Int. J. Nonlinear Anal. Appl. 10 (2019) No. 1, 229-237
ISSN: 2008-6822 (electronic)
https://dx.doi.org/10.22075/IJNAA.2019.4066

Natural scene text localization using edge color signature

Jalil Ghavidel Neycharan[a], Alireza Ahmadyfard[b,*], Morteza Zahedi[a]

[a]*Faculty of Computer Engineering, Shahrood University of Technology, Shahrood, Iran*
[b]*Faculty of Electrical and Robotic Engineering, Shahrood University of Technology, Shahrood, Iran*

(Communicated by J. Vahidi)

Abstract

Localizing text regions in images taken from natural scenes is one of the challenging problems due to variations in font, size, color and orientation of text. In this paper, we introduce a new concept so called Edge Color Signature for localizing text regions in an image. This method is able to localize both Farsi and English texts. In the proposed method first a pyramid using different scales of the input image is created. Then for each level of the pyramid an edge map is extracted. Afterward, several geometric features are employed to filter out the non-text edges from the extracted edges. At this stage we describe an edge using colors of its neighboring pixels. We use the mean-Shift algorithm to obtain the color modes surrounding each edge pixel. Subsequently, the connected edge pixels with similar color signatures are clustered using Single-Linkage clustering algorithm to construct meaningful groups. Finally, each of the clusters is labeled as text or non-text using an MLP based cascade classifier. The proposed method has been evaluated on well-known ICDAR 2013 and our Farsi dataset, the result is very promising.

Keywords: Text localization, Natural scene images, Edge color signature, Clustering, Farsi language.
2010 MSC: 68T45.

1. Introduction

Analyzing visual data has been one of the most challenging problems in the last decade. Hundreds of hours of videos are uploaded to online video sharing websites every minute and millions of images are

*Corresponding author
Email addresses: jalil_ghavidel@shahroodut.ac.ir (Jalil Ghavidel Neycharan), ahmadyfard@shahroodut.ac.ir (Alireza Ahmadyfard), zahedi@shahroodut.ac.ir (Morteza Zahedi)

Received: June 2019 *Revised:* October 2019

shared on social media. Therefore, automatic processing of this visual data for interpretation, annotation and better understanding of its concept is very important. Extraction of text regions in images taken from natural scenes can lead to better analysis of visual data. This has many applications such as helping a tourist or a blind person to better perceive surrounding places. It also improves image indexing methods. Generally, a text recognition system has three stages: *Localization*, *Extraction and enhancement* and *Recognition* [1]. The *localization* aims at finding and labeling text regions in an image or video [2]. The *text extraction and enhancement* aims to separate text region from the background and enhance text image. Finally, a text string is obtained from the enhanced binary image in the *recognition* subsystem. Since it is the prerequisite of other two subsystems, the localization subsystem plays an important role in the performance of whole system.

Since size, font, color and lighting condition for text regions in images of natural scenes vary considerably, localizing text regions in these images is a challenging problem. Moreover, Farsi text localization is more challenging in compare with Latin text; since Farsi is a cursive language, foliage and other repeating entities are more likely to be misclassified as Farsi text. Although a considerable number of methods have been proposed for localizing Latin text regions, there exist a few number of methods for Farsi text. In addition, existing methods for localizing Latin text are not applicable for Farsi mainly due to its cursive nature. For example, [3] is one of the most cited methods for Latin text localization. The authors [3] have shared a demo of their method on a website [4]. We fed a few images to the demo and the outputs of this method for Farsi and Latin has been shown in Figure 1. As it can be seen, the method successfully localizes Latin texts (Figure 1a), whereas it is unable to locate Farsi text samples (Figure 1b).

(a) (b)

Figure 1: Outputs of the method proposed in [3] for a) Images with English text b) Images with Farsi text. Detected text regions have been shown with red rectangles

This paper aims to propose a method for localizing Farsi texts from natural scene images however it can also be applied for Latin text. The main idea of our method is based on an observation that usually the set of colors surrounding a text region's edge pixels are the same. We use this intuition to propose a method for localizing text regions considering the set of colors surrounding an edge pixel as a signature to describe it. This signature is employed as a feature for grouping similar edge pixels as candidate text regions. Since the proposed method makes no assumptions about the structure of the target language, it can be used for localizing other languages as well.

This paper has been organized as follows. The related works in the literature has been reviewed in section 2. As we use the Mean-Shift algorithm to extract the color signature of each edge pixel, we

briefly explain this algorithm in section 3. Details of the proposed method are elaborated in section 4. Sections 5 is dedicated to experimental results and finally, section 6 concludes the paper.

2. Related work

The methods proposed for localizing text regions in images of natural scenes can be categorized into two main categories: based on sliding window and based on Connected Component (CC) [2]. The methods in the first group, consider a sliding window on the input image and classify image content within window as text or non-text. On the other hand, the CC-based methods aim to find candidate connected components for individual characters. Then using heuristic rules tries to cluster the extracted characters into words or text lines. The main difference between the members of this category is in the way of finding the characters, which can be based on the edges, regions energy or Extremal Regions (ER). Since finding the candidate characters is an essential step of these methods, due to cursive nature of Farsi text these methods are not as effective as for Latin. The sliding window based methods are usually more robust to noise; however, their computational complexity is usually higher than the CC-base methods [5].

Ohya et al. [6] used a locally adaptive thresholding method to detect high contrast regions as candidate text regions in grayscale space. Li et al. [7] after quantizing the input image applied a threshold to detect candidate regions for texts. Then, candidate regions are grouped using simple alignment rules. Both of these methods assume that text regions are horizontal and also these regions have a good contrast with the background. Kim et al. [8] simultaneously used color uniformity, edge and color variance to detect text regions. Takahashi et al. [9] used a graph-based method in which graph nodes and edges between nodes represent characters and pair dependencies respectively. This method is not robust to intensity changes and also it is vulnerable to complex backgrounds. Also, finding suitable weights for the edges is a challenging problem. Pan et al. [10], [11] generate a text confidence map on a pyramid of the input image using Wald-boost [12] and Histogram of Oriented Gradients (HOG) [13]. Text candidate regions are extracted using Niblack [14] binarization method. Afterward, text and non-text classification is realized using the Conditional Random Field (CRF) [15] and confidence map. The time complexity of this method is high due to the nature of CRF and the algorithm of generating text confidence map.

Epshtein et al. [16] introduce the stroke width transform (SWT) to calculate stroke width of the pixels of the input image. Subsequently region growing and heuristic rules are used to form character candidates and eliminate non-text regions. This method has been improved in [17] and [18]. The work addressed in [19] is another stroke based method which is based on an efficient comparing between intensity of a pixel with its surrounding pixels. This method is more accurate and faster than previous methods.

Clustering is another method which is frequently used for realizing candidate regions. Ghoshal et. al. [20] employ fuzzy clustering in a feature space using normalized RGB value of pixels, intensity and edge variance. Among other clustering-based methods, [21] and [22] can be cited. Coates et al. [23] use K-means as an unsupervised feature learning method. In the next step, these features are employed to train an SVM classifier. This classifier is used to detect characters in different scales using a sliding window.

Yao et. al. [24] used a modified version of Otsu's binarization method [25] for detecting text candidate regions. Subsequently structural features are adopted to eliminate non-text regions. Fabrizio et. al. [26] introduce a new operator called TMMS (Toggle Mapping Morphological Segmentation). As the name of this method implies, this operator is used to segment input image to several regions. In the next step, the non-text regions are removed using a binary classifier. Then the remaining

regions are grouped to form text regions. At last, a validation step is employed to remove false positives for text. Lu et al. [27] use edge components to locate text regions. In this method for each edge component, three features are extracted to eliminate non-text components. These features include text-specific contrast, stroke-based shape structure and edge cut counter. This method constructs a pyramid on top of input image to ensure high recall ratio. At the end, a combination of HOG and BOW (Bag of Words) are utilized for further pruning of non-text regions. In this method background of text regions need to be relatively smooth, as a result of employing edge components.

Maximally Stable Extremal Regions (MSER) is another well-known text region detector. An extremal region is a connected component, which its intensity is higher or lower than its surrounding pixels [28], [29]. Shi et al. [30] use MSER regions to find text candidate regions. These regions are considered as nodes of a graph and a loss function is defined for assigning text or non-text labels to these nodes. Due to the limitation of employed heuristic rules, this method can only detect horizontal text lines. Li et al. [31] proposed a very similar method where the major difference between these two methods is the employed features. Yin et al. [32] use MSER technique as well, but after extracting candidate regions a penalty is imposed to extract only the regions whose stability value exceed a predefined threshold. Hence, in the resulting MSER tree, only regions with significant stability are retained. After non-text elimination process, Single-Linkage [33] along with structural features, such as color and width to height ratio, are adopted to assemble text lines. This method is improved by [34] to handle multi-orientated text lines as well as horizontal ones. A Multi-Channel Multi-Resolution MSER (MC-MR MSER) is also proposed by [35]. Another multi-channel version of MSER is employed by [36]. Extremal Regions (ER) is a similar approach which has been used in several works such as [37], [3], [38], and [39]. ER regions is robust against image blur and low contrast. It is also robust to variations on illumination, color and texture.

Gin et al. [40] propose a texture-based method. Several features such as intensity mean, intensity variance and histogram of intensity in predefined blocks are utilized to train an Ada-boost classifier [41]. The time complexity of this method is high and it needs manual text region cropping for the training stage of the algorithm. Pan et al [42] improve this method by employing HOG and Multi-Scale LBP [43]. Another improvement over this method was proposed by [44]. They used more elaborate features for training, at the cost of increasing time complexity. A new version of HOG, called T-HOG, was introduced by [45] to detect text in natural scene images. In this new version, cell boundaries are realized using soft borders.

Darab et al. [46] introduce one of the first Persian text localization methods. At first, an adaptive version of Sobel's edge detection is introduced to extract edge pixels. Afterwards, the morphological dilation is applied on vertical and horizontal edges. For text regions these dilated edges are assumed to be connected. This method is applied on an image pyramid to extract text candidate regions with different scales. At last, the aforementioned approach is combined with a color clustering-based method to improve the results. It's worth mentioning that this approach is unable to detect text on complex backgrounds.

Shao et. al. [47] introduce a new text region detector called SWSR (stable Width Stroke Regions). This detector is based on the assumption that text regions have closed boundaries and uniform stroke widths. It is used as a filtering mechanism for pruning MSER regions. At the last step, mean shift clustering method with structural features is employed to form text lines. This method is unable to localize text regions within the border of the input image. It is also vulnerable to large amount of repetitive objects like bricks or windows. This method is improved in [48] using LRMR (Low Rank Matrix Recovery) and fuzzy inference system.

Because of Farsi's cursive nature, some of the most important heuristics used for Latin text such as aspect ratio of character candidates, are not effective for Farsi. Most of the aforementioned

color-based methods suffer from a common problem: all need to know the number of clusters before running clustering algorithm; however finding the number of clusters is not an easy task.

In this paper, we propose a color based approach for text localization that does not rely on clustering. We introduce the idea of color signature for each edge extracted from image. Then image edges are clustered in spatial-color space to construct edge components candidate for text regions. The idea relies on the assumption that characters in a text line usually share the same foreground or background color which most of the times is a valid assumption. The image within the bounding box of each candidate edge component is given to a MLP based cascade classifier to extract text regions.

Both Edge Color Signature (ECS) and our previous method, called Edge Color Transform (ECT) [49], use edge pixels to locate text regions, however there are several major differences between these two methods; first, ECT relies on gradient directions to assign colors to edge pixels whereas ECS employs the mean-shift algorithm to find the colors surrounding the edge pixels; second, for each input image, ECT produces two edge color maps which is one in the case of ECS; third, while ECS is based on the single-linkage clustering, ECT uses the region growing algorithm for creating groups of edge pixels. fourth, unlike ECT, prior to color assigning process, several geometrical relations are utilized by ECS to prune some of the non-text edge pixels. finally, while ECT uses a color distance metric, the distance metric of ECS algorithm is a combination of color and positional metrics.

3. The mean shift algorithm

Mean-Shift is a non-parametric mode-seeking algorithm that also can be used as a clustering method. As opposed to other clustering algorithms such as k-means, in this method [50] the number of clusters does not have to be known in advance. Given n data samples $x_i, i = 1 \dots n$ in d dimensional R^d space, estimating the probably density function with kernel K and bandwidth h is performed as:

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{n}\right) \tag{3.1}$$

and symmetric kernels are defined as:

$$K(x) = c_{k,d} k\left(||x||^2\right) \tag{3.2}$$

where $c_{k,d}$ is a normalization factor. Assuming that K is differentiable, the modes of the function f are located at the zeros of the gradient of the function f that is equal to [51]:

$$\begin{aligned} \nabla f(x) &= \left(\frac{2c_{k,d}}{nh^{d+2}}\right) \sum_{i=1}^{n} (x_i - x) g\left(\left|\left|\frac{x - x_i}{h}\right|\right|^2\right) \\ &= \left(\frac{2c_{k,d}}{nh^{d+2}}\right) \left(\sum_{i=1}^{n} g\left(\left|\left|\frac{x - x_i}{h}\right|\right|^2\right)\right) \left(\frac{\sum_{i=1}^{n} x_i g\left(\left|\left|\frac{x-x_i}{h}\right|\right|^2\right)}{\sum_{i=1}^{n} g\left(\left|\left|\frac{x-x_i}{h}\right|\right|^2\right)}\right) \end{aligned} \tag{3.3}$$

where $g(s) = -k'(s)$. The first term of equation (3.3) is proportional to density function using $G(x) = c_{k,d} g(x^2)$, and the second term is called the mean-shift vector:

$$m_h(x) = \frac{\sum_{i=1}^{n} x_i g\left(\left|\left|\frac{x-x_i}{h}\right|\right|^2\right)}{\sum_{i=1}^{n} g\left(\left|\left|\frac{x-x_i}{h}\right|\right|^2\right)} - x \tag{3.4}$$

The mean-shift vector always points at the direction of the maximum increase in density function [52]. The mean-Shift algorithm is realized through the iterative applications of the following equation:

$$x^{t+1} = x^t + m_h(x^t) \tag{3.5}$$

Eventually, the point x will converge to a mode of density function where the gradient of the density function is equal to zero. The mean-shift clustering algorithm is a practical application of the mean-shift mode-seeking algorithm in several random points in feature space. Application of the algorithm for each random point will converge to a mode. These modes are considered as the center of the clusters, and the number of clusters is equal to the number of modes.

4. The Proposed method

Since texts in natural scenes appear in such a way that attract human beings attention, texts regions usually have strong edges with good contrast with background. According to our observations on images on texts, there are usually three dominant colors around an edge pixel of text regions: color of text (foreground), color of background and color of the edge border, which is a mixture of foreground and background colors. We consider the set of dominant colors surrounding an edge as the color signature of the edge. As the color signature is almost the same for edge pixels of a text line, we use it as a cue for clustering similar edge pixels into text candidate regions. Figure 2 shows the flowchart of the proposed method.

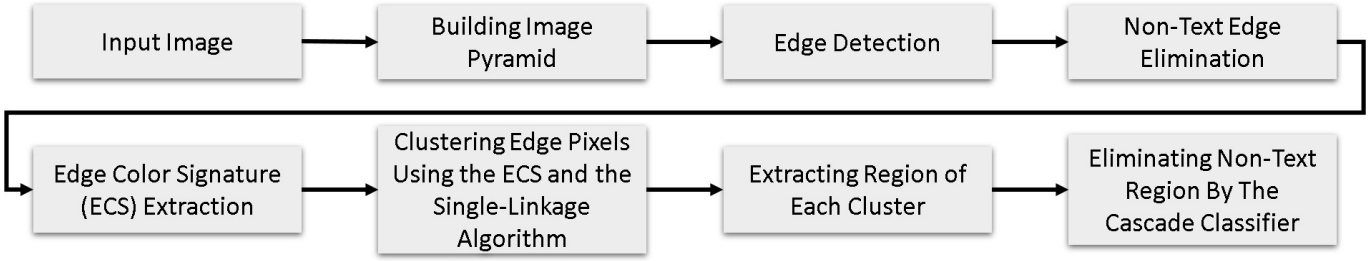


Figure 2: Flowchart of the proposed method

4.1. Creating a pyramid of the input image

An image pyramid using different scales $(0.2, 0.4, 0.6, 0.8, 1)$ of the input image is created in this step of the algorithm, and all of the later steps are applied on each scale of this pyramid. The reasons for using different scales of the input image are twofold: first, the multi-resolution representation of the input image in the pyramid enables the method to detect text regions with different font sizes; and secondly, text edges extracted from lower levels of pyramid are more suitable, in case that the input image is blurred or has low contrast. This phenomenon has been shown in Figure 3. As it can be seen from Figure 3, the extracted edge map for the downscaled image is stronger than the edge map of the original one.

4.2. Edge map extraction

In the next step of the algorithm, the edge map of the input image is extracted using the Sobel method. We noticed that for extracting text edges, the Sobel operator performs better than the Canny operator; As can be seen from Figure 4, the Sobel operator produces less non-text edge pixels compared to the Canny method.

(a) (b) (c) (d)

Figure 3: The effect of downscaling on edge map quality of blured images, a) a blurred text image, b) downscaled version of a, c) edge map of the original image, d) edge map of the downscaled image

(a) (b) (c)

Figure 4: Comparison of the Canny and the Sobel methods for edge extraction of text regions, a) The input Image, b) Edge map of Canny method, c) Edge map of Sobel method

4.3. Filtering the non-text edge pixels

Considering that connected edge pixels of text regions has moderate complexity, for each edge connected component the ratio between the width and height of its bounding box must be in a certain range. Furthermore, our observations show that the ratio between the number of the edge pixels of text regions to the area of their bounding boxes is bounded to a certain rage. In this step of the proposed method, we set a number of rules to remove some of the non-text edge pixels from the extracted edges. First, for each component of connected edges, the following conditions are checked and if a component does not satisfy one of these conditions, it is removed from the edge map.

$$\alpha_1 \leq \frac{w}{h} \leq \alpha_2 \tag{4.1}$$

$$\frac{f}{ma} \leq \alpha_3 \tag{4.2}$$

$$\alpha_4 \leq \frac{area}{w \times h} \leq \alpha_5 \tag{4.3}$$

$$\frac{area}{ma} \leq \alpha_6 \tag{4.4}$$

where w, h are the width and height for the bounding box of the connected component respectively, $area$ is the number of the pixels in the connected component, ma is the length of the main diameter and f is the ratio of the distance between the foci of the ellipse that has the same second-moments

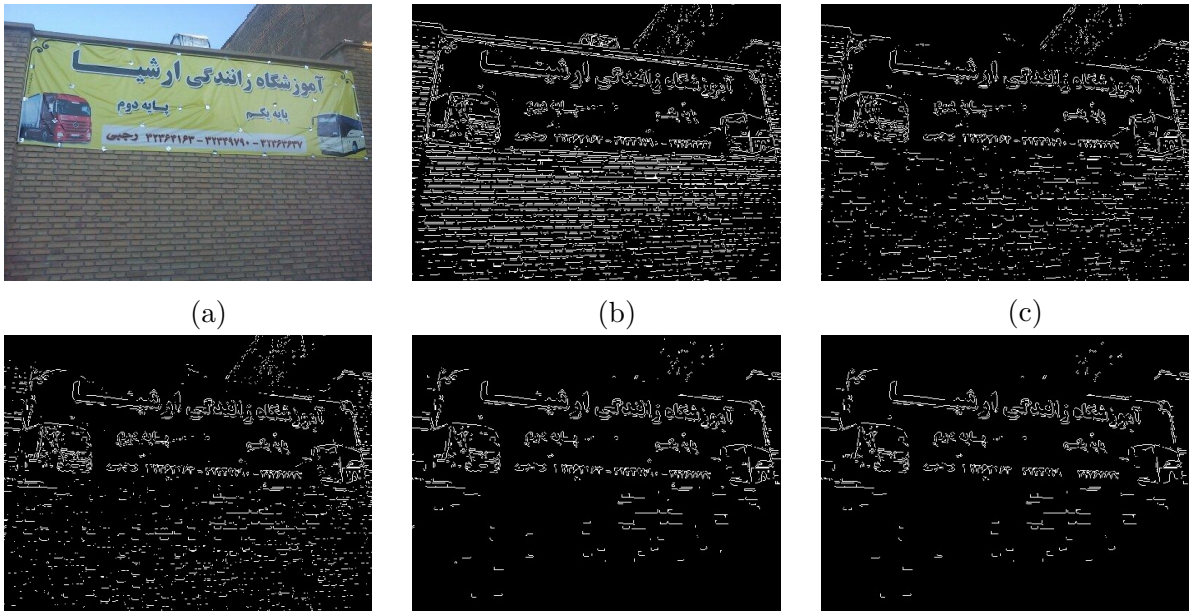


Figure 5: The effect of applying geometrical rules on a sample image, a) the input image, b) the edge map of the input image, c) The effect of applying relations (4.1) on the edge map, d) The effect of applying relations (4.2) on the edge map of the part b, e) The effect of applying relations (4.3) on the edge map of the part c, f) The effect of applying relations (4.4) on the edge map of the part e

as the region and its major axis length. The orientation and elongation of the ellipse are determined from eigen-vectors and eigen-values of the covariance matrix of coordinates for pixels on the connected component. Finally, the set of parameters $\alpha_i, i = 1 \ldots 6$ is empirically set.

The constraints (4.1) and (4.2) examine that whether the relative length and elongation of an edge connected component is within the typical range for a text edge component. The ratio (4.2) is within the range [0, 1], larger value for this ratio indicates edge components is more elongated. In fact, constraint (4.1) and (4.1) are utilized for the same purpose, except that constraint (4.2) is able to indicate elongated edge components whereas (4.1) cannot detect this. Furthermore, congestion of the edge pixels in the edge component bounding box is examined using constraint (4.3). Very crowded or sparse edge components are removed using (4.3). Finally, constraint (4.4) inspects the uniformity of each connected component. The effect of each constraint has been shown on a sample image in Figure 5.

4.4. Extraction of edge color signature

For each of the edge pixels from the edge map that passed the previous stage, a descriptor which we call *colorsignature* is extracted in this step. We define the color signature of an edge pixel as the set of dominant colors surrounding the pixel in RGB space. To extract these colors, the mean-shift algorithm is performed on an $n \times n$ neighborhood of each edge pixel in RGB feature space. As stated before, the mean-shift algorithm is able to find modes of density function without knowing the number of density modes in feature space beforehand. The output of the mean-shift algorithm is a number of clusters; after extraction of each cluster, if a cluster contains less than 10 percent of the $n \times n$ population is ignored. The mode of the dominant clusters for each edge pixel constitute a set of colors which we refer to it as the signature of edge pixels.

4.5. clustering of the edge pixels by their color signatures

In the next step of the algorithm, the pixels in the edge map are clustered by single-linkage [53] algorithm according to their spatial position and their signatures. Single-Linkage is a hierarchical clustering algorithm; this algorithm initially starts with the number of clusters equal to the number of the data points. In each step of this algorithm, the two closest clusters are merged. The distance between two clusters is measured as the distance between closest pair of data points from two clusters. With a suitable distance metric, this algorithm is able to chain edge pixels belonging to the text regions together and form text lines. We define the distance between two edge pixels p and q as follows:

$$d(p,q) = \sum_{i=1}^{c_p} \min_{j=1\ldots c_q} d_{ij}(p,q) + \beta \times ||pos_p - pos_q|| \tag{4.5}$$

$$d_{ij}(p,q) = \sum_{k\in\{r,g,b\}} |k_{p_i} - k_{q_i}| \tag{4.6}$$

$$||pos_p - pos_q|| = |p_x - q_x| + |p_y - q_y|^2 \tag{4.7}$$

where c_x is the number of colors for point x, pos_x is a 2×1 vector which represents the spatial position of x, β is a parameter which controls the relative importance of color signature and spatial distance of edge pixels in clustering, r, g and b are color channels and k_{x_i} is the k channel of cluster center (i.e. color) i for point x. As it is indicated by equation (4.5), because of elimination of smaller clusters and the essence of the mean-shift algorithm, the number and the order of the colors for two different pixels may differ; therefore for calculating color distance between two pixels min operator is employed. Also, $|x|$ indicates the absolute value of x. The power 2 in the equation (4.7) prevents adjacent text lines from merging together. As it can be observed from equation (4.5) both color and spatial distance contribute for overall distance of edge pixels; However, if the color distance between two points was bigger than a predefined threshold, C_T, the corresponding clusters would not merge. Also, the nearest clusters in the Single-Linkage algorithm merge until the minimum distance between the clusters is bigger than a predefined threshold T. Finally, clusters that contain less than 20 points are eliminated and remaining cluster are considered as candidate text edges.

4.6. Extraction of the candidate regions

In this step of the proposed method, a minimal bounding box is fitted to the edge points of each edge cluster. Then, each bounding box is rotated to align horizontally. Afterward, the region within each candidate bounding box is fed to a cascade classifier to extract text regions.

4.7. The Cascade Classifier

A cascade classifier is trained for classification of candidate text regions. Cascade classifiers have been proven to be fast and reliable in many applications such as object detection and face localization [54], [55]. The cascade classifier in the proposed method consists of five Multi-Layer Perceptrons (MLP). The difference between the MLPs of the cascade chain is in the number of hidden layer neurons and the input features. The classifier at each stage of the cascade is more sophisticated than the previous stages regarding the features used and their complexity. Therefore, the number of neurons at hidden layers are increased from the first to last stage in the cascade chain. Accordingly, we designed a cascade classifier in which the first MLP has 10 hidden layer neurons and for each following stages 10 neurons are added respect to its previous stage. Advantages of this design are twofold: most of the non-text regions can be discarded using the simple features of the early stages

of the cascade classifier, and more sophisticated features are only needed to be used in the last stages of the classifier. The features of each step of the cascade classifier are delineated in Figure 6.

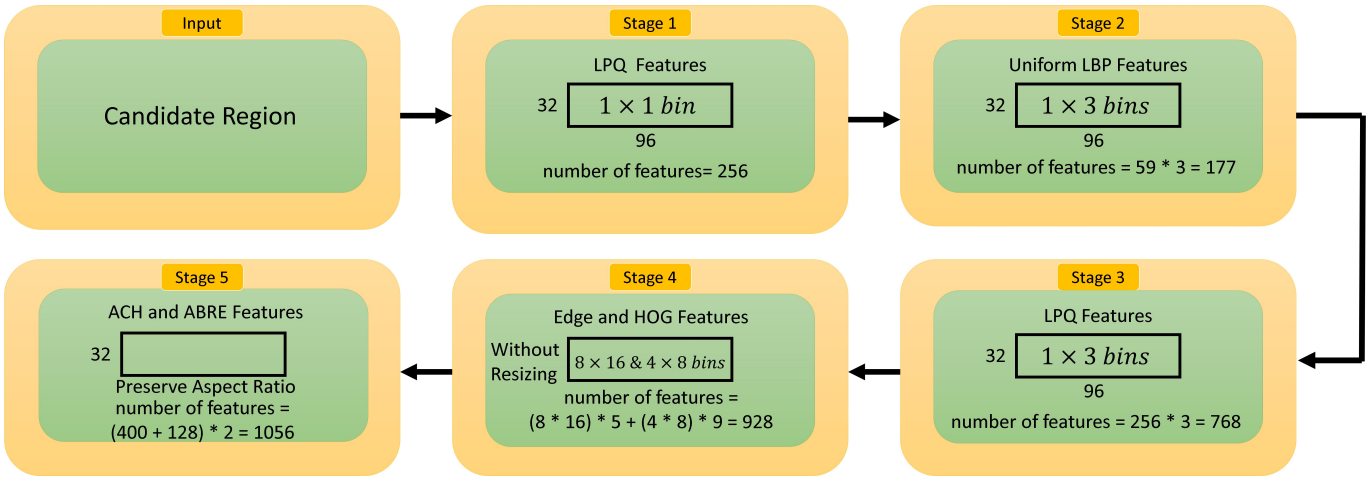


Figure 6: Details of the cascade classifier

In the first three stages of the cascade classifier, a candidate image region is resized to the fixed size 32×96 to decrease time complexity of these stages, however in the last two stages another approach is utilized to preserve more information about the candidate regions. Local Phase Quantization (LPQ) [56] is used as feature extraction algorithm for the first stage of the cascade classifier. LPQ is the result of the quantization of the discrete Fourier transform of the candidate image region. These features are easy to extract and discriminative even for blurred images [57]. In the next stage, the candidate region is divided into non-overlapping $[32 \times 32]$ regions and uniform Local Binary Pattern (LBP) [58] features are extracted from each of these sub-regions. The third stage is similar to the second stage except that it uses LPQ features instead of LBP.

In the fourth stage, Histogram of Oriented Gradient (HOG) [13] and a modified version of features introduced in [59], which we call edge features are employed for classification. For extracting HOG features the candidate region is divided into 4×8 non-overlapping regions. For extracting edge features, the magnitude and the orientation of gradient in the location of edge pixels are calculated in the intensity channel of $L * a * b$ color space [60]. For extracting edge features the candidate region is divided into 8×16 non-overlapping sub-regions and the following features are extracted from each sub-regions.

$$f_1^{ij} = \frac{avg(G_e^i j)}{std(G_e) + \varepsilon} \tag{4.8}$$

where $avg(G_e^{ij})$ is the average value of gradient magnitude in the sub-region located at row i and column j. In addition, $std(G_e)$ is the standard deviation of gradient magnitude of the candidate region and ε is a small value that is used to prevent division by zero.

The next two features are calculated using vertical, v_{prj}, and horizontal, h_{prj}, projection of the edge pixels as follows:

$$f_2^{ij} = \frac{num(v_{prj} \neq 0) + num(h_{prj} \neq 0)}{len(v_{prj}) + len(h_{prj})} \tag{4.9}$$

$$f_{\alpha+1}^{ij} = \frac{num(v_{prj} = \alpha k) + num(h_{prj} = \alpha k)}{len(v_{prj}) + len(h_{prj})}, \alpha \in \{2, 3, 5\}, k = 1, 2, \ldots \tag{4.10}$$

where $num(.)$ is the number of the elements of the input vector that satisfies the input condition, $len(.)$ is the number of the elements in the input vector, f_2 is the relative number of non-zero elements in both vertical and horizontal projections. Finally, the last feature counts multiples of two, three and five. These two features examine the distribution of the edge pixels in the region. Features f_2 is used to distinguish between zero and non-zero values and the last feature captures the overall structure of the projections. As the candidate regions are not resized for calculating these features, any arbitrary value can appear as the result of summation. Therefore, multiples of two, three and five are counted instead of pure summation. These numbers have been chosen empirically to maximize the classification performance.

In the last stage of the cascade classifier, the candidate region is resized to the height of 32 but the aspect ratio is preserved to maintain overall structure of the input region. Afterward, Discriminative Features-oriented Dictionary Learning (DFDL) [61] is employed for feature extraction from the candidate region. Extracting dense Scale Invariant Feature Transform (d-SIFT) [62] is the first step of DFDL. Then, a dictionary is learned for each class. In the test phase, d-SIFT descriptors are extracted from the candidate region. Finally, these descriptors are estimated using both text and non-text dictionaries. The sum of the estimated coefficients for the atoms of both dictionaries are the first features set of this stage. We call these features Atom Coefficients Histogram (ACH). Sum of the Absolute Bin Reconstruction Errors (ABRE) are the next set of features. Error values are calculated for each of 128 elements of the SIFT descriptor. Therefore, 128 features for text dictionary and 128 features for the non-text dictionary are obtained.

The cascade classifier is trained using a set of text and non-text images. For non-text samples a selection of images from the web was used. As the number of cropped text regions is very small, we synthesize text regions. The algorithm for synthesizing text regions has been described in the next session. For training each stage of the cascade classifier, the same number of text and non-text regions have been used. Non-text regions feeds to each stage are those which are misclassified by the previous stage.

4.8. Algorihtm of synthetic text region generatio

In order to train a cascade classifier properly, a sufficient number of training samples are required. Providing a sufficient number of non-text samples is a rather easy problem; on the other hand, preparing significant number of cropped text regions is time consuming. To extend the cropped text samples, inspired by the work [63] a method for synthesizing text samples is proposed.

Two colors c_f and c_b are chosen from the training images as foreground and background colors. A background image is selected and randomly cropped from the training database, crop; then color of each pixel in the background image is modified using the selected background color, c_b, through the following equation:

$$p_b(i,j) = \alpha \times crop(i,j) + (1-\alpha) \times c_b \tag{4.11}$$

where α is a parameter of the algorithm, randomly chosen with a uniform distribution in range $[0 \ldots 0.2]$. This process intends to produce a synthetic background image similar to natural scene images.

In the next step, a few words are randomly chosen from a predefined dictionary to add on the synthetic background image. In addition, random numbers are used to generate the numerical text images. The synthetic text images are slightly rotated by a random angle to introduce diversity into training samples. Then, a Gaussian function with random parameters is added to the text image. This function is simulating the flash effect at the night and also the sun reflection on the text regions during the day. Applying a slight motion blur or Gaussian blur is the last step for the synthesizing

Figure 7: Samples of the produced synthetic text images

text images. All of these steps are randomly applied to 10 percent of the images. Samples of the produced text images have been illustrated in Figure 7.

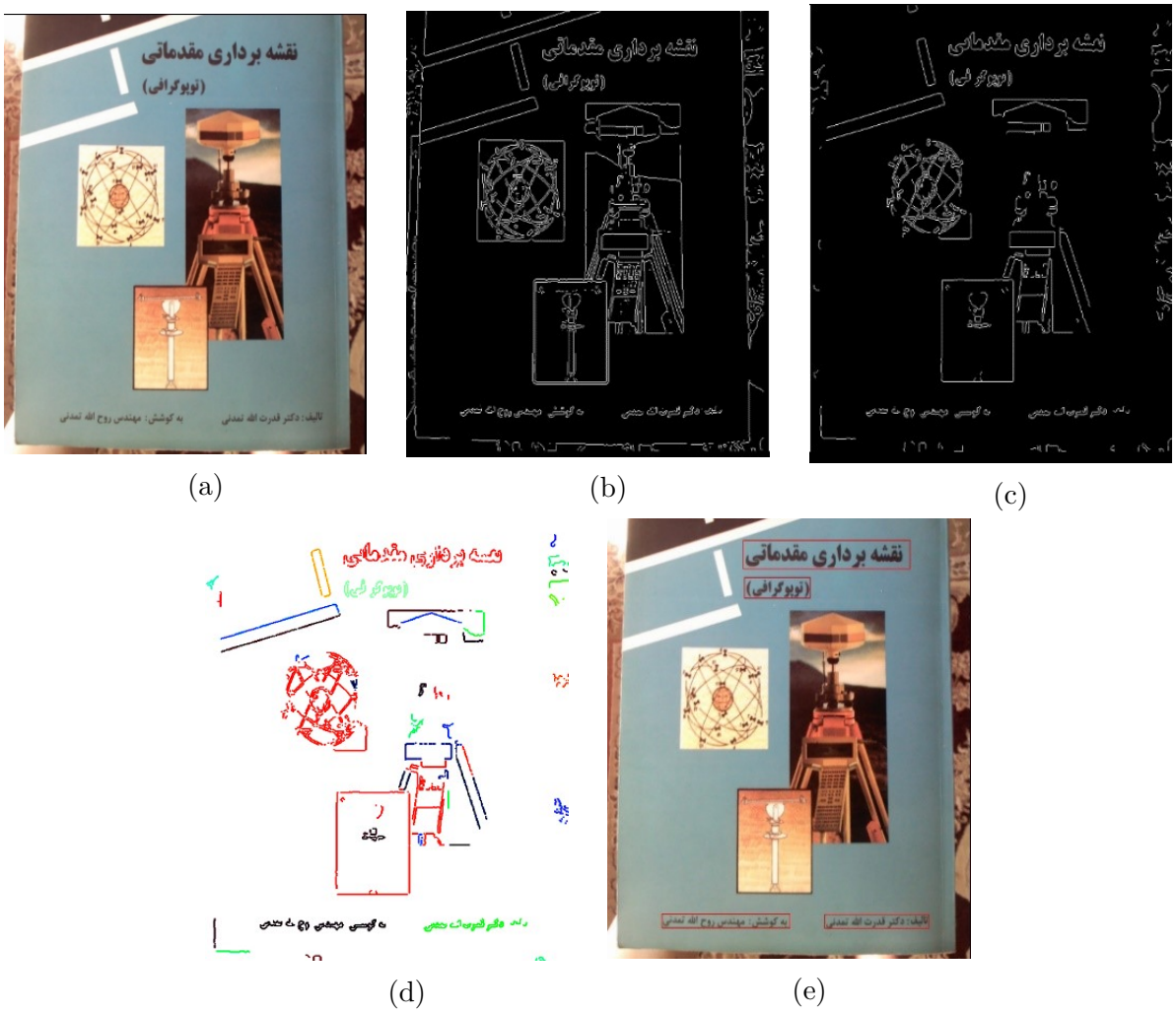
4.9. merging of the detected text regions

The result of text classification using the cascade classifier for each level of pyramid is a number of text regions. In the last step of the proposed method, all detected text regions from different levels of the image pyramid are integrated. Some of these regions contain more than a text line and should be eliminated. We refer these text lines as pseudo-text regions. Pseudo-text regions are detected by calculating the ratio of the height of each edge connected component to the height of the whole region; if the ratio is less than 0.6 for a connected component, the component is eliminated. Subsequently, if the sum of the convex-hull area of the remaining components is than the 50 percent of the whole region, the region is regarded as pseudo-text and eliminated. Successively, two regions are merged, if their overlapping ratio is greater than 0.3 and their height ratio is greater than 0.5. Outputs of the different stages of the proposed method have been shown in Figure 8 for a sample image.

5. Experimental results

In this section we introduce datasets for evaluating the proposed method and compare this method with other methods. The result of experiments are also reported in the section. Two datasets for English and Farsi text images are used for evaluating the proposed method besides other methods. For evaluating the proposed method on English text, the well-known ICDAR 2013 dataset is used. This dataset is composed of 233 train and 229 test images. Regarding Farsi text localization, to the best of our knowledge there is no shared image dataset from natural scene. Therefore, we have collected 450 Farsi natural scene text images, we call it Farset. This dataset is composed of different pictures from different cities of Iran and from other Farsi printed text scenes. These images were taken with different cameras and under different lighting conditions. The size of images varies from 320×240 to 4608×3456. These images contain different text fonts, styles, sizes, colors and backgrounds. A number of image samples in the Farset have been shown in Figure 9.

We adopted the evaluation protocol proposed in [64].This evaluation method is similar to Pascal [65] evaluation method which is based on the overlapped ratio of the detected region and the regions of ground truth. The overlapped area is calculated after aligning both detected and ground truth rectangles horizontally. For each pair of detected and ground truth rectangles, a region indicated by the proposed methods is considered as correct detection if the overlap between it and corresponding rectangle in ground truth is greater than 0.6 and also the difference between direction of these two



(a) (b) (c)

(d) (e)

Figure 8: Outputs of the different stages of the proposed method, a) The input image, b) Edge map, c) Edge map after pruning some of non-text edge pixels, d) The result of the clustering algorithm (each cluster has been shown with a different color), d) The final result

rectangles is less than 30 degrees. Accordingly, precision, recall and F-measure are defined as follows:

$$Precision = \frac{|TP|}{|E|} \tag{5.1}$$

$$Recall = \frac{|TP|}{|T|} \tag{5.2}$$

$$F - measure = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5.3}$$

where TP is the number of the correctly classified regions, and E and T respectively are the number of detected and ground truth regions.

The proposed method was implemented in Matlab. As stated in the previous section, both cropped and synthetic text images is used for the training of the cascade classifier. The cropped samples are obtained from the train set of ICDAR 2013 and Farset datasets. For creating synthetic text samples, Bijankhan corpus [66] is used. This corpus consists of 2.6 million different words from

Figure 9: Samples of the produced synthetic text images

different subjects. For English text localization, a set of 300 thousand words has been used. All of the parameters of the proposed method is chosen to produce best results based on the training sets. The proposed method has been compared to several other methods on ICDAR 2013 dataset and the results have been reported in Table 1. As it can be seen from the table, the proposed method has produced promising results compared to other methods.

In addition, the proposed method has been compared to [3] and [67] on Farset dataset. These are two well-known methods that their source codes are shared. Since in most of the methods, the exact values of some parameters are unknown, we abstained from self-implementing any other method. The result of comparing the performance of the proposed method with the mentioned methods have been reported in Table 2. Comparison results indicate that the proposed method has produced competitive results for both Farsi and English text localization. A number of image samples as the output of the proposed method have been reported in Figure 10 and Figure Figure 11.

6. Conclusion

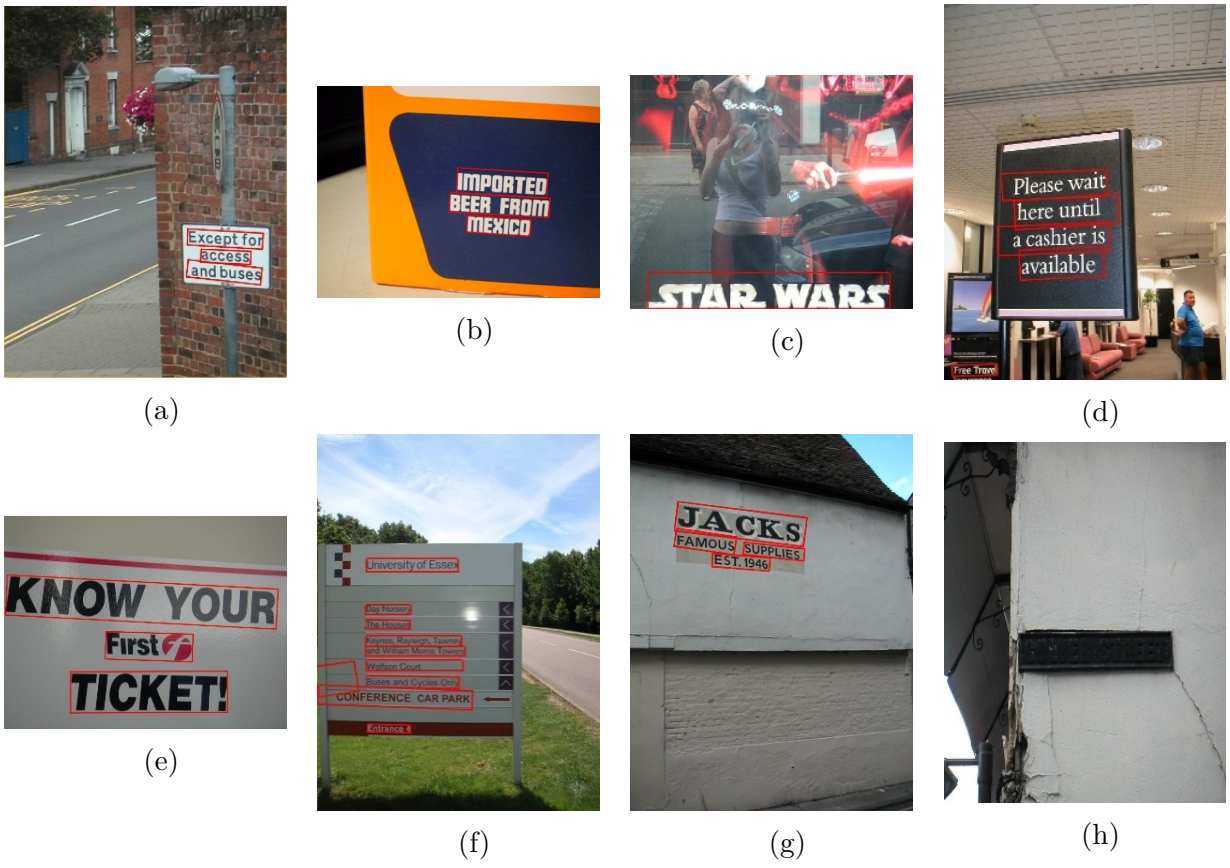
A new natural scene text localization method was proposed in this paper. The proposed method is able to detect both Farsi and English texts. Our method starts by creating a pyramid by different scales of the input images. Then edge map of the input image is extracted and some of the non-text edge pixels are removed by the means of shape features. Afterward, for each edge pixel, edge color signature (ECS) is extracted. By employing ECS and spatial position of each edge pixel as features, the Single-Linkage clustering algorithm is used to group edge pixels to from candidate text regions. Finally, a cascade classifier, which utilizes LPQ, LBP, Edge, ABRE and ACH features, is applied to eliminate non-text candidate regions. Since the proposed method makes no assumption about the

Table 1: Comparison results of the proposed method with several methods on ICDAR 2013 dataset

Method	Precision(%)	Recall(%)	F-Measure
The proposed method	90.1	75.46	82.14
Lu [59]	88.55	69.58	78.19
USTB TexStar [34]	88.47	66.45	75.89
Text Spotter [37]	87.51	64.84	74.49
Text Detector CASIA [30]	84.70	62.45	72.16
Text Detection [26]	74.15	53.42	62.10
Ghanei [48]	88.55	66.60	76.02
FASText [19]	84.00	69.30	76.80
Zheng [39]	89.50	77.63	83.14
Neuman(b) [38]	82.10	71.30	76.30

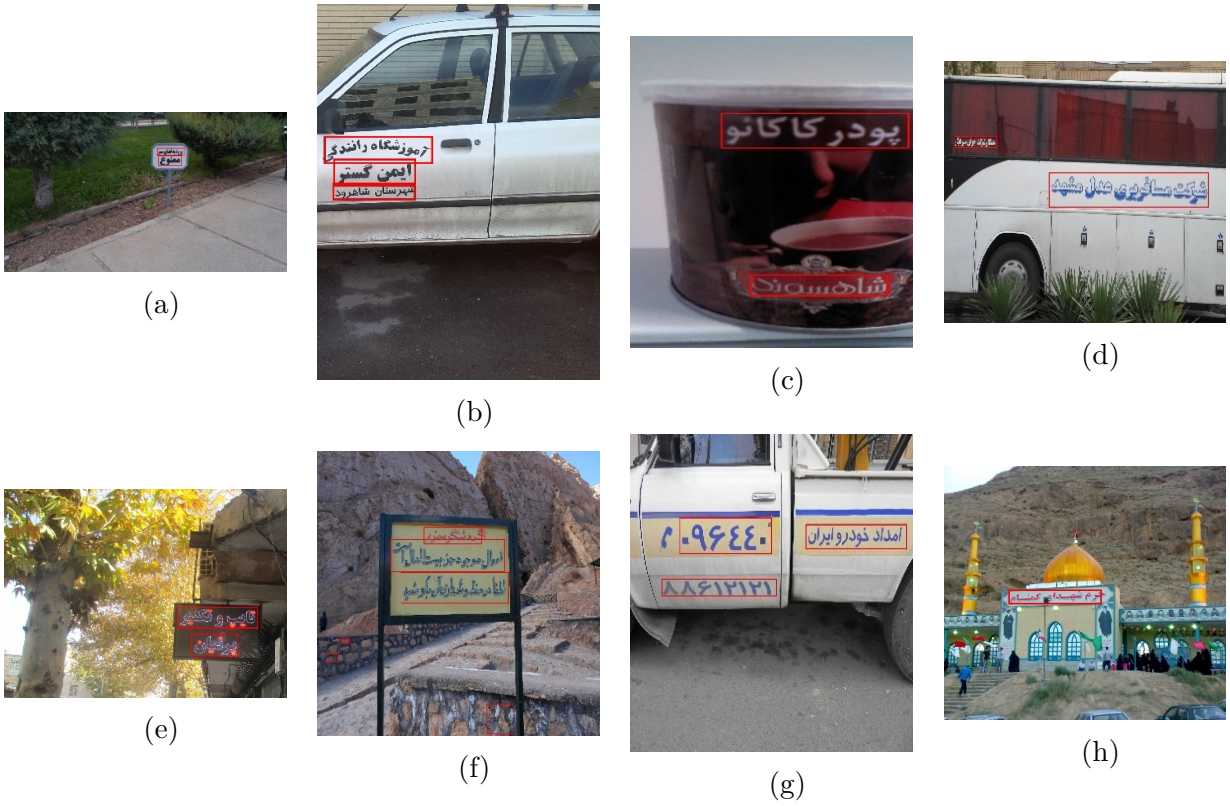
Table 2: Comparison of the proposed method with two other methods on Farset dataset

Method	Precision(%)	Recall(%)	F-Measure
Proposed method	90.84	87.54	89.16
Characterness [31]	63.56	47.85	54.60
Neuman(c) [3]	48.11	41.62	44.63



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 10: Samples of the detection results of the proposed method on ICDAR 2013 dataset



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 11: Samples of the detection results of the proposed method on Farset dataset

structure of the detected language, it is able to produce promising results on both Frasi and English datasets.

References

[1] H. Zhang, K. Zhao, Y. Z. Song, and J. Guo, Text extraction from natural scene image: A survey, Neurocomputing, vol. 122, pp. 310-323, Dec. 2013.

[2] L. Neumann and J. Matas, A Method for Text Localization and Recognition in Real-World Images, in Computer Vision - ACCV 2010, 2010, pp. 770-783.

[3] L. Neumann and J. Matas, Real-time scene text localization and recognition, in 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3538-3545.

[4] L. Neumann, Text Spotter, 2015. [Online]. Available: http://www.textspotter.org.

[5] X. Chen and A. L. Yuille, Detecting and reading text in natural scenes, in Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, 2004, vol. 2, pp. 366-373.

[6] J. Ohya, A. Shio, and S. Akamatsu, Recognizing characters in scene images, IEEE Trans. Pattern Anal. Mach. Intell., vol. 16, no. 2, pp. 214-220, Feb. 1994.

[7] C. Li, Automatic Text Location in Natural Scene Images, in Proceedings of the Sixth International Conference on Document Analysis and Recognition, Washington, DC, USA, 2001, pp. 1069-1073.

[8] K. C. Kim et al., Scene text extraction in natural scene images using hierarchical feature combining and verification, in Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, 2004, vol. 2, pp. 679-682.

[9] H. Takahashi and M. Nakajima, Region graph based text extraction from outdoor images, in Third International Conference on Information Technology and Applications (ICITA05), 2005, vol. 1, pp. 680-685.

[10] Y. F. Pan, X. Hou, and C. L. Liu, Text Localization in Natural Scene Images Based on Conditional Random Field, 2009, pp. 6-10.

[11] Y. F. Pan, X. Hou, and C. L. Liu, A Hybrid Approach to Detect and Localize Texts in Natural Scene Images, IEEE Trans. Image Process., vol. 20, no. 3, pp. 800-813, Mar. 2011.
[12] J. Sochman and J. Matas, Waldboost-learning for time constrained sequential detection, in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05), 2005, vol. 2, pp. 150-156.
[13] N. Dalal and B. Triggs, Histograms of oriented gradients for human detection, in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05), 2005, vol. 1, pp. 886-893.
[14] W. Niblack, An introduction to digital image processing. Birkeroed, Denmark: Strandberg Publishing Company, 1985.
[15] J. Lafferty, A. McCallum, and F. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in Proceedings of the eighteenth international conference on machine learning, ICML, 2001, vol. 1, pp. 282-289.
[16] 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, 2010, pp. 2963-2970.
[17] Y. Li and H. Lu, Scene text detection via stroke width, in 21st International Conference on Pattern Recognition (ICPR), 2012, pp. 681-684.
[18] A. Risnumawan, P. Shivakumara, C. S. Chan, and C. L. Tan, A robust arbitrary text detection system for natural scene images, Expert Syst. Appl., vol. 41, no. 18, pp. 8027-8048, 2014.
[19] M. Buta, L. Neumann, and J. Matas, FASText: Efficient Unconstrained Scene Text Detector, in 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1206-1214.
[20] R. Ghoshal and B. Dhara, Text Extraction from Scene Images through Color Image Segmentation and Statistical Distributions, Int. J. Comput. Appl., vol. 91, no. 9, pp. 7-10, 2014.
[21] J. Yan and X. Gao, Detection and recognition of text superimposed in images base on layered method, Neurocomputing, vol. 134, pp. 3-14, 2014.
[22] X. Liu and W. Wang, An effective graph-cut scene text localization with embedded text segmentation, Multimed. Tools Appl., Jan. 2014.
[23] A. Coates et al., Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning, Int. Conf. Doc. Anal. Recognit., pp. 440-445, 2011.
[24] J. L. Yao, Y. Q. Wang, L. Bin Weng, and Y. P. Yang, Locating text based on connected component and SVM, Proc. Int. Conf. Wavelet Anal. Pattern Recognit., vol. 3, pp. 1418-1423, 2008.
[25] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern., vol. 9, no. 1, pp. 62-66, 1979.
[26] J. Fabrizio, B. Marcotegui, and M. Cord, Text detection in street level images, Pattern Anal. Appl., vol. 16, no. 4, pp. 519-533, 2013.
[27] S. Lu, T. Chen, S. Tian, J. L. C. Tan, J. H. Lim, and C. L. Tan, Scene text extraction based on edges and support vector regression, Int. J. Doc. Anal. Recognit. IJDAR, vol. 18, no. 2, pp. 125-135, 2015.
[28] J. Matas et al., Robust wide-baseline stereo from maximally stable extremal regions, Image Vis. Comput., vol. 22, no. 10, pp. 384-393, 2004.
[29] K. Mikolajczyk et al., A comparison of affine region detectors, Int. J. Comput. Vis., vol. 65, no. 1, pp. 43-72, 2005.
[30] C. Shi, C. Wang, B. Xiao, Y. Zhang, and S. Gao, Scene text detection using graph model built upon maximally stable extremal regions, Pattern Recognit. Lett., vol. 34, no. 2, pp. 107-116, 2013.
[31] Y. Li, W. Jia, C. Shen, and A. Van Den Hengel, Characterness: An indicator of text in the wild, IEEE Trans. Image Process., vol. 23, no. 4, pp. 1666-1677, 2014.
[32] X. C. Yin, X. Yin, K. Huang, and H. W. Hao, Robust Text Detection in Natural Scene Images., IEEE Trans. Pattern Anal. Mach. Intell., pp. 1-14, Sep. 2013.
[33] A. K. Jain, M. N. Murty, and P. J. Flynn, Data clustering: a review, ACM Comput. Surv., vol. 31, no. 3, pp. 264-323, 1999.
[34] X. C. Yin, W. Y. Pei, J. Zhang, and H. W. Hao, Multi-Orientation Scene Text Detection with Adaptive Clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9. pp. 1930-1937, 2015.
[35] C. Tian, Y. Xia, X. Zhang, and X. Gao, Natural scene text detection with MC-MR candidate extraction and coarse-to-fine filtering, Neurocomputing, vol. 260, pp. 112-122, Oct. 2017.
[36] Z. Liu et al., Method for unconstrained text detection in natural scene image, IET Comput. Vis., vol. 11, no. 7, pp. 596-604, Jul. 2017.
[37] L. Neumann and J. Matas, On combining multiple segmentations in scene text recognition, in Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, 2013, no. Icdar, pp. 523-527.
[38] L. Neumann and J. Matas, Real-Time Lexicon-Free Scene Text Localization and Recognition, IEEE Trans.

Pattern Anal. Mach. Intell., vol. 38, no. 9, pp. 1872-1885, Sep. 2016.
[39] Y. Zheng, Q. Li, J. Liu, H. Liu, G. Li, and S. Zhang, A cascaded method for text detection in natural scene images, Neurocomputing, vol. 238, pp. 307-315, May 2017.
[40] A. L. Yuille, Detecting and reading text in natural scenes, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004, vol. 2, pp. 366-373.
[41] R. E. Schapire and Y. Singer, Improved boosting algorithms using confidence-rated predictions, Mach. Learn., vol. 37, no. 3, pp. 297-336, 1999.
[42] Y.-F. Pan, X. Hou, and C.-L. Liu, A Robust System to Detect and Localize Texts in Natural Scene Images, in 8th International Workshop on Document Analysis Systems, 2008, pp. 35-42.
[43] C. Chan, J. Kittler, and K. Messer, Multi-scale local binary pattern histograms for face recognition, Proc. ICB, pp. 809-818, 2007.
[44] J.-J. Lee, P.-H. Lee, S.-W. Lee, A. Yuille, and C. Koch, AdaBoost for Text Detection in Natural Scene, 2011 Int. Conf. Doc. Anal. Recognit., pp. 429-434, Sep. 2011.
[45] 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 Recognit., vol. 46, no. 3, pp. 1078-1090, 2013.
[46] M. Darab and M. Rahmati, A Hybrid Approach to Localize Farsi Text in Natural Scene Images, Procedia Comput. Sci., vol. 13, pp. 171-184, 2012.
[47] S. Ghanei and K. Faez, Robust Localization of Texts in Real-World Images, Int. J. Pattern Recognit. Artif. Intell., vol. 29, no. 07, p. 1555012, Jun. 2015.
[48] S. Ghanei and K. Faez, Localizing scene texts by fuzzy inference systems and low rank matrix recovery model, Comput. Vis. Image Underst., vol. 142, pp. 94-110, 2016.
[49] J. G. Neycharan and A. Ahmadyfard, Edge color transform: a new operator for natural scene text localization, Multimed Tools Appl, vol. 77, no. 6, pp. 7615-7636, Mar. 2018.
[50] Y. Cheng, Mean shift, mode seeking, and clustering, IEEE Trans. Pattern Anal. Mach. Intell., vol. 17, no. 8, pp. 790-799, 1995.
[51] D. Comaniciu and P. Meer, Mean shift: A robust approach toward feature space analysis, IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 5, pp. 603-619, 2002.
[52] Y. Aliyari Ghassabeh, A sufficient condition for the convergence of the mean shift algorithm with Gaussian kernel, J. Multivar. Anal., vol. 135, pp. 1-10, Mar. 2015.
[53] A. K. Jain, M. N. Murty, and P. J. Flynn, Data Clustering: A Review, ACM Comput Surv, vol. 31, no. 3, pp. 264-323, Sep. 1999.
[54] E. Rashedi and H. Nezamabadi-pour, A hierarchical algorithm for vehicle license plate localization, Multimed. Tools Appl., vol. 77, no. 2, pp. 2771-2790, Jan. 2018.
[55] P. Viola and M. Jones, Rapid object detection using a boosted cascade of simple features, in Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, 2001, vol. 1, pp. 511-518.
[56] E. Rahtu, J. Heikkila, V. Ojansivu, and T. Ahonen, Local phase quantization for blur-insensitive image analysis, Image Vis. Comput., vol. 30, no. 8, pp. 501-512, Aug. 2012.
[57] E. Sariyanidi, H. Gunes, and A. Cavallaro, Automatic Analysis of Facial Affect: A Survey of Registration, Representation, and Recognition, IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 6, pp. 1113-1133, Jun. 2015.
[58] C.-H. Chan, J. Kittler, and K. Messer, Multi-scale local binary pattern histograms for face recognition, in International Conference on Biometrics, 2007, pp. 809-818.
[59] S. Lu, T. Chen, S. Tian, J.-H. Lim, and C.-L. Tan, Scene text extraction based on edges and support vector regression, Int. J. Doc. Anal. Recognit. IJDAR, vol. 18, no. 2, pp. 125-135, Jun. 2015.
[60] M. Tkalcic and J. F. Tasic, Colour spaces: perceptual, historical and applicational background, in The IEEE Region 8 EUROCON 2003. Computer as a Tool., 2003, vol. 1, pp. 304-308.
[61] T. H. Vu, H. S. Mousavi, V. Monga, G. Rao, and U. A. Rao, Histopathological image classification using discriminative feature-oriented dictionary learning, IEEE Trans. Med. Imaging, vol. 35, no. 3, pp. 738-751, 2016.
[62] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman, SIFT Flow: Dense Correspondence across Different Scenes, in Computer Vision - ECCV 2008, 2008, pp. 28-42.
[63] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, Reading text in the wild with convolutional neural networks, Int. J. Comput. Vis., vol. 116, no. 1, pp. 1-20, 2016.
[64] S. Ghanei and K. Faez, Localizing scene texts by fuzzy inference systems and low rank matrix recovery model, Comput. Vis. Image Underst., vol. 142, pp. 94-110, Jan. 2016.
[65] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, The Pascal Visual Object Classes

(VOC) Challenge, Int. J. Comput. Vis., vol. 88, no. 2, pp. 303-338, Jun. 2010.
[66] H. Amiri, H. Hojjat, and F. Oroumchian, Investigation on a feasible corpus for Persian POS tagging, in Proceedings of the 12th International CSI Computer Conference (CSICC), 2007.
[67] Yao Li, Wenjing Jia, Chunhua Shen, and A. van den Hengel, Characterness: An Indicator of Text in the Wild, IEEE Trans. Image Process., vol. 23, no. 4, pp. 1666-1677, Apr. 2014