Multimed Tools Appl
DOI 10.1007/s11042-016-3941-x

Arbitrarily-oriented multi-lingual text detection in video

Vijeta Khare[1] · **Palaiahnakote Shivakumara**[2,3] · **Raveendran Paramesran**[1] · **Michael Blumenstein**[4]

Received: 7 February 2016 / Revised: 11 July 2016 / Accepted: 5 September 2016

Abstract Text detection in arbitrarily-oriented multi-lingual video is an emerging area of research because it plays a vital role for developing real-time indexing and retrieval systems. In this paper, we propose to explore moments for identifying text candidates. We introduce a novel idea for determining automatic windows to extract moments for tackling multi-font and multi-sized text in video based on stroke width information. The temporal information is explored to find deviations between moving and non-moving pixels in successive frames iteratively, which results in static clusters containing caption text and dynamic clusters containing scene text, as well as background pixels. The gradient directions of pixels in static and dynamic clusters are analyzed to identify the potential text candidates. Furthermore, boundary growing is proposed that expands the boundary of potential text candidates until it finds neighbor components based on the nearest neighbor criterion. This process outputs text lines appearing in the video. Experimental results on standard video data, namely, ICDAR 2013, ICDAR 2015, YVT videos and on our own English and Multi-lingual videos demonstrate that the proposed method outperforms the state-of-the-art methods.

✉ Palaiahnakote Shivakumara
hudempsk@yahoo.com; shiva@um.edu.my

Vijeta Khare
kharevijeta@gmail.com

Raveendran Paramesran
ravee58@gmail.com

Michael Blumenstein
Michael.Blumenstein@uts.edu.au

[1] Faculty of Engineering, University of Malaya, Kuala Lumpur, Malaysia

[2] Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia

[3] Computer Systems and Information Technology, University of Malaya, BS-18, Annex Building, 50603 Malaysia, Malaysia

[4] School of Software, University of Technology Sydney, Sydney, Australia

Published online: 20 September 2016

Keywords Higher order moments · Stroke width distance, dynamic window · Caption text · Region growing · Arbitrarily-oriented text detection · Multi-lingual text detection

1 Introduction

Text detection and recognition in video containing multi-scripts has drawn the attention of many researchers because, unlike text component structures in English text, multi-scripts have great variations in structure, shape and style. It is evident from the work proposed in [14, 30] that text extraction and recognition from multi-oriented and multi-lingual environments is difficult and requires great attention. Furthermore, in general, video poses different challenges, such as low resolution, complex backgrounds, font variations, and font size variations. Besides, the arbitrary orientation of the multi-script text lines makes the problem more challenging. Thus, it is considered as a challenging problem, as stated in [8]. Note: in the case of horizontal text lines, characters and words have the same orientation as horizontal text line directions, while non-horizontal text lines, characters and words have the same orientation as the direction of non-horizontal text lines. On the other hand, in arbitrarily-oriented text lines, every character in the text line has different orientations and is not the same as text line direction [26]. It is also true that the text detection of multi-scripts of any orientation in video is useful for developing real-time applications such as surveillance and video summarization etc. [19, 22, 33]. Some application examples include: developing an aid for the blind, assisting tourists requiring recognition and translation of a particular source script to a target script for reaching their destination [26]. In general, video contains two types of text: (1) graphics/caption/artificial/superimposed text, which is edited text and hence has good contrast, clarity, and visibility; (2) scene text, which is naturally embedded text on a background. Since scene text is part of an image, the characteristics, such as font, font size, contrast, orientation, and background variations etc. are unpredictable compared to graphics text. Therefore, the presence of both types of text in video makes the problem more complex [11, 23, 35].

We can see many methods [2, 28, 36] for text detection in natural scene images that are captured by high resolution cameras, where images are mostly of high contrast along with complex backgrounds. Since the images have high contrast text, the methods explore characteristics of the character shapes. For instance, Risnumawan et al. [20] proposed a method for solving arbitrary orientation of text in natural scene images by employing ellipse growing. It is noted from their work that the method is sensitive to low contrast, illumination effects and multi-script text lines. Furthermore, none of these methods have been developed with the specific objective of text detection in multi-script natural scene images.

The methods for video text detection can be classified as connected component-based, texture-based and edge or gradient-based methods [35]. Since connected component-based methods require the shape of the character, these methods may not produce a good accuracy for low resolution texts with complex backgrounds, as in the case of document analysis-based methods. For example, Yin et al. [36] proposed a method for text detection in natural scene images based on Maximally Stable Extremal Regions (MSERs). For frames with low contrast text and multi-script text lines, the MSER-based methods may not preserve the character shapes sufficiently due to large variations in structure at the character level. To overcome the problems of connected component-based methods, texture feature-based methods are proposed. However, these methods are computationally expensive and the performance relies on classifier training and the number of samples. For instance, Liu et al. [13] proposed a method

for text detection in video using a set of texture features and k-means clustering. Shivakumara et al. [23] proposed a method which combines a Fourier transform and color spaces for text detection in video. However, the primary focus of these methods is on horizontal text detection in video but not arbitrarily-oriented text in video. Recently, Liang et al. [11] proposed a method based on multi-spectral fusion for arbitrarily-oriented scene text detection in video images. The performance of the method degrades for multi-script images. In addition, the method does not utilize the temporal information for text detection, though it is available in the video.

To further achieve better efficiency and accuracy for text detection in video, edge and gradient information-based methods have been developed. These methods work well with fewer computations but are sensitive to backgrounds and hence the methods produce more false positives. For example, Shivakumara et al. [24] proposed a method based on the Laplacian in the frequency domain for video text detection, which uses gradient information to obtain text components. Shivakumara et al. [25] proposed multi-oriented video scene text detection in video using a Bayesian classifier and boundary growing. Similarly, the GVF (gradient vector flow) and the grouping-based method are proposed by [26] for arbitrarily-oriented text detection in video. The main aim of this method is to explore dense gradient vector flow at corner points of character edge components. This method considers extracted individual frames as input for text detection. As a result, the method does not utilize the temporal information. This is because the scope of the method is to work for both video frames and natural scene images. This limitation leads to inconsistent results for different data and applications. Besides, the methods are not tested on multi-script videos.

In light of the above discussion, it is noted that most of the methods reviewed focus on text detection in video containing English texts. Despite the fact that the above methods address the issue of arbitrary orientation, they do not utilize the available temporal information. Therefore, there is scope for arbitrarily-oriented multi-lingual text detection in video by exploring temporal information.

2 Related work

In the previous section, we discussed text detection in natural scene and video images where the methods consider individual frames of video as input. Since the proposed method uses temporal information for text detection in video, in this section we review papers on the methods that use temporal information for text detection.

Li et al. [9] proposed a method for video text tracking based on wavelet and moment features. This method takes advantage of wavelet decomposition, spatial information provided by the moments and with a neural network classifier for identifying text candidates. Huang et al. [4] proposed a method for scrolling text detection in video using temporal frames. This method uses motion vector estimation for detecting text. However, this method is limited to only scrolling text but not arbitrarily-oriented texts. Zhou et al. [38] exploited edge information and geometrical constraints to form a coarse-to-fine methodology to define text regions. However, the method is unable to deal with dynamic text objects. Mi et al. [15] proposed a text extraction approach based on multiple frames. The edge features are explored with a similarity measure for identifying text candidates. Wang et al. [31] used a spatio-temporal wavelet transform to extract text objects in video documents. In the edge detection stage, a three-dimensional wavelet transform with one scaling function and seven wavelet functions is applied on a sequence of video frames.

Huang [3] detected video scene text based on the video's temporal redundancy. Video scene texts in consecutive frames have arbitrary motion due to camera or object movement. Therefore, this method performs the motion detection in 30 consecutive frames to synthesize a motion image. Zhao et al. [37] proposed an approach for text detection using corners in video. This method proposes to use dense corners for identifying text candidates. Finally, optical flow has been used for moving text detection. Liu et al. [12] proposed a method for video caption text detection using stroke-like edges and spatio-temporal information. The color histogram is used for segmenting text information. Li et al. [10] proposed a method for video text detection using multiple frame integration. This method uses edge information to extract text candidates. Morphological operations and heuristic rules are proposed to extract final text information from video.

Recently, Khare et al. [8] proposed method for multi-oriented text detection in video, which uses temporal information. The main objective of the method is to introduce a new descriptor called the Histogram Oriented Moments Descriptor (HOM), which works as Histogram Oriented Gradients (HOG) for text detection in video but does not exploit the temporal coherency for text candidate detection. This is because the method uses optical flow for the text candidates to find moving text with the help of temporal information, but not text candidate detection. In other words, the method considers a frame as an individual image to find text candidates and then it uses temporal information for tracking moving text candidates in temporal frames. Therefore, the performance of the method degrades for static foregrounds and dynamic backgrounds because moving text candidates may overlap with the moving background.

Similarly, Wu et al. [32] proposed a new technique for multi-oriented scene text line detection and tracking in video. The method explores gradient directional symmetry and spatial proximity between the text components for identifying text candidates. The text lines are tracked by matching sub-graphs of text components. The ability of the method is not tested on arbitrarily-oriented text and multi-script text lines in video. It is observed from the above literature review that most of the methods focus either on horizontal or non-horizontal text and very few on arbitrarily-oriented text detection in video. Hence, the detection of multi-script text lines in video is not sufficiently addressed in the literature. However, Liu et al. [14] proposed a method for multi-lingual text extraction from scene images using a Gaussian mixture model and learning. This method uses binarization for detecting text lines through connected component analysis. This idea works for high contrast images but not low contrast images such as video frames. Huang et al. [5] also makes an attempt to solve text detection multi-scripts by exploring temporal information based on a motion perception field in consecutive frames. The methods are limited to a few languages, such as Chinese, English and Japanese. Therefore, we can conclude that the following issues require urgent attention by researchers: (1) A method that works without language or direction restrictions; (2) Determining the exact number of temporal frames for performing operations when the method uses successive frames in video and (3) Rather than using fixed windows for performing operations over an image or video, determining automatic window sizes according to font and text size in video. These issues motivated us to propose a new method to find the solution to the above-mentioned challenges.

Inspired by the work presented in [27] for multi-oriented text detection in video using the combination of wavelet and moments, where it is shown that moments help in detecting text candidates for text successfully, we propose to explore moments in a new way without wavelet support for classifying static and dynamic text clusters using temporal frames. It is noted that in [34], stroke width distance is almost the same for characters in the video, hence we utilize the same stroke width distance for determining window size automatically to make the method

invariant to different fonts, font size etc., while the methods [4, 8, 9] use a fixed window size. Most of the existing methods, which use temporal frames, utilize a fixed number based on pre-defined experiments. Hence, the intention of the methods is to use temporal frames for enhancing low contrast text information. In contrast to the above methods, the one proposed here explores moments for estimating deviations between temporal frames to automatically select temporal frames for arbitrarily-oriented-multi-script text detection in video. In addition, the proposed method introduces an iterative procedure for determining the number of temporal frames to be used out of 25–30 frames per second, unlike existing methods, which use a fixed number of frames for text detection or an individual frame [8, 26]. The same iterative procedure also helps in classifying caption and scene text. Motivated by the boundary-growing approach proposed in [27] for multi-oriented text detection in video, we propose boundary-growing without angle projection for arbitrarily-oriented text extraction in video. The main advantage of the proposed method is that it works for static text with static and moving backgrounds, as well as moving text with static and moving backgrounds. Besides, the proposed method is script independent and invariant to rotation.

3 Proposed method

As discussed in the previous section, moments assist in extracting vital clues from text pixels, such as regular spacing between text pixels and uniform color of the text pixels [27], we explore the moments to estimate deviation between temporal frames to separate caption (pixels which stay at the same location) from scene text (pixels which have small movements along the background) and background pixels. To obtain such clues for classification, we need to define a window for estimating moments. Since video contains multi-font and multi-size text, fixing a particular window does not yield good accuracy results. Therefore, we propose a new idea for determining windows automatically based on stroke width information from Sobel and Canny edge images from the input video frame. The proposed method introduces an iterative procedure for estimating deviation using moments of pixels in consecutive frames with the help of k-means clustering, which results in static and dynamic text clusters containing caption text and scene text, respectively. The output of clustering refers to text candidates of captions and scene texts in respective clusters. Due to low resolution and complex backgrounds, the clustering may misclassify non-text pixels as text pixels. Therefore, gradient direction of text candidates is analyzed to identify the potential text candidates. Furthermore, boundary growing is proposed for each potential text candidate to extract full text lines of any direction in video. The flow of the proposed method can be seen in Fig. 1.

3.1 Automatic window size detection

It is known that usually, window size is determined based on the performance of the method on sample data by varying different sizes. This generally results in the study of optimal stroke thickness of text components in images for text detection/recognition. As a result, this procedure may hamper the performance of the method when the input data contains multi-font and multi-text sizes as in video. Therefore, in this work, we propose a new idea for finding windows automatically based on stroke width and opposite direction pairs. Motivated by the work in [18] for text detection in natural scene images where it is shown that the pattern of text in the Sobel and Canny edge images of the input image shares the same symmetry property to

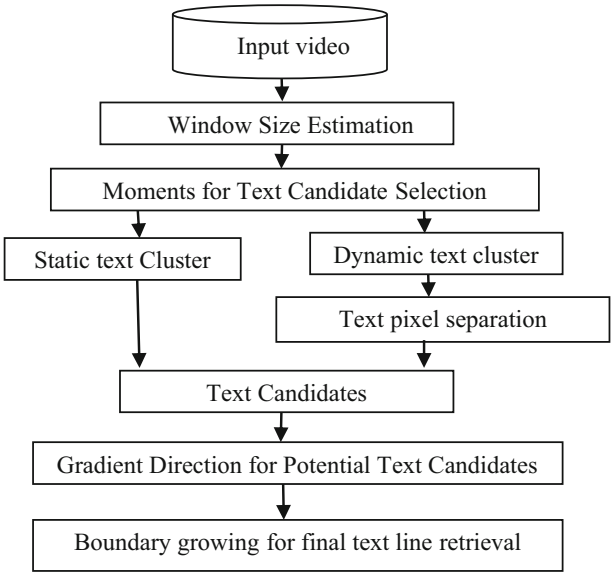


Fig. 1 Overall framework of the proposed method

identify text candidates, we propose to find common pixels in Sobel and Canny edge images of the input image, which satisfy stroke width distance and opposite direction pairs to identify the common pixels.

The proposed method finds the stroke width as stated in [2]. Let p be a pixel at the edge of the image, whereby the proposed method moves along the gradient direction of the pixel p until it reaches the opposite edge pixels, say q. The distance between p and q is considered as the Stroke Width (SW) distance. In the same way, the proposed method estimates Gradient Vector Flow (GVF) to check opposite direction arrows of pixel pairs given by SW. Gradient Vector Flow (GVF) is an improved version of gradient direction which overcomes the drawbacks of the conventional gradient direction approach. For example, the gradient does not give high responses at corners, low contrast and flat regions. The GVF provides a good response for such situations because GVF is a non-irrational external force field that points towards the boundaries when in their proximity and varies smoothly over homogenous image regions all the way to the image borders. The GVF field can be defined as the vectors $g(x, y) = (u(x, y), v(x, y))$for image f that minimizes the energy functional [26]:

$$E = \iint \mu\left(u_x^2 + y_y^2 + v_x^2 + v_y^2\right) + |\nabla f|^2 |g - \nabla f|^2 \mathrm{dxdy} \tag{1}$$

where $\mu$is a regularization parameter balancing the weight of the first and second term.

The proposed method finds common pairs of pixels in the Sobel and Canny images of the input image that satisfies the SW distance as well as the opposite direction pair, as defined in equation (2).

$$\begin{aligned} &\forall P(i,j) : CP_{(i,j)} = 1 \\ &\text{if}\left(SW_{C(i,j)} == SW_{S(i,j)}\right)\&\&\left(GVF_{(i,j)} == -GVF_{(i,j+n)}\right) \end{aligned} \tag{2}$$

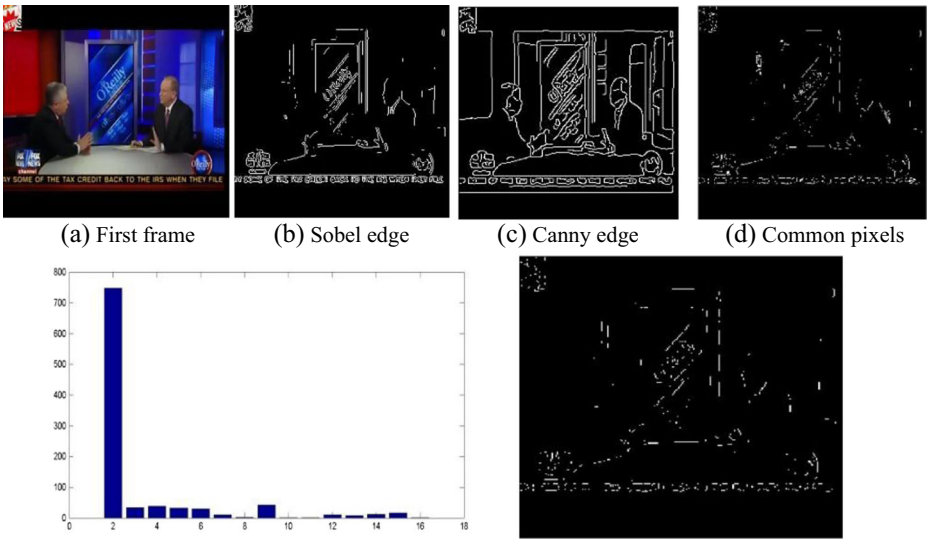
Where for all pixels (i,j) of image I, a common pixel image CP is generated if the stroke width for that pixel in a Canny image SW_Cmatches with the stroke width in the Sobel image

$SW_S$of the input. As well as the opposite direction pair, it will also be present in the opposite pair combination, stating for GVF at location (i,j), there will exist an opposite (−GVF) at location (i,$j + n$). Here n denotes pixels that represent the stroke width.

It is noted from [2] that the SW of the characters in a text line have almost uniform widths. This observation leads us to define the dimension of the window size, n automatically for the input image. To determine n automatically, on the common pixels, we perform a histogram operation for stroke width distance vs. frequencies to choose the distance which gives the highest peak, and the corresponding stroke width distance is considered as the window size for moments estimation, which results in the candidate text pixel image. It is illustrated in Fig. 2 where (a) is the input image, (b) is the Sobel edge image, (c) is the Canny edge image, (d) is the common pixels which satisfy the stroke width distance and opposite direction pair in the Sobel and Canny edge image and (e) is the histogram which we plot for a stroke width distance greater than or equal to two to choose the highest peak. Note: Pixels that have a stroke width distance less than two do not contribute to representing the text. It is noted from Fig. 2(e) that the stroke width (SW) distance of 2 gives the highest peak and hence it is considered as the window dimension for a given input image. The pixels which contribute to the highest peak are called candidate text pixels as shown in Fig. 2(f) where one can see that a few non-text pixels are removed compared to the results in Fig. 2(d), and text pixels are retained. For the image shown in Fig. 2(a), 2 × 2 dimensions are considered as the actual window dimension. In this way, automatic window size detection helps in retaining strokes that represent text in the video frame without losing significant information. However, Fig. 2(f) shows that this step misclassifies non-text pixels as candidate text pixels due to the complex background and low resolution of the video. In addition, it can be seen in Fig. 2(f) that few text pixels are also lost compared to Fig. 2(d). This loss does not have much of an effect on the text detection in this work because the restoration steps which will be presented in subsequent sections, restore the missing text pixels from the edge image while extracting text lines in the video frame. The main advantage of this step is that it gives candidate text pixels regardless of the caption and scene text type, as shown in Fig. 2(f) where we can see candidate text pixels for the caption (bottom line in Fig. 2(a)) and scene text ("Oreilly" in the middle of the image in Fig. 2(a)). Hence, this step solves two problems: window size detection and candidate text pixel identification by reducing non-text pixels. The candidate text pixels are input for identifying text candidates which will be presented in the next Section.

3.2 Static and dynamic text cluster classification

It is observed from Fig. 2(f) that the step presented in Section 3.1 misclassifies non-text pixels as candidate text pixels due to the complexity of the problem. In order to classify text pixels accurately, we propose to explore temporal information for classifying text and non-text pixels in this Section. As mentioned in [5], pixels that represent caption text stay at the same location for a few frames while pixels that represent scene text (background) have little movement; we use the same basis for classifying caption and scene text pixels. Since caption text pixels do not have movements and scene and background pixels have movements, the deviation between caption text pixels is lower than the scene and background pixels. To extract such observations, we propose moment estimation for the defined window over the first frame and its successive frames. For the deviation computed between first and second frames, we employ k-means clustering with k = 2 to classify caption pixels into one cluster (static) and background scene pixels into another cluster (dynamic). The cluster which gives the lowest mean is considered as a static cluster



(a) First frame (b) Sobel edge (c) Canny edge (d) Common pixels

(e) Histogram for SW >=2: SW vs frequencies (f) Candidate pixels of highest peak (SW = 2)

Fig. 2 Automatic window size detection using stroke width (SW) distance and opposite direction pair

(caption text) and the other is considered as a dynamic cluster (scene text and background pixels), which we call the first iteration result. In the same way, the process continues for first and third temporal frames by considering a static cluster as the input. In other words, the deviation is computed between the result of the first iteration and the third frame. Again, the k-means clustering with k = 2 is deployed to obtain static and dynamic clusters. As the iterations increase, the pixels which represent non-text in a static cluster are classified into a dynamic cluster. When all the pixels which represent non-text in a static cluster are classified into dynamic clusters after a certain number of iterations, the dynamic cluster contains nothing. This is the converging criterion for terminating the iterative process. As a result, the static cluster gives caption text and the dynamic cluster gives scene text along with non-text pixels. The same converging criterion helps in deciding the number of temporal frames unlike existing methods [15, 21, 23, 24,] that assume or have fixed the number of temporal frames to be used for text detection. The advantage of this step is that it solves two issues: It separates caption text from scene text and background pixels, at the same time it helps in deciding the number of temporal frames.

For each video sequence, $f, f+1, f+2 \ldots f+n$ as shown in Fig. 3(a), the proposed method employs an iterative procedure to separate caption text pixels from scene and background pixels. Here n denotes 30 frames per second. For the first iteration, the method considers the first two consecutive frames, say f and $f+1$ as shown in Fig. 3(b). Each frame is divided into equal-sized blocks of window size defined by the step presented in Section 3.1. For each window in f and $f+1$ frame, we compute the Higher Order Moments (HOM) as defined in equation (3).

$$HOM_{\alpha,\beta} = \sum_0^N \sum_0^N (x-\mu_x)^a \left(y-\mu_y\right)^\beta f(x,y) \tag{3}$$

where α, β are the order of HOM and $f(x, y)$ is the image intensity with the corresponding mean μ; N shows the size of the image. Then the moments of each block of the f and $f+1$ frames are compared using the Euclidean distance by the following equation.

(a) Input: Video sequence starting from *f, f+1, f+2.....fn*

(b) *f* and *f+1* frames as input for the first iteration

(c) Static cluster-1 and Dynamic cluster-1 for the first iteration

(d) Static cluster-1 and *f+2* frame as input for the second iteration

(e) Static cluster-2 and Dynamic cluster-2 for the second iteration

(f) Final, Static cluster-6 and Dynamic cluster-6 at the 6th iteration

Fig. 3 Iterative process for separating caption pixels from scene and background pixels

$$d(a,b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \tag{4}$$

where $a = (a_1, a_2, ..., a_n)$ and $b = (b_1, b_2, ..., b_n)$ are two points in the Euclidean n-space. As a result, we get the deviation matrix between the first two frames as shown below.

$$D^{N*N} = \begin{bmatrix} d(a_1, b_1) & \cdots & d\left(a_1, b_N\right) \\ \vdots & \vdots & \vdots \\ d(a_N, b_1) & \cdots & d(a_N, b_N) \end{bmatrix} \tag{5}$$

For the deviation matrix obtained from the frames, *f* and *f + 1*, the method employs a k-means clustering algorithm to classify the pixels which represent low deviation values as a static cluster from high deviation values. As a result, the k-means clustering algorithm gives two clusters: a text cluster containing the caption (Static cluster-1) and a dynamic cluster containing the scene text and background pixels (Dynamic cluster-1) as shown in Fig. 3(c) where one can observe a few non-text pixels misclassified as caption text in the static cluster-1. The method counts the number of edge components in both Static cluster-1 and Dynamic cluster-1. For the second iteration, the gray values corresponding to Static cluster-1 are considered for deviation estimation with the *f + 2* frame (the third consecutive temporal

frame) as shown in Fig. 3(d) where we can see gray pixels which represent pixels in Static cluster-1. The deviation is estimated between only gray patches in Static cluster-1 and the corresponding gray patches in the $f+2$ frame (third frame). As a result of the second iteration, we get two new clusters that are Static cluster-2 and Dynamic cluster-2 as shown in Fig. 3(e). It is observed that the number of edge components in Dynamic cluster-2 have decreased compared to the number of edge components in Dynamic cluster-1. This iterative process continues until the condition satisfies the criterion in which the number of edge components in the Dynamic cluster approaches zero. This is valid because as the iterations increase, the number of non-text pixels from the Dynamic cluster becomes zero, as shown in Fig. 3(f), where the final Static cluster contains caption text pixels and the final Dynamic cluster contains zero edge components after the sixth iteration. To validate this converging criterion, a cumulative graph for the number edge components in the Dynamic clusters is shown in Fig. 4 by adding the number of edge components of the current iteration to the number of edge components of the previous iteration. It can be observed that the number of edge components increases gradually and remains constant as the iterations increase.

The iteration at which the Dynamic cluster has no edge components, is when we consider that the iterative process has met the converging criterion. For the example shown in Fig. 3, the iterative process terminates at the sixth iteration as shown in Fig. 4. Since this step considers the output (candidate text pixels) of the step presented in Section 3.1, the iterative process terminates quickly and accurately. Therefore, it does not require much processing time for separating caption text from scene and background pixels.

It is seen from Fig. 3(f) that the iterative procedure separates caption text from scene and background pixels successfully. Next, the question is how to separate scene text from the results of the Dynamic cluster which contains both scene text and background pixels. Therefore, we consider the complement of the caption text results given by the iterative procedure by referring to the candidate text pixel results given by the step presented in Section 3.1. This results in scene text pixels with background pixels without caption text pixels as shown in Fig. 5(a) where we can see scene text pixels along with the background pixels. In order to separate scene text pixels from Fig. 5(a), we calculate the moments as discussed in the above for the pixels in Fig. 5(a). It is known that text pixels have a high contrast compared to their background [8, 27, 35]. As a result, moments give high values for text pixels and low values for non-text pixels. Since moments calculation provides a gap

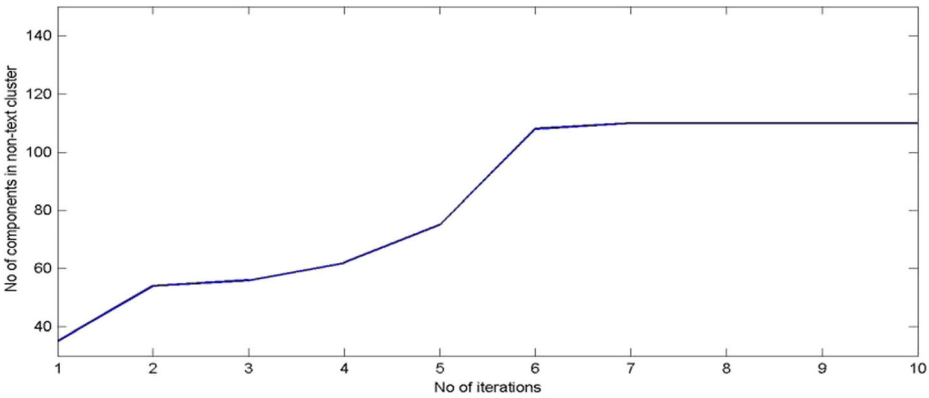
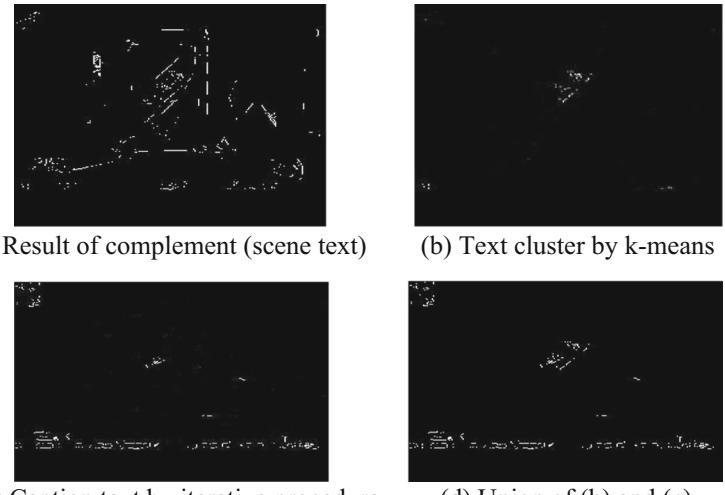


Fig. 4 Cumulative graph of the number of edge components in a Dynamic cluster for different iterations

between text and non-text pixels, we employ k-means clustering with k = 2 on the moments matrix which gives two clusters. The cluster that gives the highest mean is considered as the text cluster and the other one as a non-text cluster, which is shown in Fig. 5(b) where it can be seen that non-text pixels are removed. Furthermore, the proposed method combines caption text (Fig. 5(c)) and the text cluster (Fig. 5(b)) by a union operation as shown in Fig. 5(d) where one can see both caption and scene text pixels, which we called text candidates.

The above procedure considers the case that the caption text is static for few frames and the scene text (background) is dynamic. Suppose the case of the caption text is moving and the scene text (background) is static, in this situation, the iterative procedure misclassifies moving caption text pixels into a dynamic cluster as scene text pixels and scene text into a static cluster as caption text pixels. Since the proposed method use k-means clustering for moments computed for the pixels to classify static and dynamic clusters, both texts are segmented. Therefore, this situation does not have an effect on the final text detection. In the same way, the case of both caption and scene text being static, the iterative procedure classifies both caption and scene text pixels into a static cluster as text pixels, and conversely non-text pixels into a dynamic cluster. K-means clustering on static and dynamic clusters separates the text pixels from the non-text pixels. However, when the proposed method considers moments computed for the pixels as a complement of the output of the iterative procedure, as per the proposed method, there is a high likelihood of misclassifying non-text pixels as text pixels. Therefore, the result of the union operation may contain text and non-text pixels. In this situation, the step to be proposed in subsequent sections is to eliminate false positives. Thus this situation also does not affect final text detection. The same is valid for the case of both caption and scene text being dynamic. In this situation, the iterative procedure classifies pixels of both texts into a dynamic cluster. With the help of k-means clustering, the proposed method separates text from non-text pixels in the dynamic cluster.



(a) Result of complement (scene text) (b) Text cluster by k-means

(c) Caption text by iterative procedure (d) Union of (b) and (c)

Fig. 5 Text candidates of caption and scene texts

3.3 Potential text candidate detection for text detection

Since the considered problem is complex, it is difficult to eliminate false text candidates completely from every input video. For example, the results shown in Fig. 5(d) suffer from false text candidates and a loss of information. To eliminate such false text candidates, we propose to explore the gradient direction of the text candidate pixels. It is noted from [8, 26] that for character components in text lines, most of the gradient directions (*GD*) of pixels show inward directions and a few pixels show an outward direction. The inward direction is defined as the direction which is displayed towards the character. The outward direction is defined as the direction which is displayed towards the nearest neighbor character. Due to the influence of the nearest neighbor character, a few pixels of the text candidates show the direction away from the text candidates. Based on this observation, we formulate a rule that if the text candidate is an actual text candidate then it must have a higher number of inward directions (*ID*) compared to outward directions (*OD*), else the text candidate is considered as a non-text candidate, as defined in equation (5). One such example is shown in Fig. 6, where more pixels of the character in Fig. 6(a) show inward directions (green arrows) compared to outward directions (red arrows) and the pixels of non-text components in Fig. 6(b) do not satisfy this formulation. In addition, it is observed from Fig. 6(b) that the directions of candidate pixels are arbitrary due to the influence of the background. The effect of this formulation can be noted in Fig. 6(c) and Fig. 6(d) where Fig. 6(c) is the final result which combines (union of) the caption and scene text shown in Fig. 5(d), and Fig. 6(d), and does not have any

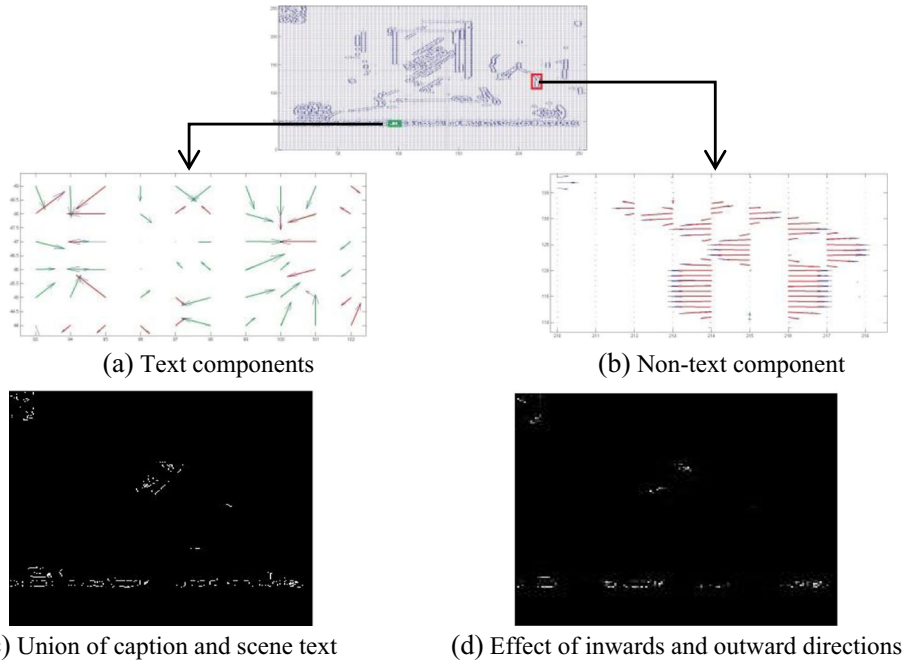


Fig. 6 Potential text candidates from the union of caption and scene text pixels

false text candidates. Therefore, the output of this step is called potential text candidates (PTC) that can be defined as:

$$PTC = \begin{cases} ID(GD_i) > OD\left(GD_{(i)}\right) & : \text{Text} \\ \text{Otherwise} & : \text{Non-text} \end{cases} \quad (6)$$

One can notice from Fig. 6(d) that the image contains a loss of text information because the above formulation and clustering presented in Section 3.2 may eliminate significant text pixels sometimes. To restore full text lines using potential text candidates, we propose boundary growing which extracts edge components from the Sobel edge image of the input frame corresponding to potential text candidates in Fig. 6(d) as shown in Fig. 7(a) where we can see edge components for the corresponding potential text candidates in Fig. 6(d). Next, the boundary growing method fixes bounding boxes for the edge component as shown in Fig. 7(b) where we can see rectangular boxes. It expands the boundary towards the outward gradient direction of the pixel of the edge component, pixel by pixel, in the Sobel edge image of the input frame until it reaches the nearest neighbor components based on the nearest neighbor criterion as shown in Fig. 7(c)-(f), where we can see the boundary of the potential candidates, which is expanded until it finds the nearest neighbor component. This process continues until it reaches the end of the text line as shown in Fig.7(g) where it can be seen that the full text line is covered. Finally, the boundary growing method extracts text from the input frame corresponding to the results in Fig. 7(g), as shown in Fig. 7(h), where full text lines of both caption and scene text are shown. This growing works well because of the fact that the space between the characters is less than the space between words and the text lines as stated in [27].

4 Experimental results

To evaluate the performance of the proposed method, we conducted experiments on standard datasets, namely, ICDAR 2013 video [6], YVT video [16] and ICDAR 2015 [7] videos. In addition, we also created our own dataset of 100 videos of 1 to 6 s that contains arbitrary-oriented, multi-script text along with English text lines because there is no standard data for evaluating arbitrary, multi-lingual text in video. ICDAR 2013 and ICDAR 2015 videos usually contain significant variations in fonts, contrast, font sizes and background variations. YVT data contains only scene text that includes a variety of backgrounds, such as buildings, greenery, sky etc. On the other hand, our data includes multi-lingual text lines of any direction. As a result, the datasets considered provides an opportunity for the proposed methods to detect text in video regardless of type, orientation and scripts.

We use standard measures proposed in [6] for calculating recall (R), precision (P) and f-measure (F) for the ICDAR 2013 and YVT video datasets, while for ICDAR 2015 video data we use the measures used in [7] where it suggests the same formula as in [6] in addition to tracking measures. Note: since the scope of the proposed work is to detect text in video, we do not consider the results estimated for the ICDAR 2013 natural scene dataset in [6].The definitions are as follows. Multiple Object Tracking Precision (MOTP), which expresses how well locations of words are estimated, and the Multiple Object Tracking Accuracy (MOTA), which shows how many mistakes the tracker system made in terms of false

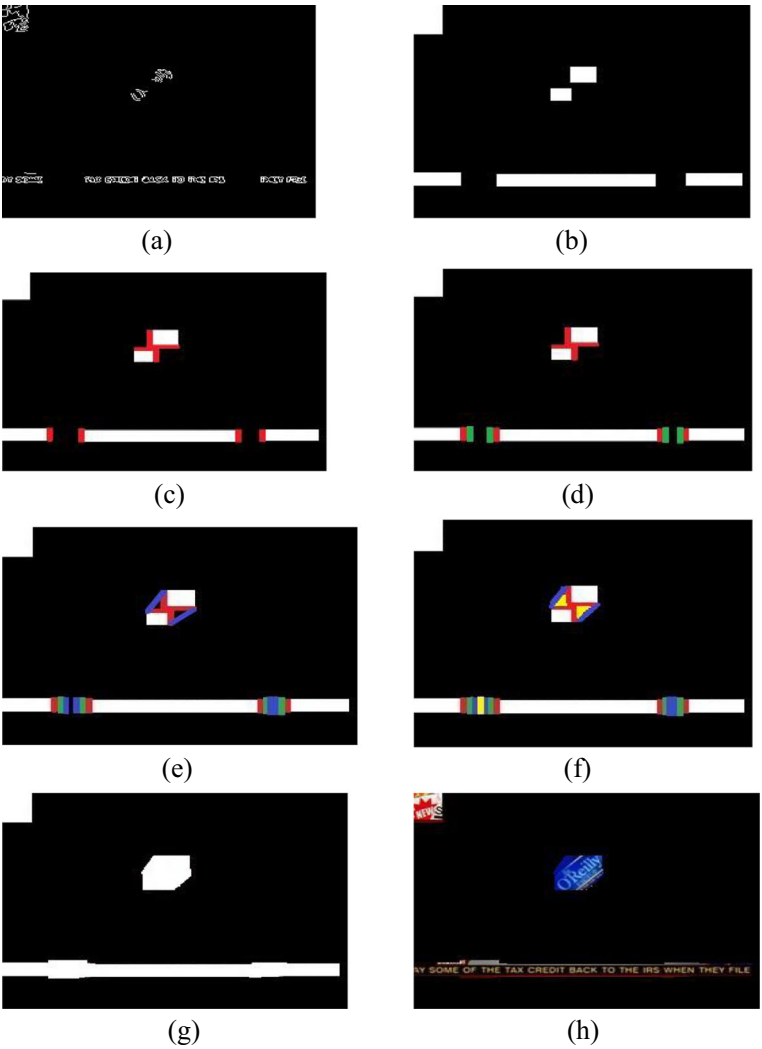


Fig. 7 Boundary growing to detect full multi-oriented text lines: (a) Restored edge components from Sobel edge image of the input frame corresponding to potential text candidates, (b) Boundary fixing for edge components, (c)-(g) Intermediate results of boundary growing towards outward direction of pixels of edge components, (g) Final growing results and (h) Text extraction results

negatives, false positives, and ID mismatches. On the other hand, the Average Tracking Accuracy (ATA) provides a spatio-temporal measure that penalizes fragmentations while accounting for the number of words correctly detected and tracked, false negatives, and false positives. More information can be found in [7]. For our data, we use the same definitions as in [6] at the line level but not the word level. The main reason for calculating measures at the line level is that fixing the bounding box for words in arbitrarily-oriented text is challenging. Besides, for different script text lines, spacing between the words is not consistent as in English for segmenting words accurately. Therefore, we calculate recall, precision and the f-measure for the lines, as it is common practice for video text detection [4, 8, 13, 23–27, 37]. Since the

ground truth is available for all standard data, ICDAR 2013 and YVT video data at the word level, we use the same ground truth for calculating the measures. More information about definitions and equations can be found in [6].

In order to show superiority compared to existing methods, we implemented state-of-the-art methods of different categories, such as the methods of natural scene images, video with temporal and without temporal frames, video with horizontal text and non-horizontal texts, etc. For example, the work in [2], which is a state-of-the-art method, proposes a stroke width transform for text detection in natural scene images, and it is considered as a benchmark method. Liu et al. [13] proposed texture features for text detection in both video and natural scene images. Shivakumara et al. [23] used the combination of Fourier and color for detecting text in both images and video without temporal information. Shivakumara et al. [25] explored a Bayesian classifier for text detection in video without temporal information. Khare et al. [8] used a HOM descriptor for detecting text in video using temporal information. Yin et al. [36] used Maximally Stable Extremal Regions (MSERs) as character candidates and the strategy of minimizing regularized variations for text detection in natural scene images. It is robust to multi-orientations, multi-fonts and multi-sized text. Mi et al. [15] used multiple frames for text detection with temporal frames. Huang, [3] used motion vectors for detecting text in video with temporal information. Zhao et al. [37] used dense corners and optical flow properties for text detection in video with temporal information. Note: For the methods which do not use temporal frames, such as natural scene text detection methods and a few video text detection methods, we extracted key frames from each video to give inputs for the methods. And for tracking results on the ICDAR 2015 video data, words are detected over individual frames by text detection methods and adding them to the tracking system to track the text as per the instructions in [7].

In this work, we use default parameter values of Sobel and Canny edge detectors for all experiments. However, for text line extraction using boundary growing presented in Section 3.3, we use greater than average word gaps plus a few pixels to terminate the boundary growing process when the growing reaches the end of the text line. For the word gaps, the boundary growing studies the distance between neighbor components while growing along the outward gradient direction.

4.1 Analysing the contributions of automatic window and deviation steps

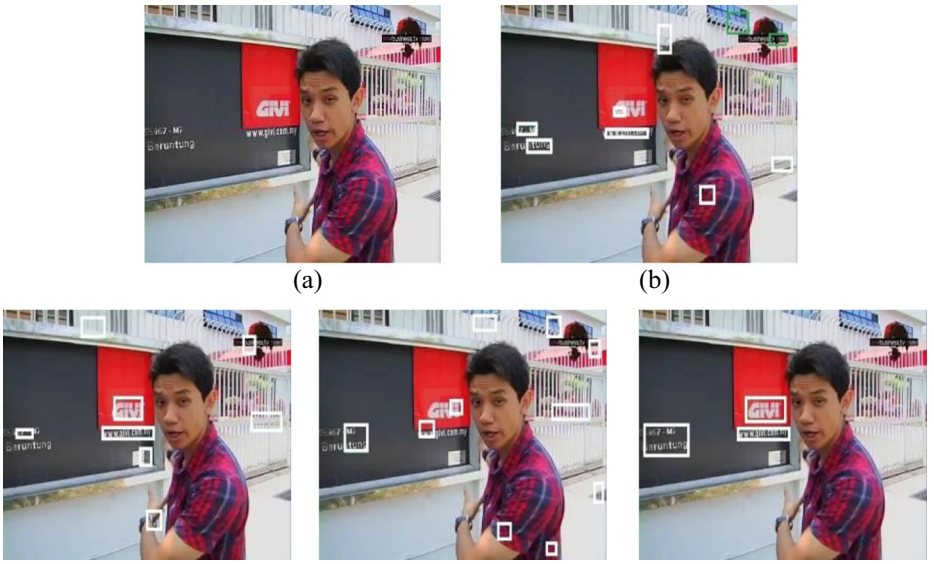
The proposed method involves two key steps for text detection of arbitrarily-oriented multi-lingual text in video. It proposes a method for finding automatic windows based on stroke width distances of the common edge components in Sobel and Canny edge images of the input frame. The same steps also help in identifying candidate text pixels. Another, an iterative procedure, is for the deviation values to the candidate text pixel for separating caption text from scene and background pixels, which in turn results in text candidates of both caption and scene text, and to determine the number of frames. Therefore, in order to know the effect of each step, we conducted experiments on different combinations of these two steps by calculating recall, precision and F-measures. For this experiment, we use our own dataset which includes English and multi-script text lines of different orientations, as discussed in the above section. The reason for only using our data is that it is more complex than the standard video datasets. The combinations of these two steps are (1) With Automatic Window (AW) and Without Deviation (D). In this step, the proposed method deploys the step of automatic window selection and the proposed method calculates the average of the moments for each

sliding window over 10 frames without using an iterative procedure. This results in a moments matrix with the same dimensions of the input frame. Then, it applies k-means clustering with k = 2 for identifying text candidates. In other words, for this experiment, the proposed method does not use an iterative procedure by calculating deviation values between the first frame and successive frames. (2) Without Automatic Window and With Deviation, where the proposed method does not use the step of automatic window selection, it fixes an 8 × 8 window and it uses an iterative procedure for text candidate detection. (3) Without Automatic Window and Without Deviation, is when the proposed method does not use both the steps. (4) With Automatic Window and With Deviation, is where the proposed method uses both the steps for calculation. The effect of these experiments is shown in Fig. 8, where one can see that for input Fig. 8(a), with automatic window and with deviation, this gives better results than other combinations as shown in Fig. 8(e). When we compare the effect of the other three combinations, without automatic window and with deviation, gives better results than with automatic window and without deviation, and without automatic window and without deviation, as shown in Fig. 8(c). It is observed from Fig. 8(b)-(e) that with automatic window detects the big font "GIVI" and the same text is missed without the automatic window. In the same way, Fig. 8(b)-(e) show that with the use of deviation, it detects almost all text lines other than the big font, while without using the deviation, text lines are missed. Therefore, we can conclude that the automatic window selection steps help in detecting multi-fonts and multi-size texts while the iterative procedure for deviation helps in detecting texts without missing any.

In this experiment, we used fixed values, namely 10 successive temporal frames and 8 × 8 dimensions for the window size to analyse the effectiveness of the proposed automatic parameters. The reason is that according to the literature review on the methods [3, 10, 12, 37], which use temporal frames for text detection and the methods [8, 9, 31], which use descriptors or feature extraction with overlapping windows, we realize that 10 successive temporal frames and 8 × 8 dimensions for the window are said to be standard parameter values. It is noted that before setting the above standard parameter values, the existing methods conduct more experiments on different parameter values. Thus we use the same parameter values for the purpose of experimentation. Overall, these two steps help to improve the performance of the proposed method. Experimental results are reported in Table 1 for the different combinations from which we can draw the same conclusions.

4.2 Experiments on arbitrarily-oriented English and multi-lingual video

Sample qualitative results of the proposed and the existing methods for our own English and Multi-lingual videos are shown in Fig. 9 and Fig. 10, respectively. It is noted from Fig. 9 and Fig. 10 that the proposed method detects almost all text and it is better than the existing methods. Fig. 9 includes sample scripts of Telugu which is a south Indian language of the Andhra Pradesh state, Bangla which is the language of West Bengal state of India, as well as Chinese and Arabic, respectively. We can observe from the frames of different scripts that each script has its own structure with different styles, fonts and shapes. This makes the problem more challenging for the method, which does not have the ability to handle multi-lingual script text lines compared to English texts. The main cause for getting poor accuracy from the existing methods is that those methods do not have the ability to handle arbitrarily-oriented text and multi-lingual text in video. As a result, the existing methods miss some texts. It is evident from the quantitative results of the proposed and existing methods for English video and Multi-lingual video reported in Table 2, that the proposed method gives better precision



(a) (b)

(c) Without AW and with D (d) Without AW and without D (e) With AW and with D

Fig. 8 Sample results for different combinations of automatic window selection and deviation for frames selection. AW denotes Automatic Window and D denotes Deviation

and F-measure for multi-lingual data compared to existing methods. For video with English text data, the proposed method is the best at recall, precision and F-measure. The methods reported in [8, 23, 25, 36] score close to the proposed method's results because these methods are robust to non-horizontal orientations and to the extent of multi-lingual script text lines compared to other methods. Table 2 shows that the methods obtained lower results for Multi-lingual video compared to English video including the proposed method. This shows that text detection in multi-lingual video is difficult compared to English video.

4.3 Experiments on standard ICDAR 2013, YVT and ICDAR 2015 videos

Sample qualitative results of the proposed and existing methods for ICDAR 2013, YVT and ICDAR 2015 video are shown in Figs.11-13, respectively, where we note that the proposed method detects almost all text lines in the video frames. The quantitative result of the proposed and existing methods for ICDAR 2013, YVT and ICDAR 2015 video data are reported in Table 3 and Table 4, respectively. Table 3 shows that the proposed method is better than the existing methods for ICDAR 2013 in terms of recall, precision and F-measure and for the YVT

Table 1 Performance of the proposed method for different combinations of Automatic Window (AW) and Deviation (D)

Measures	With AW and without D	Without AW and with D	Without AW and without D	With AW and with D
Recall (R)	55.4	52.7	48.1	65.1
Precision(P)	57.8	53.8	42.6	66.7
F-Measure(F)	56.6	53.2	45.3	65.9

Fig. 9 Sample results of the proposed and existing methods on an arbitrarily-oriented video dataset

Fig. 10 Sample results of the proposed and existing methods on our own multilingual dataset

Table 2 Text detection results of the proposed and existing methods for our English and Multi-lingual video datasets

Methods	Video with english texts			Multi-lingual videos		
	Recall	Precision	F-measure	Recall	Precision	F-measure
Epshtein et al. [2]	58.23	51.17	54.99	20.53	22.9	21.68
Liu et al. [13]	36.3	34.7	35.48	26.9	23.44	25.1
Shivakumara et al. [23]	38.12	62.32	47.30	31.2	29.9	30.92
Shivakumara et al. [25]	65.1	42.62	51.51	35.41	32.5	33.92
Zhao et al. [37]	61.49	57.3	59.33	32.4	31.52	31.96
Huang. [3]	49.3	46.9	48.07	33.6	41.7	32.63
Mi et al. [15]	49.1	49.9	49.41	46.4	46.17	46.31
Khare et al. [8]	53.9	52.7	53.29	48.4	49.3	48.84
Yin et al. [36]	52.4	64.3	57.9	**54.7**	50.31	52.4
Proposed Method	**65.12**	**66.7**	**65.9**	52.3	**53.4**	**52.8**

Bold entries indicate the best results given by the method

video, the proposed method is the best at recall compared to existing methods. The reason for the poor results of the existing method is that the YVT dataset suffers from frequent complex backgrounds while the ICDAR datasets contain low contrast text with background variations. In addition, the ICDAR datasets contain a few European script text lines but the YVT data contains English and Chinese. Table 3 shows that Shivakumara et al. [23], Yin et al. [36] and Khare et al. [8] score results close to the proposed method's results because these methods are robust to contrast variations and multi-fonts and text size. Epshtein et al. [2] developed a method for text detection in natural scene images but not video images. Liu et al.'s [13] method is sensitive to orientations as it was developed for horizontal direction text images. Shivakumara et al. [25] requires proper samples to train the Bayesian classifier. The methods described in [3, 15, 37] fix the number of temporal frames. On the other hand, the proposed method has the ability to handle multi-font, multi-size texts, arbitrary orientations and multi-scripts, and hence it gives better results compared to the existing methods. This can be verified from the illustrations drawn for recall, precision and F-measure of the proposed and existing methods on four datasets in Fig. 14 (a)-Fig. 14(c), respectively.

For ICDAR 2015 video data, we report the results according to [7] where measures are calculated based on the tracking results. In the same way, we report the results of the proposed and existing methods in Table 4. It is noted from Table 4 that the proposed method is the best at MOTA while AJOU [7] is the top performer at MOTP, and Deep2Text is the top performer at ATA. Since the proposed method is developed to handle arbitrary-oriented, multi-lingual text, it scores slightly lower in terms of accuracy for the ICDAR 2015 dataset in terms of MOTP and ATA compared to the best performer in the ICDAR 2015 robust competition. On the other hand, since AJOU, Deep2Text methods are developed for the ICDAR 2015 video data to track the words in the video, the methods achieve the best results. Since other existing methods are developed for text detection but not tracking the text in video, the methods score lower results compared to the proposed methods and the top performers in the ICDAR 2015 robust competition.

In summary, the proposed steps in Section 3.1 identify candidate text pixels based on histogram operations from stroke width distances of common pixels in Sobel and Canny edge

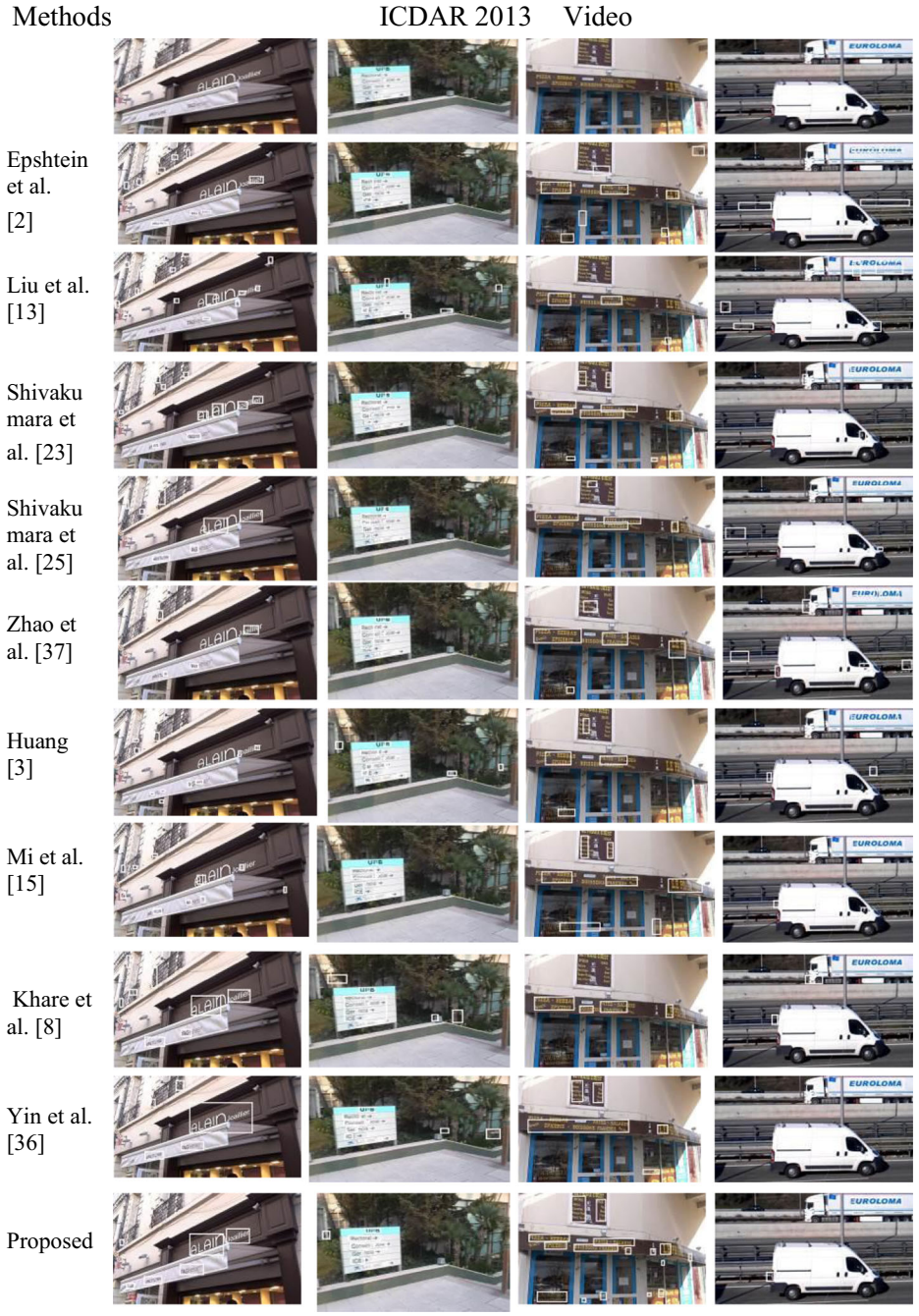


Fig. 11 Sample results of the proposed and existing methods on the ICDAR2013 video dataset

images of the input frame; Section 3.2 identifies text candidates based on an iterative procedure and k-means clustering, and Section 3.3 identifies potential text candidates based on gradient inward and outward direction relationships and the detection of text lines using

Table 3 Text detection results for standard ICDAR 2013 and YVT videos

Methods	ICDAR2013			YVT		
	Recall	Precision	F-Measure	Recall	Precision	F-Measure
Epshtein et al. [2]	32.53	39.80	35.94	40.56	46.22	43.35
Liu et al. [13]	38.91	44.60	41.62	33.73	42.35	39.09
Shivakumara et al. [23]	50.10	51.10	50.59	56.68	**54.38**	**55.43**
Shivakumara et al. [25]	53.71	51.15	50.67	54.53	50.34	52.71
Zhao et al. [37]	46.30	47.02	46.65	51.6	47.96	49.73
Huang, [3]	32.35	32.50	32.42	43.73	44.92	44.31
Mi et al. [15]	40.30	26.92	32.76	40.86	38.1	49.84
Khare et al. [8]	47.6	41.4	44.3	55.2	52.71	53.93
Yin et al. [36]	54.73	48.62	51.56	57.35	51.7	54.43
Proposed Method	**55.9**	**57.91**	**51.7**	**57.9**	52.6	55.16

Bold entries indicate the best results given by the method

boundary growing based on the nearest neighbor criterion, which are invariant to rotation, scaling and scripts. Therefore, the objective of the proposed work has been achieved.

As discussed in the Introduction Section, the primary goal of text detection is to achieve better recognition results for text in video. Since video suffers from low resolution and complex backgrounds, the classical binarization methods [1, 17, 39] give poor recognition rates when we feed the whole video text image directly to the binarization method or OCR. The main reason is that the binarization methods are developed for homogenous backgrounds with high contrast images but not for images such as video. Therefore, to improve the recognition rate of the baseline binarization methods, we segment text lines by the proposed

Table 4 Text detection results for standard ICDAR 2015 videos

Methods	ICDAR 2015		
	MOTP	MOTA	ATA
Epshtein et al. [2]	43.4	37.75	35.28
Liu et al. [13]	42.99	36.95	34.8
Shivakumara et al. [23]	70.22	49.73	39.37
Shivakumara et al. [25]	71.7	32.3	41.29
Zhao et al. [37]	64.61	45.93	34.36
Huang, [3]	44.3	36.62	34.5
Mi et al. [15]	52.86	29.48	32.37
Khare et al. [8]	72.46	**59.89**	40.1
Yin et al. [36]	**73.51**	50.47	41.26
AJOU [7]	73.25	53.45	38.77
Deep2Text-I [7]	71.01	40.77	**45.18**
Proposed Method	73.16	52.93	43.02

Bold entries indicate the best results given by the method

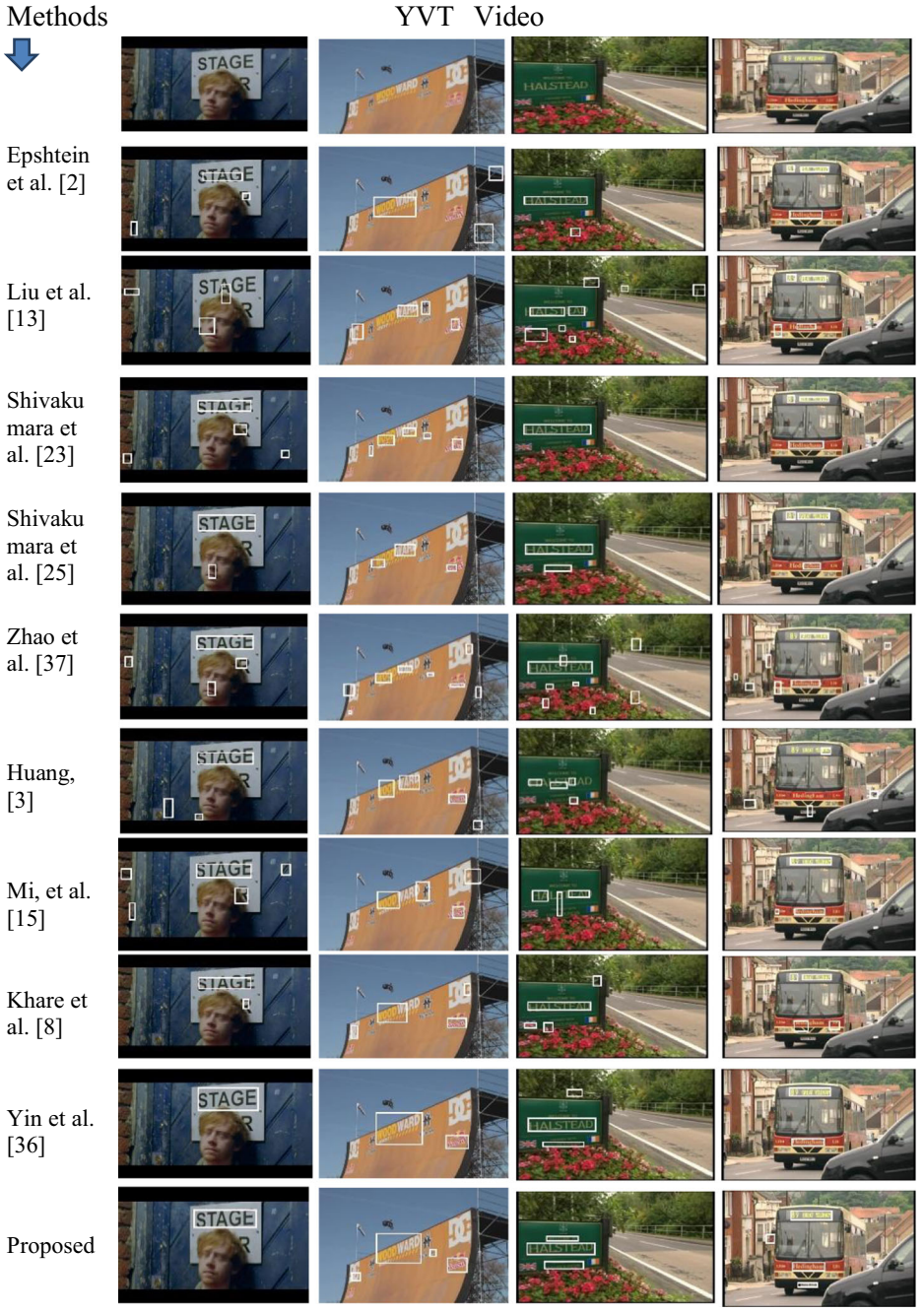
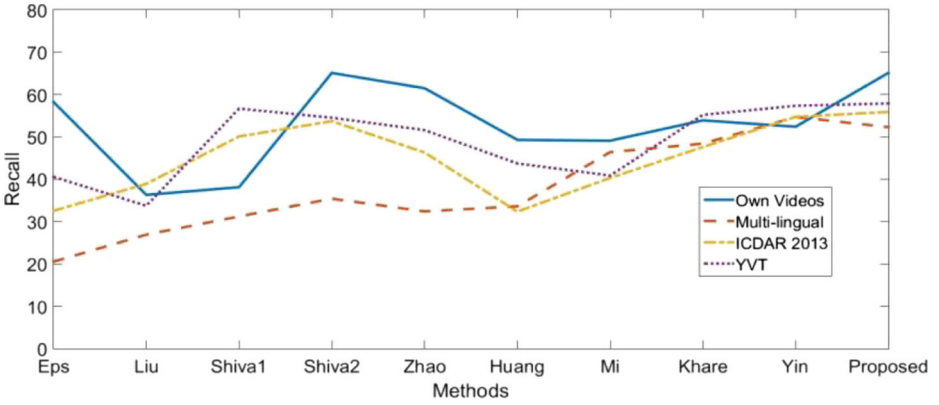


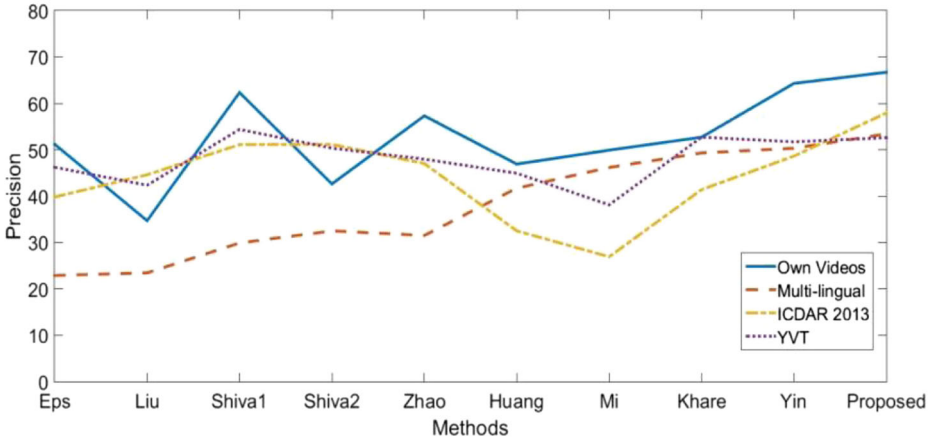
Fig. 12 Sample results of the proposed and existing methods on the YVT video dataset

method from the whole video images, which results in text lines without complex backgrounds. Therefore, we conducted experiments using baseline binarization methods, namely, Zhou et al. [39], which propose multiple criteria for tackling different situations created by low

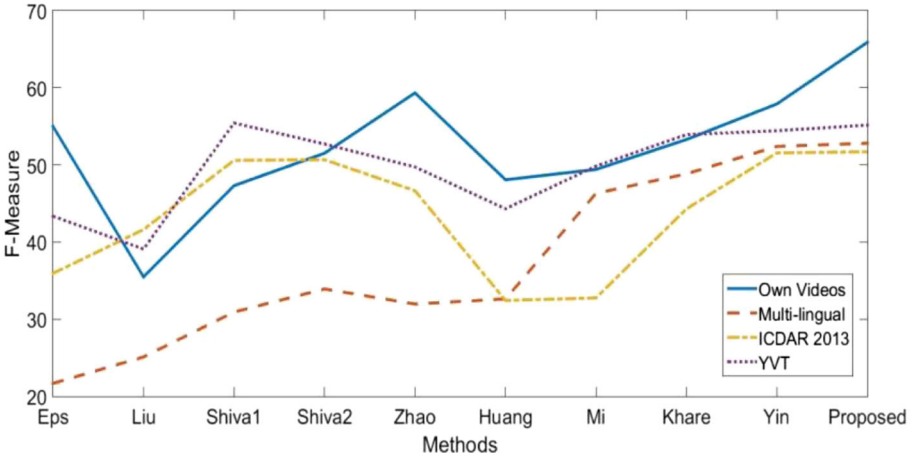
Fig. 13 Sample results of the proposed and existing methods on the ICDAR2015 video dataset



(a) Recall of the proposed and existing method for the different databases



(b) Precision of the proposed and existing method for the different databases



(c) F-measure of the proposed and existing method for the different databases

Fig. 14 Illustrating recall, precision and F-measure of the proposed and existing methods on different databases

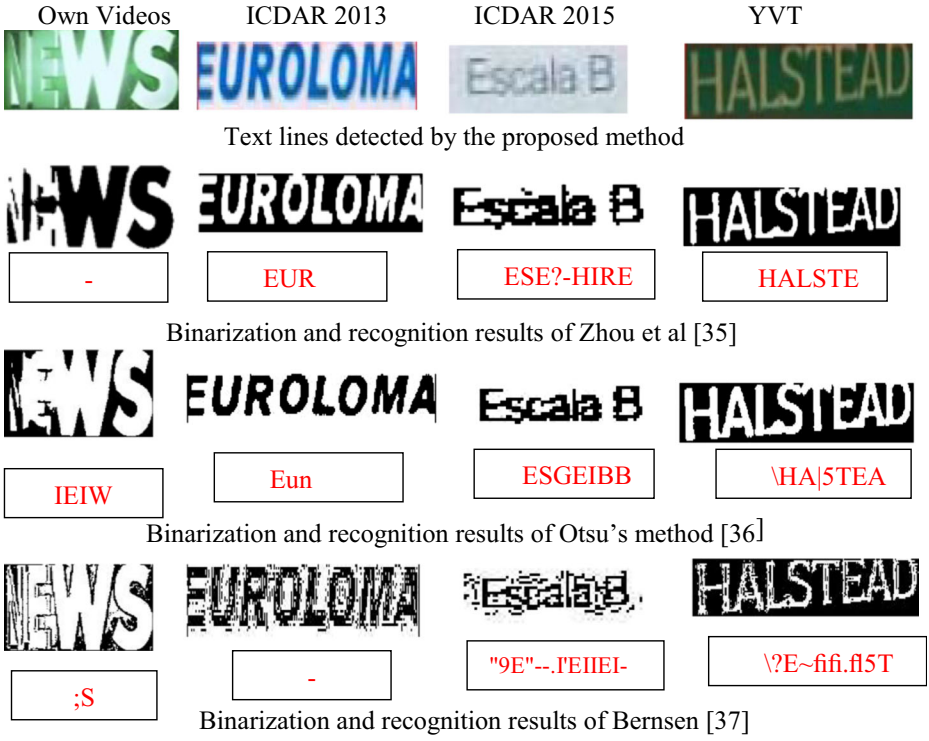


Fig. 15 OCR results for text lines of different databases using three baseline binarization methods. Note: recognition results are shown in a red colour

resolution, non-uniform illumination etc., Otsu [17] which proposes an automatic threshold based on gray values of the images, and Bernsen [1] which proposes a dynamic threshold for binarization of plain document images on text lines detected by the proposed method from the video databases. We choose 100 text lines randomly for each video database for experimentation. We consider the recognition rate at the character level as a measure to evaluate the performance of the binarization methods. For recognition, we use Tesseract OCR which is available publicly [29] by feeding the output of the binarization methods. Sample qualitative results of the three binarization methods for four video databases are shown in Fig. 15 where we can see Zhou et al. gives the better results for almost all texts compared to the other two methods. It seems that Zhou et al.'s method does not depend on threshold values unlike the other two methods, which are dependent on those. Quantitative results of the three binarization methods for different video databases are reported in Table 5 where it can be seen that Zhou et al. scores a better recognition rate as compared to the other two methods, as it works well for

Table 5 OCR results for the different datasets using baseline binarization methods at the character level (in %)

Binarization Methods	Own Videos	ICDAR 2013	ICDAR 2015	YVT
Zhou et al. [39]	78.4	72.51	77.4	76.22
Otsu's [17]	72.1	57.19	64.7	66.39
Bernsen [1]	68.3	58.23	59.29	63.81

low contrast and distorted images, which is quite common for video while the other two methods are sensitive to distortion and low contrast. Usually, when we feed the whole video image to the binarization methods without segmenting the text lines, the methods report recognition rates less than 45 % [21]. Table 5 shows that the binarization methods achieve a greater than 70 % character recognition rate for all the video databases. Therefore, we can conclude that the text line segmentation helps in improving the character recognition rate through the existing binarization methods with the available OCR.

5 Conclusions and future work

In this work, we have proposed a new method for arbitrarily-oriented, multi-lingual text detection in video based on moments and gradient directions. The proposed method introduces a new idea for finding automatic windows based on common stroke width of Sobel and Canny edge images of the input frame. The same step helps in identifying candidate text pixels. Then the proposed method explores temporal information for text candidate detection by introducing an iterative procedure based on the deviation between consecutive frames along with k-means clustering. It also finds a solution to decide the number of temporal frames for identifying text candidates. Furthermore, the gradient inward and outward directions of pixels of the edge components are used in a different way for eliminating false text candidates, which results in potential text candidates. Boundary growing is proposed for potential text candidates to extract full text lines using Sobel edge images based on the nearest neighbor criterion. The experimental results on different standard datasets show that the proposed method outperforms the existing methods. Furthermore, the experimental results show that the proposed method is independent of orientation, scripts, data, and fonts. The performance of the proposed approach may degrade for arbitrary text movements because it expects constant velocity for text detection in this work. Therefore, we plan to extend the proposed work to arbitrary and moving text detection in video in the near future by exploring moments and Fourier combinations.

Acknowledgments The work is also partly supported by the University of Malaya HIR under Grant No: UM.C/625/1/HIR/MOHE/ENG/42. The authors would like to thank the anonymous reviewers for their constructive comments and suggestions, which helped us to improve the quality and to clarify the paper significantly.

References

1. Bernsen J (1986) Dynamic thresholding of gray-level images. In Proc. ICPR, 1251–1255
2. Epshtein B, Ofek E, Wexler Y (2010) Detecting text in natural scenes with stroke width transform. In Proc CVPR, 2963–2970
3. Huang X (2011) A novel approach to detecting scene text in video. In Proc ICISP, 469–473
4. Huang W, Shivakumara P, Tan CL (2008) Detecting moving text in video using temporal information. In Proc ICPR, 1–4
5. Huang X, Ma H, Ling CX, Gao G (2014) Detecting both superimposed and scene text with multiple languages and multiple alignments in video. MTA 70:1703–1727
6. Karatzas D, Shafait F, Uchida S, Iwamura M, Boorda LGI, Mestre SR, Mas J, Mota DF, Almazan JA, De las Heras LP (2013) ICDAR 2013 robust reading competition. In Proc. ICDAR, 1115–1124
7. Karatzas D, Gomez-Bigorda L, Nicolaou A, Ghosh S, Bagdanow A, Iwamura M, Matas J, Neumann L, Chandrsekhar VR (2015) ICDAR 2015 Competition on Robust Reading. In Proc ICDAR, 1156–1160
8. Khare V, Shivakumara P, Raveendran P (2015) A new histogram oriented moments descriptor for multi-oriented moving text detection in video. ESWA 42:7627–7640

9. Li H, Doermann D, Kia O (2000) Automatic text Detection and tracking in digital video. IEEE Trans. IP 9: 147–156
10. Li L, Li J, Song Y, Wang L (2010) A multiple frame integration and mathematical morphology based technique for video text extraction. In Proc ICCIA, 434–437
11. Liang G, Shivakumara P, Lu T, Tan CL (2015) Multi-spectral fusion based approach for arbitrarily-oriented scene text detection in video image, IEEE Trans. IP 24(11):4488–4501
12. Liu X, Wang W (2012) Robustly extracting captions in videos based on stroke-line edges and spatio-temporal analysis. IEEE Trans. MM 14:482–489
13. Liu C, Wang C, Dai R (2005) Text detection in images based on unsupervised classification of edge-based features. In Proc. ICDAR, 610–614
14. Liu X, Fu H, Jia Y (2008) Gaussian mixture modeling and learning on neighboring characters for multilingual text extraction in images. Pattern Recogn 41:484–493
15. Mi C, Xu Y, Lu H, Xue X (2005) A novel video text extraction approach based on multiple frames. In Proc ICICSP, 678–682
16. Nguyen P, Wang K, Belongie S (2014) Video text detection and recognition: dataset and benchmark. In Proc WCACV, 776–783
17. Otsu N (1979) A threshold selection method from gray-level histograms, IEEE Trans. SMAC, 62–66
18. Phan TQ, Shivakumara P, Tan CL (2012) Detecting text in the real world. In Proc ACMMM, 765–768
19. Qian X, Wang H, Hou X (2014) Video text detection and localization in intra-frames of H.264/AVC compressed video. MTA 70:1487–1502
20. Risnumawan A, Shivakumara P, Chan CS, Tan CL (2014) A robust arbitrary text detection system for natural scene images. ESWA 41:8027–8048
21. Roy S, Shivakumara P, Roy PP, Pal U, Tan CL (2015) Bayesian classifier for multi-oriented video text recognition system. Pattern Recogn:5554–5565
22. Shi A, Yao C, Zhang C, Guo Z, Huang F, Bai X (2015) Automatic Script Identification in the Wild. In Proc. ICDAR, 531–535
23. Shivakumara P, Phan TQ, Tan CL (2010) New fourier-statistical features in rgb space for video text detection. IEEE Trans. CSVT 20(11):1520–1532
24. Shivakumara P, Phan TQ, Tan CL (2011) A Laplacian approach to multi-oriented text detection in video. IEEE Trans. PAMI, 33 412–419
25. Shivakumara P, Sreedhar RP, Phan TQ, Lu S, Tan CL (2012) Multi-oriented video scene text detection through Bayesian classification and boundary growing. IEEE Trans. CSVT 22:1227–1235
26. Shivakumara P, Phan TQ, Lu S, Tan CL (2013) Gradient vector flow and grouping based method for arbitrarily-oriented scene text detection in video images. IEEE Trans. CSVT 23:1729–1739
27. Shivakumara P, Dutta A, Tan CL, Pal U (2014) Multi-oriented scene text detection in video based on wavelet and angle projection boundary growing. MTA 72:515–539
28. Su F, Xu H (2015) Robust seed-based stroke width transform for text detection in natural images. In Proc. ICDAR, 916–920
29. Tesseract (2016) http://code.google.com/p/tesseract-ocr/
30. Tian S, Bhattacharya U, Lu S, Su B, Tan CL (2016) Multilingual scene character recognition with co-occurrence of histogram of oriented gradients. Pattern Recogn 51:125–134
31. Wang YK, Chen JM (2006) Detection video texts using spatial-temporal wavelet transform. In Proc. ICPR, 754–757
32. Wu L, Shivakumara P, Lu T, Tan CL (2015) A new technique for multi-oriented scene text detection and tracking. IEEE Trans. MM 17:1137–1152
33. Wu H, Zou BJ, Zhao YQ, Fu HP (2016) An automatic video text detection method based on BP-adaboost
34. Yang H, Quehl B, Sack H (2014) A framework for improved video text detection and recognition. MTA 69: 217–245
35. Ye Q, Doermann D (2015) Text detection and recognition in imagery: a survey. IEEE. Trans. PAMI 37: 1480–1500
36. Yin XC, Yin X, Huang K, Hao HW (2014) Robust text detection in natural scene images. IEEE trans. PAMI 36:970–983
37. Zhao Z, Lin KH, Fu Y, Hu Y, Liu Y, Huang TS (2011) Text from corners: A novel approach to detect text and caption in videos. IEEE Trans. IP 20:790–799
38. Zhou J (2007) A robust system for text extraction in video. In Proc ICMV, 119–124
39. Zhou Y, Feild J, Miller EL, Wang R (2013) Scene text segmentation via inverse rendering, In Proc. ICDAR, 457–461

Vijeta Khare was born in Bhopal (MP), India in 1986. She received her B. Tech degree in Computer Science & Engineering from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, India, in 2006 and M. Tech degree from ABV-Indian Institute of Information Technology and Management, Gwalior, India in 2008. She received a gold medal for her performance during M. Tech program. Currently, she is working towards her Ph. D at University of Malaya, Malaysia. She worked as an Assistant System Engineer at TCS, Hyderabad for 2 years from 2008-2010. Later she joined Amity Engineering College, New Delhi as Assistant Professor. Her research interest includes image processing, pattern recognition and video text processing

Shivakumara P received B.Sc., M.Sc., M.Sc Technology by research and Ph.D degrees from the University of Mysore, Mysore, Karnataka, India in 1995, 1999, 2001 and 2005, all in computer science. Currently, he is working as a Senior Lecturer at University of Malaya (UM), Kuala Lumpur, Malaysia. From 1999 to 2005, he was Project Associate at the University of Mysore, where he obtained M.Sc Technology by research degree and Ph.D degree in computer science and he conducted research on pattern recognition, image processing, document image processing. He worked as a Research Fellow at National University of Singapore, Singapore from 2005-2007. He worked as Research Consultant at Nanyang Technological University, Singapore. He joined back to National University of Singapore, Singapore as a Research Fellow for the period of five years from 2008-2013. Based on his work, he has published more than 150 research papers in national, international conferences and journals. He has been serving as Associate Editor for Transactions on Asian Language Information Processing (TALIP). He got Top Reviewer Recognition award from Pattern Recognition Letter Journal. He has been serving as a Program Committee Member (PCM) for the several International Conferences. His area of research includes video text understanding, document analysis and image processing related

Raveendran Paramesran (SM'03) received the B.Sc. and M.Sc. degrees in electrical engineering from South Dakota State University, Brookings, in 1984 and 1985, respectively. In 1992, he received the Ronpaku Scholarship from Japan to pursue the Doctorate in Engineering degree, which he completed with the University of Tokushima, Tokushima, Japan, in 1994. He was a Systems Designer with Daktronics, South Dakota, before joining the Department of Electrical Engineering, University of Malaya, Kuala Lumpur, Malaysia, as a Lecturer in 1986. He was promoted to an Associate Professor and Professor in 1995 and 2003 respectively. His contributions can be seen in the form of journal publications, conference proceedings, chapters in books, and an international patent to predict blood glucose levels using non-parametric model. He has successfully supervised to completion of 10 Ph.D. students and 12 students in M.Eng.Sc. (Masters by Research). His current research interests include image and video analysis, formulation of moments, analysis of electroencephalography signals. He is a Senior Member of IEEE

Michael Blumenstein is a Professor and Head of the School of Software at the University of Technology Sydney, Australia. He was previously the Head of the School of Information and Communication Technology, and also the Dean (Research) of the Science, Environment, Engineering and Technology Group at the Griffith University in Queensland, Australia. Professor Blumenstein is an internationally and nationally renowned expert in the areas of Pattern Recognition and Artificial Intelligence (specifically Machine learning and Neural Networks). He has published over 150 papers in refereed conferences, journals and books in these areas. His research and consultancy projects span numerous fields of engineering (e.g. Artificial Intelligence-based long-term bridge performance models for the Queensland bridge network), environmental science (e.g. application of artificial neural networks to a flood emergency decision support system) neurobiology (e.g. automated analysis of multidimensional brain imagery) and coastal management (e.g. a predictive assessment tool for beach conditions using video imaging and neural network analysis). Michael has successfully secured and led a number of industry consultancy and commercialization projects with funds exceeding $4.5 Million. Components of his research into the predictive assessment of beach conditions and deterioration models for bridge network maintenance strategies have been developed for use by local government agencies, coastal management authorities and in commercial

applications. He is also the co-founder of Griffith University's App Factory, which builds industry-quality mobile applications, involving student developers and academic researchers. Michael leads a research group in the area of pattern recognition and image processing, which is renowned in the state, nationally and internationally for its expertise in the field and such application areas as environmental and engineering informatics. In recognition of his international standing in his research field, Michael has been invited to serve on the Australian Research Council (ARC) College of Experts (Engineering, Mathematics and Informatics), several Journal Editorial Boards and to act as General Chair, Organising Chair, Program Chair and Committee member for dozens of national/international conferences in his areas of expertise. Following his achievements in applying Artificial Intelligence to the area of bridge engineering (where he has published widely and has been awarded federal funding), he was invited to serve on the International Association for Bridge and Structural Engineering's Working Commission 6 to advise on matters pertaining to Information Technology. Michael is the first Australian to be elected onto this committee. In addition, he was previously the Chair of the Queensland Branch of the Institute for Electrical and Electronic Engineers (IEEE) Computational Intelligence Society and is also the immediate past Chair of the Australian Computer Society's Gold Coast Chapter. He also served as a Board Member of the Australian Computer Society's Queensland Branch Executive Committee. Michael currently serves as the Chairman of the IT Forum Gold Coast and is a Board Member of IT Queensland. Michael has also been invited to serve on the Council for ICT Associations, which is the representative body for the IT industry in Queensland. Michael previously served on the Queensland State Government's ICT Ministerial Advisory Group. In 2009 Michael was named as one of Australia's Top 10 Emerging Leaders in Innovation in the Australian's Top 100 Emerging Leaders Series supported by Microsoft. Michael is a Fellow of the Australian Computer Society and a Senior Member of the IEEE