IJDAR (2015) 18:125–135
DOI 10.1007/s10032-015-0237-z

SPECIAL ISSUE PAPER

Scene text extraction based on edges and support vector regression

Shijian Lu · Tao Chen · Shangxuan Tian · Joo-Hwee Lim · Chew-Lim Tan

Received: 31 May 2014 / Revised: 22 December 2014 / Accepted: 7 January 2015 / Published online: 8 February 2015

Abstract This paper presents a scene text extraction technique that automatically detects and segments texts from scene images. Three text-specific features are designed over image edges with which a set of candidate text boundaries is first detected. For each detected candidate text boundary, one or more candidate characters are then extracted by using a local threshold that is estimated based on the surrounding image pixels. The real characters and words are finally identified by a support vector regression model that is trained using bags-of-words representation. The proposed technique has been evaluated over the latest ICDAR-2013 Robust Reading Competition dataset. Experiments show that it obtains superior F-measures of 78.19 % and 75.24 % (on atom level), respectively, for the scene text detection and segmentation tasks.

Keywords Scene text detection · Scene text segmentation · Scene text recognition · Shape features

S. Lu (✉) · T. Chen · J.-H. Lim
Institute for Infocomm Research, A*STAR, 1 Fusionopolis Way, #21-01 Connexis, Singapore 138632, Singapore
e-mail: slu@i2r.a-star.edu.sg

T. Chen
e-mail: chent@i2r.a-star.edu.sg

J.-H. Lim
e-mail: joohwee@i2r.a-star.edu.sg

S. Tian · C.-L. Tan
School of Computing, National University of Singapore, 21 Lower Kent Ridge Road, Singapore 119077, Singapore
e-mail: tians@comp.nus.edu.sg

C.-L. Tan
e-mail: tancl@comp.nus.edu.sg

1 Introduction

Scene text extraction refers to the detection and segmentation of texts in scene images or videos. It is very important to many vision-related tasks such as guidance for foreign tourists, unmanned vehicle navigation in urban environments, and content-based image indexing and retrieval [5,7–9,13]. Although optical character recognition (OCR) of the scanned document text has become mature, the recognition of texts in scenes is still a big challenge for machines.

Besides the final OCR, another two challenging tasks for an end-to-end scene text recognition system are the detection and the segmentation of texts in scenes. Compared with the well-defined document texts, texts in scenes are often more prone to different variations in terms of text size, font, color, orientation, layout, and context as illustrated in Fig. 1. In addition, texts in scenes usually suffer from different types of degradation such as shadow and low image contrast due to bad illumination, partial occlusion by other objects in scenes, and complex image background. Due to the weak text formatting and the various types of image degradation, it is much more challenging for the detection and segmentation (separation of the text from the image background) of scene texts captured by cameras compared with the detection and segmentation of texts in the scanned document images. Several benchmarking contests have been organized together with ICDAR 2003 [10], ICDAR 2005 [11], ICDAR 2011 [25], and ICDAR 2013 [6]. The best accuracy (F-measure) obtained is still lower than 80 % for both detection and segmentation tasks, which shows that there is still a large improvement space for both tasks.

A number of scene text detection techniques [2,3] have been reported in the literature. One typical approach is sliding window based which scans an image patch by patch, and at each patch, image features are computed to tell how likely

Fig. 1 Extraction of *Texts* in scenes: For the sample images in the *first row* (from the latest ICDAR2013 dataset [6]), the *second row* shows the extracted texts by using the proposed technique

the image patch is text or non-text. For example, Chen et al. [5] build a cascade Adaboost classifier that learns from different texture features such as histogram of intensity and image derivatives. In recent years, the HOG features [16] have been used for scene text detection with very promising results [17–19,29]. To skip the design of handcrafted features, unsupervised feature learning has been proposed [20,21] that first builds a dictionary through K-means clustering and then classifies text and non-text regions using multilayer neural networks. The sliding window-based approach is tolerant to image degradation but often computationally intensive and does not segment texts from the image background.

Another typical approach is connected component based [12,14,22–24,34] which first detects a set of character candidates and then identifies the real text based on different color and shape features. Epshtein et al. [12] propose a stroke width transform (SWT) operator based on the observation that texts in scenes tend to have more constant stroke width compared with most non-text objects. Gradient and color-based image partition is proposed in [24,30] that first extracts candidate character components and then groups them to text lines based on certain joint layout information. In addition, maximally stable extremal regions (MSERs) [26,28] and its extended extremal regions (ERs) [22] have been proposed for the extraction of texts in scenes. The connected component-based approach is more efficient and effective in segmenting texts, but the used components such as ERs are often sensitive to image degradation.

In this paper, we propose a scene text detection and segmentation technique as illustrated in Fig. 1. Given a scene image, a set of candidate text boundaries is first detected based on three text-specific features that are extracted from image edges. For each candidate text boundary, one or more candidate characters are then segmented with a local threshold that is estimated based on the surrounding image pixels. The real (binary) texts are finally identified by a support vector regression model that is trained through a bags-of-words (BoW) image representation method. The proposed technique has a number of contributions. First, it designs three text-specific features and integrates them into a text detector that detects texts in scenes reliably. Second, it designs an adaptive scene text segmentation technique that separates text pixels from the image background accurately. Third, it performs both text detection and text segmentation and outperforms state-of-the-art techniques greatly. In particular, the scene text segmentation technique obtains the best F-measure as reported in [6], and the scene text detection technique significantly outperforms all submissions reported in [6].

The rest of this paper is organized as follows. Section 2 presents the proposed technique including text-specific edge feature design, feature integration for text detection, scene text segmentation based on text boundary, and the final SVR-based false text removal. Experimental results are then presented in Sect. 3 based on two public datasets. Concluding remarks are finally drawn in Sect. 4.

2 Detection and segmentation of texts in scenes

Figure 2 illustrates the framework of the proposed technique. Given a scene image shown in Fig. 2a, three text-specific features shown in Fig. 2c–e are first computed over image edges shown in Fig. 2b that are detected from single-channel image using Canny's edge detector [15]. The three features are then combined to form a joint feature shown in Fig. 2f. The joint feature is integrated across three image channels shown in Fig. 2g and further multiple image scales shown in Fig. 2h. A set of candidate text boundaries shown in Fig. 2i is then determined, and candidate character components shown in Fig. 2j are further extracted based on the detected text boundaries. Finally, candidate characters are clustered into candidate words shown in Fig. 2k, and the real words shown in Fig. 2l are identified by a trained support vector regression model.

2.1 Text-specific edge features

We design three text-specific features to differentiate text and non-text edges. The first feature is text-specific image contrast which is defined as follows:

$$F_1 = \frac{\mu(G_e)}{\sigma(G_e) + \xi} \tag{1}$$

where G_e stores the gradient magnitude of all pixels of the edge component e, $\mu(G_e)$ and $\sigma(G_e)$ denote the mean and standard deviation of G_e, and ξ denotes a small positive number which ensures a nonzero denominator. Different from the conventional image contrast, the image gradient variance is

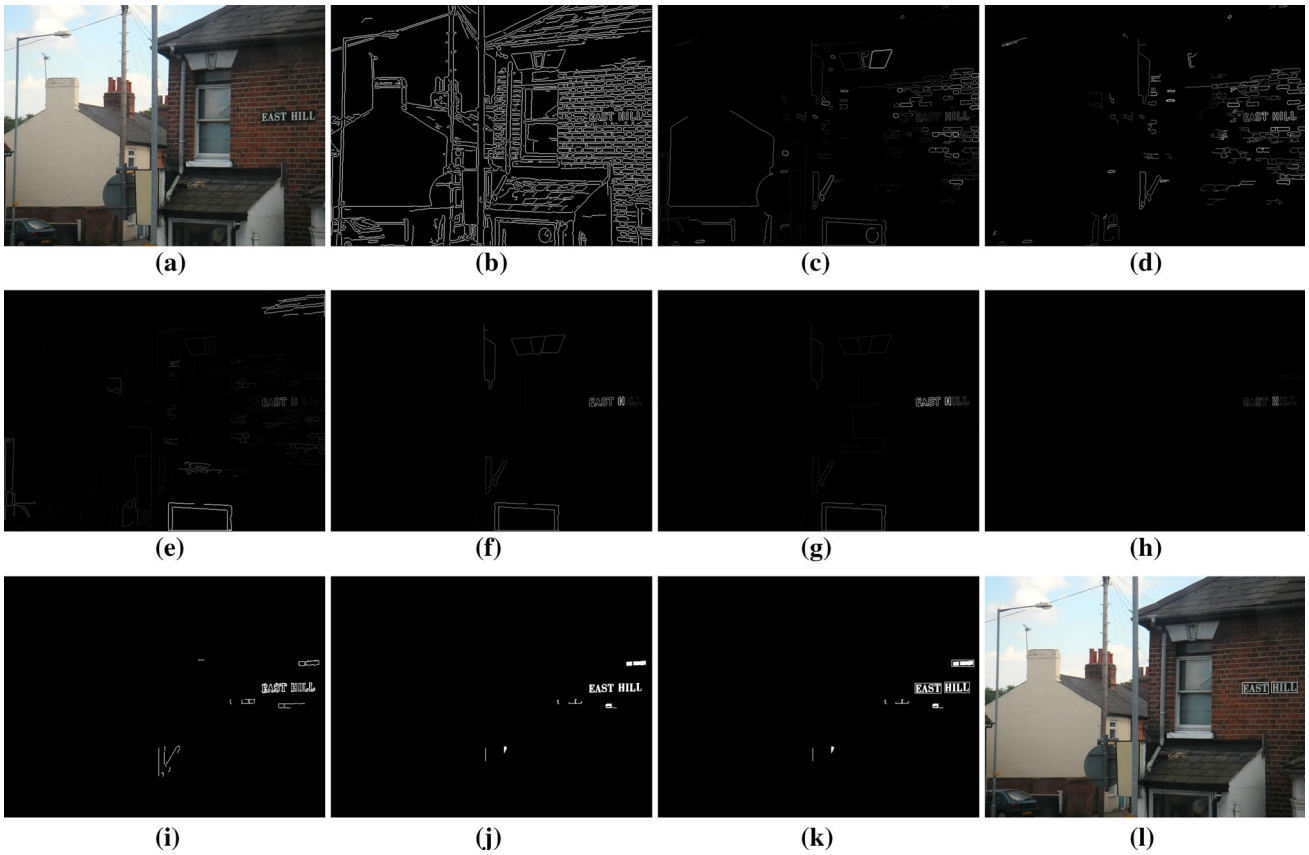

Fig. 2 Process of the proposed scene text localization technique: For the sample image in (**a**), **b**–**e** detected Canny edges and the three text-specific features, respectively. **f**–**h** The combined features (channel Y at the original image scale) and the integration across multiple image channels and multiple image scales, respectively. **i**–**k** Extracted text boundary, the segmented text, and the detected word candidates, respectively. **l** The finally located words within the original scene image (after the false alarm removal using the supported vector regression (SVR) to be described in Sect. 2.4)

incorporated in Eq. 1 to discriminate text and non-text edges as texts in scenes often have a more homogeneous image background, and hence a smaller gradient variance along the text boundary.

The second feature captures a text-specific shape structure that a character often has more than one stroke and accordingly two edge cuts in either horizontal or vertical direction (an edge cut means an intersection between an edge and a horizontal or vertical scan line as illustrated in Fig. 3). This feature is defined as follows:

$$F_2 = R^{\left(\sum_{i=1}^{H} f(cn_i) + \sum_{j=1}^{W} f(cn_j)\right) / (W+H)} \quad (2)$$

where W and H denote the edge width and edge height, respectively, i.e., the width and height of the corresponding edge bounding box, and cn_i and cn_j are the numbers of edge cuts in the ith row and the jth column of the edge, respectively. Function $f(x)$ returns 1 if x is larger than 2 and 0 otherwise. The numerator thus gives the number of edge rows and edge columns that have more than two edge cuts. Compared with non-text edges, text edges often have a much larger F_2 as text edges often have more rows or columns that have more than two edge cuts as illustrated in Fig. 3. Parameter R is used to control the weight of this feature in the finally integrated feature. Extensive experiments on various scene images with texts show that it does not affect the detection performance much when it is set between 10 and 30 (it is fixed at 20 in our implemented system).

Note that there is a smoothing process that sets F_2 of each edge component as the mean F_2 of a number of neighboring edge components. The neighboring edge components are determined based on their relative size and distance to the edge component under study. In particular, a neighboring edge component is detected if its size lies within [0.5 2] of the size of the interested edge component, and its distance to the interested edge component is smaller than the sum of the width and height of both edge components. The smoothing process helps to ensure that digits and letters with a simple structure such as 1, i, I will have a high feature value F_2.

The third feature captures another text-specific structure that each character stroke has a pair of edges, and hence, text edges should have a larger number of rows and columns with

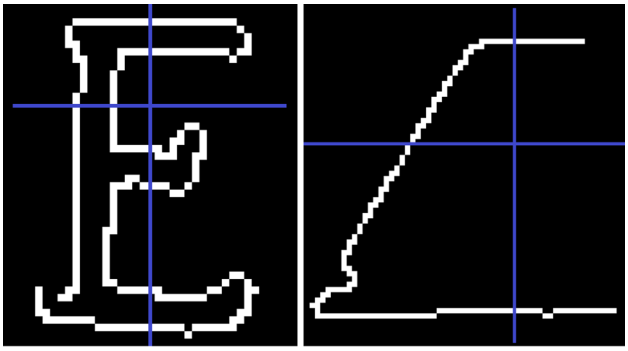

Fig. 3 Illustration of the second and third text-specific edge features: The graph in (**a**) shows edge of the *letter* "E" in the *word* "LITTER" in the second sample image in Fig. 1 and the one in (**b**) shows the edge of a non-text object—the tile around the same height as the word "LITTER" on the *left* but neighboring to the dustbin. The *vertical* and *horizontal lines* in *blue* color are added to illustrate the (number of) cuts between image edges and a *horizontal* and *vertical scan line* for text and non-text objects (color figure online)

an even number of edge cuts compared with non-text edges as illustrated in Fig. 3. This feature is defined by:

$$F_3 = \frac{\sum_{i=1}^{H} g(cn_i) + \sum_{j=1}^{W} g(cn_j)}{W + H} \tag{3}$$

where W, H, cn_i, and cn_j are the same as defined in Eq. 2. Function $g(x)$ returns 1 if x is an even number and 0 if odd. The numerator thus gives the number of edge rows and columns that have an even number of edge cuts. Compared with non-text edges, text edges often have a larger F_3 as they usually have a larger number of rows and columns (normalized by edge width and height) that have an even number of edge cuts as illustrated in Fig. 3.

For the sample image in Fig. 2a, b shows the Canny edges that are detected from the Y channel image (under the YUV color space) at the original image scale. For Canny edges, we simply use the Canny edge detection function that is implemented in MATLAB. The parameters such as the low threshold and high threshold are also set by default in MATLAB. For the Canny edges as shown in Fig. 2b, Fig. 2b–e shows the computed three text-specific features, respectively.

2.2 Feature integration for candidate text detection

The three text features are extracted from edges of different channel images at different image scales. We use the YUV color space where channel Y captures the image lightness information and channels U and V capture the image color information. The YUV is used because a certain amount of text in scenes is printed in colors and text features such as contrast may become very weak within the gray-scale image which captures the intensity/lightness only. Note that all three channel images are first normalized to be [0, 1] before the feature computation as described in the last subsection.

In addition, the three text-specific features are computed at images of different scales. The multi-scale approach is helpful in two different aspects. First, it helps to detect texts of different size because text edges may only be detected properly at certain scales and can be neither too large (edges could be broken) nor too small (edge topology could be lost). Second, it helps to improve the robustness to different types of image degradation, as text edges missed at one image scale, say, due to the complex image background, could be detected properly at another image scale. In our implemented system, we use eight image scales that increase from 0.2 to 1.6 of the original image scale with a step size of 0.2.

For each channel image at one specific image scale, the three text-specific features of each image edge are first combined as follows:

$$S_{i,j} = \prod F_{i,j,k} \tag{4}$$

where $F_{i,j,k}$ denotes the kth feature that is extracted from edges of the jth channel image at the ith scale. Note that $F_{i,j,k}$ is actually a feature image as illustrated in Fig. 2c–e where each pixel has a feature value as computed in Eqs. 1–3 if the pixel corresponds to an edge pixel. The three features are combined through multiplication to discriminate text and non-text edges, as text edges often have fair/high values across the three features but non-text edges do not.

The text-specific features can thus be integrated across images of different channels at different scales as follows:

$$M = \frac{1}{C \times S} \sum_{i=1}^{S} \sum_{j=1}^{C} S_{i,j} \tag{5}$$

where C is the number of image channels and S is the number of image scales. $S_{i,j}$ denotes the combined feature for the jth channel image at the ith scale as defined in Eq. 4. Note that the $S_{i,j}$ computed at different image scales need to be scaled back to the original scale before the averaging.

For the sample image in Fig. 2a, Fig. 2f shows the joint features that combine the three text-specific features in Fig. 2c–e by multiplication. As Fig. 2f shows, the combination of the three text-specific features enhances the discrimination between text and non-text edges greatly. Figure 2g, h shows the integrated features across Y, U, and V channels at the original scale and further across multiple image scales, respectively. As Fig. 2g–h shows, the integration of features across image channels and scales further enhances the discrimination between the text and non-text edges clearly.

Candidate text boundaries can thus be detected through global thresholding of the integrated text features [33]. As text edges usually have much higher feature values than non-

text edges, we simply set the threshold as the mean of the integrated feature image. Besides, we use the external character boundary for scene text segmentation to be described in the next subsection. The internal boundary for characters with holes such as "O" and "B" needs therefore to be identified and removed first. The internal boundary can be identified by those that are completely enclosed by another candidate text boundary. Note that some falsely detected candidate text boundaries such as rectangular traffic sign boundary could enclose multiple real text boundaries and accordingly cause problems. In the proposed technique, such falsely detected text boundaries are identified (and filtered out) if they enclose more than three candidate text boundary components (as a real external character boundary seldom encloses more than three internal boundary components). Similar idea has also been proposed in [31] that makes the use of text boundary topology for false text candidate removal.

For the integrated feature image in Fig. 2h, Fig. 2i shows the binary image after the thresholding and filtering where most text boundaries are detected, whereas non-text boundaries are removed. Note that the detected text boundaries are often more than 1 pixel thick as text edges detected at different image scales may lie at slightly different positions while scaled back to the original image scale. In addition, some non-text boundaries are also detected most of which can be removed by a SVR model to be described in the ensuing sections.

2.3 Scene text segmentation based on text boundary

Segmentation of texts in scenes usually requires adaptive thresholding which estimates a local threshold for each pixel based on the pixels within a neighborhood window. Two critical issues need to be addressed for adaptive thresholding of texts in scenes. First, the relative text brightness needs to be identified as texts in scenes could be either brighter or darker than the background. Second, the size of the neighborhood window which often affects the thresholding performance greatly needs to be estimated adaptively as the size of texts in different scenes could be dramatically different.

We estimate the relative text brightness by detecting a number of text pixels and the surrounding background pixels. In particular, each candidate text boundary is first scanned row by row and column by column as illustrated in Fig. 4. Take horizontal scanning in Fig. 4b as an example. For each scanned row, the first and the last text boundary pixels (in red) neighboring to the background are first located. A pair of background pixels (in green color) is then located 1–3 pixels on the left and right of the located first and last candidate text boundary pixels. A pair of text stroke pixels (in blue) can then be detected that lie 1–3 pixels on the right and left of the text boundary pixels (in yellow) that are dual to the first and last detected text boundary pixels (in red), respectively. The

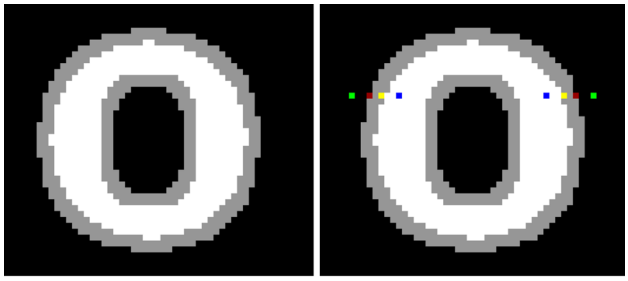

Fig. 4 Illustration of scene text segmentation based on the text boundary: **a** the internal and external text boundaries for character O; **b** candidate text and background pixels detected

relative text brightness can be estimated based on whether the average intensity of the text pixels is larger than that of the background pixels.

The size of the neighborhood window is very important to the performance of the adaptive thresholding. If the neighborhood window is too large, certain neighboring non-text objects could be included which will affect the statistics (such as the mean and variance) of the included pixels and hence the local threshold estimation. If it is too small, certain text pixels may not be segmented properly, e.g., the interior pixels of character strokes will be missed when the size of the neighborhood is smaller than the stroke width (which often happens as texts in scenes could have an ultra-large size). Ideally, the size of the neighborhood window should be adaptive to the size of texts in scenes.

In our proposed technique, the size of the neighborhood window can be adaptively estimated based on the size of the detected candidate text boundary which gives a very close estimation of the size of texts in scenes. In our system, the size of the neighborhood window is set at 1/5 of the height of the text boundary under processing. In addition, we use Niblack's adaptive thresholding technique [1] that estimates a local threshold based on the mean and variance of the surrounding pixels within a neighborhood window as follows:

$$T = \mu + k \cdot \sigma \tag{6}$$

where μ and σ refer to the mean and variance of the values of the neighboring pixels, respectively. Parameter k is set at 0.4 in our system. For the detected text boundary in Fig. 2i, Fig. 2j shows the segmented texts where most text pixels are extracted correctly.

2.4 SVR-based false text removal

Text lines and words can then be located based on the size and spatial layout of the segmented binary components. We first detect text lines through an iterative searching process that exhaustively merges neighboring binary components with

similar height, width, and distance. Specifically, two neighboring components are merged if the nearest distance and the height difference between them are concurrently smaller than 2 and 1 times of the mean of their heights. Words are then detected based on the distance between neighboring component pairs within each detected text line. In particular, an inter-word blank is detected when the nearest distance between two neighboring components is larger than 2 times of the median distances between all neighboring component pairs within the detected text line. For the segmented binary text candidates in Fig. 2j, k shows the detected word candidates where a certain amount of non-text words are also falsely detected.

The real words are identified from the detected word candidates through the combination of bag-of-word (BoW) representation and a support vector regression (SVR) model. In particular, the dense histogram of oriented gradient (HoG) features are extracted from multiscale cells (4×4; 8×8; 12×12) within each training word bounding box. Bag-of-words (BoW) algorithm is then adopted to cluster these HoGs into a histogram to represent each bounding box. The reason to adopt BoW for bounding box representation is that the detected word bounding boxes could have very different sizes and aspect ratios (e.g., the aspect ratio can vary from 1 to 16 in ICDAR 2013 dataset) and should not be normalized to a fixed size which could introduce severe text shape deformation and distortion.

A code book containing k codewords is constructed by k-means clustering of the dense HoG that are extracted from a set of training word bounding boxes. By varying k in a large range from 100 to 1,000, we found that the best performance is achieved when k is set to 400 as illustrated in Fig. 5. A new candidate word bounding box can be quantized into a BoW histogram by matching the extracted HoG with the generated code book. The matching process is described as follows. For each HoG descriptor, the Euclidean distance between each codeword in the codebook and the HoG descriptor is calculated, and the codeword with the closest distance is considered as the matched word for the descriptor. The match count for this codeword is accordingly increased by 1. After all HoG features extracted from the bounding box found their matches with the codebook, a BoW histogram that records the appearance number of each codeword in the bounding box is determined. Note that the training bounding boxes in the proposed technique are those candidate word bounding boxes that are detected from all training images with the training dataset.

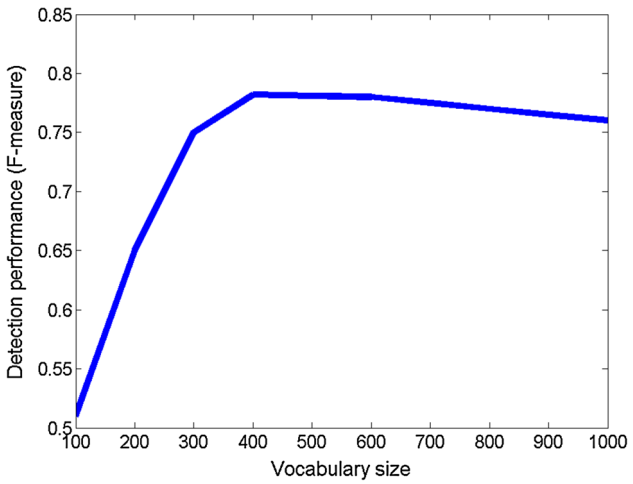

Fig. 5 The relationship between the scene text detection performance and the vocabulary size of the learned code book

A SVR model is designed and applied where a soft score is computed for each detected bounding box. Different from the conventional approach such as support vector machines (SVM) that labels the training examples in a binary manner, we compute a soft score ranging from 0 to 1 to each detected bounding box, where a higher score indicates that the bounding box is more likely to be a real word bounding box. The soft score approach is designed because the detected bounding boxes often have only partial overlap with the ground truth word bounding boxes, e.g., some character within a word is missed to be detected or certain amount of neighboring background is included in the detected word bounding box. Compared with SVM, SVR allows the training samples to have different soft scores which reflect their different importance for model training.

The soft score is computed according to the overlap between the detected bounding box and the corresponding ground truth bounding box. Larger scores indicate that the training samples have more overlap with their ground truth texts and thus are more important for SVR training, while lower scores indicate that the training samples have less overlap and thus less important. The soft score is defined as follows:

$$s(R, G) = \frac{(|R \cap G|)}{(|R \cup G|)} \in [0, 1] \tag{7}$$

where R and G refer to the detected bounding box and its ground truth, respectively. The numerator part is thus the intersection area between the two boxes, and the denominator is the union area of the two bounding boxes. The training box with higher overlap with the ground truth will therefore be assigned with a higher score during SVR learning.

With the BoW histograms H_i and the soft score $s_i, i = 1, 2, \ldots, L$, where L is the number of training bounding boxes, we adopt support vector regression (SVR) [32] to learn a model to correlate the histograms H and the soft scores s for the training samples. The objective functions for SVR are formulated as follows:

$$\min_{\omega, b, \xi_i, \xi_i^*} \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{L}\xi_i + C\sum_{i=1}^{L}\xi_i^* \tag{8}$$

Subject to

$$\begin{aligned} &\omega^T \oslash (H_i) + b - s_i \leq \epsilon + \xi_i \\ &s_i - \omega^T \oslash (H_i) - b \leq \epsilon + \xi_i * i \\ &\xi_i, \xi * i \geq 0, i = 1, 2, \ldots, L \end{aligned} \tag{9}$$

where ω is the normal vector of the hyperplane, b is the bias which is a scalar, and ξ_i and ξ_i^* are the slack variables that are related to prediction errors in SVR. C is a regularization parameter that controls the balance between margin maximization and carried prediction error. Extensive experimentation has been carried out by varying C at different values, e.g., $2^{-2}, 2^{-1}, \ldots, 2^8$. Experiments show that the optimal performance can be obtained when C is set around 128 (2^7). $\oslash(H_i)$ maps H_i into a higher-dimensional space. By introducing the Lagrange multiplier, the above formula can be transformed into its dual form as follows:

$$\begin{aligned} &min_{\alpha,\alpha^*} \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(H_i, H_j) \\ &+\epsilon \sum_{i=1}^{L} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{L} (\alpha_i - \alpha_i^*) \end{aligned} \tag{10}$$

subject to

$$0 \leq \alpha_i, \quad \alpha_i^* \leq C, \quad i = 1, \ldots, L \tag{11}$$

where K is the kernel function. Here, we use norm-2 Chi-square distance function as the kernel which demonstrates great effectiveness for histogram-based distance calculation as described in [35,36].

After solving the above problem, the score d of the BoW histogram H of a detected bounding box can be predicted using following approximation function:

$$d = \sum_{i=1}^{L} s_i (\alpha_i^* - \alpha_i) K(H_i, H) + b \tag{12}$$

After obtaining the scores for all the detected bounding boxes, bounding boxes with low scores can be removed, and this reduces the number of false positive boxes and accordingly improves the detection precision significantly. Figure 2l shows the finally identified words where non-text candidates are identified and removed correctly.

3 Experiments

The proposed technique has been evaluated and compared with state-of-the-art techniques. Two evaluation datasets are first described. Experimental results on the two datasets are then presented and discussed.

3.1 Dataset

The proposed technique has been evaluated extensively by using the ICDAR-2013 public benchmarking dataset that was used in the Robust Reading Competition 2013[1] organized together with the International Conference on Document Analysis and Recognition. The dataset consists of 229 training images and 233 testing images with resolutions from 422×102 to $3{,}888 \times 2{,}592$ pixels. It includes most images from the ICDAR-2003 dataset [10] with a number of new images added. All images in the ICDAR-2013 dataset are color images as illustrated in the first row of Fig. 6. For each image, the text location is provided by the coordinates of the top-left and bottom-right vertices of a bounding box. In addition, text segmentation ground truth is also provided for the training images where a binary image labeling the text regions is given for each training image. In the experiments, the scene text detection is evaluated using the 233 testing images and the scene text segmentation is evaluated using the 229 training images.

The proposed technique has also been evaluated on the Street View Text (SVT) dataset [18]. The SVT dataset consists of 350 scene images with resolutions from $1{,}024 \times 768$ to $1{,}914 \times 898$ pixels. The images within this dataset are all color images and extracted from the Google Street View. Different from the images in the ICDAR-2013 dataset, the SVT images suffer more from noise, poor lighting, and low image contrast as illustrated in the first row of Fig. 7.

3.2 Experimental results

For the scene text detection task, the evaluation method in [27] is used which was specifically designed for scene text detection evaluation. This method is widely used because it considers not one-to-one but many-to-one and one-to-many matches with more details described in [6]. For the scene text segmentation task, the method in [4] is used which evaluates the segmentation performance from both pixel level and atom level as described in [6].

Figures 6 and 7 show qualitative results of the proposed technique on the two public datasets. In particular, the four rows in each figure show several sample images from the ICDAR 2013 dataset and SVT dataset (in Figs. 6, 7), the computed overall feature images, the extracted scene texts, and the finally identified words, respectively. As Figs. 6 and 7 show, the proposed technique is robust and tolerant to the variation in text scale, text fonts, lighting, image contrast, and image context.

On the other hand, the proposed technique may not work well under several scenarios. In particular, it could miss detection and/or segmentation when text edges cannot be

[1] http://dag.cvc.uab.es/icdar2013competition/.

Fig. 6 Illustration of the proposed scene text extraction technique on the ICDAR-2013 dataset: For six sample images shown in the *first row*, the *second*, *third*, and *fourth rows* show the corresponding feature images (integrated across multiple image channels and scales), the segmented text candidates, and the finally identified *words*, respectively

Fig. 7 Illustration of the proposed scene text extraction technique on the SVT dataset: For six sample images shown in the *first row*, the *second*, *third*, and *fourth rows* show the corresponding feature images (integrated across multiple image channels and scales), the segmented text candidates, and the finally identified *words*, respectively

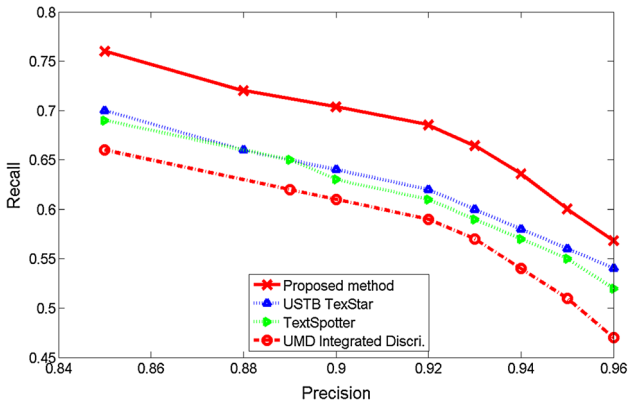

Fig. 8 Receiver operating characteristic (ROC) curves of the proposed scene text detection technique and several best-performing techniques that were submitted to the ICDAR 2013 Robust Reading Competition as listed in Table 1

Table 1 Performance of scene text detection methods on the ICDAR-2013 dataset

Methods	Recall (%)	Precision (%)	F-measure (%)
The proposed method	**69.58**	**89.22**	**78.19**
USTB_TexStar	66.45	88.47	75.89
Textspotter	64.84	87.51	74.49
UMD_Integrated discri.	62.26	89.17	73.33
CASIA_NLPR	68.24	78.89	73.18
Text_detector_CASIA	62.85	84.70	72.16
I2R_NUS_FAR	69.00	75.08	71.91
I2R_NUS	66.17	72.54	69.21
TH-TextLoc	65.19	69.96	67.49
Text detection	53.42	74.15	62.10
Baseline	34.74	60.76	44.21
Inkam	35.27	31.20	33.11

detected properly, typically when (1) texts have an ultra-small character size (e.g., texts within the white color sign in the second image in Fig. 7), (2) texts suffer from severe perspective distortion (e.g., white color texts in the poster at the top-left corner of the last image in Fig. 7, (3) texts have a very complex background (e.g., text in the advertisement board in the third image in Fig. 6), and (4) texts are cluttered by other objects (e.g., street sign text in the fourth image in Fig. 6, etc). In addition, non-text objects may be falsely extracted when they demonstrate certain text-like image and shape characteristics such as the window structures in the fifth image in Fig. 6, the wheel structures in the sixth image in Fig. 6, and the two white color flags in the fifth sample image in Fig. 7.

Quantitative results are also obtained for the ICDAR-2013 dataset. For the scene text detection task, Fig. 8 shows the receiver operating characteristic (ROC) curves of the proposed technique and several best-performing scene text detection techniques (submitted to the Robust Reading Competition 2013 and also listed in Table 1) that are determined by multiple combination of precision and recall. In particular, the optimal F-measure is up to 78.19 % when the precision and recall are 89.22 and 69.58 %. It is clearly higher than the best F-measure as obtained by submissions to the Robust Reading Competition [6] and also significantly outperforms our submitted method (I2R_NUS_FAR) which just obtains a F-measure of 71.91 % as shown in Fig. 8 and Table 1. The big improvement is largely due to the BoW and SVR-based false alarm reduction as described in Sect. 2.4 where most detected non-text candidates are identified and removed accurately.

For the scene text segmentation task, we obtain the best F-score in both pixel level and atom level as shown in Table 2 as extracted from [6]. The good text segmentation performance is largely due to the text boundary-based segmentation approach that is more tolerant to different types of image degradation such as shadow and low image contrast resulting from uneven lighting as described in Sect. 2.3. As Table 2 shows, the proposed technique outperforms all submissions to the Robust Reading Competition 2013 including our submitted method I2R_NUS_FAR. The segmentation improvement over the I2R_NUS_FAR method is largely due to the better scene text localization results as obtained through the SVR-based false alarm reduction as described in Sect. 2.4.

For the SVT dataset, our experiments show that a F-measure of 45.96 % is obtained for the text detection task. This F-measure is much lower compared with those obtained on the ICDAR-2013 dataset. One major cause of the lower F-measure is due to the low text–background contrast where texts in many SVT images are printed in blue, green, or red colors as illustrated in Fig. 7. Another major cause of the lower F-measure is the text font where texts in many SVT images use various art fonts for different special effects. These art font texts are often associated with complex boundary patterns, shadow effects, pop-up effects, etc., as illustrated in Fig. 7, which make the text detection and segmentation a much more challenging task. Note that many state-of-the-art techniques also report much lower accuracy on the SVT dataset. For example, a F-measure of 24.1 % and 37 % is reported in [22,24], respectively.

3.3 Discussion

The proposed technique still has several limitations especially for the scene text detection task as illustrated in Figs. 6 and 7. In particular, the detection may fail to locate texts with a very cluttered background because text edges could be connected to the edge of the surrounding non-text objects. In addition, some non-text objects such as bricks, windows, and car wheels could be falsely detected as some of them have similar shape patterns as texts. Furthermore, it may also fail when scene texts have a very small size (more detection fail-

Table 2 Performance of scene text segmentation methods on the ICDAR-2013 dataset

Methods	Pixel level (%)			Atom level (%)		
	Recall	Precision	F-score	Recall	Precision	F-score
The proposed method	**77.27**	**82.10**	**79.63**	**70.12**	81.20	**75.24**
I2R_NUS_FAR	74.73	81.70	78.06	68.63	80.36	74.03
NSTextractor	60.71	76.28	67.61	63.38	**83.70**	72.14
USTB_FuStar	69.58	74.45	71.93	68.03	72.79	70.33
I2R_NUS	73.57	79.04	76.21	60.33	76.69	67.53
NSTsegmentator	68.41	63.95	66.10	67.98	54.14	60.28
Text_Detection	64.74	76.20	70.01	62.03	57.40	59.63
OTCYMIST	46.11	58.53	51.58	41.77	31.49	35.91

ure is observed when the height of characters is smaller than 15 image pixels) or when the lighting is extremely poor.

Several issues will be further investigated. First, the system extracts the text boundary through global thresholding which may introduce false alarms especially when the image has no texts. The false alarms could be reduced by training a classifier based on the proposed features together with others such as histogram of gradients (HoG). Second, the current system integrates the three features by multiplication, but it is not clear how much each feature contributes to the final text detection. The contribution of each of the four features will be studied, targeting weighted feature integration for better detection performance. Third, recognition of the segmented scene texts will be studied. One major issue is the integration of gray-level and segmented binary texts for better recognition accuracy.

The computational cost of the proposed technique can be broke down into three parts including one for edge feature extraction for the scene text detection, one for the scene text segmentation together with text line and word grouping, and one for SVR-based false alarm reduction. Currently the edge feature extraction part is most time-consuming as it involves the Canny edge detection and text-specific feature computation at multiple image channels and scales. For the SVR-based false alarm reduction, its computational complexity is quadratic with respect to the number of training samples which only occurs in the offline training stage. In the online testing stage, the complexity is linear to the number of testing samples. Overall our system (implemented in MATLAB) takes up to 10 s on average for the processing of an image within the ICDAR-2013 dataset. The computation should be much faster when the system is ported in C/C++.

4 Conclusion

This paper presents a scene text extraction technique. In the proposed technique, a number of text-specific image and shape features are designed to detect the candidate text regions first. BoW model and SVR-based classifier are then trained to remove the falsely detected text regions. Experiments on the ICDAR-2013 dataset show that the proposed technique obtains superior scene text detection and segmentation performance.

References

1. Niblack, W.: An Introduction to Digital Image Processing. Prentice-Hall, Englewood (1986)
2. Liang, J., Doermann, D., Li, H.: Camera-based analysis of text and documents: a survey. Int. J. Doc. Anal. Recognit. **7**(2–3), 84–104 (2005)
3. Jung, K., Kim, K.I., Jain, A.K.: Text information extraction in images and video: a survey. Pattern Recognit. **37**(5), 977–997 (2004)
4. Clavelli, A., Karatzas, D., Llados, J.: A framework for the assessment of text extraction algorithms on complex colour images. In: IAPR International Workshop on Document Analysis Systems, pp. 19–26 (2010)
5. Chen, X., Yuille, A.: Detecting and reading text in natural scenes. In: IEEE International Conference on Computer Vision and Pattern Recognition, pp. 366–373 (2004)
6. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., Gomez i Bigorda, L., Robles Mestre, S., Mas, J., Fernandez Mota, D., Almazan Almazan, J., de las Heras, L.-P.: ICDAR 2013 robust reading competition. In: International Conference on Document Analysis and Recognition, pp. 1484–1493 (2013)
7. Lienhart, R., Wernicke, A.: Localizing and segmenting text in images and videos. IEEE Trans. Circuit Syst. Video Technol. **12**(4), 256–268 (2002)
8. Jain, A.K., Yu, B.: Automatic text location in images and video frames. Pattern Recognit. **31**(12), 2055–2076 (1998)
9. Kim, H.K.: Efficient automatic text location method and content-based indexing and structuring of video database. J. Vis. Commun. Image Represent. **7**(4), 336–344 (1996)
10. Lucas, S.M., Panaretos, A., Sosa, L., Tang, A., Wong, S., Young, R.: ICDAR 2003 robust reading competitions. In: International Conference on Document Analysis and Recognition, pp. 682–687 (2003)
11. Lucas, S.M.: ICDAR 2005 text locating competition results. In: International Conference on Document Analysis and Recognition, pp. 80–84 (2005)
12. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: IEEE International Conference on Computer Vision and Pattern Recognition, pp. 2963–2970 (2010)

13. Datta, R., Joshi, D., Li, J., Wang, James Z.: Image retrieval: Ideas, influences, and trends of the new age. ACM Comput. Surv. **40**(2), 1–60 (2008)
14. Shivakumara, P., Phan, T.Q., Tan, C.L.: A Laplacian approach to multi-oriented text detection in video. IEEE Trans. Pattern. Anal. Mach. Intell. **33**(2), 412–419 (2011)
15. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern. Anal. Mach. Intell. **8**(6), 679–698 (1986)
16. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE International Conference on Computer Vision and Pattern Recognition, pp. 886–893 (2005)
17. Mishra, A., Alahari, K., Jawahar, C.V.: Top-down and bottom-up cues for scene text recognition. In: IEEE International Conference on Computer Vision and Pattern Recognition, pp. 2687–2694 (2012)
18. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: IEEE International Conference on Computer Vision, pp. 1457–1464 (2011)
19. Wang, K., Belongie, S.: Word spotting in the wild. In: European Conference on Computer Vision, pp. 591–604 (2010)
20. Coates, A., Carpenter, B., Case, C., Satheesh, S., Suresh, B., Wang, T., Wu, D.J., Ng, A.Y.: Text detection and character recognition in scene images with unsupervised feature learning. In: International Conference on Document Analysis and Recognition, pp. 440–445 (2011)
21. Wang, T., Wu, David J., Coates, A., Ng, A.Y.: End-to-end text recognition with convolutional neural networks. In: International Conference on Pattern Recognition, pp. 3304–3308 (2012)
22. Neumann, L., Matas, J.: Real-time scene text localization and recognition. In: IEEE International Conference on Computer Vision and Pattern Recognition, pp. 3538–3545 (2012)
23. Yao, C., Bai, X., Liu, W., Ma, Y., Tu, Z.: Detecting texts of arbitrary orientations in natural images. In: IEEE International Conference on Computer Vision and Pattern Recognition, pp. 1083–1090 (2012)
24. Yi, C., Tian, Y.: Text string detection from natural scenes by structure-based partition and grouping. IEEE Trans. Image Process. **20**(9), 2594–2605 (2011)
25. Shahab, A., Shafait, F., Dengel, A.: ICDAR 2011 robust reading competition challenge 2: reading text in scene images. In: International Conference on Document Analysis and Recognition, pp. 1491–1496 (2011)
26. Chen, H., Tsai, S.S., Schroth, G., Chen, D.M., Grzeszczuk, R., Girod, B.: Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In: International Conference on Image Processing, pp. 2609–2612 (2011)
27. Wolf, C., Jolion, J.: Object count/area graphs for the evaluation of object detection and segmentation algorithms. Int. J. Doc. Anal. Recognit. **8**(4), 280–296 (2006)
28. Shi, C., Wang, C., Xiao, B., Zhang, Y., Gao, S.: Scene text detection using graph model built upon maximally stable extremal regions. Pattern Recognit. Lett. **34**(2), 280–296 (2012)
29. Pan, Y.F., Hou, X., Liu, C.L.: A hybrid approach to detect and localize texts in natural scene images. IEEE Trans. Image. Process. **20**(3), 800–813 (2011)
30. Yi, C., Tian, Y.: Localizing text in scene images by boundary clustering, stroke segmentation, and string fragment classification. IEEE Trans. Image. Process. **21**(9), 4256–4268 (2012)
31. Kasar, T., Kumar, J., Ramakrishnan, A.G.: Font and background color independent text binarization. In: International workshop on Camera Based Document Analysis and Recognition (workshop of ICDAR), pp. 3–9 (2007)
32. Basak, D., Pal, S., Patranabis, D.C.: Support vector regression. Neural Inf. Process. **11**(10), 203–224 (2007)
33. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–65 (1979)
34. Huang, W., Lin, Z., Yang, J., Wang, J.: Text localization in natural images using stroke feature transform and text covariance descriptors. In: IEEE International Conference on Computer Vision, pp. 1241–1248 (2013)
35. Chen, T., Yap, K.-H., Zhang, D.J.: Discriminative soft bag-of-visual phrase for mobile landmark recognition. IEEE Trans. Multimed. **13**, 612–622 (2014)
36. Li, T., Mei, T., Kweon, I.-S., Hua, X.S.: Contextual bags-of-words for visual categorization. IEEE Trans. Circuits Syst. Video Technol. **21**, 381–392 (2010)