Research Article

Fengying Peng, Runmin Wang*, Yiyun Hu, Guangyi Yang, and Ying Zhou

Feature fusion-based text information mining method for natural scenes

https://doi.org/10.1515/dema-2022-0255
received February 11, 2023; accepted June 9, 2023

Abstract: As a crucial medium of information dissemination, text holds a pivotal role in a multitude of applications. However, text detection in complex and unstructured environments presents significant challenges, such as the presence of cluttered backgrounds, variations in appearance, and uneven lighting conditions. To address this issue, this study proposes a text detection framework that leverages multistage edge detection and contextual information. This framework deviates from traditional approaches by incorporating four primary processing steps, including text visual saliency region detection to accentuate the text regions and diminish background interference, multistage edge detection to enhance the conventional stroke width transform results, a texture-based and connected components-based integration to accurately distinguish text from the background, and a context fusion step to recover missing text regions and improve the recall of text detection. The proposed method was evaluated on two widely used benchmark datasets, i.e., the international conference on document analysis and recognition (ICDAR) 2005 dataset and the ICDAR 2011 dataset, and the results indicate the advancedness of the method.

Keywords: scene text detection, multistage edge detection, hierarchical identification strategy, context information

MSC 2020: 68Txx

1 Introduction

The widespread use of digital image capture devices has resulted in an increased demand for image retrieval and understanding. As a result, text detection has become a critical task in content-based image analysis techniques. Unlike in document images, where text characters are typically organized in neat arrangements with proper resolutions, detecting text in natural scenes is a challenging task due to the variations in font, size, color, and alignment orientation, as well as the impact of complex backgrounds, illumination changes, image distortion, and degradation. Given the crucial role of text detection in natural scenes in various applications, substantial attention has been devoted to this problem in recent years. Numerous detection methods have been proposed and can be roughly divided into two categories: texture-based methods and connected components (CC)-based methods.

*** Corresponding author: Runmin Wang,** School of Information Science and Engineering, Hunan Normal University, Changsha, 410081, China; School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, 430074, China, e-mail: runminwang@hunnu.edu.cn
Fengying Peng: School of Foreign Languages, Changsha Normal University, Changsha, 410100, China, e-mail: hunanpfy0731@163.com
Yiyun Hu: Department of Mathematics, University of WAshington, Seattle, WA, 98105, United States, e-mail: yyh155@uw.edu
Guangyi Yang: Information Center, Hunan Institute of Metrology and Test, Changsha, 410014, China, e-mail: guangyiyang_himt@163.com
Ying Zhou: Data Information Management Center, Hunan Children's Hospital, Changsha, 410007, China, e-mail: yingzhou_hch@163.com

In comparison to the background, the text region exhibits distinctive textural characteristics. With machine learning techniques having gained widespread application [1–3], the effective extraction of these features and the selection of appropriate classifiers are considered to be the two crucial components in texture-based methods. Some hand-engineered features have been designed to describe text candidates, e.g., T-HOG [4], eHOG [5], HOM [6], and discrete wavelet transform features [7]. In recent years, Liao et al. [8], Liu et al. [9], and Cai et al. [10] have adopted learning methods to obtain the features for avoiding randomness of the hand-engineered features. The feature vectors extracted from each text candidate region are fed into a trained classifier; the text likelihood of the candidate regions is estimated; and the text candidates are distinguished by the aforementioned trained classifiers. For example, Wei and Lin [7] and Ye et al. [11] adopted the support vector machine (SVM) in their work, Hanif and Prevost [12] trained the AdaBoost classifier, Xu and Su [13] adopted the Random forest classifier, Wang et al. [14] and Sun et al. [15] used the Neural Networks, etc. In the second stage of the process, texture-based methods prove to be effective in addressing complex backgrounds with dissimilar texture structures in relation to the text regions. However, these approaches consistently utilize a trained classifier that performs a comprehensive scan of the images through the use of multi-scale sliding windows, resulting in a high demand for thousands of predictions. In addition, the training of the classifier requires a substantial number of training samples, which can be both time-consuming and resource-intensive. As a result, the computational complexity inherent in these methods limits their practical application in large databases.

The implementations of CC-based methods are founded on the premise that characters in text regions exhibit specific attributes, such as approximate constant color, proximate pixel values, and similar stroke widths, among others. These methods segment an image into a collection of CCs, and the ultimate CCs are classified as text or background based on the analysis of their geometric properties. CC-based methods typically comprise two distinct phases, namely, the detection of CCs and the verification of CCs. So far, various approaches have been used to extract the CCs, e.g., Shi et al. [16] and Yin et al. [17] have adopted the maximally stable extremal regions, Mancas-Thillou and Gosselin [18] have used the color clustering method, Shivakumara et al. [19] have performed K-means clustering in the Fourier-Laplacian domain, Sun et al. [20] have designed the color-enhanced contrasting extremal regions, Epshtein et al. [21] and Xu et al. [22] have adopted the Stroke Width Transform (SWT) processing to obtain the CCs. CC-based methods are recognized for their relative speed; however, their efficacy in accurately removing text components without prior knowledge of text position and scale is limited. Furthermore, the design of a reliable CC analyzer is a challenging task due to the presence of numerous non-text components and the difficulties that arise when text is noisy, multicolored, and textured.

The central objective of text detection, as a typical pattern recognition problem, is to differentiate texts from their backgrounds. The identification of suitable candidates and their accurate classification are crucial considerations in text detection. Detection accuracy and execution efficiency are the two paramount metrics used to evaluate the efficacy of a text detection method. It is widely acknowledged that the extraction of text candidates through the use of multi-scale sliding windows is computationally intensive, and relying solely on geometrical characteristics for candidate classification can lead to inaccurate results. As a result, the integration of texture-based and CC-based methods may offer a promising solution by capitalizing on their respective strengths.

Because the characters are regions of similar stroke width, the SWT processing proposed by Epshtein et al. [21] has been widely used to obtain the CCs. Epshtein et al. [21] carried out the SWT processing based on the Canny edge map, and the false positives existing in the Canny edge maps would undoubtedly weaken the processing results according to the principle of SWT. Based on the aforementioned discussion, how to obtain an accurate edge map needs to be solved.

The false positives in the results will decrease the detection precision, and suppressing the background region will help reduce the false alarm effectively. In fact, text is a carrier for interpersonal communication in human society, and they are typically designed in natural scenes to attract attention. Judd et al. [23] showed that scene texts can receive many eye fixations, and Wang et al. [14] have also performed psychophysical experiments to confirm the text regions in the images to be salient, the aforementioned studies evaluate existing visual attention models for text detection. Some methods [5,24,25] use the salient regions to detect the texts by leveraging intensity, color, orientation, size, and curvature saliency cues. Meanwhile, some work [26,27] proposed a weakly supervised approach for scene text detection, and promising detection performance

has been reported in their work. Inspired by their work, a novel method exploiting features of color and luminance is adopted to detect the texts' visual saliency region in our work.

Traditional methods of character analysis tend to result in incorrect classifications due to their individualized approach. In reality, texts in natural scenes often exhibit a distinct layout structure, frequently appearing in clusters and lines. The proximity of candidates to text regions often leads to their misclassification as texts. The specific spatial distribution of texts provides a compelling reason to incorporate text context information. Thus, in our work, the utilization of context information is employed to improve classification robustness. Specifically, key regions are formed by the detection of adjacent text regions, followed by the retrieval of missing text regions surrounding the key regions through the implementation of constraint rules.

Drawing upon the previously discussed considerations, this work presents a novel text detection approach for natural scenes that utilizes context information and multistage edge detection. The primary contributions of this study are as follows:

(1) The text regions are emphasized through the use of visual saliency detection to increase visibility.
(2) A multistage edge detection methodology is established to acquire a precise edge map.
(3) The integration of texture-based and CC-based methods is utilized to assess text candidates for optimal results.
(4) A hierarchical identification approach is proposed to minimize texture distortion and improve the robustness of the method.

2 Our methodology

Based on the fact that the texture-based and CC-based methods are complementary, the two methods are reasonably integrated into our work, and an overview of the proposed method is shown in Figure 1.

2.1 Pre-processing

Texts are typically designed in natural scenes to attract attention, and some research [14,23] evaluated existing visual attention models for text detection to show that scene texts are salient. The saliency cues are adopted in our approach to prevent an exhaustive spatial and scale search over all possible regions. The augmented precision of the edge map consistently exerts a profound impact on the SWT outcome, as the SWT operator exhibits a linear correlation with the quantity of edge pixels. A multistage edge detection method is proposed

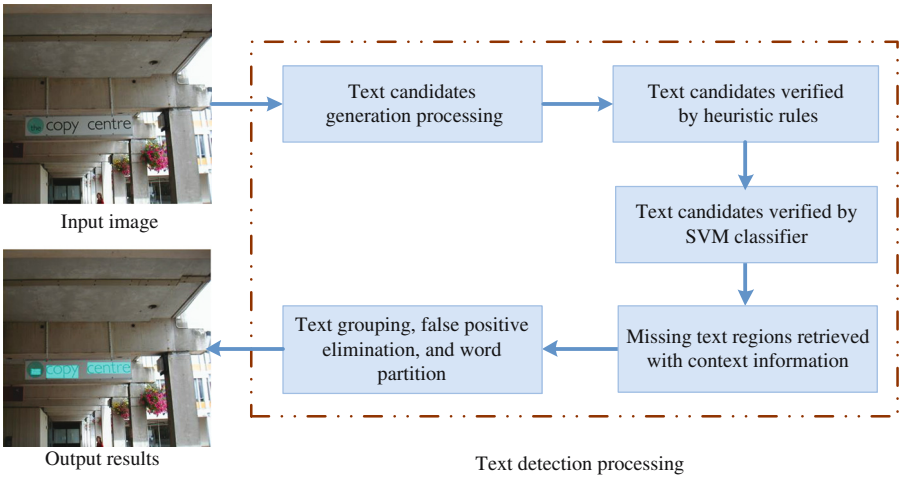


Figure 1: Flowchart of the proposed method.

to obtain an accurate edge map based on the gradient image with the assistance of the visual saliency regions in our work.

2.1.1 Computing saliency

Visual saliency refers to the distinctive perceptual attributes that cause an object to stand out from its surrounding elements, thereby attracting human attention. In our research, we aim to discover a technique for obtaining precise edge maps to enhance the results of the SWT. In this study, we adopt the concise and efficient model proposed by Achanta et al. [28] to calculate saliency for the outputs. The methodology is deemed effective, resulting in full-resolution saliency maps with clearly defined boundaries.

The saliency map S of the image I can be formulated as follows:

$$S(x, y) = |I_{\mu} - I_{\omega \mathrm{hc}}(x, y)|, \tag{1}$$

where I_{μ} is the arithmetic mean pixel value of the image and $I_{\omega \mathrm{hc}}$ is the Gaussian blurred version of the original image, which eliminates fine texture details as well as noise and coding artifacts. The norm of the difference is used to obtain the magnitude of the differences. The features of color and luminance are used in this method, and equation (1) is extended and rewritten as follows:

$$S(x, y) = ||I_{\mu} - I_{\omega \mathrm{hc}}(x, y)||, \tag{2}$$

where, I_{μ} is the mean image feature vector, $I_{\omega \mathrm{hc}}$ is the corresponding image pixel vector value in the Gaussian blurred version (using a 3 × 3 separable binomial kernel) of the original image, and $||\ ||$ is the $L2$ norm. Applying the Lab color space, each pixel location is a $[L;\ a;\ b]^T$ vector, and the $L2$ norm is the Euclidean distance.

2.1.2 Obtaining the accurate edge map by using the multistage edge detection

The Canny edge detection method is characterized by its utilization of two thresholds to identify both strong and weak edges. This method is known to be less susceptible to being impacted by noise when compared to other edge detection techniques such as the Sobel and Prewitt methods. The Canny method operates by identifying local maxima in the gradient of an image, thereby detecting edges. However, it should be noted that this approach may also result in the detection of non-text edges. Efforts to minimize such false detections by adjusting the segment threshold may unfortunately result in the loss of true text edges.

Fortunately, the characters have obvious differences with the backgrounds in text regions. The strokes of the character have similar fixed widths, and the gradient distribution of the character has obvious regularity, (as shown in Figure 2). We can obtain the accurate text edges[1] in the image while dismissing noisy and foliage edges with the following steps. First of all, we strengthen the regular boundary and weaken the irregular one, which is produced mainly by the non-text targets, (i.e., calculating the synthesis gradient value of every pixel with 3 × 3 neighborhoods), and then the edge filter mask is obtained by using binarization processing and morphological processing. At last, the accurate edge is obtained using the following equation:

$$\mathrm{A}_{\mathrm{ccurate\_edge}}(i, j) = \mathrm{O}_{\mathrm{rig\_edge}}(i, j) \times \mathrm{E}_{\mathrm{dge\_filter}}(i, j), \tag{3}$$

where $\mathrm{A}_{\mathrm{ccurate\_edge}}(i, j)$ shows the pixel value of the accurate edge map at position (i, j), $\mathrm{O}_{\mathrm{rig\_edge}}(i, j)$ and $\mathrm{E}_{\mathrm{dge\_filter}}(i, j)$ show the pixel value of the original Canny edge map and the edge filter mask at position (i, j), respectively, $i \in (1, M), j \in (1, N)$, and M and N show the row and column of the gray image, respectively.

1 It should be pointed out that we can qualitatively analyze the accurate edge map, but it is very difficult to do quantitative analysis for unrealistically annotating the text edge images manually. In order to quantitatively evaluate the accurate edge map obtained in our work, we adopted the SWT results and text detection results to verify the effectiveness of the accurate edge map. The evaluation protocol and results are introduced in Section 3.3.

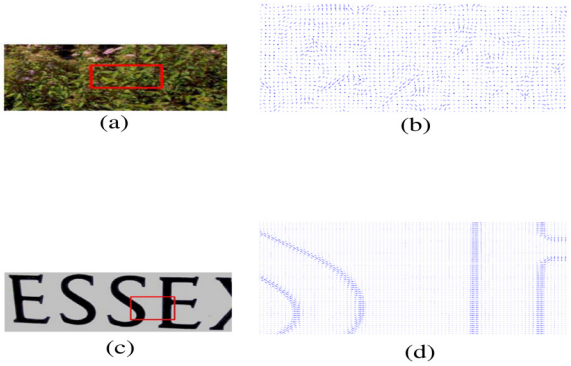


Figure 2: Gradient direction images of the text and the non-text regions. (a) non-text region, (b) gradient direction image of the non-text region, (c) text region, and (d) gradient direction image of the text region.

2.2 Text candidates generation processing

The characters always have a nearly fixed stroke width in the text region. The SWT performs per-pixel computations to estimate the most probable stroke width, resulting in an output image that maintains the same dimensions as the input image. In the SWT image, each element contains the width of the stroke associated with the pixel. The SWT operator is linear with the number of edge pixels, and an accurate edge map often completely enhances SWT results. To describe the effectiveness of the applied method, we obtain accurate edges by using the multistage edge detection method proposed in our method. The SWT images are guided by the original edge map and the accurate edge map separately, and the SWT images are shown in Figure 3. Applying the SWT image guided by the accurate edge map, we can obtain better results in our work.

In text regions, the stroke width of characters is typically consistent. The SWT algorithm computes the most probable stroke width for each pixel, producing an output image of equal size to the input image. Each element in the SWT image corresponds to the stroke width associated with the respective pixel. The computational complexity of the SWT operator is linear with the number of edge pixels present in the image. The accuracy of the SWT results can be significantly improved by incorporating an accurate edge map. To evaluate the performance of the proposed method, we employ a multistage edge detection technique to obtain accurate edges. The SWT images are generated using both the original edge map and the accurate edge map as guides. The results are depicted in Figure 3. It can be observed that the use of the accurate edge map as a guide for the SWT results in improved performance in our study.

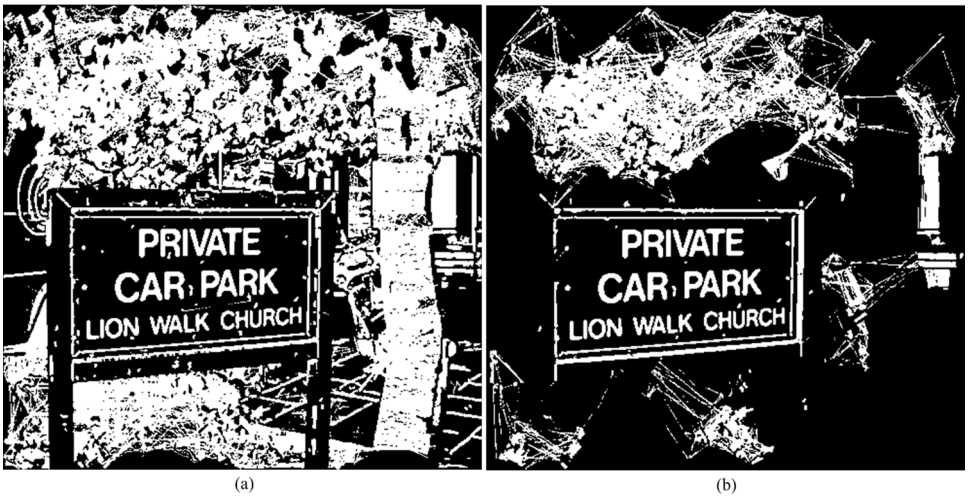


Figure 3: SWT images guided by different edge maps. (a) SWT guided by the original edge, and (b) SWT guided by the accurate edge.

2.3 False positives elimination processing

The SWT generates an output image, where each pixel corresponds to the width of the most probable stroke. While the SWT processing eliminates certain text regions that are unlikely to be text (e.g., by setting a maximum stroke width threshold of 350 pixels in our work, pixels with a stroke width exceeding this threshold are discarded), there may still exist non-text regions that persist and hinder subsequent processing.

2.3.1 False positives elimination by heuristic rules

To differentiate between text and non-text regions, the geometric and textural properties of individual CCs are characterized. In this study, we primarily utilized seven types of component features, as listed in Table 1, to eliminate certain non-text regions.

- *Stroke width changes rate*: This feature, coined as the stroke width variance-to-mean ratio, quantifies the extent of stroke width variability by comparing the variance to the mean stroke width. Its purpose is to facilitate the removal of candidate components characterized by significant deviations in stroke width.
- *Area and bounding box ratio*: This feature is used to remove candidate regions whose foreground pixel density is too small, which is defined as the total number of foreground pixels to the external rectangle area.
- *Height size and width size*: The characters in images always have a certain range of size; this feature is defined to eliminate the candidate regions that have large sizes or small sizes.
- *Containing CCs number*: For the text components, there are not any other CCs inside it, and we define this feature to eliminate some candidate components.
- *Aspect ratio*: This feature, defined as the maximum ratio between component width and height or from its height to width, is used to filter out non-text components whose shape is too long or too narrow. Note that, adhesion problems may occur between the adjacent characters in the horizontal direction, and the aspect ratio constraint of the component is relatively loose, only some outliers with a very large aspect ratio will be removed under this condition.
- *CC area*: The number of foreground pixels of a text region always has a certain range, and this feature is defined to remove candidate regions, whose foreground pixel number is too large or too small.
- *Color range*: In fact, the green leaves are bothersome objects and often appear in the natural scene images. This feature is specially defined to filter out the green leaves in hue, saturation, and value color space.

The parameters in Table 1 are empirically determined based on the international conference on document analysis and recognition (ICDAR) 2005 training dataset. $S_{\text{WCR\_T}}$= 0.45, $R_{\text{ab\_min}}$ = 0.1, $R_{\text{ab\_max}}$ = 0.7, H_{min} = 10, H_{max} = 0.9 × $H_{\text{\_img}}$, W_{min} = 6, W_{max} = 0.8 × $W_{\text{\_img}}$, $C_{\text{CCN\_T}}$ = 4, $A_{\text{R\_max}}$ = 10, $C_{\text{CA\_min}}$ = 50, $C_{\text{CA\_max}}$ = 0.85 × ($H_{\text{\_img}}$ × $W_{\text{\_img}}$), $H_{\text{ue\_T}}$ = 0.25, $S_{\text{aturation\_T}}$ = 0.4, $H_{\text{ue\_deta\_T}}$ = 0.05, and $S_{\text{aturation\_deta\_T}}$ = 0.15, where $H_{\text{\_img}}$ and $W_{\text{\_img}}$ are the height and the width of the input images and $H_{\text{ue\_ave}}$, and $S_{\text{aturation\_ave}}$ are the average hue and

Table 1: Features for a candidate region in heuristic rules

Feature type	Feature	Definition
Stroke	Stroke width change rate	$S_{\text{WCR}}(x_i) < S_{\text{WCR\_T}}$
Spatial	Area and bounding box ratio	$(R_{\text{ab}}(x_i) = a_{x_i}/b_{x_i}) \in (R_{\text{ab\_min}}, R_{\text{ab\_max}})$
	Height size and width size	$H(x_i) \in (H_{\text{min}}, H_{\text{max}})$ & $W(x_i) \in (W_{\text{min}}, W_{\text{max}})$
	Containing CC number	$C_{\text{CCN}}(x_i) < C_{\text{CCN\_T}}$
	Aspect ratio	$(A_R(x_i) = \max(w_i/h_i, h_i/w_i)) < A_{R\_\text{max}}$
	Connect component area	$C_{\text{CA}}(x_i) \in (C_{\text{CA\_min}}, C_{\text{CA\_max}})$
Color	Color range	$H_s(x_i) \notin (H_{\text{ue\_}T}, S_{\text{aturation\_}T}, H_{\text{ue\_deta}}, S_{\text{aturation\_deta}})$ where, $\lvert H_{\text{ue\_ave}} \ H_{\text{ue\_T}}\rvert < H_{\text{ue\_data\_}T}$ & $\lvert S_{\text{aturation\_ave}} \ S_{\text{aturation\_}T}\rvert > S_{\text{aturation\_deta\_}T}$

the average saturation of the candidate region. In order to prove the robustness of the proposed method, the same parameter values in these heuristic rules are used for detecting texts in both the ICDAR 2005 and ICDAR 2011 test datasets.

2.3.2 False positives elimination by classifier

The aforementioned heuristics are effective in eliminating certain non-text regions. However, some non-text candidates that possess similar geometric and textural properties to text regions may still persist. To address this issue, we employ an offline trained classifier to distinguish between text and non-text candidates. In natural scene text images, texts often appear in groups or lines, and image binarization processing can result in adhesion between adjacent characters. This adhesion can cause significant variations in the length of candidate regions, making it challenging to use a fixed size to describe them. To mitigate this problem, we present a novel hierarchical identification strategy in our study, aimed at reducing the texture distortion caused by image normalization processing and effectively addressing the variation in length of candidate regions.

- If the aspect ratio of the CC region is between 2 and 4, we recognize the CC region directly.
- In our approach, if the aspect ratio of a CC region exceeds 4, we intercept it from the left, right, and middle positions, respectively, in order to maintain an aspect ratio of ~3 for each sub-region. A voting-based method is then employed to classify the CC region. If more than one sub-region is identified as text, the entire CC region will be deemed a text region. Otherwise, it will be classified as a non-text region.
- In our approach, if the aspect ratio of a CC region is below 2, we first determine the number of adjacent CCs with similar heights in the horizontal direction. If the number of adjacent regions is greater than 2, the CC region is immediately classified as text. This decision strategy is based on the observation that text lines typically contain more than three characters, while non-text regions rarely have multiple adjacent regions with similar heights. If the number of adjacent regions is not greater than 2, the CC region is joined with itself until the aspect ratio approaches 3, and the resulting composite region is then evaluated by the trained classifier.

In order to ensure a fair comparison, we trained text classifiers on the training datasets of the ICDAR 2005 and the ICDAR 2011, and we applied these trained classifiers to the respective test sets of ICDAR 2005 and ICDAR 2011. To develop the training sets, 2,338 positive samples and 2,709 negative samples were collected from the ICDAR 2005 training dataset, whereas 3,488 positive samples and 3,738 negative samples were collected from the ICDAR 2011 training dataset. The training samples are normalized to 48 × 144. The Histograms of Gradients (HOG) feature is selected, and the SVM classifier is trained with this feature by the offline method. In our research, each detection window is divided into cells of 8 × 8 pixels and each group of 2 × 2 cells is integrated into a block, and each cell consists of nine orientation bins.

2.4 Retrieving the missing text regions around the key regions

Despite the potential loss of text regions through prior processing, it is fortunate that text in natural scenes frequently appears in groups or lines. As a result, candidates in proximity to text regions are highly likely to be considered as text. To take advantage of the specific spatial distribution of text, we utilize context information to recover missing text regions.

In order to describe the algorithm conveniently, we illustrate this question aided by Figure 4. In our work, the key region is defined as the region of the remaining adjacent candidates in the horizontal direction. As shown in Figure 4, the words “AR PARK” highlighted with a yellow background are the key region, the red dashed box indicates the horizontal search area, and the yellow dashed box indicates the vertical search area. Meanwhile, we further imagine that the candidate regions are lost after a series of ahead processing, which are shown in the left and right parts of Figure 4.

Figure 4: Retrieving the missing text regions by using context information.

In our study, we leveraged context information to retrieve the missing text regions by utilizing key regions. To formulate context descriptors, we employ four distinct features, namely stroke width, common part, CC width, and CC height. We denote all the missing candidate regions after a series of ahead processing as $C_r = \{c_r^1, c_r^2, \ldots, c_r^i, \ldots, c_r^M\}$, where M is the total number of missing candidate regions. The c_r^i is the ith missing candidate region, and we define the state of c_r^i as $\{c_s^i, c_c^i, c_w^i, c_h^i\}$, where the c_s^i is the stroke width and c_c^i is the common part with the key region (i.e., it is the common width with the key region in the vertical search area, and the common height in the horizontal search area). Note that c_w^i is the CC whole width and c_h^i is the whole height. Meanwhile, we denote the key region $K_r = \{k_r^1, k_r^2, \ldots, k_r^i, \ldots, k_r^N\}$, where N is the total number of characters in the key region. The K_s^{ave} is the average stroke width of the characters in the key region, the K_w^{ave} is the average width, and the K_h^{ave} is the average height.

The missing text candidates will not be retrieved in the search area, if they meet any one of the following conditions:

$$\begin{cases} c_c^i < T_1 \times k_w^{ave}, & \text{when in vertical area} \\ c_c^i < T_2 \times k_h^{ave}, & \text{when in horizontal area} \end{cases} \tag{4}$$

$$\min(c_s^i, k_s^{ave}) / \max(c_s^i, k_s^{ave}) < T_3 \tag{5}$$

$$\min(c_w^i, k_w^{ave}) / \max(c_w^i, k_w^{ave}) < T_4 \tag{6}$$

$$\min(c_h^i, k_h^{ave}) / \max(c_h^i, k_h^{ave}) < T_5 \tag{7}$$

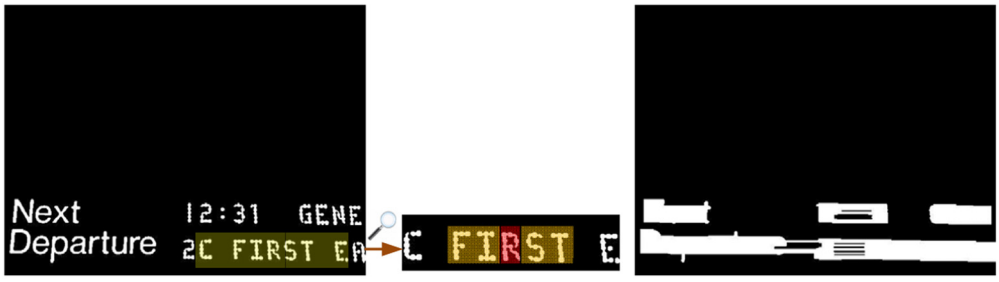
Note that, the first row of equation (4) is one of the requirements to find the missing text regions in the vertical searching area, and the second row of equation (4) corresponds to the horizontal search area. Based on the pixel-wise annotation of the ICDAR 2003 training dataset,[2] the parameters are empirically selected, and $T_1 = T_2 = 0.5$ and $T_3 = T_4 = T_5 = 0.75$ in our work.

2.5 Grouping characters into text line and further verification

The objective of this step is to group the set of characters identified in the previous steps into coherent lines of text. The characters within the same text line are anticipated to exhibit similar stroke width, character width, and character height. Our work operates under the assumption that the text lines in natural scenes tend to be horizontal or slightly tilted. To address this, we introduce the concept of an "influence field." For example, the influence field for the character R in Figure 5(b) is depicted by the yellow background. The CC candidates within the influence field are compared to the reference character based on their CC width and height. If these CCs meet specified constraint conditions, they are merged together.

$$\min(c_h^i, c_h^{seed}) / \max(c_h^i, c_h^{seed}) > T_1 \tag{8}$$

2 Available at: http://graphics.cs.msu.ru/en/research/projects/msr/text.



(a) (b) (c)

Figure 5: The results of grouping characters into text line. (a) Before grouping characters into text line, (b) the influence field of a character, and (c) after grouping characters into text line.

$$\min(c_{\mathrm{w}}^{i}, c_{\mathrm{w}}^{\mathrm{seed}}) / \max(c_{\mathrm{w}}^{i}, c_{\mathrm{w}}^{\mathrm{seed}}) > T_2 \tag{9}$$

Note that, for the ith CC in the influence field, the c_{w}^{i} is the whole width and c_{h}^{i} is the whole height. The parameter $c_{\mathrm{w}}^{\mathrm{seed}}$ is the width and $c_{\mathrm{h}}^{\mathrm{seed}}$ is the height of the specific candidate. For every specific candidate, the morphology structure element is dynamically selected as $1 \times (T \times c_{\mathrm{h}}^{\mathrm{seed}})$. The morphology close processing is implemented between the specific candidate and the CCs in its influence field. For every CC that is taken into account, the text line is developed by merging the sub-images, which are secured by using morphology close processing. The before and after grouping characters in text lines are shown in Figure 5(a) and (c), respectively, and in this section, $T = 1.5$ and $T_1 = T_2 = 0.65$.

In the task of text region recognition, the selection of appropriate descriptors is critical for effectively differentiating between text regions and background interferences. The SVM classifier utilized in the previous stage has demonstrated remarkable efficacy in eliminating background interferences. However, text lines in natural scene images often exhibit significant layout variations, making it challenging to train a comprehensive classifier using generic descriptors. As a result, some persistent background interferences may not be fully removed in previous processing steps. To address this challenge, a different approach is adopted. Guided by the work of Wang et al. [14], an unsupervised feature learning algorithm is utilized to automatically extract features from the training samples, and a convolutional neural networks (CNN) classifier is applied for candidate region classification. Although the CNN classifier presented in Wang et al. [14] demonstrates exceptional classification capabilities, it is not employed for text detection in the original image due to its time-consuming nature as it performs multi-scale evaluations of candidate regions. In light of these considerations, an off-the-shelf CNN classifier is utilized in this work for efficient verification of candidate text lines with smaller areas than the original image.

2.6 Segmenting text line into words

In the context of scene text detection, while word segmentation is not the primary concern, it is nonetheless necessary to segment the detected text lines into separate words for the purpose of evaluating performance using the strategies employed in the ICDAR 2005 and ICDAR 2011 robust reading competitions. To address this requirement, a heuristic rule has been developed to facilitate the segmentation process. This rule is based on the calculation of the average distance between adjacent CCs within a given text line. The minimum word spacing distance T is estimated by equation (10). A separation will occur if the distance between adjacent CCs is over T, and the intra-word characters would be separated from the inter-word characters.

$$T = \alpha \times D_{\mathrm{ave}} + \beta, \tag{10}$$

where D_{ave} is the average distance value. Based on the previous research, we empirically set $\alpha = 1.75$ and $\beta = 3$.

3 Performance evaluation

3.1 Experimental datasets

In order to comprehensively assess the performance of the proposed method in comparison with other state-of-the-art text detection methods, we carried out experiments on the ICDAR 2005 dataset[3] and ICDAR 2011 dataset.[4] The ICDAR 2005 dataset consists of ~499 natural scene images that have been annotated with ground truth, with varying resolutions, and its TrialTest subset contains a total of 249 images. The ICDAR 2011 dataset was specifically collected for the text detection competition at the ICDAR 2011 conference, and it comprises 484 natural scene images that have been annotated with ground truth, with a corresponding test dataset of 255 images.

3.2 Evaluation protocol

The performance of our method is quantitatively measured by Precision (P), Recall (R), and *f*-measure (F). They are separately computed using the definitions provided in the studies by Lucas et al. [29], Lucas [30], and Shahab et al. [31]. The output is a set of rectangles designated with bounding boxes for detected texts, and a set of ground-truth rectangles are also provided in these datasets. The match m_p between two rectangles is defined as the area of the intersection divided by the area of the minimum bounding box containing both rectangles. The number has the value one for identical rectangles and zero for rectangles that have no intersection. The closest match of each resulting rectangle with the set of truths is calculated.

The best match $m(r; R)$ for a rectangle r in a set of rectangles R is defined as follows:

$$m(r; R) = \max\{m_p(r; r0)|r0 \in R\}. \tag{11}$$

The Precision and Recall are defined as follows:

$$\text{Precision} = \frac{\sum_{r_e \in E} m(r_e, T)}{|E|} \tag{12}$$

$$\text{Recall} = \frac{\sum_{r_t \in T} m(r_t, E)}{|T|}, \tag{13}$$

where T and E represent the sets of ground truth and result rectangles, respectively. The *f*-measure, which is a single measure of algorithm performance, is a combination of the two measures, and it is defined as follows:

$$f = \frac{1}{\frac{a}{\text{Precision}} + \frac{1-a}{\text{Recall}}}, \tag{14}$$

where the parameter α is the relative weight of the Precision and the Recall, and $\alpha = 0.5$ in our work.

3.3 Evaluation of SWT results obtained by using the multistage edge detection

To prove the effectiveness of the multistage edge detection method, we adopted the ICDAR 2005 annotation test dataset[5] as the baseline. Compared with the baseline in pixel level, the detection results are quantitatively measured by the Recall and the Precision. Note that, in this section, the two parameters are defined as follows:

3 Available at http://algoval.essex.ac.uk/icdar/Datasets.html#RobustReading.html.

4 Available at http://robustreading.opendfki.de/wiki/SceneText#TrainingDataset.

5 Available at http://graphics.cs.msu.ru/en/research/projects/msr/text.

Table 2: Comparative results by using multistage edge detection

Method	Recall	Precision
With multistage edge detection	0.88	0.21
Without multistage edge detection	0.95	0.11

$$\text{Recall} = \frac{\sum(I_{\text{mg\_dec}} \cap I_{\text{mg\_gt}})}{\sum I_{\text{mg\_gt}}} \tag{15}$$

$$\text{Precision} = \frac{\sum(I_{\text{mg\_dec}} \cap I_{\text{mg\_gt}})}{\sum I_{\text{mg\_dec}}}, \tag{16}$$

where $I_{\text{mg\_dec}}$ is our segmentation result obtained by binarizing the SWT image and $I_{\text{mg\_gt}}$ is the corresponding ground truth. In order to prove the effectiveness of the multistage edge detection algorithm, we adopted the original Canny edge map and the accurate edge map to obtain the SWT image, respectively. As shown in Table 2, we can greatly increase the Precision with a little cost of the Recall reduction.

There are total 1.08×10^7 pixels annotated as texts in the ICDAR 2005 annotation test dataset. With the original Canny edge map for SWT processing, we can obtain 1.1×10^8 candidate text pixels, and there are 9.77×10^6 true text pixels in it. However, with the accurate edge map for SWT processing, we can obtain 5.0×10^7 candidate text pixels with 8.15×10^6 true text pixels in them. The aforementioned results show that we can greatly remove the false positive pixels with a little cost of the true positive pixels reduction. In addition, the average time consumption of performing SWT processing guided by the original edge maps is 16.6 s; however, the average time consumption is 10.8 s by using the accurate edge maps.[6]

To further illustrate this problem, we first, we adopt the original Canny edge map and the accurate edge map to obtain the SWT image, respectively, and then the same exact processing steps as the subsequent pipelines will follow the SWT processing to detect the scene texts. For the original Canny edge map case, we can obtain the Recall 0.61, the Precision 0.66, and the *f*-measure 0.63. However, for the accurate edge map case, we can obtain the Recall 0.62, the Precision 0.73, and the *f*-measure 0.67. It is obvious that better text detection performance has been obtained by using the accurate edge map obtained by our multistage edge detection method.

3.4 Evaluation of retrieving the missing text regions

In order to further verify the effectiveness of retrieving the missing text regions by using the context information, a comparison experiment has been designed on the ICDAR 2005 TrialTest dataset. The results, which are obtained before retrieving and after retrieving the missing text regions, are compared with the ICDAR 2005 annotation test dataset at the pixel level, and they are quantitatively measured by the Recall and the Precision defined in equations (15) and (16). Before retrieving the missing text regions, we can obtain the Recall 0.66 and the Precision 0.53; however, we can obtain the Recall 0.72 and Precision 0.52 after retrieving the missing text regions. It is obvious that we can greatly improve the Recall rate with a little cost of the Precision reduction.

3.5 Comparison with other approaches

In order to verify the effectiveness of the text detection scheme proposed in this work, experiments have been designed and carried out on two public datasets, and the detected results are evaluated at the word level. The performance of our method is quantitatively measured by Precision (P), Recall (R), and *f*-measure (F), and they

6 In order to compare time consumption of performing SWT processing guided by the original edge map and the accurate edge map, respectively, we implemented our method in Matlab on an Intel(R) Core(TM)2 Duo CPU 2.8 GHz and 3 GHz RAM desktop. In order to accommodate both bright text on dark background and vice-versa, we apply the SWT processing twice.

Table 3: Experimental results on the ICDAR 2005 dataset

Method	P	R	F
Proposed method	**0.73**	**0.62**	**0.67**
Epshtein et al. [21]	0.73	0.60	0.66
Li et al. [33]	0.62	0.65	0.63
1st ICDAR 2005 [30]	0.62	0.67	0.62
Yi and Tian [34]	0.71	0.62	0.62
Meng et al. [35]	0.66	0.57	0.61
Yao et al. [36]	0.64	0.60	0.61
2nd ICDAR 2005 [30]	0.60	0.60	0.58
Wang et al. [37]	0.58	0.60	0.57
Fabrizio et al. [38]	0.63	0.50	0.56*
Zhang and Kasturi [39]	0.67	0.46	0.55*
1st ICDAR 2003 [29]	0.55	0.46	0.50
2nd ICDAR 2003 [29]	0.44	0.46	0.45

* The values are calculated by us based on the P and R results reported by Fabrizio et al. [38] and Zhang and Kasturi [39] in their studies.

are computed by using the definitions provide in the studies by Lucas et al. [29], Lucas [30], and Shahab et al. [31]. It is imperative to acknowledge that the evaluation methodology employed in the ICDAR 2011 competition differs from that of the preceding ICDAR 2005 competition. The revised evaluation scheme was introduced by Wolf et al. [32]. The performances of the proposed method and other state-of-the-art methods on the ICDAR 2005 and the ICDAR 2011 databases are shown in Tables 3 and 4, and our results in each table are highlighted with bold font.

We can know from Tables 3 and 4, that our method achieved the Precision 0.73, Recall 0.62, and *f*-measure 0.67, respectively, on the ICDAR 2005 dataset. Meanwhile, a similar competitive performance is obtained on the ICDAR 2011 dataset, in which we achieved the Precision 0.69, Recall 0.58, and *f*-measure 0.63. Comparing our approach with other state-of-the-art methods listed in Tables 3 and 4, our approach has achieved competitive performances with the most state-of-the-art methods on the ICDAR 2005 and the ICDAR 2011 datasets. Specifically, as shown in Table 3, our results on all performance indicators are better than the method of Epshtein et al. [21]. The SWT processing was first proposed in the study by Epshtein et al. [21], and they had achieved great success in scene text detection.

Figure 6 shows some typical results obtained by our method, and the detected text regions are bounded by red rectangles. These results indicate that our system is robust against large variations in text font, color, size,

Table 4: Performance comparison of text detection methods on the ICDAR 2011 dataset

Method	P	R	F
1st ICDAR 2011 [31]	0.83	0.63	0.71
Li et al. [33]	0.63	0.68	0.65
Proposed method	**0.69**	**0.58**	**0.63**
2nd ICDAR 2011 [31]	0.67	0.58	0.62
TH-TextLoc System*	0.67	0.58	0.62
Li et al. [40]	0.59	0.62	0.61
Neumann et al.*	0.69	0.53	0.60
TDM IACS*	0.64	0.54	0.58
LIP6-Retin*	0.63	0.50	0.56
KAIST AIPR System *	0.60	0.45	0.51
ECNU-CCG Method*	0.35	0.38	0.37
Text Hunter*	0.50	0.26	0.34

* The results of these methods, which are the participants in the ICDAR 2011 text detection competition, are reported in the study by Shahab et al. [31].

Figure 6: Typical results obtained by our method on the two public datasets.

and geometric distortion. In addition, one of the advantages of our method is that we can detect text regions with less than three characters (as shown in the bottom row of Figure 6), while these text regions are always lost by the other proposed methods (e.g., [34]) for the assumption that the text regions always have more than three characters.

3.6 Weakness

Although some satisfactory detection results have been obtained by using our method, there are still some false positives that cannot be eliminated, because these text candidates are very like the genuine texts, e.g., the rectangular windows in Figure 7(a) and the wire fences in Figure 7(b) are very like the digital 1; The intra-word characters may be incorrectly separated from the inter-word characters (as shown in Figure 7(c)). Our method has difficulty in detecting some exaggerated art texts (as shown in Figure 7(d)), and the proposed method does not work well when the text regions have a poor resolution (as shown in Figure 7(e) and (f)). Meanwhile, the

Figure 7: Some poor results were obtained by our method. (a) false alarm due to rectangular windows, (b) false alarm derived from wire fences, (c) wrongly segmented texts, (d) missing texts from exaggerated art texts, (e) missing texts due to dark resolution, and (f) missing texts from glare regions.

hand-designed heuristic rules in Section 2.3.1 may affect the robustness of the algorithm. In our future work, we will try to resolve these problems by improving the detection method.

In addition, we found in our experiments that, as a feature extraction method, the SWT transform is strongly influenced by the accuracy of edge extraction, while the traditional Sobel edge detection operator based on differential operations lacks robustness in dealing with noisy images. Accordingly, this poses difficulties for the subsequent elimination of false alarms and for improving the performance of the algorithm. In fact, deep learning has substantially improved the accuracy of edge detection in recent years, so obtaining high-precision text edges by deep learning and then obtaining stroke width features by SWT are expected to improve text detection performance.

In Table 1, the establishment of discriminative conditions for candidate regions is predicated upon their geometric features, yielding the algorithm with diminished robustness primarily owing to the empirical selection of hyperparameter thresholds. An Interval Type-3 Fuzzy System and a new online fractional-order learning algorithm are proposed in the literature [41], which has fewer rules compared to traditional methods and learning iteration results, which has better stability for prediction tasks. Based on this, we will try to adopt the work proposed in the literature [41] for fast classification of text candidate regions in our future work.

4 Conclusion and future work

In this study, we propose a novel approach for detecting text in natural scenes. The method consists of four stages: (1) the utilization of an effective visual attention model, which effectively highlights text regions and suppresses the background; (2) the implementation of a multistage edge detection process to produce a precise edge map; (3) the verification of text candidates through heuristic rules and an offline trained classifier; and (4) the incorporation of contextual information to retrieve missing text regions and group characters into text lines, followed by further verification and word segmentation. The advancedness of the proposed approach is demonstrated through experimental evaluations on the ICDAR 2005 and the ICDAR 2011 benchmark datasets. In addition, the results indicate that the multistage edge detection process significantly improves the results obtained through the SWT algorithm. Future work aims to extend the scope of the proposed method toward arbitrary text detection in natural scenes. The planned additions of new and enhanced features will further distinguish texts from their background. In addition, a machine learning-based fuzzy system is planned to be developed in order to fast classification of text candidate regions without human intervention. Furthermore, a more effective segmentation strategy is planned to be designed to accurately separate intra-word characters from inter-word characters.

Funding information: The authors would like to thank the Foreign Language Research Joint Project of the Social Science Foundation of Hunan Province (Grant No. 2021WLH35), the Key Research and Development Program of Changsha Science and Technology Bureau (Grant No. kq2004050), the Scientific Research Foundation of this Education Department of Hunan Province of China (Grant No. 21A0052), and the Science and Technology Program of the State Administration for Market Regulation under Grant 2021MK080 for their support.

Author contributions: All authors contributed to this article. All authors have accepted responsibility for the entire content of the manuscript and approved its submission.

Conflict of interest: The authors state that there is no conflict of interest.

Data availability statement: Data sharing is not applicable to this article.

Ethical approval: The conducted research is not related to either human or animal use.

References

[1] M. A. Klllopotek, *On a deficiency of the fci algorithm learning Bayesian networks from data*, Demonstr. Math. **33** (2000), no. 1, 181–194.

[2] R. Pugliese, S. Regondi, and R. Marini, *Machine learning-based approach: Global trends, research directions, and regulatory standpoints*, Data Sci. Management **4** (2021), 19–29.

[3] J. Liu, J. He, Z. Tang, Y. Xie, W. Gui, T. Ma, et al., *Frame-dilated convolutional fusion network and GRU-based self-attention dual-channel network for soft-sensor modeling of industrial process quality indexes*, IEEE Trans. Syst. Man Cybernet. Sys. **52** (2022), no. 9, 5989–6002.

[4] R. Minetto, N. Thome, M. Cord, N. J. Leite, and J. Stolfi, *T-HOG: An effective gradient-based descriptor for single line text regions*, Pattern Recognition J. Pattern Recognition Soc. **46** (2013), no. 3, 1078–1090.

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

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

[7] Y. C. Wei and C. H. Lin, *A robust video text detection approach using SVM*, Expert Syst. Appl. **39** (2012), no. 12, 10832–10840.

[8] M. Liao, Z. Zou, Z. Wan, C. Yao, and X. Bai, *Real-time scene text detection with differentiable binarization and adaptive scale fusion*, IEEE Trans. Pattern Anal. Machine Intell. **45** (2022), no. 1, 919–931.

[9] Y. Liu, R. Wang, G. Zhu, M. Liu, C. Han, X. He, et al., *EWST: an extreme weather scene text detector with dehazing and localization refinement*, J. Electr. Imag. **32** (2023), no. 1, 013007.

[10] Y. Cai, Y. Liu, C. Shen, L. Jin, Y. Li, and D. Ergu, *Arbitrarily shaped scene text detection with dynamic convolution*, Pattern Recognition **127** (2022), 108608.

[11] Q. Ye, Q. Huang, W. Gao, and D. Zhao, *Fast and robust text detection in images and video frames*, Image Vision Comput. **23** (2005), no. 6, 565–576.

[12] S. M. Hanif and L. Prevost, *Text detection and localization in complex scene images using constrained AdaBoost algorithm*, in: 2009 10th International Conference on Document Analysis and Recognition, IEEE, 2009, pp. 1–5.

[13] H. Xu and F. Su, *A robust hierarchical detection method for scene text based on convolutional neural networks*, in: 2015 IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2015, pp. 1–6.

[14] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, *End-to-end text recognition with convolutional neural networks*, in: Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), IEEE, 2012, pp. 3304–3308.

[15] L. Sun, Q. Huo, W. Jia, and K. Chen, *Robust text detection in natural scene images by generalized color-enhanced contrasting extremal region and neural networks*, in: 22nd International Conference on Pattern Recognition, IEEE, 2014, pp. 2715–2720.

[16] C. Shi, C. Wang, B. Xiao, Y. Zhang, and S. Gao, *Scene text detection using graph model built upon maximally stable extremal regions*, Pattern Recognition Letters **34** (2013), no. 2, 107–116.

[17] X. C. Yin, X. Yin, K. Huang, and H. W. Hao, *Robust text detection in natural scene images*, IEEE Trans. Pattern Analysis Machine Intell. **36** (2013), no. 5, 970–983.

[18] C. Mancas-Thillou and B. Gosselin, *Color text extraction with selective metric-based clustering*, Computer Vision Image Understanding **107** (2007), no. 1–2, 97–107.

[19] P. Shivakumara, T. Q. Phan, and C. L. Tan, *A Laplacian approach to multi-oriented text detection in video*. IEEE Trans Pattern Analysis Machine Intell. **33** (2010), no. 2, 412–419.

[20] L. Sun, Q. Huo, W. Jia, and K. Chen, *A robust approach for text detection from natural scene images*. Pattern Recognition **48** (2015), no. 9, 2906–2920.

[21] B. Epshtein, E. Ofek, and Y. Wexler, *Detecting text in natural scenes with stroke width transform*, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 2963–2970.

[22] H. Xu, L. Xue, and F. Su, *Scene text detection based on robust stroke width transform and deep belief network*, in: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1–5, 2014, Revised Selected Papers, Part II 12, Springer International Publishing, 2015, 195–209.

[23] T. Judd, K. Ehinger, F. Durand, and A. Torralba, *Learning to predict where humans look*. in: 12th International Conference on Computer Vision, IEEE, 2009, pp. 2106–2113.

[24] S. Karaoglu, J. C. Van Gemert, and T. Gevers, *Object reading: text recognition for object recognition*. in: Computer Vision-ECCV 2012. Workshops and Demonstrations: Florence, Italy, October 7–13, 2012, Proceedings, Part III 12, Springer, Berlin Heidelberg, 2012, pp. 456–465.

[25] Q. Sun, Y. Lu, and S. Sun, *A visual attention based approach to text extraction*, in: 20th International Conference on Pattern Recognition, IEEE, 2010, pp. 3991–3995.

[26] C. Xue, W. Zhang, Y. Hao, S. Lu, P. H. Torr, and S. Bai, *Language matters: A weakly supervised vision-language pre-training approach for scene text detection and spotting*, in: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII, Springer Nature Switzerland, Cham, 2022, pp. 284–302.

[27] C. Gu, S. Wang, Y. Zhu, Z. Huang, and K. Chen, *Weakly supervised attention rectification for scene text recognition*, in: 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, pp. 779–786.

[28] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, *Frequency-tuned salient region detection*, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 1597–1604.

[29] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, et al., *ICDAR 2003 robust reading competitions: entries, results, and future directions*, Int. J. Document Analysis Recognition (IJDAR) **7** (2005), 105–122.
[30] S. M. Lucas, *ICDAR 2005 text locating competition results*, in: 8th International Conference on Document Analysis and Recognition (ICDAR'05), IEEE, 2005, pp. 80–84.
[31] A. Shahab, F. Shafait, and A. Dengel, *ICDAR 2011 robust reading competition challenge 2: Reading text in scene images*. in: 2011 International Conference on Document Analysis and Recognition, IEEE, 2011, pp. 1491–1496.
[32] C. Wolf and J. M. Jolion, *Object count/area graphs for the evaluation of object detection and segmentation algorithms*, Int. J. Document Analysis Recognition (IJDAR) **8** (2006), no. 4, 280–296.
[33] Y. Li, C. Shen, W. Jia, and A. Van Den Hengel, *Leveraging surrounding context for scene text detection*, in: IEEE International Conference on Image Processing, IEEE, 2013, pp. 2264–2268.
[34] C. Yi and Y. L. Tian, *Text string detection from natural scenes by structure-based partition and grouping*. IEEE Transactions on Image Processing, **20** (2011), no. 9, 2594–2605.
[35] Q. Meng and Y. Song, *Text detection in natural scenes with salient region*, in: 10th IAPR International Workshop on Document Analysis Systems, IEEE, 2012, pp. 384–388.
[36] J. L. Yao, Y. Q. Wang, L. B. Weng, and Y. P. Yang, *Locating text based on connected component and SVM*, 2007 International Conference on Wavelet Analysis and Pattern Recognition, IEEE, vol. 3, 2007, pp. 1418–1423.
[37] R. Wang, N. Sang, R. Wang, and X. Kuang, *A hybrid approach for text detection in natural scenes*, in: MIPPR 2013: Pattern Recognition and Computer Vision, vol. 8919, SPIE, 2013, pp. 137–142.
[38] J. Fabrizio, B. Marcotegui, and M. Cord, *Text detection in street level images*, Pattern Analysis Appl. **16** (2013), 519–533.
[39] J. Zhang and R. Kasturi, *Text detection using edge gradient and graph spectrum*, in: 20th International Conference on Pattern Recognition, IEEE, 2010, pp. 3979–3982.
[40] Y. Li and H. Lu, *Scene text detection via stroke width*. in: Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), IEEE, 2012, pp. 681–684.
[41] A. Mohammadzadeh, M. H. Sabzalian, and W. Zhang, *An interval type-3 fuzzy system and a new online fractional-order learning algorithm: theory and practice*, IEEE Trans. Fuzzy Syst. **28** (2020), no. 9, 1940–1950.