Multimed Tools Appl (2017) 76:26201–26223
DOI 10.1007/s11042-016-4099-2

A hierarchical recursive method for text detection in natural scene images

Xiaobing Wang[1] · Yonghong Song[1] · Yuanlin Zhang[1] · Jingmin Xin[1]

Received: 6 April 2016 / Revised: 26 August 2016 / Accepted: 27 October 2016/
Published online: 12 December 2016

Abstract Text detection in natural scene images is a challenging problem in computer vision. To robust detect various texts in complex scenes, a hierarchical recursive text detection method is proposed in this paper. Usually, texts in natural scenes are not alone and arranged into lines for easy reading. To find all possible text lines in an image, candidate text lines are obtained using text edge box and conventional neural network at first. Then, to accurately find out the true text lines in the image, these candidate text lines are analyzed in a hierarchical recursive architecture. For each of them, connected components segmentation and hierarchical random field based analysis are recursively employed until the detected text line no more changes. Now the detected text lines are output as the text detection result. Experiments on ICDAR 2003 dataset, ICDAR 2013 dataset and Street View Dataset show that the hierarchical recursive architecture can improve text detection performance and the proposed method achieves the state-of-art in scene text detection.

Keywords Scene text detection · Hierarchical recursive architecture · Hierarchical random field · Text edge box

✉ Yonghong Song
songyh@mail.xjtu.edu.cn

Xiaobing Wang
wxbxj@stu.xjtu.edu.cn

Yuanlin Zhang
ylzhangxian@mail.xjtu.edu.cn

Jingmin Xin
jxin@mail.xjtu.edu.cn

[1] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, 710049, China

1 Introduction

Recently, visual content analysis attracts much attention in computer vision including object detection [3], action detection [27], event detection [30] and so on. As a special kind of object, text can be widely applied in multimedia retrieval, mobile based text recognition, navigation, industrial automation and so on. Text detection becomes a research focus these years. Similar as event detection which may involves different objects, scenes and actions, text detection involves various texts in complex scenes while actions are not involved. Therefore, while optical character recognition (OCR) is viewed as a solved problem which achieves 99 % recognition rates on scanned documents, text detection in natural scene images is still a challenging problem. Scene images usually have complex backgrounds and texts in them have different sizes, fonts, colors, languages, arrangement orientations and illumination conditions. Meanwhile, blurring, degradation and distortion may occurs in them. These situations make detecting text from scene images difficult.

In our previous work [29], a two-steps text detection method based on multi-layer segmentation and higher order conditional random field was proposed, which could detect various texts in complex scenes. However, this method could not detect text with strong light or too complex background, especially text connected to background targets. Meanwhile, in some detected text lines several characters are missed. Considering that texts in natural scenes usually are not alone and arranged into lines for easy reading, we try to find candidate text lines first. And then segmentation is done to these candidate text lines. Now texts with complex background are more likely to be separated from the background. Meanwhile, recursive analysis is done to each candidate text line. With this the complete text lines can be detected and detection accuracy can be improved. Therefore, a text line based hierarchical recursive text detection method is proposed in this paper.

The proposed text detection method includes two steps: candidate text lines generation and hierarchical recursive candidate text lines analysis. In candidate text lines generation, text proposals are obtained using text edge box, which is a modified edge box for text. Candidate text regions are obtained using a conventional neural network (CNN) based text detector to classify text proposals, and then are linked into candidate text lines. In hierarchical recursive candidate text lines analysis, each candidate text line is recursively processed. The processing text line is segmented into connected components (CCs) first, which are then verified with a hierarchical random field (HRF) model. Connected component (CC) segmentation and CC analysis are recursively employed until the detection result of this line no more changes. The detected text lines are grouped into words for easy evaluation. The proposed method is distinguished by the following three contributions:

- Text edge box is proposed in this paper, which is a modified edge box and suitable for text proposals. Candidate text regions are obtained using text edge box and CNN based text detector.
- CCs in the candidate text lines are analyzed with a HRF model. Unary CCs, CC pairs and CC lines are hierarchical layers of the graph in this model. The information from these three levels is integrated for distinguishing text from non-text.
- Each candidate text line is recursively processed to detect text. CC segmentation and CC analysis are recursively used until the detection result no more changes.

The rest of this paper is organized as follows. Some related works about text detection in natural scene images are introduced in Section 2. A overview of the proposed method is presented in Section 3. The details of candidate text lines generation and hierarchical recursive

candidate text lines analysis are separately described in Sections 4 and 5. Experiments for performance evaluation of the proposed method are shown in Section 6. Finally, conclusions are stated in Section 7.

2 Related work

Early text detection methods in natural scene images are simple, which use text features such as edges, texture and so on to obtain candidate text regions and then identify text ones from them. Ye et al. [32] employ edge features and morphology operation to locate edge-dense image blocks as candidate text regions. Then, wavelet-based features and a support vector machine (SVM) classifier are used to identify text from the candidate ones.

With the development of computer vision, many researchers attempt to detect text which is a special object using object detection methods. However, common object detection methods only can process one kind of object while text is more complex, which has different characters, fonts, languages and so on. Therefore, text detection is more difficult than common object detection. Chen and Yuille [5] use sliding windows to get local image regions as candidate text regions and a cascade with 4 Adaboost classifiers containing 79 features to remove non-text regions. Zhu et al. [37] propose a Non-Linear Niblack method to segment the input image into CCs, which are candidate text. Then, a cascade of Adaboost classifiers containing 12 CC features is used to remove non-text. Epshtein et al. [8] propose Stroke Width Transform (SWT) to compute the stroke width of each pixel according to which the image is segmented into CCs. Then CC analysis is employed to identify text. Due to strokes are basic elements of text, this research is important for text detection. Neumann and Matas [16] employ Maximally Stable Extremal Regions (MSER) as candidate text. Then effective pruning is used to group character regions by an exhaustive enumeration of all character sequences. Higher-order properties of text such as word text lines are exploited to verify text. MSER is useful for finding candidate text and many MSER-based methods [17, 33] are proposed these years.

As time goes on, much work has been devoted to text detection. Many modified methods are proposed for solving this problem, which are based on the previous researches and have better performances. Wang et al. [26] use a CNN with two convolutional layers to classify sliding windows and then text regions are obtained by applying non-maximal suppression (NMS) to the remanning windows. Pan et al. [20] propose a hybrid approach to localize texts in natural scene images by integrating region-based information into a robust CC-based method. Candidate text regions are obtained using a region-based method and then they are segmented into CCs by Niblack. Text is identified using CC analysis with unary CC properties and binary contextual CC relationships integrated by CRF model. Yao et al. [31] propose a method which detects texts of arbitrary orientations in natural images. SWT is employed to obtain candidate text CCs. These CCs are verified through CC analysis and CC chain analysis to identify text. Neumann and Matas [18] propose a novel feature based on Stroke Support Pixels (SSPs) to select text CCs from MSERs. Text lines are generated by linking them. Then, each text line is further refined through an iterative segmentation approach of GrabCut to obtain text.

Though many methods have been proposed to solve this problem, text detection in natural scenes is still challenging and the gap between the technical status and the required performance is large. Researches on this problem should be go on to improve the performance.

3 Overview of the proposed method

To robust detect various texts in complex scenes, a hierarchical recursive text detection method is proposed in this paper. As shown in Fig. 1, the proposed method includes two steps:

- **Candidate text lines generation**. When an input image is given, text edge box is used to obtain text proposals, which are then processed through a CNN based text detector to obtain candidate text regions. By linking these regions, candidate text lines are generated.
- **Hierarchical recursive candidate text lines analysis**. For each candidate text line, two CC segmentation methods including multi-layer segmentation and color clustering are combined to segment CCs for initialization. Text CCs are identified through CC analysis based on a HRF model. Then, CC segmentation combining graph cut based segmentation and multi-layer segmentation and HRF based analysis are recursively done to the text line until the detected text no more changes. Now the detected text in candidate text lines make up the final detection result of the input image.

4 Candidate text lines generation

4.1 Text edge box

Edge box [36] is a useful method for locating object proposals using edges. Its observation is that the number of contours wholly enclosed by a bounding box is indicative of the likelihood of the box containing an object. Edges tend to correspond to object boundaries, and as such boxes that tightly enclose a set of edges are likely to contain an object. A simple box objectness score is proposed that measures the number of edges that exist in the box minus those that overlap the box's boundary. This score can be computed as follows:

$$h_b = \frac{\sum_i w_b(s_i) m_i - \sum_{p \in b^{in}} m_p}{2(b_w + b_h)^k}, \tag{1}$$

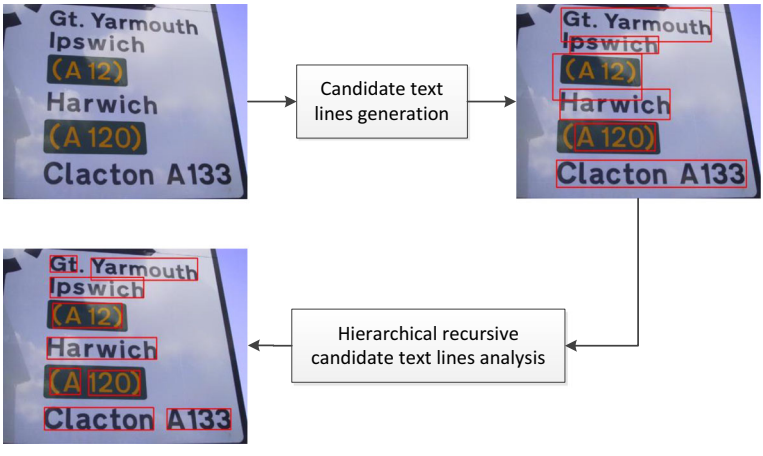

Fig. 1 The flowchart of the proposed method for text detection in natural scene images. The detected text lines are grouped into words for easy evaluation

where m_i, m_p are gradient magnitudes of edges, $w_b(s_i)$ indicates whether the edge s_i is wholly contained in the box b, b_w and b_h are the box's width and height and $k = 1.5$ is used to offset the bias of larger windows having more edges on average.

Edge box has been used for text proposals generation and get better performance than other region proposal methods [9]. However, this method is not designed for text and many boxes containing non-text objects are extracted as text proposals. Therefore, edge box is modified for text proposals and text edge box is proposed in this paper.

The proposed text edge box is a modified edge box for text, which adds a textness weight based on gradient orientation histogram. Text composes of strokes, whose edge pixels usually have similar gradient magnitudes and opposing gradient directions. Therefore, the textness weight of a box can be computed as follows:

$$w_{text} = 1 - \frac{abs(w_1 - w_3) + abs(w_2 - w_4)}{\sum_{i=1}^{4} w_i}, \tag{2}$$

where edge pixels in this box are grouped into 4 types according to their gradient orientations θ, Type 1: $0 < \theta \leq \pi/4$ *or* $7\pi/4 < \theta \leq 2\pi$, Type 2: $\pi/4 < \theta \leq 3\pi/4$, Type 3: $3\pi/4 < \theta \leq 5\pi/4$, Type 4: $5\pi/4 < \theta \leq 7\pi/4$ and w_i is the sum of the edge gradient magnitudes in this group.

When the edge map of an input image is computed using the edge detector based on structured forests [7], the object score h_b of a box b can be obtained using (1) and its textness weight w_{text} can be obtained using (2). Then, the text edge box score of the box can be computed as follows:

$$h_{tb} = w_{text} * h_b. \tag{3}$$

Though the search for candidate bounding boxes using a sliding window search over position, scale and aspect ratio with a score threshold, many candidate boxes are obtained. Based on their text edge box scores, these boxes are sorted. After performing NMS of the sorted boxes, text proposals are generated.

Compared with edge box, the number of extracted text proposals using text edge box is much less while text included in them is not less, as shown in Fig. 2. Many text proposals whose main parts are non-text are removed. That means that the number of objects which are needed to be processed in the next step is much reduced while the recall of text detection is not lower with text edge box.

4.2 Candidate text regions generation and linking them into lines

After text proposals are extracted from the input image using text edge box, there are some non-text proposals included in them. Therefore, these proposals should be processed to get candidate text regions.

Usually texture features of proposals (bounding boxes) such as Histograms of Oriented Gradients (HOG) [20], Local Binary Pattern (LBP) [19] and so on are used to distinguish text and non-text. However, these features can't well represent text. Moreover, with the development of deep learning, deep learning methods especially conventional neural network (CNN) are widely used for object detection and recognition [1]. CNN has been used for text detection and gained better performance [9, 26]. Therefore, CNN is employed as a text detector to classify text proposals here.

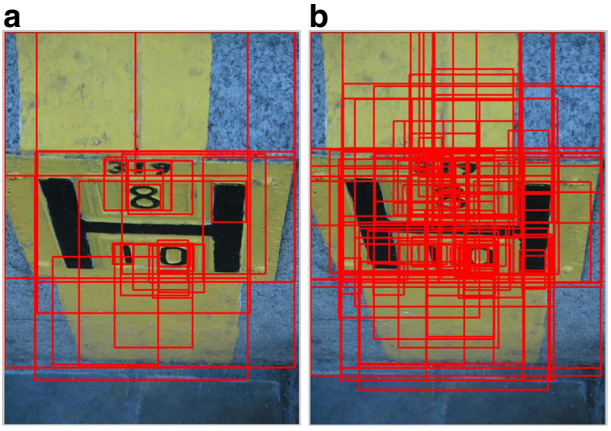

Fig. 2 Text proposals extracted with text edge box and edge box. With the score threshold is set to 0.05, the number of extracted proposals is 30 with text edge box as shown in (**a**) while the number is 134 with edge box as shown in (**b**)

As shown in Fig. 3, the CNN based text detector has three layers with two conventional ones and one full connection one. Each conventional layer contains two steps: convolution and sub-sampling. As the size of the input of CNN based text detector is $32 * 32$, text proposals obtained above are resized to $32 * 32$ first. Then, these proposals are classified using the CNN. These proposals classified as text are candidate text regions, based on which candidate text lines are generated.

Usually, texts in natural scene images are not alone and they are arranged into horizontal lines for easy reading. Moreover, these text in the same text line always have similar properties such as sizes, colors and so on and they are nearby in the spatial space. Based on these observations, candidate text lines can be generated by linking candidate text regions similar as linking CCs into lines [28].

First, candidate text regions are grouped into candidate text pairs if the following three constrains are employed in the proposed method:

$$\begin{aligned} H_{ab}/max(W_a, W_b) &< T_h, \\ V_{ab}/max(H_a, H_b) &< T_v, \\ min(H_a, H_b)/max(H_a, H_b) &> T_s. \end{aligned} \tag{4}$$

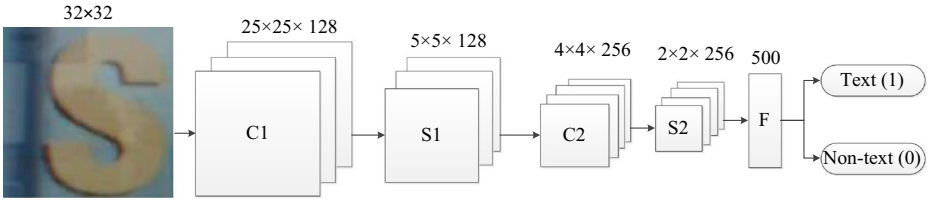

Fig. 3 The structure of the CNN used for generating candidate text regions. It contains two conventional layers and one full connection layer

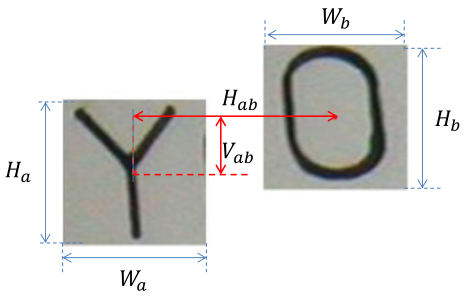

Fig. 4 Two nearby candidate text regions that can be grouped into a candidate text pair. The spatial and size constrains between them should be satisfied

Here, as shown in Fig. 4, H_{ab} is the horizontal distance of the centers of the two candidate text regions a and b, V_{ab} is the vertical distance and H_a, H_b, W_a, W_b are their heights and widths. T_h, T_v and T_s are set as 2, 0.5 and 0.5 in the proposed method.

Then, candidate text pairs are linked into candidate text lines. For two candidate text pairs, if they have the same end and the difference between the incline angels from one region to the other one in the two pairs is not larger than $\pi/12$, they will be linked into a line. Linking candidate text pairs into lines until no pairs can be merged, candidate text lines are generated, as shown in Fig. 5.

5 Hierarchical recursive candidate text lines analysis

When candidate text lines are obtained, they should be analyzed for detecting text in them. To accurately find out text lines, a hierarchical recursive analysis is applied to each candidate text line. For each of them, CC segmentation and CC analysis are done to distinguish text and non-text line. In CC analysis, a HRF model based analysis is employed which integrates information from several hierarchical layers including unary CCs, CC Pairs and CC lines to classify text and non-text and improves the classification accuracy. Meanwhile, each text line is recursively analyzed for robust detection. Moreover, some candidate text lines only overlap parts of the true text lines as shown in Fig. 6, recursive analysis can detect the complete text in these lines.

As shown in Fig. 7,the process of hierarchical recursive analysis for each candidate text line is as follows:

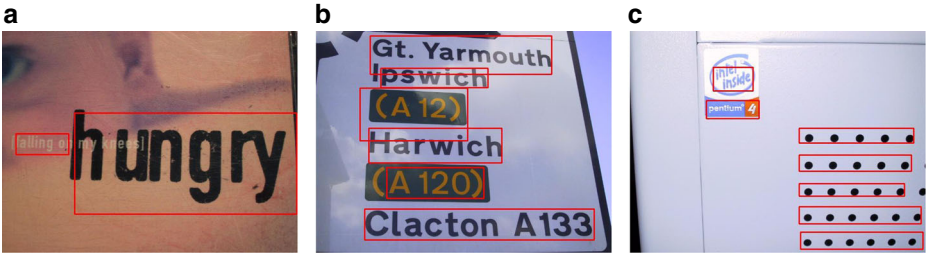

Fig. 5 Examples of candidate text lines generated. While text in these images are included in these lines, some non-text objects are also included

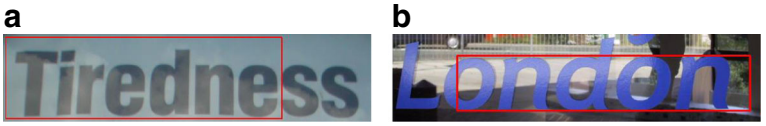

Fig. 6 Examples of candidate text lines which only overlap parts of the true text lines. **a** A candidate text line miss some characters of the text line. **b** A candidate text line overlaps part of the true character in the line

1) Multi-layer segmentation and GMM-based color clustering are combined to segment the candidate text line into CCs for initial analysis. Overlapped CCs selection are applied to these CCs generated. The selected CCs are candidate text CCs in the line.
2) HRF based analysis are employed to verify text. CCs classified as text are linked into lines to generate text lines. If no text line is detected in this line, the process for this line stops.
3) For the text line detected, a GMM is used to represent the color distributions in the foreground and background. With the pixels in the detected CCs are foreground and the other pixels are background, the parameters of the GMM are learned.
4) For the text line detected, its area is expanded by ex_h pixels and ex_v pixels in horizontal and vertical direction and a new text line with expanded bounding box is obtained.

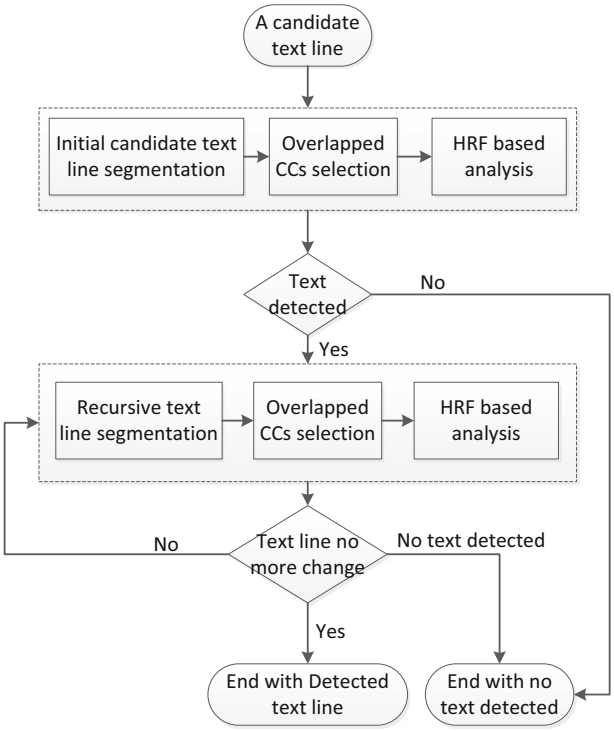

Fig. 7 The flowchart of hierarchical recursive analysis for each candidate text line. When no text detected in the recursive analysis, the process is end with no text detected. Otherwise, the hierarchical recursive analysis repeats until the detected text line no more changes, the process is end with a text line detected

5) Graph cut based segmentation and multi-layer segmentation are combined to segment CCs. In the graph cut based segmentation, the GMM learned based on the detected text line is used to compute the unary potentials. Overlapped CCs selection is also applied to these CCs generated to obtain candidate text CCs.
6) HRF based analysis is also employed to verify text. CCs classified as text are linked into lines to generate text lines. If no text line is detected in this line, the process for this line stops. If text line is detected and the overlap ratio of the current result to last analysis result is not smaller than $TH_{overlap}$, the detected text line is output as a text line. If text line is detected and the overlap ratio is smaller than $TH_{overlap}$, repeat from Step 3 until convergence.

Here the overlap ratio of the current result to last analysis result means the overlap ratio of pixels between the detected text line in consecutive two times recursive analysis. The overlap ratio can be computed as follows:

$$Ratio = \frac{L1 \cap L2}{L1 \cup L2}, \tag{5}$$

where $L1 \cap L2$ means the number of overlapped pixels of the two detected lines and $L1 \cup L2$ means the number of pixels in the two lines. Moreover, the threshold of the overlap ratio $TH_{overlap}$ is set to 0.9 in the proposed method.

When candidate text lines are obtained, hierarchical recursive analysis will be applied to each of them as shown in Fig. 8 and text lines satisfying convergence condition are detected as the final text detection result of the input image.

In initial candidate text line segmentation of hierarchical recursive analysis, multi-layer segmentation which integrates contrasts in RGB channels to segment CCs from an image, was proposed in our previous work [29]. Moreover, GMM-based color clustering is used to segment a candidate text line into three layers, which corresponds to text, background, and mixed pixels. For each candidate text line, two CC segmentation result maps are obtained using the two segmentation methods, as shown in Fig. 9. Therefore, some CCs are overlapped with some others. Among of these overlapped CCs, the ones with lower probabilities of being text should be removed. Previous text detection methods only use the probabilities computed based on the features of individual CCs to select overlapped CCs while context information between nearby CCs is also considered in the proposed method. Because text in natural scene images usually is not alone and nearby texts always have similar properties such as color, size, stroke width and so on, selection with both probability of each CC being text and context similarities between CCs taken into account is more accurate.

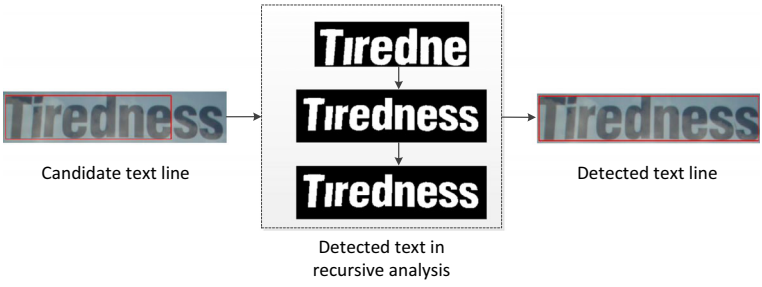

Fig. 8 The hierarchical recursive analysis for a candidate text line. Text line is detected until it no more changes

Fig. 9 Examples of initial candidate text line segmentation results. **a** The candidate text lines. **b** The segmentation results obtained using multi-layer segmentation. **c** The segmentation results obtained using GMM-based color clustering. **d** The selected CCs after overlapped CCs selection

The probability of each CC being text and the similarity between two nearby CCs can be computed through two support vector machine (SVM) classifiers [4], which can be found in our previous work [29]. Then, the probabilities of CCs being text can be modified by taking similarities between pairwise CCs into account, which can be computed as follows:

$$P(c) = (P(c \in text) + S(c))/2, \tag{6}$$

where $S(c) = \max(S(c, d)), d \in N_c$ is the highest similarity between a CC and its pairwise CCs. According to the modified probability $P(c)$ of each CC, CCs with higher values than others overlapped with them are selected as candidate texts as shown in Fig. 9 (d), which will be verified in HRF based analysis.

5.1 HRF based analysis

When candidate text CCs in a line are obtained, they are analyzed based on a Hierarchical Random Field (HRF) model [12]. In previous, Pan et al. [20] used Conditional Random Filed (CRF) to integrate information from unary CCs and CC pairs for CC analysis. However, text line information is not considered which is important and useful for text classification. Here, with HRF used CC analysis is formulated into a binary labeling problem of CCs and information from unary CCs (characters), CC pairs (words) and CC lines (text lines) is integrated for classification as shown in Fig. 10.

With HRF used in the proposed method, CC analysis is formulated into a binary labeling problem of CCs: text (labeled 1) and non-text (labeled 0). The energy of the HRF model with labeling $\mathbf{f}$ follows the form:

$$E(\mathbf{f}) = \sum_{c \in C} E_c(f_c) + \sum_{c,d \in C, d \in N_c} E^p_{cd}(f_c, f_d) + \sum_{l \in L} E_l(\mathbf{f}, f_l), \tag{7}$$

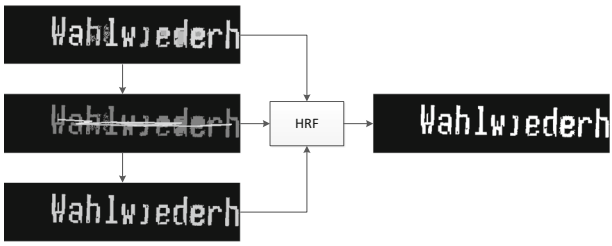

Fig. 10 The HRF based analysis of the candidate text CCs in a text line. Here the intensities of the images mean the probabilities of unary CCs and CC line being text and the similarities between pairwise CCs

where $E_c(f_c)$ reflects the probability of a CC c being text , $E^p_{cd}(f_c, f_d)$ reflects the similarity of two pairwise CCs c and d and $E_l(\mathbf{f}, f_l)$ reflects the probability of a CC line l being text.

In the energy function of HRF model, the potentials from unary CCs, CC pairs and CC lines are included and the definitions of these potentials are separately introduced in the following.

The unary potential $E_c(f_c)$ is constructed based on the probability of each CC being text. For each candidate text CC in a text line, the probability of it being text is computed using a SVM classifier and several CC features such as contour gradient, stroke width variance and so on. With the probability $P(c \in text) \in [0, 1]$ of a candidate text CC c is obtained, the unary potential from this CC is defined as:

$$E_c(f_c) = \begin{cases} sigmod(P(c \in text) - 0.5), & f_c = 0 \\ sigmod(0.5 - P(c \in text)), & f_c = 1, \end{cases} \tag{8}$$

where, $sigmod(x) = 1/(1 + exp(-x))$, $f_c = 0$ means the CC is labeled as non-text while $f_c = 1$ means it is labeled as text. The unary potential $E_c(f_c)$ is a penalty of unsuitable labeling and it reflects the probability of the CC being text. If the CC has a high probability of being text, the penalty is small when it is labeled as text, otherwise the penalty is large.

The pairwise potential $E^p_{cd}(f_c, f_d)$ is defined based on the similarity of two pairwise CCs. Candidate text CCs are grouped into pairs based on their distances and then the similarities between pairwise CCs are computed using a SVM classifier and several pairwise CC features such as color difference, stroke width difference and so on. With the similarity $S(c, d) \in [0, 1]$ between two CCs c and d is obtained, if $S(c, d) \geq 0.5$, the pairwise potential from the CC pair is defined as:

$$E^P_{cd}(f_c, f_d) = K * S(c, d) * \Delta(f_c \neq f_d). \tag{9}$$

Here $K = 2$ is the weight of the pairwise potential, $\Delta(\cdot)$ is 1 if the condition is true and 0 otherwise. The pairwise potentials mean that the CCs with high similarities should be labeled same, otherwise large penalty will be generated.

The higher order potentials $E_l(\mathbf{f}, f_l)$ are from CC lines. Therefore, CC lines should be generated first. CC lines can be obtained by linking candidate text pairs into lines. In the CC pairs obtained above, the pairs with $S(c, d) \geq 0.5$ is candidate text pairs. Then, for two candidate text pairs, if they have the same CC and the difference between the incline angels from one CC to the other one in the two pairs is not larger than $\pi/12$, they can be linked into a line.

Because CCs in the same line are more likely to take the same label, the higher order potentials $E_l(\mathbf{f}, f_l)$ in the HRF model take the form of a Robust P^N model [11] defined as:

$$E_l(\mathbf{f}, f_l) = \min\{\gamma_{max}, \gamma_i + \sum_{c \in l} \omega_c k^i \Delta(f_c \neq i)\}, \tag{10}$$

where ω_c is the weight of the CC c in the line l, $\Delta(\cdot)$ is 1 if the condition is true and 0 otherwise, and the variables satisfy $\gamma_i \leq \gamma_{max}, \forall i \in \{0, 1\}$. By adding a free label L_F to represent the situation when there is no dominant label in the line and its cost is γ_{max}, the potential $E_l(\mathbf{f}, f_l)$ can be written as:

$$E_l(\mathbf{f}, f_l) = E_l(f_l) + \sum_{c \in l} E_l(f_c, f_l), \tag{11}$$

where $E_l(f_l)$ associates the cost γ_i with f_l taking a label i, and γ_{max} with the free label L_F. Moreover, $E_l(f_c, f_l)$ is defined as:

$$E_l(f_c, f_l) = \begin{cases} 0, \ if \ f_l = L_F \ or \ f_c = f_l \\ \omega_c k^{f_l}, \ otherwise. \end{cases} \tag{12}$$

For a CC line l, the higher order potential $E_l(f_l)$ is defined based on the probability of it being text. When CC lines are generated by linking CCs, the probability of each CC line being text can be computed. By extracting the features of a CC line and inputting them into a SVM classifier, the probability of the CC line being text is obtained. Texts in the same text line always have similar properties such as size, stroke width and so on and the distances between consecutive texts are uniform. Therefore, the features including height variance, stroke width variance and distance variance are extracted to represent text. Meanwhile, considering texts have distinct properties such as aspect ratio, compactness and so on with the background, the features including mean probability, mean aspect ratio, mean compactness, mean occupy ratio are also employed here. The details about these text line features can be found in our previous work [29]. Moreover, the THOG feature [15] is added because text lines have distinct textures with the background. Then, with its probability $P(l \in text) \in [0, 1]$ computed, the higher order potential $E_l(f_l)$ from it can be defined as:

$$E_l(f_l) = \begin{cases} num(l) * sigmod(P(l \in text) - 0.5), & f_l = 0 \\ num(l) * sigmod(0.5 - P(l \in text)), & f_l = 1 \\ num(l), & f_l = L_F, \end{cases} \tag{13}$$

where $sigmod(x)$ is defined above and $num(l)$ is the number of CCs in the line and the weight of the potential. The potential means a CC line should be labeled as text if it has high probability of being text. Meanwhile, CCs in the same line are more probable to have the same labels, otherwise large penalty will be generated.

A CC line consists of several CCs while the line and CCs in it both have labels. If their labels are different, it will not make sense. Therefore, the label of the CC line and the labels of the CCs in it should keep consistent, which is reflected by the potential $E_l(f_c, f_l)$. Because the higher order potentials in the HRF model take the form of Robust P^N model, the variables in (12) can be defined as: $\omega_c = 1$ which means a CC and $k^{f_l} = 2 * (E_l(L_F) - E_l(f_l))/num(l)$ which means the penalty when the label of the CC and the label of the CC line are different. Now the connective potential $E_l(f_c, f_l)$ between the CC line and a CC in it can be written as:

$$E_l(f_c, f_) = \begin{cases} 0, \quad if \ f_l = L_F \ or \ f_c = f_l \\ 2(E_l(L_F) - E_l(f_l))/num(l), otherwise. \end{cases} \tag{14}$$

When the potentials of the energy function are defined, it can be minimized by the range move algorithms [12]. Then, the labels of the CCs in the line are obtained and CCs labeled 1 are text while the ones labeled 0 are non-text, as shown in Fig. 10. By linking text CCs into lines, text lines are obtained which will be recursively analyzed.

5.2 Recursive text line segmentation and analysis

Text lines are obtained after HRF based analysis. However, some of them may be non-text lines while some may only overlap part of the true text lines. To solve this problem, these text lines should be recursively analyzed while their areas should be dynamically changed. Therefore, recursive text line segmentation should be applied to them to segment CCs for analysis.

For each text line, its area should be expanded first in each time recursive analysis. Therefore, a new bounding box for each text line is obtained by expanding ex_h pixels and ex_v pixels in horizontal and vertical direction. Here, ex_h is set to the half of the width of the text line and ex_v is set to a third of its height.

For each text line with a new bounding box, a graph cut based method is recursively employed for text line segmentation. With the observation that texts in the same line have similar colors, a text line can be segmented into CCs according to the colors in its current analysis result. With graph cut used, text line segmentation is formulated into a binary labeling problem: 1 for foreground (text) and 0 for background. The energy function of the graph cut based framework [2] is as follows:

$$E(\mathbf{f}) = \sum_{i \in L} E_D(f_i) + \sum_{i,j \in L, j \in N_i} E_S(f_i, f_j), \tag{15}$$

where $E_D(f_i)$ is the unary potential which means the probability of each pixel belonging to foreground according to the color distributions for foreground and background, $E_S(f_i, f_j)$ is the pairwise potential which make sure neighbour pixels with similar colors labeled same, $\mathbf{f}$ is the vector of the labels for each pixel in the text line. The unary potential $E_D(f_i)$ for each pixel is computed based on a GMM whose parameters are learned with the pixels in the CCs detected as text are foreground and the other pixels in the current analysis result are background. The pairwise potential $E_S(f_i, f_j)$ is computed based on the RGB color distance of the two neighbour pixels. Then, the labels of the text line with a new bounding box can be obtained by minimizing the energy function and the text line is segmented with the CCs in the foreground are candidate text.

However, text in text lines may not be separated from the background only use graph cut based segmentation. Therefore, multi-layer segmentation introduced above is also recursively employed to segment text lines with areas expanded.

Based on the detection result of last time, the text line is expanded and then segmented into CCs with graph cut based segmentation and multi-layer segmentation as shown in Fig. 11. Then, overlapped CCs selection and HRF based analysis are applied to the text line in the following. Based on the new analysis result, the text line may be analyzed again for text detection.

6 Experiments

The proposed method is compared with several state-of-art methods on three public datasets. Moreover, the experiments on the components (text edge box, CNN based text detector

Fig. 11 The recursive segmentation for a text line. **a** Its area is expanded based on the last time detection result. **b** The segmentation result using graph cut based segmentation. **c** The segmentation result using multi-layer segmentation

and recursive analysis) of the proposed method are also presented. The proposed method is implemented with Matlab 2015b and runs on a computer with 3.5GHz 4-core CPU, 32G RAM and Nvidia Quadro K1200 for all the experiments.

6.1 Datasets

ICDAR datasets are the benchmarks for text detection in natural scene images, including ICDAR 2003 dataset, ICDAR 2011 dataset and ICDAR 2013 dataset, which are from the ICDAR robust reading competitions held in different years. ICDAR 2003 dataset [13] contains 509 fully annotated images with 258 images for training and 251 for testing. The images are captured from natural scenes and contain texts in a variety of colors, sizes and fonts with complex backgrounds and various orientations. ICDAR 2011 dataset [22] is extended from ICDAR 2003 dataset including about 100 new captured images with digital camera using auto focus and natural lighting while the others are chosen from ICDAR 2003 dataset. Totally, ICDAR 2011 dataset consists of 484 images including 255 test images and 229 train images. With a small number of duplicated images excluded from ICDAR 2011 dataset, ICDAR 2013 dataset [10] is obtained, which contains 462 fully annotated images with 229 images for training and 233 images for testing. Because ICDAR 2011 dataset and ICDAR 2013 dataset are almost the same, the ICDAR 2011 dataset is not used here. Therefore, the proposed method is evaluated in two ICDAR datasets including ICDAR 2003 dataset and ICDAR 2013 dataset.

Besides, the proposed method is evaluated on a more challenging Street View Dataset (SVT) [8]. The dataset contains 307 images, which are much harder than ICDAR images, due to the presence of vegetation, repeating patterns, such as windows, bricks and so on, which are virtually undistinguishable from text without OCR.

6.2 Performance evaluation

Precision, recall and f-measure are the criteria for performance evaluation in text detection. Among them, precision is the number of correct estimates divided by the total number of estimates, recall is the number of correct estimates divided by the total number of targets

and f-measure is adopt to combine precision and recall into a single measure of quality. However, different performance evaluation methods are adopted for the datasets above.

For ICDAR 2003 dataset and Street View Dataset, an object detection evaluation method [14] is adopted, in which precision and recall are defined by finding the best matches between the detected and the ground truth boxes and only one-to-one matches are considered.

For ICDAR 2013 dataset, the evaluation method proposed by Wolf and Jolion [25] is used, taking into account the quality of each match between detected and ground truth boxes. Matches are determined based on area overlapping, given certain minimum quality thresholds. Not only one-to-one matches but also one-to-many and many-to-one matches are also considered here. Meanwhile, "don't care" regions are needed to be handled, inside which any detected regions that fall are removed from the results list. Therefore, no false alarms are reported if a detection method marks a don't care region as text.

6.3 Experiments on ICDAR datasets

In the proposed method the CNN text detector used for generating candidate text regions is trained using 200K samples from ICDAR 2013 dataset and Char74k English dataset [6]. The training samples include 100K positive samples and 100K negative samples, which are all resized to $32 * 32$ pixels. Meanwhile, the SVM classifiers used in hierarchical recursive candidate text lines analysis are trained on the training images of the ICDAR 2013 dataset. Because the images in ICDAR 2003 dataset and ICDAR 2013 dataset are similar, these SVM classifiers are also used for text detection in ICDAR 2003 dataset.

To evaluate the performance of the proposed method, it is tested on ICDAR 2003 dataset and ICDAR 2013 dataset. Meanwhile, its performance is compared with several state-of-art methods. Moreover, the results of our previous work and the winning methods in ICDAR 2013 robust reading competition such as Text Spotter and CASIA NLPR are also included for performance comparison. The results of some early proposed methods [5, 37] are not presented while some methods which are proposed for text recognition [9, 26] can't be used for performance comparison here.

As shown in Table 1, the proposed method achieves the highest recall (0.74) and high precision (0.79) on ICDAR 2003 dataset. Compared with the other methods in the table, the recall is obviously higher while the second higher recall is 0.71. Meanwhile, the precision is only lower than the methods of Opitz et al. [19] and Yuan et al. [35] while their recalls are much lower than the proposed method. Therefore, the proposed method outperforms the other methods on ICDAR 2003 dataset.

Table 1 Performance comparison on ICDAR 2003 dataset

Method	Precision	Recall	F-measure
The proposed method	0.79	0.74	0.76
Opitz et al. [19]	0.82	0.68	0.74
Our previous work [29]	0.76	0.71	0.73
Wang et al. [28]	0.71	0.70	0.70
Yuan et al. [35]	0.84	0.60	0.70
Pan et al. [20]	0.67	0.70	0.69
Yao et al. [31]	0.69	0.66	0.67
Epshtein et al. [8]	0.73	0.60	0.66

As shown in Table 2, the proposed method also achieves the highest recall (0.75) on ICDAR 2013 dataset. Meanwhile, its precision (0.82) is not low for text detection. Though the precision is common compared to the other methods, the recall of the proposed method is higher. The proposed method also achieves better performance on ICDAR 2013 dataset compared to the other methods.

From the performances of the proposed method on ICDAR 2003 dataset and ICDAR 2013 dataset, it can be found that the proposed method outperforms several state-of-art methods in natural scene text detection. It can detect various text from complex backgrounds. Compared with the method of Neumann and Matas [18], the proposed method has same precision but higher recall, which means more text can be detected with the proposed method. In the method of Neumann and Matas text lines are recursively refined with Grabcut while in the proposed method hierarchical recursive analysis is applied to each text line. That means hierarchical recursive analysis can help detecting more text.

Figure 12 shows some samples of the text detection results with the proposed method, which succeeds in most natural scene images with various text and complex background. However, it fails in detecting text with low contrast, strong light or occlusions caused by other objects such as fences. These situations should be considered in the future.

6.4 Experiments on Street View Dataset

The proposed method is also evaluated on Street View Dataset, which is much more challenging than ICDAR datasets. Its performance is compared with several state-of-art methods and our previous work on this dataset. The CNN text detector trained based on ICDAR 2013 dataset and Char74k English dataset is also employed here while the SVM classifiers used in hierarchical recursive candidate text lines analysis are retrained on the street view text dataset proposed by Wang et al. [24], which has similar images as the Street View Dataset. This is because the images in the Street View Dataset are different from the images in ICDAR datasets and using the SVM classifier trained on ICDAR images for text detection in Street View Dataset is not suitable.

As shown in Table 3, the proposed method achieves the highest recall (0.55) and second highest precision (0.61) on the Street View Dataset. Compared to the method of Yin et al. [33], the precision is lower but the recall is much higher and much better performance is achieved by the proposed method. Compared to other methods, the proposed method has both higher recall and precision, which means the proposed method outperforms them. The proposed method achieves better performance than the other methods on the Street

Table 2 Performance comparison on ICDAR 2013 dataset

Method	Precision	Recall	F-measure
The proposed method	0.82	0.75	0.78
Neumann and Matas [18]	0.82	0.72	0.77
Yin et al. [33]	0.88	0.66	0.76
Our previous work [29]	0.82	0.68	0.74
Text Spotter [10]	0.88	0.65	0.74
Yu et al. [34]	0.84	0.65	0.73
CASIA NLPR [10]	0.79	0.68	0.73

Fig. 12 The samples of text detection results on ICDAR datasets using the proposed method. **a** are successful samples while (**b**) are failed ones

View Dataset, which is much harder than ICDAR datasets. The means that the proposed method can robust detect various text from complex background such as street view, as shown in Fig. 13a. With the hierarchical recursive analysis used for text detection, more text in complex scenes can be detected.

Though the proposed method has the best performance on Street View Dataset, it is still bad compared to the performances on ICDAR datasets. This is because the images in the Street View Dataset have many very small and blur text as shown in Fig. 13b. Nearby small characters are probably segmented together into one CC while blur text is difficult to be separated from the background. Therefore, the proposed method fails in detecting such text. These are open problems in text detection and much attention should be paid to them in the future.

Table 3 Performance comparison on Street View Dataset

Method	Precision	Recall	F-measure
The proposed method	0.61	0.55	0.58
Our previous work [29]	0.59	0.48	0.53
Yin et al. [33]	0.66	0.41	0.51
Phan et al. [21]	0.50	0.51	0.51
Epshtein et al. [8]	0.54	0.42	0.47
Yu et al. [34]	0.27	0.35	0.31

Fig. 13 The samples of text detection results on the Street View Dataset using the proposed method. **a** are successful samples while (**b**) are failed ones

6.5 Experiments on text edge box

In the proposed method text edge box is proposed for text proposals. To show that text edge box is more suitable for text proposals than edge box, the two methods are tested on the three datasets as shown in Table 4. Here the recall is the ratio of the detected pixels to the ground-truth pixels in a bounding box. For the two text proposal methods, the score threshold is set to 0.05 and other parameters are default values. The results show that the proposal numbers using text edge box are much smaller than using edge box while the recalls are almost the same. That means using text edge box many non-text proposals are removed and the following computation is much reduced.

6.6 Experiments on CNN-based text detector

In the proposed method the CNN-based text detector is trained on 200K samples from ICDAR 2013 dataset and Char74k English dataset. Here the CNN model is implemented

Table 4 Performance comparison of text edge box and edge box on three public datasets

Method	Datasets	Recall	Proposal number
Text edge box	ICDAR 2003	0.98	81950
	ICDAR 2013	0.98	59929
	SVT	0.89	106601
Edge box	ICDAR 2003	0.98	248572
	ICDAR 2013	0.98	222659
	SVT	0.90	438772

Table 5 The training results of CNN-based text detector

Training sample number	Error	Runtime(s)
50000	0.027	10964
100000	0.025	20700
200000	0.020	43192

using MatConvNet [23]. Moreover, the training results of the CNN model with different training samples are shown in Table 5 with learning rate set as 0.001 and epoch number set as 300. The runtime of training CNN modes is computed with GPU used. It shows that the performance is better with more training samples. In the proposed method the performance of the trained CNN model is not good enough for distinguish text and non-text CCs and it is only used for generating candidate text regions. However, we will collect more training samples and design a CNN model with more layers to improve performance and consider use the CNN model to distinguish text and non-text CCs in the future.

6.7 Experiments on recursive analysis

In the proposed method, the key idea is the hierarchical recursive analysis to text lines. In our previous work [29] experiments have shown that the hierarchical analysis with information from unary CCs, CC pairs and CC lines integrated for text verification is helpful for improving text detection performance. Therefore, only experiments on recursive analysis are presented here to show whether it works or not. To achieve this target, the proposed method without recursive analysis is evaluated on the three public datasets.

As shown in Table 6, the performance comparison of the proposed method with and without recursive analysis shows that the recall and precision is much lower without recursive analysis, which is useful for improving text detection performance. Moreover, some candidate text lines generated in the proposed method only overlap parts of the true text lines. Some detected text lines are not complete while some non-text lines are detected as text without recursive analysis, as shown in Fig. 14. Therefore, recursive analysis is essential in the proposed method. Otherwise, the recall and precision are much lower. Meanwhile, recursive analysis is a little time consuming and the runtime is about 1 to 2 minutes for one input image with the proposed method implemented in Matlab. However, i it is implemented and modified in C++, the runtime will be much reduced.

Table 6 Performance comparison of recursive analysis on three public datasets

Method	Dataset	Precision	Recall	F-measure
With recursive analysis	ICDAR 2003	0.79	0.74	0.76
	ICDAR 2013	0.82	0.75	0.78
	SVT	0.61	0.55	0.58
Without recursive analysis	ICDAR 2003	0.51	0.64	0.53
	ICDAR 2013	0.52	0.68	0.59
	SVT	0.28	0.48	0.35

Fig. 14 The samples of text detection results without recursive analysis. Compared with the results with recursive analysis, text lines detected are not complete while some non-text lines exist

7 Conclusions

To robust detect various texts in complex scenes, a hierarchical recursive method for text detection in natural scene images is proposed in this paper. In the proposed method a modified edge box called text edge box is used to obtain text proposals, from which candidate text regions are generated by using a CNN text detector. Linking candidate text regions into lines, candidate text lines are generated. For each candidate text line, hierarchical recursive analysis is applied to them, in which CC segmentation and HRF based analysis are recursively employed for detecting text. When the detected texts in each line no more changes, they are output as the final text detection result. The proposed method is evaluated on three benchmarks and achieves better performances than several state-of-art methods. Experiments show that hierarchical recursive architecture is helpful for improving text detection performance.

However, text detection in natural scene images is still challenging. For improving text detection performance in the future, the features of several objects in different levels such as characters, words and text lines can be auto learned from deep learning methods. Meanwhile, the situations of blur or low contrast text should be considered too.

Acknowledgments This work is supported by the National Natural Science Foundation of China (91520301).

References

1. Bengio Y (2009) Learning Deep Architectures for AI. Foundations and Trends in Machine Learning 2(1):1–27
2. Boykov Y, Kolmogorov V (2004) An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. IEEE Trans Pattern Anal Mach Intell 26(9):1124–1137
3. Cabrera CR, Sastre RJ, Rodriguez JA, Bascon SM (2012) Surfing the point clouds: Selective 3D spatial pyramids for category-level object recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp 3458–3465
4. Chang CC, Lin CJ (2011) LIBSVM: A Library for Support Vector Machines. ACM Trans Intell Syst Technol 2(3):27:1–27:27
5. Chen XR, Yuille AL (2004) Detecting and Reading Text in Natural Scenes. Proc. IEEE Conf. on Computer Vison and Pattern Recognition, pp II366-II373
6. De Campos TE, Babu BR, Varma M (2009) Character recognition in natural images. In: Proceedings of the 4th International Conference on Computer Vision Theory and Applications, pp 273–280

7. Dollr P, Zitnick CL (2015) Fast edge detection using structured forests. IEEE Trans Pattern Anal Mach Intell 37(8):1558–1570
8. Epshtein B, Ofek E, Wexler Y (2010) Detecting Text in Natural Scenes with Stroke Width Transform. In: Proceedings IEEE Conf. on Computer Vison and Pattern Recognition, pp 2963–2970
9. Jaderberg M, Simonyan K, Vedaldi A, Zisserman A (2016) Reading Text in the Wild with Convolutional Neural Networks. Int J Comput Vis 116(1):1–20
10. Karatzas D, Shafaity F, Uchidaz S, Iwamurax M, Bigorda LG, Mestre SR, Mas J, Mota DF, Almazan JA, De LasHerasJP (2013) ICDAR 2013 Robust Reading Competition. In: Proceedings of the twelfth International Conference on Document Analysis and Recognition, pp 1484–1493
11. Kohli P, Ladicky L, Torr PH (2009) Robust Higher Order Potentials for Enforcing Label Consistency. Int J Comput Vis 82:302–324
12. Ladicky L, Russell C, Kohli P, Torr PH (2014) Associative hierarchical random fields. IEEE Trans Pattern Anal Mach Intell 36(6):1056–1077
13. Lucas SM, Panaretos A, Sosa L, Tang A, Wong S, Young R (2003) ICDAR 2003 robust reading competitions. In: Proceedings of the Seventh International Conference on Document Analysis and Recognition, pp 682–687
14. Mariano VY, Min J, Park JH, Kasturi R, Mihalcik D, Li HP, Doermann D, Drayer T (2002) Performance evaluation of object detection algorithms. In: Proceedings of the 16th International Conference on Pattern Recognition, pp 965–969
15. Minetto R, Thome N, Cord M, Leite NJ, Stolfi J (2013) T-HOG: An effective gradient-based descriptor for single line text regions. Pattern Recogn 46(3):1078–1090
16. Neumann L, Matas J (2011) Text Localization in Real-world Images using Efficiently Pruned Exhaustive Search. In: Proceedings of the 11th International Conference on Document Analysis and Recognition, pp 687–691
17. Neumann L, Matas J (2012) Real-Time Scene Text Localization and Recognition. In: Proceedings IEEE Conf. on Computer Vison and Pattern Recognition, pp 3538–3545
18. Neumann L, Matas J (2015) Efficient Scene Text Localization and Recognition with Local Character Refinement. IEEE Conf. on Computer Vison and Pattern Recognition
19. Opitz M, Diem M, Fiel S, Kleber F, Sablatnig R (2014) End-to-End Text Recognition using Local Ternary Patterns, MSER and Deep Convolutional Nets. In: Proceedings of the 11th IAPR International Workshop on Document Analysis Systems, pp 186–190
20. Pan YF, Hou XW, Liu CL (2011) A Hybrid Approach to Detect and Localize Texts in Natural Scene Images. IEEE Trans Image Process 20(3):800–813
21. Phan TQ, Shivakumara P, Tan CL (2012) Detecting text in the real world. In: Proceedings of the 20th ACM international conference on Multimedia, pp 765–768
22. Shahab A, Shafait F (2011) Dengel A (2011) ICDAR Robust Reading Competition Challenge 2: Reading Text in Scene Images. In: Proceedings of the eleventh International Conference on Document Analysis and Recognition, pp 1491–1496
23. Vedaldi A, Lenc K (2015) MatConvNet: Convolutional neural networks for MATLAB. In: Proceedings of the 2015 ACM Multimedia Conferenc, pp 689–692
24. Wang K, Babenko B, Belongie S (2011) End-to-end scene text recognition. In: Proceedings of the 13th IEEE International Conference on Computer Vision, pp 1457–1464
25. Wolf C, Jolion JM (2006) Object count/area graphs for the evaluation of object detection and segmentation algorithms. International Journal of Document Analysis 8(4):280–296
26. Wang T, Wu DJ, Coates A, Ng AY (2012) End-to-End Text Recognition with Convolutional Neural Networks. In: Proceedings of the 21st International Conference on Pattern Recognition, pp 3304–3308
27. Wang S, Yang Y, Ma ZG, Li X, Pang CY, Hauptmann AG (2012) Action recognition by exploring data distribution and feature correlation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp 1370–1377
28. Wang XB, Song YH, Zhang YL (2013) Natural Scene Text Detection with Multi-channel Connected Component Segmentation. In: Proceedings of the twelfth International Conference on Document Analysis and Recognition, pp 1375–1379
29. Wang XB, Song YH, Yuan LZ, Xin JM (2015) Natural scene text detection with multi-layer segmentation and higher order conditional random field based analysis. Pattern Recogn Lett 60-61:41–47
30. Yang Y, Ma ZG, Xu ZW, Yan SC, Hauptmann AG (2013) How Related Exemplars Help Complex Event Detection in Web Videos? In: Proceedings of IEEE International Conference on Computer Vision, pp 2104–2111
31. Yao C, Bai X, Liu WY, Ma Y, Tu ZW (2012) Detecting Texts of Arbitrary Orientations in Natural Images. In: Proceedings IEEE Conf. on Computer Vison and Pattern Recognition, pp 1083–1090

32. Ye QX, Gao W, Wang WQ, Zeng W (2003) A robust text detection algorithm in images and video frames. In: Proceedings of the 2003 Joint Conference of the 4th International Conference on Information, Communications and Signal Processing and 4th Pacific-Rim Conference on Multimedia, vol 2, pp 802–806
33. Yin XC, Yin YW, Huang KZ, Hao HW (2014) Robust Text Detection in Natural Scene Images. IEEE Trans Pattern Anal Mach Intell 36(5):970–983
34. Yu C, Song YH, Zhang YL, Liu Y (2016) Scene text localization using edge analysis and feature pool. Neurocomputing 175:625–661
35. Yuan J, Wei BG, Liu YH, Zhang Y, Wang LD (2015) A method for text line detection in natural images. Multimedia Tools and Applications 74(3):859–884
36. Zitnick CL, Dollar P (2014) Edge Boxes: Locating Object Proposals from Edges. In: Proceedings of the 13th European Conference on Computer Vision part V Lecture Notes in Computer Science, pp 391–405
37. Zhu KH, Qi FH, Jiang RJ, Xu L (2007) Automatic character detection and segmentation in natural scene images. J Zheijang Univ Sci A 8(1):63–71

Xiaobing Wang received the B.E. degree in information and communication engineering from Xi'an Jiaotong University, Xi'an, China, in 2009. He is currently working toward the Ph.D. degree with the Department of Control Science and Engineering, Xi'an Jiaotong University. His current research interests include pattern recognition and image processing.

Yonghong Song received Bachelor degree in Computer Science in 1989 in Xi'an Jiaotong University. She is a senior engineer at School of Electronic and Information Engineering in Xi'an Jiaotong University. Her research interests include document image processing, image video retrieval, visual target tracking and human-computer interaction.

Yuanlin Zhang received the B.E. degree In Electronics Engineering from Dalian University of Technology, China, in 1989 and the M.E. degree in signal and information processing from National University of Defense Technology, China, in 1997. His research interests include image processing and computer vision.

Jingmin Xin received the B.E. degree from Xi'an Jiaotong University, in 1988 and the M.S. and Ph.D. degrees from Keio University, Japan, in 1993 and 1996, respectively. His research interests are in the areas of adaptive filtering, statistical and array signal processing, system identification, and pattern recognition.