when dealing with severely perspectively distorted text.

Fig. 10 shows some empirical samples and results on the dataset of ICDAR 2015 Robust Reading Competition Challenge 3 (Text Detection in Scene Videos). Our tracking based scene text detection method can deal with the arbitrary motion challenge in video. For example, in Fig. 10, though the scale changes a lot in the first row and the view varies very much in the second row, our proposed method has an impressive performance for text detection and tracking. As described above, in the third row in Fig. 10, there are severely perspective and skew distortions, and some text regions are missed by our approach.

VII. Conclusion

In this paper, we construct a tracking based text detection system in scene videos by locating character candidates extensively and searching text regions globally. In our approach, the multi-information fusion strategy can precisely detect multi-orientation text regions with fair recall by extensively extracting character candidates from multiple channels and scales and powerfully filtering non-text region candidates. The tracking based text detection technique can robustly detect video text with high average tracking accuracy by tracking text with multiple tracking strategies and integrating detection results with dynamic programming. Moreover, the experiments on a variety of public datasets verify the effectiveness of our proposed method. Impressively, our proposed approach won the ICDAR 2015 Robust Reading Competition (video text detection).

As shown in the experiments of our system, the degradation of text tracking always occurs with severe motion blur and perspective distortions. As one future work, with investigations of recent multi-target tracking methods [58] and multi-graph matching methods [59], [60], the tracking technique in our system will be extended to improve robustness for some challenges such as severe motion blur and occlusions. Another issue is about the computational cost of multi-orientation text detection with multi-information fusion. Basically speaking, the computational cost of the proposed method is proportional to the number of the channels and scales used. The average speed of our previous method with one channel and one scale (original size) [25] is 1.4s per image on MSRA-TD500 database. So, if several (e.g., 3) channels and (e.g., 2) scales are used in our system, the overall computational performance will be comparative to other state-of-the-art text detection approaches in images. Basically speaking, the computational cost of the whole tracking based scene text detection system is accumulation of the time cost of text detection and text recognition in individual frames, and text tracking. In general, scene text detection in video (across a lot of frames) is rather time consuming. [9] which is also an future topic.

Acknowledgments

The authors are grateful to Li-Yu Meng for helpful efforts in the experiments. The authors are also grateful to the associate editor Prof. Peter Tay and the anonymous reviewers for their constructive comments. The research is partly supported by National Natural Science Foundation of China (61473036).

References

[1] X.-C. Yin, Z.-Y. Zuo, S. Tian, and C.-L. Liu, "Text detection, tracking and recognition in video: A comprehensive survey," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2752–2773, Jun. 2016.

[2] 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.

[3] K. Iwatsuka, K. Yamamoto, and K. Kato, "Development of a guide dog system for the blind with character recognition ability," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 1. 2004, pp. 453–456.

[4] N. Ezaki, M. Bulacu, and L. Schomaker, "Text detection from natural scene images: Towards a system for visually impaired persons," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 2. 2004, pp. 683–686.

[5] H. Shiratori, H. Goto, and H. Kobayashi, "An efficient text capture method for moving robots using DCT feature and text tracking," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 2. 2006, pp. 1050–1053.

[6] M. Tanaka and H. Goto, "Text-tracking wearable camera system for visually-impaired people," in *Proc. 19th Int. Conf. Pattern Recognit. (ICPR)*, 2008, pp. 1–4.

[7] H. Goto and M. Tanaka, "Text-tracking wearable camera system for the blind," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, 2009, pp. 141–145.

[8] P. Sanketi, H. Shen, and J. M. Coughlan, "Localizing blurry and low-resolution text in natural images," in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Jan. 2011, pp. 503–510.

[9] I. Haritaoglu, "Scene text extraction and translation for handheld devices," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2. Dec. 2001, p. II-408.

[10] X. Shi and Y. Xu, "A wearable translation robot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Apr. 2005, pp. 4400–4405.

[11] M. Petter, V. Fragoso, M. Turk, and C. Baur, "Automatic text detection for mobile augmented reality translation," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV)*, Nov. 2011, pp. 48–55.

[12] V. Fragoso, S. Gauglitz, S. Zamora, J. Kleban, and M. Turk, "TranslatAR: A mobile augmented reality translator," in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Jan. 2011, pp. 497–502.

[9] We also conducted some initial experiments with time cost on the video dataset of ICDAR 2015 Challenge 3 (on a Linux laptop with a 2.20 GHz processor). The time cost of text detection and recognition in each frame is about 1 second, and the total time of video text extraction (including detection, recognition and tracking) is about 30 frames per minute.

[13] R. Minetto, N. Thome, M. Cord, N. J. Leite, and J. Stolfi, "Snoopertrack: Text detection and tracking for outdoor videos," in *Proc. 18th IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2011, pp. 505–508.

[14] Y.-T. Cui and Q. Huang, "Character extraction of license plates from video," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1997, pp. 502–507.

[15] S. H. Park, K. I. Kim, K. Jung, and H. J. Kim, "Locating car license plates using neural networks," *Electron. Lett.*, vol. 35, no. 17, pp. 1475–1477, Aug. 1999.

[16] W. Wu, X. Chen, and J. Yang, "Incremental detection of text on road signs from video with application to a driving assistant system," in *Proc. 12th Annu. ACM Int. Conf. Multimedia (ACM MM)*, 2004, pp. 852–859.

[17] W. Wu, X. Chen, and J. Yang, "Detection of text on road signs from video," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 4, pp. 378–390, Dec. 2005.

[18] D. Létourneau, F. Michaud, J.-M. Valin, and C. Proulx, "Textual message read by a mobile robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 3. Oct. 2003, pp. 2724–2729.

[19] D. Létourneau, F. Michaud, and J.-M. Valin, "Autonomous mobile robot that can read," *EURASIP J. Appl. Signal Process.*, vol. 17, pp. 2650–2662, Dec. 2004.

[20] X.-C. Yin, H.-W. Hao, J. Sun, and S. Naoi, "Robust vanishing point detection for MobileCam-based documents," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, 2011, pp. 136–140.

[21] D. Karatzas *et al.*, "ICDAR 2015 competition on robust reading," in *Proc. 13th Int. Conf. Document Anal. Recognit. (ICDAR)*, 2015, pp. 1156–1160.

[22] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2012, pp. 1083–1090.

[23] 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.

[24] L. Kang, Y. Li, and D. Doermann, "Orientation robust text line detection in natural images," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 4034–4041.

[25] 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.

[26] P. Shivakumara, T. Q. Phan, S. Lu, and C. L. Tan, "Gradient vector flow and grouping-based method for arbitrarily oriented scene text detection in video images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 10, pp. 1729–1739, Oct. 2013.

[27] A. Risnumawan, P. Shivakumara, C. S. Chan, and C. L. Tan, "A robust arbitrary text detection system for natural scene images," *Expert Syst. Appl.*, vol. 41, no. 18, pp. 8027–8048, 2014.

[28] G. Liang, P. Shivakumara, T. Lu, and C. L. Tan, "Multi-spectral fusion based approach for arbitrarily oriented scene text detection in video images," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4488–4501, Nov. 2015.

[29] V. Khare, P. Shivakumara, and P. Raveendran, "A new histogram oriented moments descriptor for multi-oriented moving text detection in video," *Expert Syst. Appl.*, vol. 42, no. 21, pp. 7627–7640, 2015.

[30] V. Khare, P. Shivakumara, P. Raveendran, and M. Blumenstein, "A blind deconvolution model for scene text detection and recognition in video," *Pattern Recognit.*, vol. 54, pp. 128–148, Jun. 2016.

[31] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, 2012, pp. 3304–3308.

[32] Z. Zhang, W. Shen, C. Yao, and X. Bai, "Symmetry-based text line detection in natural scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2558–2567.

[33] L. Gómez and D. Karatzas, "MSER-based real-time text detection and tracking," in *Proc. 22nd Int. Conf. Pattern Recognit. (ICPR)*, 2014, pp. 3110–3115.

[34] 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.

[35] S. Tian, W.-Y. Pei, Z.-Y. Zuo, and X.-C. Yin, "Scene text detection in video by learning locally and globally," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2016, pp. 2647–2653.

[36] 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.

[37] C. Yi and Y. Tian, "Text extraction from scene images by character appearance and structure modeling," *Comput. Vis. Image Understand.*, vol. 117, no. 2, pp. 182–194, Feb. 2013.

[38] L. Neumann and J. Matas, "Scene text localization and recognition with oriented stroke detection," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2013, pp. 97–104.

[39] 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.

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

[41] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proc. Brit. Mach. Vis. Conf.*, vol. 1. 2002, pp. 384–393.

[42] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Accurate and robust text detection: A step-in for text retrieval in natural scene images," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2013, pp. 1091–1092.

[43] W. Huang, Z. Lin, J. Yang, and J. Wang, "Text localization in natural images using stroke feature transform and text covariance descriptors," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp.1241–1248.

[44] Y. Li, W. Jia, C. Shen, and A. van den Hengel, "Characterness: An indicator of text in the wild," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1666–1677, Apr. 2014.

[45] 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.

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

[47] X. Wang, Y. Song, and Y. Zhang, "Natural scene text detection with multi-channel connected component segmentation," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, 2013, pp. 1375–1379.

[48] S. Tian, X.-C. Yin, Y. Su, and H.-W. Hao, "A unified framework for tracking based text detection and recognition from Web videos," *IEEE Trans. Pattern Anal. Mach. Intell.* [Online]. Available: http://ieeexplore.ieee.org/document/7895141/

[49] Z.-Y. Zuo, S. Tian, W.-Y. Pei, and X.-C. Yin, "Multi-strategy tracking based text detection in scene videos," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, 2015, pp. 66–70.

[50] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *Proc. 13th Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 127–141.

[51] X. Yin, X.-C. Yin, H.-W. Hao, and K. Iqbal, "Effective text localization in natural scene images with MSER, geometry-based grouping and AdaBoost," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, 2012, pp. 725–728.

[52] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 26th Annu. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.

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

[54] 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.

[55] K. Zhang, L. Zhang, M.-H. Yang, and D. Zhang. (2013). "Fast tracking via spatio-temporal context learning." [Online]. Available: https://arxiv.org/abs/1311.1939

[56] C. Zauner, M. Steinebach, and E. Hermann, "Rihamark: Perceptual image hash benchmarking," *Proc. SPIE*, vol. 7880, p. 78800X, Feb. 2011.

[57] S. M. Lucas, "ICDAR 2005 text locating competition results," in *Proc. ICDAR*, 2005, pp. 80–84.

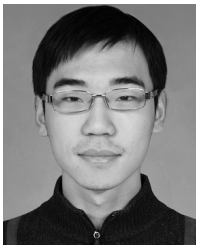
[58] A. Milan, K. Schindler, and S. Roth, "Multi-target tracking by discrete-continuous energy minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2054–2068, Oct. 2016.

[59] J. Yan, J. Wang, H. Zha, X. Yang, and S. Chu, "Consistency-driven alternating optimization for multigraph matching: A unified approach," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 994–1009, Mar. 2015.

[60] J. Yan, M. Cho, H. Zha, X. Yang, and S. M. Chu, "Multi-graph matching via affinity optimization with graduated consistency regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 6, pp. 1228–1242, Jun. 2016.

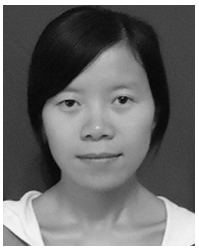
Chun Yang received the B.Sc. degree in computer science from the University of Science and Technology Beijing, China, in 2011, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology. His current research interests include pattern recognition, classifier ensemble, and document analysis and recognition.

Xu-Cheng Yin (M'10–SM'16) received the B.Sc. and M.Sc. degrees in computer science from the University of Science and Technology Beijing, China, in 1999 and 2002, respectively, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, in 2006. From 2006 to 2008, he was a Researcher with the Information Technology Laboratory, Fujitsu Research and Development Center. He was a Visiting Researcher with the School of Computer Science, University of Massachusetts, Amherst, USA, from 2013 to 2014. He was a Visiting Professor with the Department of Quantitative Health Sciences, University of Massachusetts Medical School, USA, in 2016. He is currently a Professor with the Department of Computer Science and Technology, University of Science and Technology Beijing, China. He has authored over 50 research papers (IEEE TPAMI, IEEE TIP, PLoS ONE, Information Fusion, Information Sciences, IJCAI, SIGIR, CIKM, ICMR, ICDAR, and ICPR). His current research interests include pattern recognition, computer vision, image processing, information retrieval, and document analysis and recognition. His team received the first place of Text Localization in Real Scenes and Text Localization in Born-Digital Images in the ICDAR 2013 Robust Reading Competition, the first place of End-To-End Text Recognition in Real Scenes (Generic) and End-To-End Text Recognition in Born-Digital Images (Generic) in the ICDAR 2015 Robust Reading Competition, and the first place of Video Text Detection in the ICDAR 2015 Robust Reading Competition.



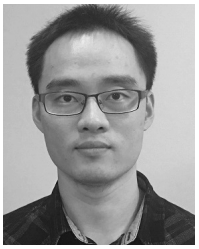
Wei-Yi Pei received the B.Sc. degree in computer science from the University of Science and Technology Beijing, China, in 2010, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology. His current research interests include pattern recognition, computer vision, and scene text detection and recognition.

Shu Tian received the B.Sc. and Ph.D. degrees in computer science from the University of Science and Technology Beijing, China, in 2010 and 2016, respectively. He is currently a Faculty Member with the School of Computer and Communication Engineering, University of Science and Technology Beijing. He has authored over ten research papers (IEEE TPAMI, IEEE TIP, IJCAI, and ICDAR). His current research interests include object tracking, pattern recognition, and multimedia understanding.



Ze-Yu Zuo received the B.Sc. degree in computer science from the University of Science and Technology Wuhan, Hubei, China, in 2013, and the M.Sc. degree in computer science from the University of Science and Technology Beijing, China, in 2016. She is currently an Engineer with Sina Weibo, China. Her current research interests include video text tracking, multimedia understanding, and retrieval.

Chao Zhu received the bachelor's degree in automation from Xidian University, Xi'an, China, in 2005, the master's degree in system engineering from Xi'an Jiaotong University, Xi'an, in 2008, and the Ph.D. degree in computer science from the Ecole centrale de Lyon, Lyon, France, in 2012. He was a Post-Doctoral Fellow with the Multimedia Information Processing Laboratory, Institute of Computer Science and Technology, Peking University, Beijing, China, from 2013 to 2015. He is currently a Faculty Member with the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His current research interests include object detection and recognition, feature extraction, and image/video classification.



Junchi Yan received the M.S. and Ph.D. degrees from the Department of Electronic Engineering, Shanghai Jiao Tong University, China, in 2011 and 2015, respectively. He is currently with the Department of Computer Science and Technology, East China Normal University, China, and also a Research Staff Member and a Master Inventor with IBM China Research Center. His current research interests include computer vision, machine learning, and their applications. He received the IBM Research Accomplishment and the Outstanding Accomplishment Award in 2013 and 2014, respectively.