Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 10, No. 4, 2022, 449-466.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

A bilingual text detection in natural images using heuristic and unsupervised learning

Somayye Bayatpour and Mehran Sharghi*

Faculty of Engineering and Technology, Alzahra University, Tehran, Iran.

Article Info

Article History:
Received 27 August 2021
Revised 25 January 2022
Accepted 03 August 2022

DOI:10.22044/jadm.2022.11089.2260

Keywords:
Text Detection, Natural Images, Mean Shift Clustering, Bilingual Text, Heuristic, Unsupervised Learning.

**Corresponding author: msharghi@alzahra.ac.ir (M. Sharghi).*

Abstract

Digital images are being produced in a massive number every day. A component that may exist in digital images is text. Textual information can be extracted and used in a variety of fields. Noise, blur, distortions, occlusion, font variation, alignments, and orientation are among the main challenges for text detection in natural images. Despite many advances in text detection algorithms, there is not yet a single algorithm that addresses all of the above problems successfully. Furthermore, most of the proposed algorithms can only detect horizontal texts, and a very small fraction of them consider the Farsi language. In this paper, a method is proposed for detecting multi-orientated texts in both the Farsi and English languages. We define seven geometric features to distinguish text components from the background, and propose a new contrast enhancement method for text detection algorithms. Our experimental results indicate that the proposed method achieves a high performance in text detection on natural images.

1. Introduction

A large number of digital images are produced daily by various imaging devices. Natural images are digital images taken from real-world scenes with no human manipulation. An element that may be found in natural images is text. Textual information contained in images can be used in different fields including document analysis, pictorial searches, automatic driving, human-machine interactions, intelligent blind assistant, etc. Detecting and recognizing texts in natural images has established an active research field in the recent years, and scientific communities have witnessed tremendous research and advances in this field. However, there are still challenges in this area including noise, blur, distortions, occlusion, font variation, alignments, and orientation, which should be addressed. Despite advances in text detection algorithms, there is still no single algorithm capable of dealing with all the above problems. Furthermore, most of the proposed algorithms can only detect horizontal text, and only a few methods have considered the Farsi language. The pre-processing steps including noise reduction, contrast enhancement, image enhancement, and segmentation have been exploited by the researchers to overcome the problems of noise and blur in images [1-3]. Various image segmentation techniques have also been proposed specifically for text recognition to be applied to localized texts, also reduce the detrimental effects of non-uniform light and digital sensors [4-7].

The text detection process has two main stages: “text localization” and “text extraction and enhancement”. Text localization involves determining the location of text regions in an image. Typically, in order to facilitate recognition, the text extraction phase includes a step to remove the background. This means that the extracted text turns into a binary image before being sent to OCR. Multi-directional strokes are connected in texts of different languages, allowing texts to have a unique and similar texture as well as geometric properties. These characteristics distinguish texts from other elements of images, and can be used to identify them. The Farsi and English text elements

are geometrically similar, which makes it possible to detect them using a similar technique. In this research work, a hybrid method is presented for text detection in natural images, which exploits the connected component analysis and the textural information of the image. In the connected component analysis, we applied some heuristics including MSER and stroke width transform (SWT) plus geometric and relational features thresholding. The method detects multi-directional texts on flat areas, written in straight or curved lines with different fonts and scales in both Farsi and English. In the proposed algorithm, an innovative method is presented for enhancing the contrast of the image to address non-uniform illumination. Also the edges of textual elements are detected using a new method. Our method overcomes the problems caused by noise, blur, distortions, and occlusion by applying various techniques in a preprocessing phase. It eliminates noises using image smoothing techniques, and it removes blur and distortions by performing histogram equalization. The occlusion problem is addressed by running histogram equalization at the pre-processing phase and using the MSER technique at the edge detection phase. Our proposed method is robust against text alignment and orientation. This is because most of the features that are being used are not depending on alignment or orientation. The method also exhibits good performance in detecting text written in different fonts because our features are chosen based on the structures and measures that are common in most printed fonts. In order to distinguish text components, seven new geometric features are used. Text elements with a similar scale and color are grouped using mean-shift clustering. Additionally, we define four relational features for text components, grouping clustered components into text and non-text by thresholding. In summary, the main contributions of this work include:

-Detecting multi-directional and curved text lines in different fonts and scales.

-Detecting texts written in both the Farsi and English Languages.

-Presenting an innovative method for enhancing the contrast of images.

-Presenting a new special method for detection edges of textual elements.

-Introducing seven new geometric and four new relational features for text components.

This paper is organized as what follows. In Section 2, we provide a summary of the related works in the field of text detection in natural images. The proposed method is presented in Section 3 including details of pre-processing, edge-detection, background removal, component clustering, and text extraction. In Section 4, the methodology and experimental results are evaluated. Finally, Section 5 concludes the paper.

2. Previous Works

Unlike textual documents, which usually have a uniform font style, texts within natural images may have different fonts, colors, scales, and rotations. The background in natural images can be very complicated. Elements such as signs, branches, and grass are hard to distinguish from texts, leading easily to errors and false detection. In order to overcome these challenges, a rich body of text detection methods have been proposed. Text detection approaches in natural images are classified into four major categories: texture-based methods, connected component-based methods, hybrid methods, and deep learning-based methods.

2.1. Texture-based methods

Texture-based methods are designed based on the different textures of text regions. This property separates them from other parts of the image. Typically, in texture-based methods, the features of the specified areas are extracted, following which classifier examines the existence of text in them. These methods can handle noise efficiently; however, they are usually compute-intensive and their performance depends on the text alignment [8]. Features used in the texture-based approaches consist of three groups: color, edge and gradient, and texture.

Text detection using color features

Texts usually have uniform colors with high contrast against the background. Chen *et al*. [9] have proposed a method for detecting text using a Gaussian mixture model in RGB color space. It has been found that mean shift clustering in the color layer can reduce the number of colors in the image and background complexity, thus improving the text detection performance [10].

Text detection using gradient and edge features

Edges are reliable features for detecting texts. A text has clear and noticeable edges that are hard to fade with non-uniform illumination. The method proposed by Liu *et al*. [7] segments all possible

text areas using an edge-detection strategy. They obtained gradient and geometric properties of all region's boundaries by analyzing texture and then separating text areas from non-text. Wu *et al*. [11] have used Gaussian's derivative to extract horizontal edges. The aggregation of these edges was used to generate the corresponding blocks of text strings. If there was a "short path" between the two edges, that block was assumed as text. The methods introduced by Liu *et al*. and Ou *et al*. [12, 13] have provided a text detection technique based on the edge features, where the density, thickness, and variance of edge curvature were used to extract text. This method is suitable for a variety of sizes, fonts, rotations, and text layouts.

Text detection using texture features

Characters usually have a specific density. Yousfi *et al*. [14] have exploited the textural features of text for detecting Arabic/Farsi texts. They used local binary patterns to identify text areas as rectangular boxes. Their proposed texture properties were Haar-like features extracted based on the difference between average intensities in the rectangular areas and the integral image technique. In the Aradhya *et al*. work [15], wavelet transform was used to extract texture properties from the image, and then the Gabor filter was applied to segment the resulting image textures into the text and non-text candidate regions. K-means was subsequently used to highlight the candidate text regions, following which text lines were detected with morphological operators. In the H.Goto *et al*. method [16], a method was proposed for text localization using the DCT feature and Fisher discrimination analysis (FDA). Kim *et al*. [17] have proposed a method for detecting text using texture features and SVM. In their method, positive pixels were compiled using a mean shift algorithm to form textual areas. Pan *et al*. [18] have introduced a method that locates the text at a high speed. This method combined learning-based region filtering, boosting classifiers, and a multivariate classifier. Regions of images were grouped into text and non-text via coarse filtering followed by a fine region enhancement. In the enhancement stage, popular texture features in text detections were used for assessment including Gabor, DCT, LBP, and HOG. Imani *et. al.* [57] described word images using 3 textural features extracted from a sliding window divided into 9 horizontal cells. The features were: image intensity, horizontal and vertical components obtained by the Sobel operator.

2.2. Connected component-based methods

These methods first divide the image into its connected components (CCs) and then extract the text candidate components using a variety of techniques. These methods eliminate the non-text components with heuristic rules or training classifiers based on their geometric properties. The textual components are usually segmented through color clustering or edge detection. These methods require low computational resources due to the relatively small number of candidate textual connected components. However, designing such methods is very challenging. This is due to a large number of non-text connected components in natural images. Moreover, a prior knowledge of the possible textual elements in the images is required. Due to the superior performance of connected component approaches to texture-based approaches, the ICDAR competitions in the text locating [19] and image evaluation campaign [20] ranked these approaches as first [21]. Different connected component-based techniques have been proposed by the researchers that are summarized hereafter.

Text detection by training classifier

These methods select appropriate features for text detection and extract them from image patches containing characters in training sets. The classifiers are then trained and used to detect text in natural images. Wang *et al*. [22] have introduced a coarse to fine method to extract large and small texts. Their method separated the colors of images into homogeneous colored layers, analyzing each connected component in every layer through Block Adjacency Graph (BAG). Finally, all the potential characters were identified by heuristic rules. Zhao *et al*. [23] have built a sparse dictionary from training data, and used it to decide whether an area is a text area or not. However, it had limitations in generalizing its dictionary, and was incapable of handling rotations and scale variations. Zhao *et al*. [24] have found text candidate areas in a sequence of video images by corner points and their integration according to their densities and geometric properties. The key-frames for pixels in the text candidate areas motion vector and features of optical flow were extracted, and the text was detected using a decision tree.

Text detection by heuristic

These methods find textual connected components according to their geometric properties or their stroke widths in a variety of ways. Heuristics such as MSER and stroke width transform (SWT) have been used extensively for detecting the textual-connected components. Shivakumara *et al*. [25]

have proposed a method for detecting multi-oriented texts, which extracted the candidate regions of a text through clustering in the Fourier-Laplace domain and dividing the areas into separate CCs using skeletons. This method did not capture characters or words directly but only specified the text blocks. Bouman *et al*. [26] have provided a lightweight method tailored for mobile applications, which localized texts on the boards and road signs. In their method, the homogeneous regions were first found using morphological operators, and those with cavities were separated. In order to find small characters, the algorithm was repeated with smaller block sizes. In the Moradi *et al*. work [27], dealing with detecting Farsi/Arabic text, dilation in different directions was initially performed to create more corners in the letters. The corner coefficient was subsequently defined, and a corner map was created to compute a corner histogram. Finally, the non-text areas were discarded using three features: complexity, max, and the average of the corner histogram.

MSER-based methods

The text elements usually have a high contrast, and they tend to create a homogeneous colored region. MSER defines specific features for the connected components that are highly efficient in the segmentation of text elements. Text discrimination is done by specifying thresholds for these features of CCs. Yin *et al*. [28] have introduced a multi-part clustering algorithm to group the MSER components and detect the multi-oriented texts. The method presented in the Kang *et al*. work [29] considered each MSER element as a graph vertex. Thus the text detection problem turned into a graph partitioning problem. Chen *et al*. [1] have exploited several features for background removal including MSER areas, aspect ratio, size, and edge direction of CCs. The non-text CCs were eliminated by thresholding the standard deviation of stroke width (SW) values of each CC. Finally, the text lines were identified based on the height, SW, solidity, and distance between the connected components.

Stroke Width Transform (SWT)-based methods

These methods assume the text elements as the components with parallel edges. Accordingly, they have a relatively invariable stroke width along the element length. Thus the discriminating criterion is the variance of opposite edge distances in each element. In this regard, Kumar *et al*. [30] have provided a method based on MSER and SWT, and Epshtein *et al*. and Mosleh *et al*. [31, 32] have proposed methods that use SWT to detect text. Mosleh *et al*. [32] have introduced bandlet-based edge detection, which enhances text edges to improve SWT. Also Yao *et al*. [33] have presented a framework for the detection and recognition of text based on MSER and SWT. Huang *et al*. [34] have introduced a new algorithm based on SWT called Stroke Feature Transform (SFT). Their main contribution was to resolve the edge points matching in stroke width transform. The efficiency of detection with SFT in standard datasets was reported superior in comparison to similar methods. However, the method suffers a limitation in that it can only detect a horizontal text.

2.3. Hybrid methods

In these methods, a combination of connected components and texture features analysis is used in the detection of text. Neumann *et al*. [35] have defined each character as a set of oriented strokes and their positions. They modeled each stroke as a response to directional filters in a gradient projection scale space. They also modeled each stroke condition in a matrix by sampling responses. The characters were separated by a known pattern produced by a trained classifier with artificial training data. Mansouri *et al*. [36] have proposed a multi-stage technique to detect the Farsi/Arabic text in natural images. First, CCs were obtained by an MSER algorithm and subscription of MSER regions with canny edge detection. These components were connected using a morphological closing operator, and text lines were discriminated using Hough transform and histogram peaks. Darab *et al*. [37] have also proposed a method for Farsi/Arabic text detection. First, it produced an edge map with a Sobel filter, where the text areas include vertical and horizontal edges. Next, the contextual lines were detected based on their pixel count, the height of the region, and the thresholding aspect ratio. The image pyramid was used to find the lines and letters with abnormal sizes. An SVM was then used to select the true lines. Yin *et al*. [38] have selected text candidate MSER areas using MSER tree pruning and hybrid features. Le *et al*. [39] first segmented the image with mean-shift clustering, and extracted parallel edges with a gradient-based method. The width features of parallel edges were then computed, and all the connected components with many parallel edges were labeled as text by thresholding. Neumann *et al*. [5] have found extremal regions using a method with a two-step classification via Gaussian pyramid and multiplex projection.

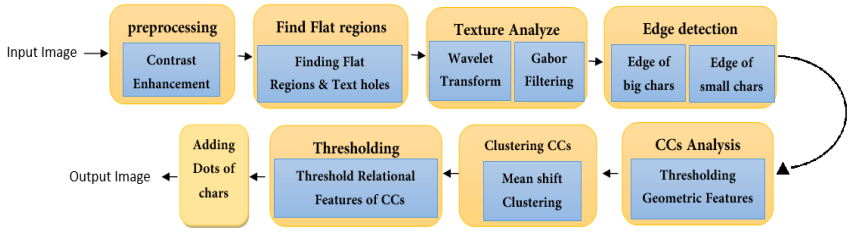

Figure 1. Diagram of steps of proposed algorithm.

These regions were categorized into text and non-text by an SVM classifier and scale-invariant features obtained from extremal regions. Yi *et al*. [41] initially segmented an image into connected components using the gradient-based and color-based features. Following this, they eliminated some of the non-text elements by thresholding three features of the components; aspect ratio, size, and the number of cavities. Finally, they grouped the text candidate elements based on their relational properties such as sizes and distances between characters.

2.4. Detecting text by deep learning and convolutional neural networks

Recently, the CNN neural networks have been developed to identify and eliminate the non-textual elements from images with great precision. This has opened a new window for the relevant researchers. Bin Ahmad [42] has proposed a method for detecting Arabic text in natural images, which performed feature extraction and character detection simultaneously by a CNN. Li *et al*. [43] have proposed a multilingual text detection technique, which used common features of text strokes in different languages and CNN. In the training phase, the features were extracted from image patches that contain text using random selection. These features were used as the initial seeds in K-means clustering. Contrary to the traditional CNNs, in which the core of the first layer is chosen randomly, they replaced the first layer with K-means clustering. The last layer was completely connected to the SVM that classified the image. Liu *et al*. [44] have introduced a new CNN called Deep Matching Prior Net (DMPN). They used the sliding window in the middle layer to mark potential areas that contain text. Following this, a Monte-Carlo method was used to calculate the rectangular areas that contained text. Finally, these rectangular areas were improved to rectangular boxes that tightly contained text. Zhang *et al*. [45] have proposed a method with a fully convolutional neural network that produced a text and non-textual pixel map. Their method provided high-performance text detection on general datasets. Wang *et al*. [40] have proposed an architecture called Deep Scale Relationship Network (DSRN). They used a module to transfer multi-scale convolutional features to a unified dimension. Also they proposed a scale relationship module to aggregate the information through a bi-directional convolution operation. Xie *et al*. [50] have presented convolutional attention networks (CAN), which are different from CNNs and RNNs, and have an encoder-decoder architecture. They also proposed a novel spatial mechanism using average pooling, which was applied in every layer of the decoder. Their work involved position embedding. Chng *et al*. [51] have presented a method based on the DeconvNet architecture [54], and they reduced the training phase to a one-step process. Liao *et al*. [52] have proposed an end-to-end fast scene text detector named Textboxes. It performed parameter down-sampling on some of the CNN layers. Finally, Lyu *et al*. [53] have proposed a trainable deep neural network based on Mask R-CNN. Table 1 presents the advantages and disadvantages of the text detection approaches.

3. Proposed method

Our proposed method is a hybrid procedure that uses connected components analysis and texture properties together to detect the text in natural images. It can localize multi-oriented texts written in both Farsi and English alphabets. The algorithm consists of seven steps including pre-processing, finding flat regions and holes, textural analysis, edge detection, connected component analysis, clustering CCs, and thresholding relational features of CCs, as illustrated in Figure 1.

3.1 Pre-processing: contrast enhancement

In natural images, the contrast between foreground and background may be low or non-uniform due to non-uniform lighting or color fades. As a result, the edges of the texts are not well detected. To improve this, we consider contrast enhancement in a pre-processing step. We introduced a new method for enhancing the contrast of text images.

Approach		Advantages	Disadvantages
Texture based		They have good performance, in noisy spaces.	Low speed and Layout-related performance
Texture-based features	color	Simplicity in extraction and application	They lose their performance in non-uniform light
	Edge and gradient	Insensitive to non-uniform light and multi-colored letters	They do not work in complex backgrounds.
	textural	Insensitive to the noise and color of the letters	They do not detect spars letters that have poor textural properties
Connected Component-based		They work well in texts with different scales and rotations. Separated text elements can be directly recognized	Without prior knowledge, they cannot find text. Their design is very difficult because a large number of CCs are easily confused with the text.
Hybrid		The ability to recognize different types of text (dense and sparse characters)	They cannot find text without prior knowledge
Deep Learning		They eliminate non-textual elements with high speed and great precision	They require a vast amount of training data, Their training is difficult, time-consuming, and demands powerful computational resources. They may fall into local extrema

Table 1. Advantages and disadvantages of text detection approaches.

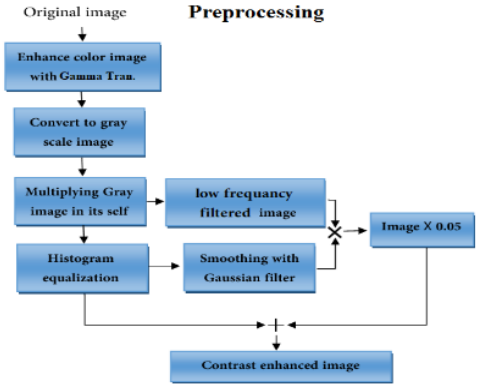

Figure 2. Steps of pre-processing phase.

Increasing the contrast to extract text should be done in a way that the internal pixels of a region do not differ internal pixels of a region do not differ significantly in their intensity, while pixels closer to the border, especially in textual areas, become more prominent against the background. Figure 2 shows the steps involved in our pre-processing phase. In the first step, the intensity of the colored input image improves with a Gamma transform wherein low and high threshold values for intensities are selected automatically. Then the gray image of the enhanced input is created and multiplied by itself increasing the contrast, thus making it darker.

Figure 3. (a) Original image (b) Gamma transform (c) Gray image (d) Multiplied image by itself (e) Histogram equalization (f) Output.

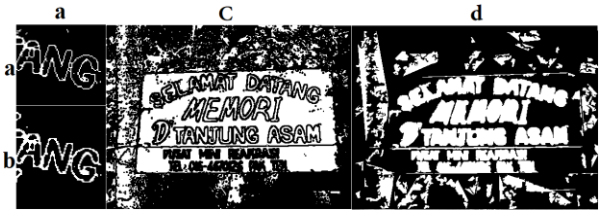

Figure 4. (a) Ragged edges (b) Edge smoothing and connecting (c) Flat regions (d) Text candidates.

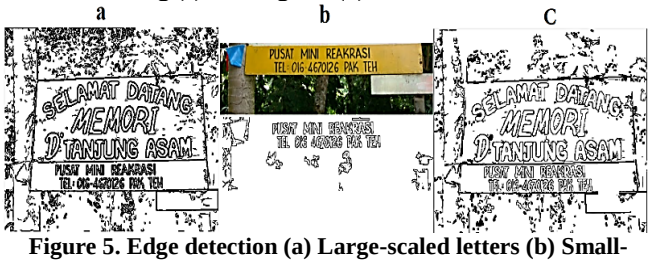

Figure 5. Edge detection (a) Large-scaled letters (b) Small-scaled letters (c) Final output.

A histogram equalization then is performed to return balance in illumination and illustrate edges. As a result, this increases noise and unnecessary details. In order to address this problem, we create a mask and add a small fraction of it to the image (multiplied by an adjustment value that is 0.05). The goal here is to increase the smoothness of internal regions, while reducing differences in their intensities. The mask image is obtained by multiplying two images. The first image is a smoothed version of the result of the histogram equalization step, and it is obtained by applying a Gaussian filter. The second one is a smoothed version of the image before the histogram equalization, and it is obtained by applying a low-pass Butterworth filter with a cutoff frequency of 8000Hz. We found, in our experiments, that the cut-off frequency of 8 kHz produces appropriately smoothed regions, while keeping the main details preserved enough for the later stages of processing. Multiplying these two smoothed images increases the intensities to approximately the power of two of the original values. Hence, a small fraction of the mask image is added to the

last image to smooth the inner regions without making significant changes to the other areas. Figure 3 depicts the outcome of each step in this procedure.

3.2 Finding flat regions

In our method, we assume that the backgrounds of text areas are flat and homogeneous. In order to identify the homogeneous regions, the standard deviation of the image is obtained in 3 × 3 neighborhoods. The small neighborhood size of the 3 x 3 precisely is capable of finding the homogenous regions without adverse effects, whereas a larger window (e.g. 5 x 5) shifts the edges, and causes a lot of distortion in the image. The result is normalized, then thresholded with the value 0.1 to identify all the high entropy areas or edges (Figure 4-a). The broken edges are smoothed to produce the connected edges by a frequency filter (Figure 4-b). Finally, a high threshold of 0.1 is applied to identify homogenous regions as depicted in Figure 4-c. The cavities of flat regions include text components. To find them, we obtained black parts of each flat region's components in their convex areas and then aggregate their negatives.

3.3 Textual analysis and determination of text candidate regions

Flat regions that contain text are not completely uniform, hence we only consider those that contain holes in them. In [15], a technique was proposed to find quasi-text areas that have many corners. We have adapted this technique to identify the potential regions that contain text. Initially, the wavelet transform is applied to the gray image followed by the Gabor's filter. We used the discrete wavelet transform 'db1' from the Daubechies family that has a finite number of filter parameters and fast implementations. Regarding the Gabor transform, we use the wavelength of 2, the special frequency if sqrt (1/2). The filter output in four orientations 0, 45, 90, and 135 degrees are added together to obtain the final result. In the resulted image, the text regions are very bright. This image is smoothed by an averaging filter, and then a threshold is applied to remove the darker non-text areas (0.35 was found to be the best threshold value in our experiments). Finally, the common white regions of this image and the cavities image are marked as candidate textual regions (Figure 4-d).

3.4 Edge detection

In the proposed method, the connected component analysis is used to isolate the text elements with different sizes, colors, and contrast. Thus the edges should be carefully detected for these elements in a way that they can be cut from the image to obtain their attributes. Two different methods are used for edge detection of large- and small-scale letters, which minimizes the distortion and fractures of their edges. Two groups of character sizes overlap with each other, and some character sizes satisfy both. Thus the operations support all sizes of characters. For the large-scale letters, Laplacian of the original image is obtained, and a weighted value of it (by weight of ½) is added to the gray level image to enhance the edges. Next, its gradient image is calculated, which specifies the edges for the large-scale letters. We have selected pieces larger than 20 pixels in order to preserve only larger characters because the small characters are broken into small pieces due to the thickness of gradient edges (Figure 5-a).

For edge detection of small-scale letters, the regional MSER features are used. Firstly, the unsharp masking technique-an edge sharping technique-is applied, and the connected components of the image are obtained using MSER zoning. This procedure separates small objects with a high precision. Objects of over 40 pixels and less than 1500 pixels are selected as small-scale letters. Edges of small letters are initially identified by a Laplacian filter. Small holes may appear near their edge due to the noises or variable lighting, which leads to the formation of ragged edges (see Figure 6-a). In order to address this problem, the MSER regions became filled, and their edges were added to the initial edge image of small letters. This helps achieve connected and strong edges (Figure 6-b). We correct the zigzags that appear on the edges of relatively larger MSER regions by multiplying them with components larger than 500 pixels in the large-scale characters. It removes the outer parts of large MSER CCs. The final image of the edge detection is multiplied by the candidate regions of the text obtained in step (2) to exclude the non-textual objects (Figure 5-c).

Figure 6. (a) Small holes near edges (b) Filled MSER regions.

3.5 Features of textual connected components

Textual CCs have geometric features that distinguish them from other elements in the image. At this phase, we specify 10 features for

text elements of which 7 are our contributions, and 3 have been proposed by the other researchers. A large number of non-text components are eliminated using these features. These features are derived from the geometric properties such as the area, length, and width of CCs. The Farsi and English text elements have very similar geometric attributes. This makes it possible to apply the same features for both. There are also some differences between them. For example, the English text elements are composed of separated glyphs, which have aspect ratios that are limited to a certain range. On the other hand, in Farsi, the words are usually composed of connected letters; hence, the text elements are stretched horizontally, and it is not possible to determine a particular aspect ratio range for them. However, different thresholds might be required when using these common features to detect text in any of the two languages. In the Farsi letters, some of the characters including ژ،ز،ر and L and I in English have some geometric attributes contrary to the traits of other characters. These special cases are thus isolated in a separate stage and added to the final output. Each connected component usually contains a single letter or a connected part of the text or a non-textual area of the image. We use the words element, component, or connected component with the same meaning in describing our geometric features.

Feature 1: Stroke width uniformity

The text elements have parallel edges. Thus elements with a relatively stable stroke width (SW) can be considered as the text candidates. We used the method presented in [46] to obtain the stroke width transform (SWT) of all elements. For each element, SWT is represented by an array of values that contain the distance transform of the pixels along the medial axis or skeleton of that element. This array has a small variance for text elements. Thus the SWT values for text elements are close to the average. Given this point, our first feature is defined as follows:

$$Feature1 = \frac{\sum_i SW_i \; / \max_i \left(SW\right)}{8 \times Extent} \tag{1}$$

In this formula, SW is the stroke width array, and the extent is the ratio of the area of a component to the area of its surrounding bounding box. In the textual elements, the max and average value of the stroke width array are almost equal. Thus for such elements, the sum of the stroke width values divided by the maximum stroke value is approximately equal to the length of the medial axis of that element. With a simple thresholding technique, the non-text components can be eliminated. The extent parameter is added to the denominator for two reasons. Firstly, in the case of textual elements, it has a normalization effect with respect to the medial axis length of letters. Secondly, for non-textual elements with no cavities or parallel edges, due to their greater extent value, the value of this feature drops below the threshold causing the element to be marked as non-text. Note that the extent value for the Farsi text elements is usually greater compared to the English characters due to the presence of stretched horizontal elements. The coefficient 8 reduces the effect of the skeleton length in the equation. For the Farsi and English characters, the thresholds for discriminating text and non-text elements were determined as 2.25 and 2.5, respectively. The elements with a feature smaller than the threshold value are labeled as non-text, and eliminated. The stated values were determined by experiments on numerous images. Figure 7 illustrates the effect of this feature, where some non-text elements in the left-hand side image have been eliminated on the right-hand side image by applying feature 1.

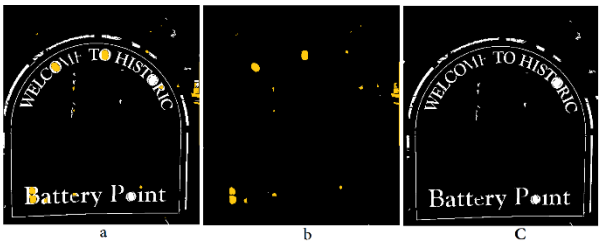

Figure 7. (a) Initial image (b) Differences (c) After thresholding with feature 1.

Feature 2: Stroke local variation

This feature is based on the fact that the stroke width for text has small variations along its path. We define our second geometric feature as follows:

$$Feature\,2 = \frac{mean\left(Local\_Stadard\_Deviation\left(SWs\right)\right)}{Length\_of\_Skeleton} \tag{2}$$

We calculate the standard deviation of stroke width using a sliding window of 9 pixels along the medial axis of each component. Feature 2 uses the average value of the standard deviations of stroke width. This average value should be small for the textual components though its value also depends on the size and thickness of each component. Hence, it is normalized by dividing it by the length of the component skeleton. For all letters in Farsi and English, the threshold was set at 0.05. The value of feature 2 for the text elements should be less than 0.05. Figure 8 displays the effect of

feature 1, feature 2, and features 1 and 2 together. As it can be seen from the figure, each of these features has removed some of the non-text components from the image.

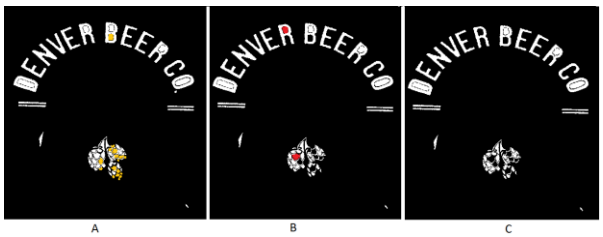

Figure 8. (a)After thresholding with feature 1 and (b) Feature 2 (c) Result.

Feature 3: Expected area distance

For the text elements, the product of the mean value of the stroke width of the element and its skeleton length is approximately equal to its area. For the text elements, the difference between this approximation and the actual area of the element is very small, and also proportional to the size of the element. In order to normalize this difference, we divide it by the square of equivalent diameter (i.e. the diameter of the circle with the same area as the element) to create our third feature, which is presented in Equation 3. The corresponding threshold value for this feature is set to 0.5 for all textual components in Farsi and English.

$$Feature\,3 = \frac{Area - (Length\_of\_Skeleton \times mean(SWs))}{EquDiam^{2}} \quad (3)$$

The effect of applying this feature is shown in Figure 9.

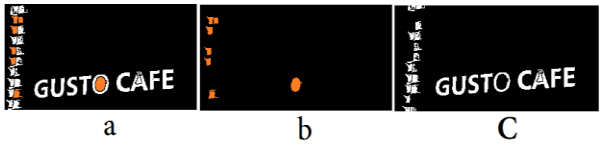

Figure 9. (a) Initial image (b) Differences (c) After thresholding by feature 3.

Feature 4: Bounding ellipse to element area ratio

This feature represents the ratio of the area of each component to the area of its bounding ellipse.

$$Feature\,4 = \frac{element\_area}{inscribed\_ellipse\_area} \quad (4)$$

Text elements usually have strokes in different directions spreading across its bounding oval. The textual element strokes are usually proportional to their size. As the area of a text element increases, the occupied part of its bounding ellipse also grows; thus this ratio remains almost constant. It should be noted that there are exceptions such as large letters with a very narrow stroke or small letters with a large stroke. The threshold for this feature was set between 0.02 and 0.85 for Farsi and English text elements, respectively.

The effect of applying this feature is shown in Figure 10

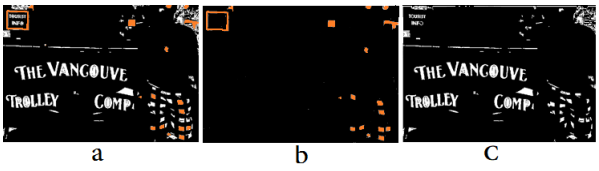

Figure 10. (a) Initial image (b) Differences (c) After thresholding by feature 4.

Feature 5: Elongation

This property shows the elongation of a connected component. The eccentricity of a connected component can be calculated as follows:

$$ecc = \frac{sqrt(major\_Axis\ \wedge 2 - mior\_Axis\ \wedge 2)}{major\_Axis} \quad (5)$$

where the major axis and minor axis are the axes of the bounding ellipse. Eccentricity shows the difference between the focal lengths of the oval surrounding the element, which specifies its elongation. This value is close to 1 for stretched elements and is close to 0 for round elements. The bounding boxes for English/Persian letters are usually wide rectangles (except ا in Farsi and L and I in English). In our fifth geometric feature, the ratio of height and width of the bounding box multiplied by eccentricity is used, as shown in Equation 6. This ratio is a small number for the text elements.

$$Feature\,5 = \frac{\max(w, h)}{\min(w, h)} \times ecc \quad (6)$$

W and h are equal to the width and height of the bounding box, respectively. The value of this feature, for Farsi characters, is between 0.33 and 7, while, for English characters, is between 0.1 and 4. The effect of applying this feature is shown in Figure 11.

Figure 11. (a) Initial image (b) Differences (c) After thresholding by feature 5.

Feature 6: improved stroke width metric

As our sixth feature, we exploit a metric proposed in [46] by Li *et al*. They used the ratio of the standard deviation of stroke width to the average value of stroke width as a text element discriminator. In some characters with several strokes in different directions, the stroke width is sometimes different in each direction. Thus the mean value of stroke width is not a good approximation for the actual stroke width in different directions. In order to overcome this, we used local averages of stroke width in a small sliding window. Our enhanced version of the metric proposed by [47] is defined as follows:

$$Feature\,6 = std\,(SWs) \times mean\,(\frac{1}{Local\ \_\ average\,(SWs)}) \quad (7)$$

The threshold for this feature is between 0 and 0.13 and between 0 and 0.141 for the Farsi and English texts, respectively. Figure 12 shows the result of applying this feature.

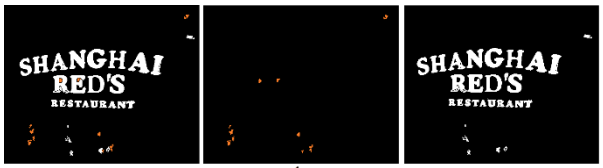

Figure 12. (a) Initial image (b) Differences (c) After thresholding by feature 6.

Feature 7: Symmetric property

Text elements are elements with many corners, parallel edges, and curvature. This property reduces the area occupied in their bounding box. After reviewing all letters and characters in Farsi and English, as well as the connected sub-words in Farsi, we observed that if the bounding boxes divide into 4 sections with lines of symmetry, only one or two sections will be relatively filled full and other sections would be significantly vacant. The symmetry property is defined according to this fact. Based on this property, we define our seventh feature to have a binary value that is equal to one for text elements and zero for non-text elements.

The value of this feature is determined as follows. For each textual element, we consider a white rectangle with the size of the bounding box of the element, and then divide it into 4 equal parts from its half-length. We then consider each element and its convex hull and match these two for each quarter of the bounding box (see Figure 13). We calculate the area ratio array by dividing the area occupied by the element in each quarter by the corresponding convex hull image in that quarter to obtain area ratio entries.

$$AreaRatio = \frac{Area\ \_of\ \_quarter\ \_of\ \_element}{Area\ \_of\ \_convex\ \_image} \quad (8)$$

Feature 7 will be 1 for elements with at most two area ratio entries greater than 0.85, while it is zero for all other elements (see Fig. 14).

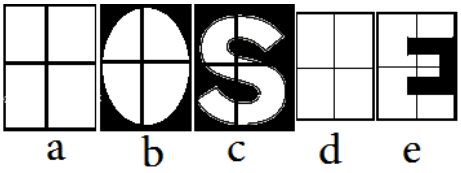

Figure 13. (a) Divide element bounding box (b) Convex image of element S (c) Image of element S (d) Convex image of element E (e) Image of element E.

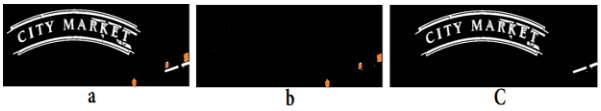

Figure 14. (a) Initial image (b) Differences (c) After thresholding with feature 7.

We have also applied three more features previously proposed by the other researchers to discriminate textual components. These are solidity, aspect ratio, and border ratio.

$$Solidity = \frac{Area}{Convex\ \_Area} \quad (9)$$

$$aspect\ \_Ratio = \frac{h}{w} \quad (10)$$

$$border\ \_Ratio = \frac{perimeter}{Area} \quad (11)$$

Given that the edges of text CCs are parallel lines, the ratio of the perimeter to the area will be constant. Since the text is usually written in fixed thicknesses, the border ratio can be set. The solidity range is obtained between 0.1 and 0.987, and the border ratio range is obtained from 0.06 to 1 for both the Farsi and English texts. The aspect ratio range is determined as 0.3 to 3 for the English elements.

Figure 15. Stretched letters.

The letters I and L in English, and آ in Farsi, are similar in shape, being narrow and rectangular. These letters have different geometric features than other letters. They fill almost all the space inside their bounding box, and their aspect ratio and elongation are larger than those of other letters. Applying the aforementioned features will eliminate these letters from the image. Thus we apply the following rules to identify and add any occurrences of these letters back to the image. These are components that satisfy these four rules.

1. By experimenting on multiple images and texts with various fonts and scales, it is observed that the value of solidity for letters I, L, and ا is greater than 0.6. Their eccentricity is due to a low width and stretched length greater than 0.81. Hence, we use a different threshold value for feature 3 and feature 4 to identify these special characters. Any component with a feature 3 value less than 0.6 and a feature 4 value between 0.85 and 1 is identified as a candidate.

2. The three letters under discussion have no holes in their shape. We use the ratio of the area of a component before and after filling holes to find them. Considering the presence of noise, which may produce small cavities, any component with a ratio greater than 0.95 is a possible candidate.

$$\frac{Area}{Filled\_Area} > 0.95 \quad (12)$$

3. The proportion of the width of letters I, L, and ا to their length is far smaller than that of other letters. To ignore the direction restriction, we consider the ratio of the width to the length of their surrounding ellipse. The components with a ratio of less than 0.7 are selected.

4. Letters I, L, and ا have a small mean of local standard deviations of stroke width. We calculate a ratio by normalizing this value, as shown in the following equation. The components with a value of less than 0.5 for this ratio have been extracted.

$$\frac{mean(Local\_std(SW))}{major\_Axis} \quad (13)$$

Detection of letters ژ،ر،ز

These constitute another set of special Farsi letters whose differences are only in the number of their dots. They may also appear in the form of a narrow rectangle. We identify any occurrences of these letters by applying a set of special features to restore them. The solidity value of letters is between 0.65 and 0.8, and their eccentricity is greater than 0.81. The threshold values for the third feature are the same as those for letters I, L, and ا as formerly discussed. For the fourth feature, due to its small curvature, the threshold is set between 0.6 and 0.85. Additionally, these letters have the same small width variations; hence, the corresponding rule (number 4) is applied to them, as with the previous letters. After the pre-processing phase and application of all geometric features, we obtained one image for both Farsi/English languages that contain components belonging to them. These connected components (CCs) are not completely isolated and non-textual CCs may still exist among them. In the next phase, we cluster these elements using the shape and color attributes to remove non-text elements.

3.6 Clustering of connected components

At this step, we cluster the connected components using mean shift clustering to identify the potential text lines. The set of features used for clustering purposes consists of a shape feature (i.e. stroke width) and three-color features (i.e. mean intensity of each color channel R, G, B). Due to the differences between Farsi and English, we use an additional shape feature (i.e. EquivDiameter as explained before) for the English texts. The clustering output is a set of images, each of which contains either text elements of similar color, size, and fonts or non-text elements. Figures 16-b through 16-d show three clusters obtained from the original image in Figure 16-a. In the final step, the clusters related to text lines are identified using thresholds, as explained in the next section.

d c b a

Figure 16. (a) Original (b-d) Obtained clusters.

3.7 Thresholding clusters and discrimination of text lines

Now cluster components are the same according to color and size (for the English components). To discriminate textual clusters and eliminate others, we perform a thresholding operation. Firstly, each group of connected components in a cluster, close in their spatial positions, is isolated based on their Euclidian distance. Components with distances less than a certain threshold in an image are put in the same group. In our experiment, the threshold

is set as 25 pixels. This is the maximum distance between letters in a text line in our experimental datasets which contains a large number of images containing representative texts in different shapes and fonts. If an image contains text in an extremely large font size this parameter can be adjusted accordingly. Even if the parameter is used as it is, text in a very large font can still be detected but in separate groups which will be dealt with at a later stage. Following this, each group is examined against the following criteria. Those components within a group that satisfy all of the criteria are marked as text.

1- Standard deviation of the mean intensities of elements in a group in each of the color channels R, G, and B should be less than the threshold value of 0.4 to be considered as textual elements.

2- Distance transform for components edges in the group is calculated, and the inner values in each component are selected with their mean calculated. If the standard deviation of all the means obtained for the components in a group was less than 0.5, it could be classified as a text group. Here, we assume that the text elements belong to a group associated with the same stroke width.

3- If the ratio of the total area of the individual elements in a group to the convex area of the group is between 0.1 and 0.7, the group can be considered textual.

4- We define a signal feature of stroke width as follows to represent the stroke width changes for the components in each group. Also 0.5 is the threshold to pass this criterion.

Signal feature of stroke width

Due to the nature of our clustering, the mean values of stroke width for the element in a cluster (and hence in a group) of components are close to each other. Thus, in our stroke width analysis, we consider all elements of a group as a single element and treat the accumulated stroke width as a 1D signal, and define a signal feature accordingly. The degree of disturbance and variations in the stroke width signal of text elements are far less than in the non-text elements. Due to the uneven edges in non-text elements, their signal has strong and frequent fluctuations, while, in-text elements, these fluctuations are limited, and the signal is much smoother, as it can be observed in Figure 17 (row-b). Based on this fact, the stroke width signal analysis was performed as follows. In our signal analysis, we first down-sample the values by finding the local maxima. This is to facilitate the analysis by extracting the main trend in the stroke signal. In the non-textual elements, a large stroke variation might be seen in a small neighborhood. By choosing a small neighborhood, these variations can be detected (Figure 17: row-c). Then we used a derivative operator (Figure 17: row-d). This results in a signal that preserves major stroke changes well, and hence, highlights the stroke fluctuations very effectively. The mean value of peaks of the derivative signal (Figure 17: row-e) forms our stroke signal feature. The last two stages, i.e. clustering and signal processing, are performed separately for Farsi and English. This results in two separate output images added to form the final image of the identified textual regions.

3.8 Adding dots of letters

At this step, dots of letters are added back to the extracted text. First, the area that might potentially contain dots is identified. To begin with, we consider the rectangular bounding boxes obtained by performing a morphological close operation to the components of the previous stage. We extend the height of these bounding boxes by 40% above and below. The potential dot(s) in these areas is subsequently searched for using two operations. Note that dots in English and Farsi appear in different shapes, including circles, diamonds, and rectangles. In the first operation, the edge image obtained in the earlier stage is used to identify any CCs with a surface area between 9 and 80 pixels, and solidity greater than 0.85. Such elements may include cavities inside the letters. The cavities are identified easily, by subtracting the original image from the whole filled image, and then eliminated. In the second operation, gray-level images are denoised by applying a Gaussian filter. MSER is then used with the threshold of 0.2 to identify dots within the range of 9 to 30 pixels and with the threshold of 1.5 for dots within the range of 30 to 50 pixels. The cavities are eliminated as aforementioned. All the dots are added to the output image. The final outputs of our proposed algorithm can be seen in Figure 18.

4. Evaluation and Results

We evaluated the proposed method using the Wolf's approach [47] on the two datasets ICDAR

2017-Total Text [55] and DNIFT [56]. The first dataset contains the natural images of English texts, while the second dataset includes natural images of the Farsi and English texts. The ground truths (GT) of ICDAR are labeled text areas displayed word by word (each word is marked with a rectangle that covers it). The DNIFT images are collected and labeled by us in the same manner as ICDAR.

4.1 DNIFT: Dataset of natural images with Farsi text

In order to evaluate Farsi text detection in our algorithm, we used a dataset containing natural images of Farsi texts. This dataset contains 100 natural images collected from the Internet, and contains the Farsi and English texts. The collected images are composed of road, street, and alley signs, shops, and fronts of public places as well as banners and posters, etc. (Figure-19). We created ground truth images manually by drawing rectangles for each word on the image. Each rectangle was converted into arrays X and Y containing coordinates of polygon vertices stored as .mat files, similar to the data structure of standard datasets.

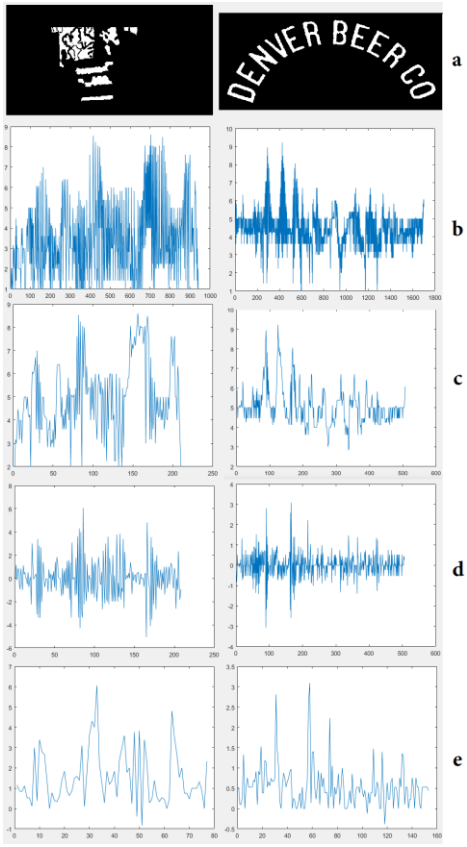

Figure 17. (Columns) left, non-text and right, text elements group (Rows) (a) Elements group (b) Stroke width signal (c) Local means of stroke width signal (d) Derivative signal (e) Peaks of derivative signal.

Figure 18. Output images of our proposed method.

Figure 19. Dataset of natural images with Farsi text.

4.2 Wolf's approach

The Wolf evaluation method named 'DetEval' compares rectangles containing words in detection images with GT rectangles. In order to determine whether a text rectangle is detected, some criteria based on the overlap of GT and detection result rectangles are used. The first matrices σ and τ are computed as described in [47].

The rows and columns of these matrices correspond to the ground truth and detection, respectively. Entries of σ and τ represent the area recall and area precision, respectively, and are equal to the proportion of overlaps on rectangles of GTs and detections.

$$\sigma_{ij} = R_{Ar}\left(G_i, D_j\right), \tau_{ij} = P_{Ar}\left(G_i, D_j\right) \quad (14), (15)$$

In general, a non-null value in element *(i, j)* represents the overlap of rectangle *i* in the GT with rectangle *j* in the detections. These two rectangles are only matched (one to one match) when their overlapped area is greater than the pre-defined thresholds t_p and t_r:

$$\sigma_{ij} > t_r, \ \tau_{ij} > t_p \quad (16)$$

In 'one to many' match, a GT rectangle matches a set of detection rectangles S; if:

1-A large enough percentage of the GT rectangle was detected:

$$\sum j \in S_o \ \ \sigma_{ij} \geq t_r \quad (17)$$

2-Each detection rectangle overlaps with a GT rectangle sufficiently to be considered part of it.

$$\sum j \in S_o \quad \tau_{ij} \geq t_p \quad (18)$$

In many to one matches, a detection rectangle matches the set of GT rectangles S; if:

1-A large enough percentage of the GT rectangle was detected:

$$\sum j \in S_m \quad \sigma_{ij} \geq t_r \quad (19)$$

2- Each GT rectangle that exists in the detection result has a certain level of area precision.

Many to many match, is not supported by Wolf's algorithm. Based on this matching strategy, ROB and POB are defined as follows:

$$R_{ob}(G, D, t_r, t_p) = \frac{\sum_i Match_G(G_i, D, t_r, t_p)}{|G|} \quad (20)$$

$$P_{ob}(G, D, t_r, t_p) = \frac{\sum_j Match_D(D_j, G, t_r, t_p)}{|D|} \quad (21)$$

where $Match_D$ and $Match_G$ are the functions that evaluate various types of matches:

$$Match_D(D_j, G, t_r, t_p) = \begin{cases} 1 & if\ D_j\ matches\ against\ a\ single\ detected\ rectangle \\ 0 & if\ D_j\ does\ not\ match\ against\ any\ detected\ rectangle \\ f_{sc}(k) & if\ D_j\ matches\ against\ several\ (\rightarrow k)\ detected\ rectangle \end{cases}$$

$$Match_G(G_i, D, t_r, t_p) = \begin{cases} 1 & if\ G_i\ matches\ against\ a\ single\ detected\ rectangle \\ 0 & if\ G_i\ does\ not\ match\ against\ any\ detected\ rectangle \\ f_{sc}(k) & if\ G_i\ matches\ against\ several\ (\rightarrow k)\ detected\ rectangle \end{cases}$$

In these formulas, *fsc(k)* is a parametric function of the evaluation schema, which controls the penalty for each matching case of scattering, merging or splitting. The value 1 means no penalty will be charged while lower values apply more fines. One drawback of the Wolf's method is that it calculates precision very strictly. In other words, although it is based on the area match, it gives an equal score to small mismatch and large matches. If there is a tiny connected component in the output image, it is counted as one FP, which reduces precision. Meanwhile, each word that contains several connected components and occupies a large area is counted as only one TP (Figure 20).

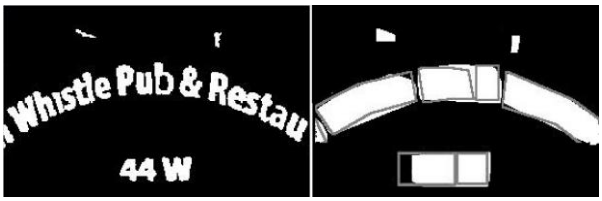

Figure 20. (Left): Input image, (Right): Detected text (white areas) and ground truth.

4.3 Evaluation of features' performances

In this section, the performance of the seven geometric features that eliminate non-text elements as formerly explained are evaluated. The evaluation is performed on two sets, each containing ten images from the two datasets. For each image, the initial steps of pre-processing, homogeneous regions, texture analysis, and edge detection is executed first. Following this, the resulting images are thresholded using two properties, solidity, and border ratio, and saved as the base images. We then add each of the features individually to a copy of the base image and

compare the resulting image with the GT using Wolf's algorithm. Precision, Recall, and F-measures are calculated for each group of images. Table 2, Table 3, Figure 21, and Figure 22 display the results of the individual feature evaluations for DNIFT and Total Text, respectively. Note that the values of precision, recall, and F-measure show the effect of each geometric feature individually, and do not reflect the overall performance of the proposed algorithm. Reviewing these results, it can be observed that the value of recall is almost independent of the specific feature applied; however, precision improves compared to the base image with the addition of each feature. As it can be seen from the results, the performance criteria for DNIFT are relatively low in comparison to Total Text. This is because the selected Farsi test images contain more logos or guide signs close to the text in flat regions. It should, however, be noted that these performances are achieved by terminating the algorithm at an early stage. Running the full cycle will produce

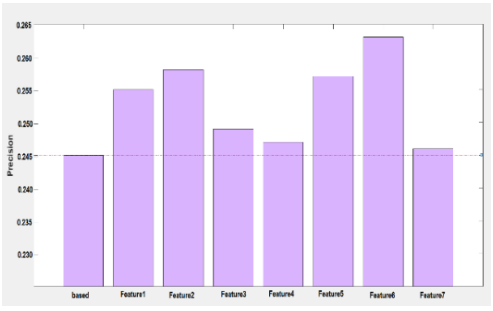

Figure 21. Geometric features improved precision on DNIFT.

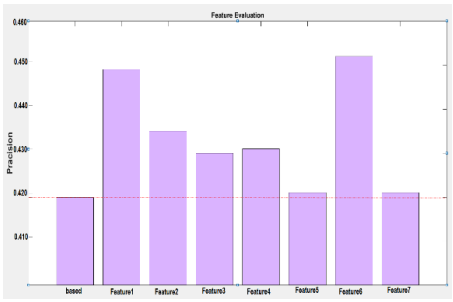

Figure 22. Geometric features improved precision on Total Text.

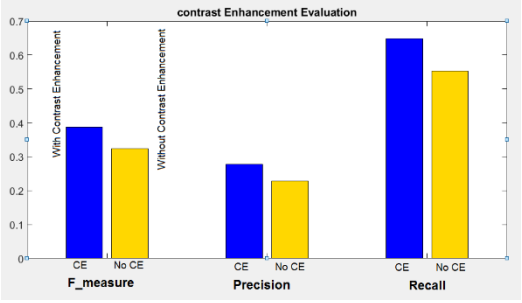

Figure 23. Contrast enhancement improved performance.

greater performance. Our contrast enhancement stage was evaluated by selecting 10 images with Farsi and English text from the DNIFT dataset. We obtained the base images in the same way as described in the previous section but without applying contrast enhancement.

Note that the components in these images are obtained by thresholding based on two properties, solidity and border ratio. Precision, recall, and F-measure are calculated for these images, and compared with the values for images obtained by including contrast enhancement. Table 4 and Figure 23 reveal the results. As it can be seen, applying contrast enhancement improves detection performance dramatically.

Table 2. Evaluation of geometric features of CCs on DNIFT.

Image	F-Score	Precision	recall
Based	0.359	0.246	0.661
Feature1 (uniformity)	0.373	0.259	0.661
Feature2 (local variation)	0.363	0.250	0.661
Feature3 (Expected area diff.)	0.361	0.248	0.661
Feature4 (Area ratio)	0.373	0.258	0.677
Feature5 (Elongation)	0.369	0.256	0.661
Feature6 (stroke width metric)	0.377	0.264	0.661
Feature7 (symmetric)	0.359	0.247	0.661

Table 3. Evaluation of geometric features of CCs on Total Text.

Image	F-Score	Precision	recall
Based	0.583	0.419	0.957
Feature1 (uniformity)	0.610	0.448	0.957
Feature2 (local variation)	0.597	0.434	0.957
Feature3 (Expected area diff.)	0.593	0.429	0.957
Feature4 (Area ratio)	0.593	0.430	0.957
Feature5 (Elongation)	0.583	0.420	0.957
Feature6 (stroke width metric)	0.613	0.451	0.957
Feature7 (symmetric)	0.584	0.420	0.957

4.4 Evaluation and comparison of the proposed algorithm

In order to evaluate the overall performance of the proposed algorithm, we run our experiments on a Windows 10 system with an x64-based processor, Intel(R) Core (TM) i7-5500U 2/40 GHz RAM 8/00 GB, and MATLAB R2016. We evaluated our detection algorithm on the ICDAR 2017-Total Text dataset. Using the Wolf approach, the evaluation results were calculated as 84% for recall, 69% for precision, and 0.76 for F-measure. The evaluation results applying the algorithm on the DNIFT dataset were calculated as 81% for recall, 63% for precision, and 0.71 for F-measure.

Most of the related work on Farsi text detection has been based on a video sequence or inserted captions. Thus we chose two multi-lingual algorithms to compare against our algorithm. Yin *et al*. [38] have provided a method for detecting multilingual text that uses analysis of connected components and MSER zoning. They used Wolf's algorithm to evaluate their method. Li *et al*. [43] have provided a method for localizing multilingual text, and evaluated it separately for the English, Arabic, and Chinese languages. Table 5 reports the detection performance of our method and the aforementioned competitors.

In order to compare the proposed algorithm for English text detection, we chose several methods [51, 52, 53]. They evaluated their methods using Wolf's Protocol 'DetEval' on the Total-Text dataset. Table 6 compares the detection performance of these techniques.

Table 5. Evaluation using multilingual datasets.

Method	Recall	Precision	F-measure
Yin ***et al***. [38]	63%	79%	0.70
Li ***et al***. [43]	72%	70%	0.66
Proposed	81%	63%	0.71

Table 6. Evaluation based on Total-Text.

Method	Recall	Precision	F-measure
DeconvNet [51]	33%	40%	0.36
TextBoxes [52]	42.5%	47.2%	0.45
Mask text spotter [53]*	55%	69%	0.61
Proposed	84%	69%	0.76

5. Conclusion

Over the course of the last decade, the detection of text in natural images has become a widespread research issue with a rapid development and sustainable growth. A variety of algorithms have hence been proposed in this regard. The four main approaches to text detection include texture-based, connected component-based, hybrid, and deep learning approaches. Most of the proposed text detection algorithms can only detect horizontal text. Moreover, only a small number of methods and benchmarks have dealt with the Farsi/Arabic languages. In the age of communication and globalization, designing the systems that can extract multilingual text is essential. Our approach for detecting text in natural images was a hybrid approach, benefitting from the connected components and texture analysis. The proposed method was able to find multi-oriented text within natural images. Further, it identified texts in both the Farsi and English languages. Evaluation of the proposed method using Wolf's algorithm demonstrated improved detection performance compared to other methods. Considering the generality of text features used in the proposed method, it can be extended to detect text written in other languages, handwriting, and abnormal fonts. Setting the threshold values for geometric features can also be performed using an automated approach such as association rule mining. Deep learning has recently gained attention in detecting text within natural images and it is in its infancy. However, it requires a vast amount of data and training models. Furthermore, it is not clear what is happening at the neuronal levels and it may fall into local extrema. Finally, its training is difficult, time-consuming, and demands powerful computational resources.

References

[1] H. Chen, S.S. Tsai, G. Schroth, D.M. Chen, R. Grzeszczuk, and B.Girod, "Robust Text Detection in Natural Images With Edge-Enhanced Maximally Stable Extremal Regions," in *Proc. 18th IEEE Int. Conf. Image Processing (ICIP)*, pp. 2609-2612, Sep.(2011).

[2] Q. Ye and D. Doermann, "Scene Text Detection via Integrated Discrimination of Component Appearance and Consensus," Camera-Based Document Analysis and Recognition CBDAR, Lecture Notes in *Computer Science*, Vol. 8357. Springer, Cham, pp 47-59 (2013).

[3] S. Liu, Y. Xian, H. Li, and Z. Yu, "Text Detection in Natural Scene Images Using Morphological Component Analysis and Laplacian Dictionary," *IEEE/CAA Journal of Automatica Sinica* (2017).

[4] S. Lee, M.S. Cho, K. Jungz, and J.H. Kim, "Scene Text Extraction with Edge Constraint and Text Collinearity in Pattern Recognition," *(ICPR) 20th International Conference on. IEEE*, pp. 3983-3986 (2010).

[5] L. Neumann, and J. Matas, "Real-time Lexicon-free Scene Text Localization and Recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 38, No. 9, pp. 1872-1885 (2016).

[6] J.L. Field and E.G. Learned-Miller, "Improving Open-Vocabulary Scene Text Recognition," in *Proc. IEEE Int. Conf. Document Analysis and Recognition*, pp. 604-608, (2013)

[7] Y.X. Liu, and T. Ikenaga, "A Contour-based Robust Algorithm for Text Detection in Color Images," *IEICE Trans. on Information and Systems*, Vol. 89(3), pp. 1221–1230 (2006).

[8] H. Zhang, K. Zhao, Y. ZheSong, and J. Guo, "Text Extraction from Natural Scene Image: A survey," *Neuro-computing*, Vol. 122, pp. 310–323 (2013).

[9] X. Chen, J. Yang, J. Zhang, and A. Waibel, "Automatic Detection and Recognition of Signs from Natural Scenes," *IEEE Trans. Image Processing*, Vol. 13, No. 1, pp. 87–99 (2004).

[10] N. Nikolaou, and N. Papamarkos, "Color Reduction for Complex Document Images," *Int. J. Imaging Systems and Technology*, Vol. 19, pp. 14–26 (2009).

[11] V. Wu, R. Manmatha, and E.M. Riseman, "Text Finder: An Automatic System to Detect and Recognize Text in Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 11, pp. 1224–229 (1999).

[12] X. Liu, and J. Samarabandu, "Mult-scale Edge-based Text Extraction from Complex Images," in *IEEE International Conference on Multimedia and Expo*, pp. 1721–1724 (2006).

[13] W. Ou, J. Zhu, and C. Liu, "Text Location in Natural Scene," *Journal of Chinese Information Processing* (2004).

[14] S. Yousfi, A. Berrani, and C. Garcia, "Boosting-based Approaches for Arabic Text Detection in News Videos," in *11th IAPR International Workshop on Document Analysis Systems (DAS'14), Tours, France* (2014).

[15] V.N.M. Aradhya, and M.S. Pavithra, "A Comprehensive of Transforms, Gabor filter and k-means Clustering for Text Detection in Images and Video," *Applied Computing and Informatics* Vol. 12, pp. 109–116 (2016).

[16] H. Goto, and M. Tanaka, "Text-Tracking Wearable Camera System for The Blind," in *Proc. IEEE Int. Conf. Document Analysis and Recognition*, pp. 141–145 (2009).

[17] K.I. Kim, K. Jung, and H. Kim, "Texture-based Approach for Text Detection in Images using Support Vector Machines and Continuously Adaptive Mean Shift Algorithm," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 12, pp. 1631–1639 (2003).

[18] Y.F. Pan, C.L. Liu, and X.Hou, "Fast Scene Text Localization by Learning-based Filtering and Verification," in *17th IEEE International Conference on Image Processing (ICIP), IEEE*, pp. 2269–2272 (2010).

[19] S. Lucas, "ICDAR 2005 Text Locating Competition Results," in *Eight Int. Conf. Document Analysis and Recognition* (2005)

[20] "Image evaluation campaign," Imag. Eval. (2006) [online] Available: https://zoxh.com/i/imageval.com

[21] T. Retornaz, and B. Marcotegui, "Scene Text Localization based on the Ultimate Opening," in *Int. Symposium on Mathematical Morphology*, Vol. 1, pp. 177-188 (2007).

[22] K. Wang, and J.A. Kangas, "Character Location in Scene Images from Digital Camera," *Pattern Recognition*, Vol. 36 (10), pp. 2287–2299 (2003).

[23] M. Zhao, S. Li, and J. Kwok, "Text Detection in Images using Sparse Representation with Discriminative Dictionaries," *Image and Vision Computing*, Vol. 28 (12), pp. 1590–1599 (2010).

[24] X. Zhao, K.H. Lin, Y. Fu, Y. Hu, Y. Liu, and T.S. Huang, "Text from Corners: A Novel Approach to Detect Text and Caption in Videos," *IEEE Trans. Image Processing*, Vol. 20, No. 3, pp. 790-799 (2011).

[25] P. Shivakumara, T.Q. Phan, and C.L. Tan, "A Laplacian Approach to Multi-oriented Text Detection in Video," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 33 (2), pp. 412–419 (2011).

[26] K.L. Bouman, G. Abdollahian, M. Boutin, and E.J. Delp, "A Low Complexity Sign Detection and Text Localization Method for Mobile Applications," *IEEE Trans. on multimedia*, Vol. 13, No. 5, Oct. (2011).

[27] M. Moradi, S. Mozaffari, and A. Oruji, "Farsi/Arabic Text Extraction from Video Images by Corner Detection," in *The 6th Iranian Machin Vision and Image Processing Conference* (2010).

[28] X.C. Yin, W.Y. Pei, J. Zhang, and H.W. Hao, "Multi-orientation Scene text Detection with Adaptive Clustering," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 1, pp. 1–1.

[29] L. Kang, Y. Li, and D. Doermann, "Orientation Robust Text Line Detection in Natural Images," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, pp. 4034-4041 (2014)

[30] S. Kumar, and A. Perrault, "Text Detection on Nokia N900 using Stroke Width Transform," (2011) [online] available: https://www.cs.cornell.edu/courses/cs4670/2010fa/projects/final/results/group_of_arp86_sk2357/Writeup.pdf

[31] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. IEEE Int. Conf. Pattern Recognition*, pp. 2963–2970 (2010).

[32] A. Mosleh, N. Bouguila, and A. Ben Hamza, "Image Text Detection using A Bandlet-based Edge Detector and Stroke Width Transform," in *Proc. Conf. British Machine Vision*, pp. 1–2 (2012).

[33] C. Yao, X. Bai, and W. Liu, "A Unified Framework for Multi-oriented Text Detection and Recognition," *IEEE Trans. On Image Processing*, Vol. 23 (11), pp. 4737–4749 (2014).

[34] W. Huang, Z. Lin, J.C. Yang, and J. Wang, "Text Localization in Natural Images using Stroke Feature Transform and Text Covariance Descriptors," in *Proc. Int. Conf. IEEE on Computer Vision*, pp. 1241–1248 (2013).

[35] L. Neumann, and J. Matas, "Scene Text Localization and Recognition with Oriented Stroke Detection," in *IEEE Int. Conf. Computer Vision (ICCV)*, pp. 97-104 (2013).

[36] S. Mansouri, M. Charhad, and M. Zrigui, "A Heuristic Approach to Detect and Localize Text in Arabic News Video," *Computación y Sistemas*, Vol. 22, No. 1, pp. 75–82 (2018).

[37] M. Darab, and M. Rahmati, "A Hybrid Approach to Localize Farsi Text in Natural Scene Images," in *Proc. Int. Neural Network Society Winter Conference (INNS-WC)* (2012).

[38] X.C. Yin, X. Yin, K. Huang, and H. Hao, "Robust Text Detection in Natural Scene Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 36, No. 5, pp. 970–983 (2014).

[39] H.P. Le, N.D. Toan, S.C. Park, and G. Lee, "Text Localization in Natural Scene Images by Mean-shift Clustering and Parallel Edge Feature," in *Proc. 5th Int. Conf. Ubiquitous Information Management and Communication, Seoul, Korea*, pp. 21-23 (2011).

[40] Y. Wang, H. Xie, Z. Fu, and Y. Zhang, "DRSN: A Deep Scale Relationship Network for Scene Text Detection," *IJCAI*, pp. 947-953, (2019).

[41] C. Yi, and Y.L. Tian, "Text String Detection from Natural Scenes by Structure Based Partition and Grouping," *IEEE Trans. Image Processing*, Vol. 20 (9), pp. 2594–2605 (2011).

[42] S.B. Ahmed, S. Naz, M.I. Razzak, and R. Yousaf, "Deep Learning based Isolated Arabic Scene Character Recognition," *proc. Arabic Script Analysis and Recognition, ASAR, IEEE Xplore* (2017).

[43] L. Li, S. Yu, L. Zhong, and X. Li, "Multilingual Text Detection with Non-linear Neural Network," *Mathematical Problems in Engineering*, Article ID 431608, 7 pages (2015).

[44] Y. Liu and L. Jin, "Deep Matching Prior Network: Toward Tighter Multi-oriented Text Detection," *Computer Vision and Pattern Recognition* (2017).

[45] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented Text Detection with Fully Convolutional Networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016.

[46] Li, Yao, and Lu, "Scene Text Detection via Stroke Width," in *21st IEEE Int. Conf. Pattern Recognition (ICPR),* (2012).

[47] C. Wolf, and J.M. Jolion, "Object count/area Graphs for the Evaluation of Object Detection and Segmentation Algorithms," *Int. Journal of Document Analysis and Recognition*, Vol. 8 (4), pp. 280–296, (2006).

[48] C. Yao, X. Bai, W. Liu, Y. Ma, and Y. Tu, "Detecting Texts of Arbitrary Orientations in Natural Images," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1083–1090 (2012).

[49] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," *Computer Vision and Pattern Recognition*, (2017).

[50] H. Xie, S. Fang, Z.J. Zha, Y. Yang, Y Li, and Y. Zhang, "Convolutional Attention Networks for Scene Text Recognition," *TOMCCAP* 15(1s), pp. 3:1-3:17 (2019).

[51] C. K. Chng, and C. S. Chan, "Total-Text: A Comprehensive Dataset for Scene Text Detection and Recognition," in *14th IAPR International Conference on Document Analysis and Recognition,* (2017).

[52] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Text Boxes: A Fast Text Detector with a Single Deep Neural Network," *AAAI*, pp. 4161-4167, (2017).

[53] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai. "Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," in *Proceedings of European Conference on Computer Vision (ECCV),* (2018).

[54] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," *ICCV* (2015).

[55] S. Chan, "Total-Text" Dataset [online]. Available: https://github.com/cs-chan/Total-Text-Dataset

[56] S. Bayatpour "DNIFT: Dataset of Natural Images with Farsi Text" [online]. Available: https://github.com/s-bytpr/-Dataset-of-natural-images-with-Farsi-text-DNIFT-

[57] Z. Imani, Z. Ahmadyfard, and A. Zohrevand, "Holistic Farsi handwritten word recognition using gradient features," *Journal of AI and Data Mining, Vol. 4, pp. 19-25*, (2016)

یک روش تشخیص متن دو زبانه در تصاویر طبیعی با استفاده از هیوریستیک و یادگیری بدون ناظر

سمیه بیات‌پور و مهران شرقی*

گروه هوش مصنوعی، دانشگاه الزهراء، تهران، ایران.

ارسال ۲۰۲۱/۰۸/۲۷؛ بازنگری ۲۰۲۲/۰۱/۲۵؛ پذیرش ۲۰۲۲/۰۸/۰۳

چکیده:

روزانه حجم بسیار زیادی تصویر دیجیتالی تولید می‌شود. یکی از عناصری که ممکن است در تصاویر دیجیتالی وجود داشته باشند؛ نوشته‌ها هستند. از اطلاعات درون نوشته‌ها در تصاویر طبیعی، می‌توان در زمینه‌های متنوعی بهره برد. مهمترین چالشهایی که برای تشخیص متن در تصاویر وجود دارد. عبارتند از: وجود نویز، تاری، اغتشاش، چسبیدن عناصر به یکدیگر، تنوع در فونت و طرحبندی و چرخش و... . با وجود پیشرفتهای زیادی که در شاخه تشخیص متن صورت گرفته است؛ هنوز الگوریتم واحدی طراحی نشده است که بتواند در رویارویی با همه مشکلات گفته شده با موفقیت عمل نماید. علاوه بر آن اکثر الگوریتمهای ارائه شده، تنها می‌توانند متنهای افقی را تشخیص دهند و تنها تعداد ناچیزی از آنها به زبان فارسی پرداخته‌اند. در این مقاله روشی برای تشخیص متنهای دارای چرخش به دو زبان فارسی و انگلیسی ارائه شده است. ما هفت ویژگی هندسی برای جداسازی عناصر متنی از پس زمینه معرفی نموده‌ایم و یک روش جدید بهبود کنتراست تصویر مخصوص الگوریتمهای تشخیص متن در تصاویر ارائه داده‌ایم. نتایج ارزیابی روش ارائه شده، کارایی بالای آن را روی تصاویر طبیعی نشان می‌دهد.

کلمات کلیدی: تشخیص متن، تصاویر طبیعی، خوشه بندی mean shift، متن دو زبانه، هیوریستیک، یادگیری بدون ناظر.