

Farsi Font Detection using the Adaptive RKEM-SURF Algorithm

Zahra Hossein-Nejad

Department of Electrical Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Iran.
hoseinnejad.zahra@yahoo.com

Hamed Agahi*

Department of Electrical Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Iran.
agahi@iaushiraz.ac.ir

Azar Mahmoodzadeh

Department of Electrical Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Iran.
mahmoodzadeh@iaushiraz.ac.ir

Received: 29/Feb/2020

Revised: 24/Aug/2020

Accepted: 01/Sep/2020

Abstract

Farsi font detection is considered as the first stage in the Farsi optical character recognition (FOCR) of scanned printed texts. To this aim, this paper proposes an improved version of the speeded-up robust features (SURF) algorithm, as the feature detector in the font recognition process. The SURF algorithm suffers from creation of several redundant features during the detection phase. Thus, the presented version employs the redundant keypoint elimination method (RKEM) to enhance the matching performance of the SURF by reducing unnecessary keypoints. Although the performance of the RKEM is acceptable in this task, it exploits a fixed experimental threshold value which has a detrimental impact on the results. In this paper, an Adaptive RKEM is proposed for the SURF algorithm which considers image type and distortion, when adjusting the threshold value. Then, this improved version is applied to recognize Farsi fonts in texts. To do this, the proposed Adaptive RKEM-SURF detects the keypoints and then SURF is used as the descriptor for the features. Finally, the matching process is done using the nearest neighbor distance ratio. The proposed approach is compared with recently published algorithms for FOCR to confirm its superiority. This method has the capability to be generalized to other languages such as Arabic and English.

Keywords: Adaptivity; Feature Extraction; Font Detection; Redundant Keypoint Elimination Method (RKEM); Speeded-Up Robust Features (SURF).

1- Introduction

Farsi is the official language of Iran, Tajikistan and Afghanistan. Farsi is among the first three languages of the world in terms of the number and variety of proverbs [1]. With vocabulary coming from Arabic (and other languages like Greek, Aramaic, Turkish, etc.) into Farsi, it has become one of the richest languages in terms of the word count [2]. Farsi is the ninth most widely used language in web content, and higher than Arabic, Turkish and other Middle Eastern languages [3]. To understand written Farsi texts by computers, new particular algorithms should be generated.

Optical character recognition (OCR) is a process by which printed documents or scanned pages are converted to recognizable characters. OCR is one of the most important sectors of e-government. Most of the work done in the field of OCR is related to English, Chinese, and Japanese texts with dramatic improvements in recent years. While, Farsi OCR has continued to thrive despite the relatively high volume of academic research and the urgent need for

government agencies. Farsi OCR has still a long way from its intended desire, and yet no completely acceptable system has been developed. In other words, the aim is to generate Farsi systems that are comparable in accuracy and performance to the English OCRs.

Font detection is one of the most useful pre-processing steps in improving the OCR performance for systems which deal with typeset-printed-scanned texts consisting several different fonts [4, 5]. Font detection is the process through which text language and font type, size and style could be identified. Although many methods can be found in the literature for font detection, most of them are composed of two major stages, i.e., feature extraction and font recognition [6, 7]. In general, feature extraction phase of font detection methods is categorized into two approaches including typographical and textural features [8]. In typographical features extraction methods, character weights and space widths are used to analyze textual images [9, 10]. While, in textural feature extraction methods, local and global features are used to describe textual images [6]. Overall, textural based methods are more accurate compared to the typographical ones, with more applications in OCR software [11, 12]. Gabor filter,

*Corresponding Author

wavelet transform and local detectors are examples of textural features widely used in font detection [13]. Although many studies have been conducted to detect English, Chinese and Japanese fonts, few ones have been done for Farsi font detection [14-16]. Due to the complexities in Farsi texts including continuous writing, the variations of the letters with respect to their relative position in words and the difference in the shape of the characters in different fonts, direct application of English-font detection methods for Farsi fonts is not possible. On the other hand, no effective method has been developed for Farsi font detection which is comparable to those for English in terms of recognition accuracy. Given the importance and wide spread use of OCR and low accuracy of existing methods, proposing operative Farsi OCRs is mandatory and challenging. This paves the ground for the motivation to propose approaches which improve the Farsi font detection system so that it reaches acceptable detection rates.

One of the most common feature extraction methods in the font detection applications is the scale-invariant feature transform (SIFT). This algorithm is robust against scale and rotation changes and also intensity variations, affine distortion and noise [17]. These advantages have made this algorithm significant and widely used in the image processing tasks. Meanwhile, the imperative problem of the SIFT is the creation of redundant points, which lead to similar descriptors and consequently possible interference in the matching process. Recently, the RKEM has been proposed by Hossein-Nejad and Nasri [18], which aims to identify and eliminate redundant points in the SIFT using a redundancy index. The RKEM-SIFT could well remove useless keypoints and result in very good attainments in the image registration. Applying the RKEM to problems such as image registration [18], copy-move forgery detection [19] and image mosaicking [20] validated that this algorithm identifies important features and removes unnecessary ones.

This paper proposes an approach for Farsi font detection, which works based on an improved version of the SURF algorithm. For this purpose, first an Adaptive RKEM (A.RKEM) is presented to eliminate redundant keypoints of the SURF algorithm. The proposed method operates based on the adaptive calculation of the threshold in the RKEM-SURF method. In fact, the threshold value is determined based on the amount of dispersion (variance) of keypoints distances. Hence, the type of the images and the between distortions are considered. These points can lead to the improvement in the RKEM-SURF efficiency via eliminating redundant points and consequently enhancing the image matching performance. Another improvement with respect to the RKEM-SURF is that the threshold value in the input and pattern images are found separately, again leading to a more successful image matching process. Afterwards, feature descriptors are

extracted using the SURF algorithm. This descriptor is robust against rotation, scale and brightness. It gives vectors of length 64 and has high processing speed. Finally, the matching process is done using the nearest neighbor distance ratio (NNDR) to assign a font type to a query text. Simulation results on a database provided by the authors and standard databases demonstrated that the proposed method achieves higher recognition rates compared to the RKEM-SIFT [18], SURF [21] and also recently published algorithms.

The organization of the rest of paper is as follows. In Section II, review of literature and research method are described. Section III introduces the proposed A.RKEM-SURF algorithm, adapted to the font detection problem. Experimental results and comparisons are presented in Section IV. Finally, the paper is concluded in Section V.

2- Review of Literature and Research Method

In this section, related works is described briefly, and then the RKEM-SIFT algorithm and its problems in font detection are investigated.

2-1- Related Work

In recent years, there has been an increasing interest in the font detection. A considerable amount of literature has been published on this issue, some of which are referred to in this section. In [6], the statistical analysis of edge pixels relationship was used to detect Arabic fonts. In [9], the third and fourth-order moments were used as global texture features to recognize eight types of Spanish fonts. In [13], the authors proposed Sobel-Roberts gradient in sixteen dimensions for feature extraction to detect Farsi fonts. To classify fifteen Arabic fonts, the authors of [8] used scale-invariant detectors such as SIFT and DOG to extract keypoints and used the SIFT descriptor to describe the features. In [22], the authors used sixteen channels of Gabor filter in four directions and four sizes for feature extraction to detect eight types of English fonts and six types of Chinese fonts. In [23], Gaussian mixture model was used to extract features for detecting ten types of Arabic fonts. In [24], correlation coefficients were used to extract the features to detect the Farsi fonts. In [25], wavelet transform and neural network were used for feature extraction and classification respectively, to detect Arabic fonts. In [26], stack and points were used to extract features for detecting seven fonts. In [27], the holes in the characters and horizontal projection profile of text lines were used to extract features in detecting Farsi font. In [28], the SIFT algorithm was used to identify and describe the features and the matches based on the nearest neighbor technique to detect Farsi and Arabic fonts. In [29], redundant oriented LBP features were used for features

recognition and the nearest neighbor for the classification in the process of detecting ten types of Arabic fonts.

2-2-RKEM-SIFT Algorithm and Its Problems in Font Detection

In this section, the RKEM-SIFT algorithm is described briefly, and then the disadvantages of this algorithm in the font detection application are reported.

2-2-1-RKEM-SIFT Algorithm

The RKEM-SIFT is an improved form of the SIFT algorithm; which is used to remove redundant features generated by the SIFT algorithm. The feature extraction step in this algorithm consists of four phases, respectively including: extracting scale-space extrema, improving accuracy of localization and eliminating unstable extrema, allocating orientation to each generated feature, and removing unnecessary keypoints. In the step of removing redundant keypoints, it computes the distances between the keypoints in each image. Afterwards, for any pair of keypoints with a distance less than a pre-determined threshold value, one keypoint is deleted and the other one is kept for the matching process, according to a redundancy index. For more details, we refer the reader to [18].

2-2-2-RKEM-SIFT Algorithm Problems in Font Detection

Despite the high performance of the RKEM-SIFT algorithm, experiments of the present study showed that

this method has some problems in the font detection task. The major shortcoming is that the mentioned threshold value for the keypoints' distances should be determined experimentally. In choosing the threshold value in the RKEM-SIFT, the image type (e.g., natural images, texts image, etc.) is not considered as an operational factor. Accordingly, some redundant keypoints may not be eliminated or some useful ones be removed unwantedly. Furthermore, this threshold value is considered the same for both the query and training images and the distortion between images is not considered. In addition, the RKEM-SIFT is not suitable for real-time applications. As one of the improved versions of the SIFT, the SURF algorithm was proposed by Bay [21] in 2006 for the feature matching in images. This algorithm has advantages including time efficiency, robustness against scale and rotation changes and also intensity variations; but it detects several redundant keypoints.

3- Proposed Method

The proposed font detection process consists of three phases, as Fig. 1 shows. At first, the initial features of each image (input images, pattern image) are detected using the proposed A.RKEM-SURF method and then the descriptors are extracted using the SURF algorithm. Finally, the matching process is done by the nearest neighbor distance ratio criteria.

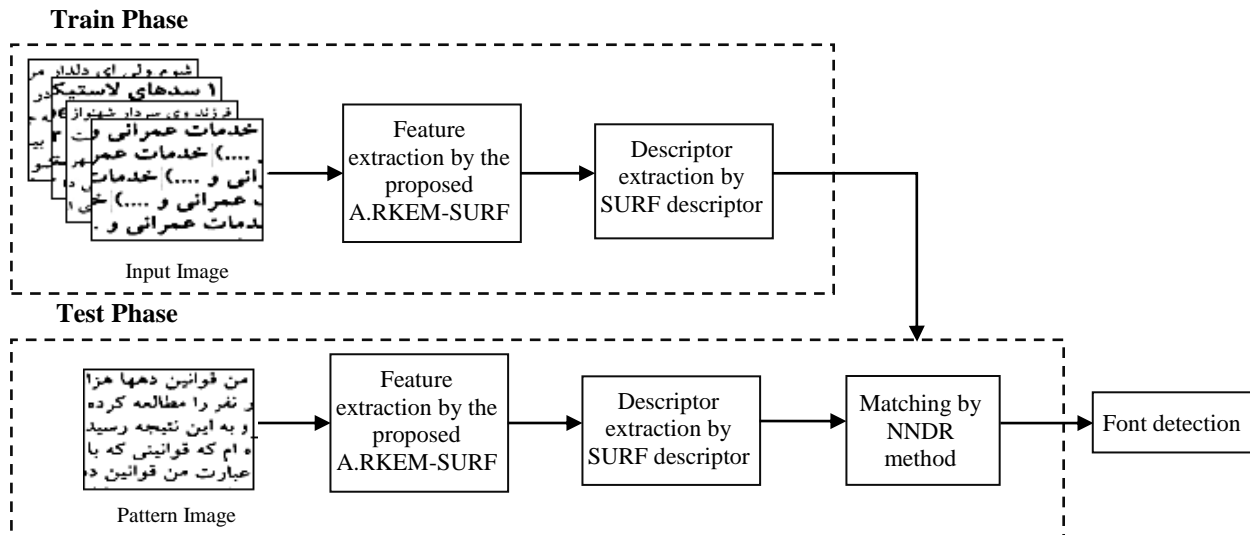


Fig.1 Block diagram of proposed method for Farsi font detection

3-1-Feature Extraction

In this subsection, the A.RKEM method is proposed and then used for the feature extraction. The details are described as follows.

3-1-1-Initial Futures Detection

At the first step, the scale-space extrema are detected, the keypoints are accurately localized and the orientation for each keypoint is assigned based on the classical SURF algorithm [17].

3-1-2-Final Futures Detection using the A.RKEM Method

In this step, the A.RKEM method is proposed and used for the identification of the final features. The aim is to find feature vectors which are too close to each other; then remove unnecessary ones and keep the rest. Notice that the following stages are performed for each image individually. The Manhattan distance between any two keypoints (e.g., p_m and p_j) is calculated (i.e., $d(p_m, p_j)$) for all the keypoints according to (1) [18].

$$d(p_m, p_j) = \sum_{i=1}^l |p_m(i) - p_j(i)| \quad (1)$$

In which, $p_m(i)$ and $p_j(i)$ are the i^{th} coordinate of the keypoints p_m and p_j , respectively and l is the length of any keypoint vector in the image.

An integer value n_{\max} is determined to represent the maximum number of threshold integer values (i.e., $1, 2, \dots, n_{\max}$). Now, the following two stages are performed for all the n_{\max} number of threshold values:

- 1st stage: If the distance of p_m to any other keypoint is greater than the threshold value n , then p_m is kept in R_n . The set R_n collects those keypoints whose distances to all others are greater than n .

$$R_n = \{p_m | d(p_m, p_j) > n; m, j = 1, 2, \dots, M\} \quad (2)$$

$$n = 1, 2, \dots, n_{\max}$$

In (2), M is the total number of keypoints.

- 2nd stage: The variance of the set R_n is calculated according to (3).

$$\sigma_n^2 = \text{var}(R_n) \quad n = 1, 2, \dots, n_{\max} \quad (3)$$

The purpose of this algorithm is selecting the optimal threshold value for removing redundant keypoints. It deserves to be noticed that the original RKEM sets this value experimentally at 3 [18], without considering the type or other specifications of the images. To the optimal selection aim, the threshold value with minimum variance of the keypoints' distances is selected according to (4). The reason is that smaller dispersion of keypoints is due to existence of less redundant points.

$$\text{Optimal Thershold} = \arg \min(\sigma_n^2) \quad (4)$$

If the distance between each two distinct keypoints in the image is less than the optimal threshold, the unnecessary keypoint should be removed. In this condition, the keypoint with higher Redundancy Index (RI), defined in (5), is considered redundant and thus removed [18].

$$RI(p_m) = 1/SD(p_m) \quad (5)$$

In (5), $SD(p_m)$ is the summation of the distance values between the keypoint p_m and all other ones.

The presented A.RKEM method automatically finds the threshold value for each image, independent of the others. Accordingly, the image type and distortions are considered when adjusting the threshold value. This method leads to accurate removal of redundant keypoints and ultimately increases the matching accuracy. The A.RKEM is not limited to the font detection application, as it can be used in any task that SIFT and its variants are applied.

3-2-Descriptor Extraction

To carry out the image matching, different descriptors could be applied. These descriptors are generally categorized into three groups: distribution-based descriptors, special frequency techniques-based descriptors, and differential descriptors [30]. The distribution-based descriptors use histograms to represent different appearance characteristics. They are robust against geometric aberrations. One main disadvantage of these descriptors is that their dimensions are large. Shape context, SIFT and its improved versions (e.g., SURF and GLOH) are some examples of these descriptors. The special frequency techniques-based methods describe the frequency content of an image. Fourier transform is one of the basic techniques of this group of descriptors that breaks an image content down into basic functions. However, the spatial relationships between points are not clear and the basic functions are unlimited; thus, it is not suitable for adapting local approaches. Other examples of these descriptors are Gabor and Wavelet filters; which overcome the mentioned problems in the Fourier transform. But a large number of these filters are needed to describe small changes in frequency. Differential descriptors use image derivatives for description. Steerable and complex filters are two examples of these descriptors. Among the above-mentioned methods, the distribution-based descriptors such as the SIFT, SURF and the GLOH have higher matching accuracies than others; while the differential descriptors perform the least [31].

SURF is an example of the distribution-based descriptors that was proposed by Herbert Bay in 2006 [21]. As an extended speed-up version of the SIFT, the SURF is both a detector and a descriptor. The SURF algorithm consists of four stages including (1) keypoints detection, (2) keypoints positioning, (3) direction assignment, and (4) descriptors

creation for keypoints. SURF descriptor uses integral images in conjunction with Haar wavelet filters in order to increase the robustness and decrease computation time [21]. Haar wavelets are simple filters which can be used to find gradients in the x and y directions. Extracting the descriptor can be divided into two distinct steps. The first step is to construct a square window around the required point. This square window contains the pixels which form entries in the descriptor vector and its size is 20σ ; where σ refers to the scale at which the point was detected. Furthermore, the window is oriented along a computed direction. Since all subsequent calculations are relative to this direction, this direction is important to be found to be repeatable under varying conditions. To determine the orientation, Haar wavelet responses of size 4σ are calculated for a set of pixels within a radius of 6σ of the detected point. The specific set of pixels is determined by sampling those inside the circle using a step size of σ . The responses are weighted with a Gaussian function, centered at the required point. In keeping with the rest of the Gaussian is dependent on the scale of the point and chosen to have standard deviation equal to 2.5σ . Once weighted the responses are represented as points in vector space, with the x - responses along the abscissa and the y -responses along the ordinate. The dominant orientation is selected by rotating a circle segment covering an angle of $\pi/3$ around the origin. At each position, the x and y -responses within the segment are summed and used to form a new vector. The longest vector lends its orientation the interest point. The descriptor window is divided into 4×4 regular sub regions. Within each of these sub regions Haar wavelets of size 2σ are calculated for 25 regularly distributed sample points. If we refer to the x and y wavelet responses by dx and dy respectively, then for these 25 sample points, we collect the vector $[\sum dx, \sum dy, \sum |dx|, \sum |dy|]$ for each sub-region to create the 64-D descriptor vector. The resulting SURF descriptor is invariant to rotation, scale, brightness. In this paper, the SURF descriptor is used. After creating one descriptor for each feature, matching between the target page image descriptor and the training site pages images.

3-3-Matching

In this paper, the matching operation is performed based on the descriptors of each feature. By calculating the Euclidean distance between descriptors in both images and using an appropriate criterion, matching is done. In general, there are three criteria for correct matching between descriptors in two images: threshold-based matching, nearest-neighbors-based matching, and the nearest neighbor distance ratio (NNDR), each of which is described in the following [30].

- Threshold-based matching: if the distance between the descriptors of two keypoints in two images is less than a threshold, the two keypoints are matched. This method, however, has disadvantages; e.g., a descriptor can have several matches.
- Nearest-neighbor-based matching: two regions A and B are matched if the D_B descriptor is the closest neighbor to D_A , and the distance between the two descriptors is less than the threshold. Through this method, a descriptor has only one match.
- Nearest-neighbor distance ratio (NNDR): this method is similar to the nearest-neighbors-based matching. In this method, A and B keypoints are matched if (6) is satisfied [31].

$$\frac{\|D_A - D_B\|}{\|D_A - D_C\|} < T_{ED} \quad (6)$$

In which, D_B is the descriptor of the first nearest neighbor to the descriptor D_A , and D_C is that of the second nearest neighbor to D_A . If the ratio of ‘the distance between the first nearest neighbor to the descriptor’ to ‘the distance of the second nearest neighbor to a given one’ is smaller than a threshold value T_{ED} , the matching is done. The value of T_{ED} is considered equal to 0.8. Since the matching based on the ratio of the first and second nearest neighbor is more accurate than other methods, this criterion is used in this paper for matching [30].

Each pattern page is compared with all the pages in the database. The font of the training page with the maximum number of matches is assigned to the pattern page. If this number for a pattern page is less than a pre-determined threshold, the font of that page is not included in the font bag of the training database.

4- Experimental Results

In this section, we evaluate the performance of the proposed A.RKEM-SURF algorithm and compare it with the RKEM-SIFT[18], also the method of method [28], the Sobel–Roberts features in [13] and the SIFT [8]. All the experiments are performed on a personal computer with a 2.28 GHz Intel Core i7, 16G RAM using the MATLAB[®] 2015A software. The database, evaluation criteria, and experimental results are presented in the next subsections.

4-1-Databases

We used four datasets to evaluate the proposed *Adaptive* RKEM-SURF method. The first dataset is provided by the authors of this paper via printing and scanning the Farsi translation of ‘Le Petit Prince’ (The Little Prince) book, written by Antoine de Saint-Exupéry originally in French. It contains 46 pages with 425×550 pixels printed in 20 different Farsi fonts. Each font is written in four

sizes: 6, 10, 14, 18 with four different styles: normal, **bold**, *italic* and **bold-italic**. The second dataset includes 1400 text images. It contains 500 pages printed in 10 different Farsi fonts; each of which is written in sizes 11-16 [13]. The third and the fourth databases are the printed/scanned versions of the Arabic and English translations of ‘The Little Prince’ book, produced by this paper authors.

4-2-Evaluation Criteria

Classical evaluation criteria including the recognition rate (matching accuracy) and recall according to (7-8) are used to evaluate the effectiveness of the font detection methods [32].

$$Accuracy = TP/m \quad (7)$$

$$Recall = TP/P \quad (8)$$

In (7-8), TP is the number of correct matches, P is the number of correspondences and m is the total number of matches. If the accuracy and recall are high, the performance of the system is acceptable.

4-3-Setup of Experiments

Six sets of experiments were performed to evaluate the performance of the proposed *Adaptive* RKEM-SURF method. In the first and second sets, the performance on different sizes and different styles scenarios were evaluated, respectively. In the third set, the performance was investigated in both different sizes and styles. In the fourth set, comparison with other classical methods such as RKEM-SIFT [18], methods of [28], the Sobel–Roberts features in [13] and the SIFT [8] were presented.. In the fifth set, the performance in multi-language texts was evaluated. Finally, to evaluate the performance of the proposed A.RKEM-SURF method on images with simulated noise was evaluated.

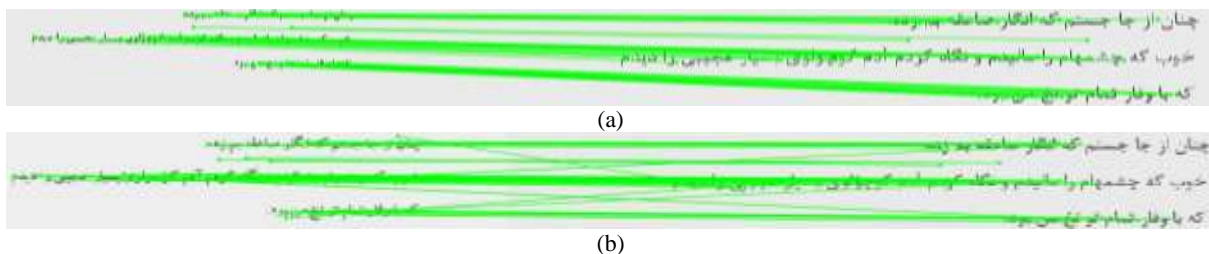


Fig. 2 Font detection using the proposed method for texts with different sizes. (a) texts with sizes (10,18), (b) texts with sizes (14,18). The first/second number in the parenthesis represents the size of the text characters in the pattern/database image.

Table 1: Font detection results on text images with different sizes.

Sizes	Method	Recall (%)	Accuracy (%)
(6,10)	RKEM-SIFT	94.6	98.2
	A.RKEM-SURF	97	99.5
(6,18)	RKEM-SIFT	83.15	89.3
	A.RKEM-SURF	92.7	97.7

4-3-1-Experiments on Different Sizes

In this test, text images with different sizes are used, and the performance of the proposed A.RKEM-SURF method is assessed. Fig. 2 and Table 1 validated that the effect of applying the proposed A.RKEM-SURF method is appropriate in the font detection application, according to the high rates of the recognition (accuracy) and the recall criteria.

4-3-2-Experiments on Different Styles

In this experiment, text images with different styles are used, and the performance of the A.RKEM-SURF is compared with the RKEM-SIFT. The results shown in Table 2 confirm than the performance of the proposed A.RKEM-SURF method is high in detecting the fonts for texts with different styles.

4-3-3-Experiments on Different Sizes and Styles

In this test, we used images with different sizes and styles to evaluate the performance of the proposed font detection method. The results of this test are shown in Fig. 3; from which, it is easy to conclude that the proposed A.RKEM SURF works well on images with different sizes and styles.

4-3-4-Comparison to other Methods

An experiment was conducted to compare the performance of the proposed font-detection method with other classical methods, such as [8], [13], and [28]. The results are reported in Table 3, which show that the proposed A.RKEM-SURF method outperformed the methods of [8] and [13], in term of the recognition rate. Although the proposed method achieved the accuracy value same as that of [28]; it was faster as the last column demonstrates.

(10,14)	RKEM-SIFT	78	87.7
	A.RKEM-SURF	88.1	95.8
(10,18)	RKEM-SIFT	72.65	85.1
	A.RKEM-SURF	85.42	94.6
(14,18)	RKEM-SIFT	68.09	84.2
	A.RKEM-SURF	80.6	93.2

Table 2: performance comparison in different styles scenario.

Sizes	Method	Recall (%)	Accuracy (%)	Run time (s)
Normal	RKEM-SIFT	91.03	100	14.62
	A.RKEM-SURF	98.78	100	10.45
Italic	RKEM-SIFT	90	99.9	13.81
	A.RKEM-SURF	95.43	100	11.94
Bold	RKEM-SIFT	86.52	100	16.43
	A.RKEM-SURF	92	100	13.07
Bold-Italic	RKEM-SIFT	75.18	97.6	18.72
	A.RKEM-SURF	89.32	99.2	15.61

Table 3: Comparison of the proposed method with some others

Method	Number of fonts	Recognition rate	Required time per sample (ms)
[8]	15	99.5	4.02
[13]	10	94	3.78
[28]	20	100	3.25
A.RKEM-SURF	20	100	2.14

Table 4: Recognition rates for texts written in English and Arabic

Language	Method	Accuracy (%)
English	A.RKEM-SURF	100
Arabic	Method of [8]	98.1
	A.RKEM-SURF	100

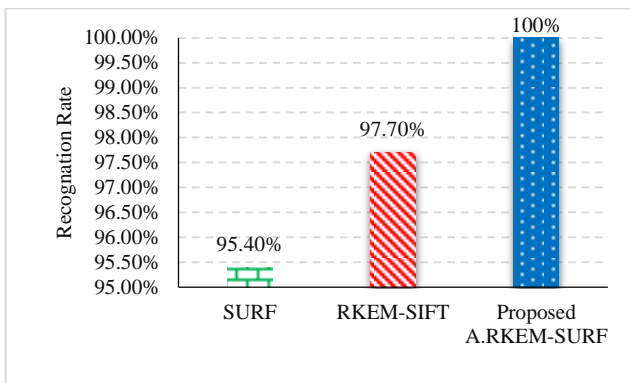


Fig. 3 Overall recognition rate in images with different sizes and styles

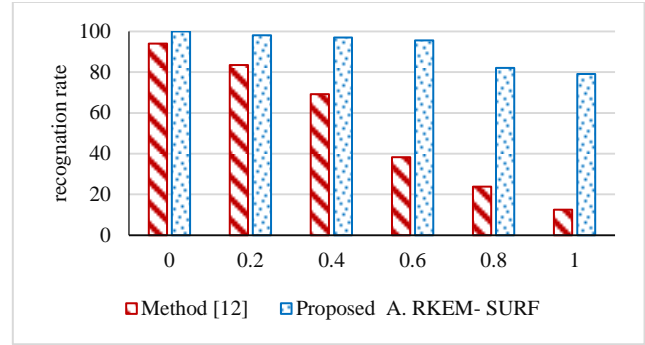


Fig. 4 Comparison of font detection rate on images contaminated by additive Gaussian noise with one mean and different variances.

4-3-5-Experiments on other Languages

In this test, the performance of the proposed method in detecting fonts of texts written in English and Arabic is assessed. The results are shown in Table 4, which show that the proposed A.RKEM-SURF method works well in the font detection task for English and Arabic texts. Also, this method performs better than the method [8].

4-3-6-Experiments on Images with Simulated Noise

In this test, to evaluate the performance of the A.RKEM-SURF method on noisy images, Gaussian noise with a mean of one and a variance between zero to one is added to the text images with ten types of fonts. This test is important since images are usually taken using low quality scanners. The results are shown in Fig. 4. It is easy to infer that the font detection rate in both compared methods decreases with the increase of the noise variance. Yet, the accuracy of the A.RKEM-SURF method is higher than that of the method of [13]. This indicates the appropriate functioning of the proposed A.RKEM-SURF method against noise.

5- Conclusion

Farsi language has challenging characteristics for OCR that elevates the need for the FOFR. Font detection is an essential step in the OCR systems. Thus, one main phase of recognizing Farsi characters is to detect the Farsi font of the written text. In this paper, a new three-step algorithm is presented for the purpose. The A.RKEM-SURF is introduced and used for the feature extraction step. Then SURF is used as the descriptor and NNDR is utilized for the matching step. The simulation results of the proposed method show a promising performance in the font detection task. Not only very good recognition rates are obtained in general, but also particular fonts (e.g. Tabassom) which are known to be weak points for other FOFR methods, are successfully identified. Additionally,

the proposed method was shown to work well in recognizing fonts in texts written in English and Arabic.

Acknowledgments

The authors would like to thank the Journal of Information Systems and Telecommunication (JIST) and Applications Associate Editor and the anonymous reviewers for their valuable comments.

References

- [1] Mostafa Rahmandoust, "The Death of Pitcher: Persian Parables and its Stories," Tehran: School Publications, 1387.
- [2] University of Texas: New Persian Language. https://w3techs.com/technologies/overview/content_language/all/
- [4] A. Zramdini and R. Ingold, "Optical font recognition using typographical features," *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 20, pp. 877-882, 1998.
- [5] M. Rajdev, D. Sahay, S. Khare, and S. Nainan, "Optical Character and Font Recognizer," in *International Conference on Advances in Computing and Data Sciences*, 2019, pp. 477-486.
- [6] B. Bataineh, S. N. H. S. Abdullah, and K. Omar, "A statistical global feature extraction method for optical font recognition," in *Asian Conference on Intelligent Information and Database Systems*, 2011, pp. 257-267.
- [7] S. Ghosh, P. Roy, S. Bhattacharya, and U. Pal, "Large-Scale Font Identification from Document Images," in *Asian Conference on Pattern Recognition*, 2019, pp. 594-600.
- [8] M. A. Mousa, M. S. Sayed, and M. I. Abdalla, "An efficient algorithm for Arabic optical font recognition using scale-invariant detector," *International Journal on Document Analysis and Recognition (IJ DAR)*, Vol. 18, pp. 263-270, 2015.
- [9] C. Avilés-Cruz, R. Rangel-Kuoppa, M. Reyes-Ayala, A. Andrade-Gonzalez, and R. Escarela-Perez, "High-order statistical texture analysis—font recognition applied," *Pattern Recognition Letters*, Vol. 26, pp. 135-145, 2005.
- [10] V.-C. Juan and A.-C. Carlos, "Font recognition by invariant moments of global textures," in *Proceedings of international workshop VLVB05 (very low bit-rate video-coding 2005)*, 2005, pp. 15-16.
- [11] M. Lutf, X. You, Y.-m. Cheung, and C. P. Chen, "Arabic font recognition based on diacritics features," *Pattern Recognition*, Vol. 47, pp. 672-684, 2014.
- [12] G. D. Joshi, S. Garg, and J. Sivaswamy, "A generalised framework for script identification," *International Journal of Document Analysis and Recognition (IJ DAR)*, Vol. 10, pp. 55-68, 2007.
- [13] H. Khosravi and E. Kabir, "Farsi font recognition based on Sobel–Roberts features," *Pattern Recognition Letters*, Vol. 31, pp. 75-82, 2010.
- [14] J. H. AlKhateeb, J. Ren, S. S. Ipson, and J. Jiang, "Knowledge-based baseline detection and optimal thresholding for words segmentation in efficient pre-processing of handwritten Arabic text," in *Fifth International Conference on Information Technology: New Generations (itng 2008)*, 2008, pp. 1158-1159.
- [15] C. ERGÜN and S. Norozpour, "Farsi document image recognition system using word layout signature," *Turkish Journal of Electrical Engineering & Computer Sciences*, Vol. 27, pp. 1477-1488, 2019.
- [16] A. Mohammadhani, F. Marvari, and M. Moshki, "Font Recognition of Persian Letters and Printed Words," *16th Iranian Electrical Engineering Student Conference*, Kashan, Electrical Engineering Student Scientific Organization, 2013.
- [17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, Vol. 60, pp. 91-110, 2004.
- [18] Z. Hossein-Nejad and M. Nasri, "RKEM: Redundant Keypoint Elimination Method in Image Registration," *IET Image Processing*, Vol. 11, pp. 273-284, 2017.
- [19] Z. Hossein-Nejad and M. Nasri, "Copy-Move Image Forgery Detection Using Redundant Keypoint Elimination Method," in *Cryptographic and Information Security Approaches for Images and Videos*, S. Ramakrishnan, Ed. Boca Raton: CRC Press, pp. 773-797, 2019.
- [20] Z. Hossein-Nejad and M. Nasri, "Natural Image Mosaicing based on Redundant Keypoint Elimination Method in SIFT algorithm and Adaptive RANSAC method," *Signal and Data Processing*, 2020.
- [21] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer vision and image understanding*, Vol. 110, pp. 346-359, 2008.
- [22] Y. Zhu, T. Tan, and Y. Wang, "Font recognition based on global texture analysis," *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 23, pp. 1192-1200, 2001.
- [23] F. Slimane, S. Kanoun, A. M. Alimi, R. Ingold, and J. Hennebert, "Gaussian mixture models for arabic font recognition," in *2010 20th International Conference on Pattern Recognition*, 2010, pp. 2174-2177.
- [24] E. Rashedi, E. Nezam abadipour, and a. S. Saryazdi, "Farsi font recognition using correlation coefficients (in Farsi)," *4th Conf. on Machine Vision and Image Processing*, Ferdosi Mashhad, 2007.
- [25] N. E. B. Amara and S. Gazzah, "Une approche d'identification des fontes arabes," in *Conférence Internationale Francophone sur l'Écrit et le Document (CIFED 04)*, 2004.
- [26] Y. Pourasad, H. Hassibi, and A. Ghorbani, "Farsi font recognition using holes of letters and horizontal projection profile," in *International Conference on Innovative Computing Technology*, 2011, pp. 235-243.
- [27] Y. Pourasad, H. Hassibi, and A. Ghorbani, "Farsi font recognition in document images using PPH features," *International Journal of Natural and Engineering Sciences (IJNES) E-ISSN: 2146-0086*, Vol. 2, pp. 17-20, 2011.
- [28] M. Zahedi and S. Eslami, "Farsi/Arabic optical font recognition using SIFT features," *Procedia Computer Science*, Vol. 3, pp. 1055-1059, 2011.
- [29] A. Nicolaou, F. Slimane, V. Maergner, and M. Liwicki, "Local binary patterns for arabic optical font recognition," in *2014 11th IAPR International Workshop on Document Analysis Systems*, 2014, pp. 76-80.
- [30] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *Pattern Analysis and Machine*

Intelligence, IEEE Transactions on, Vol. 27, pp. 1615-1630, 2005.

- [31] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," IEEE transactions on pattern analysis and machine intelligence, Vol. 27, pp. 1615-1630, 2005.
- [32] H. Agahi, A. Mahmoodzadeh, and M. Salehi, "Handwritten Digits Recognition Using an Ensemble Technique Based on the Firefly Algorithm," Information Systems & Telecommunication, pp. 136, 2018.

Zahra Hossein-Nejad received the B.Sc. and M.Sc. degrees in Electrical Engineering from the Islamic Azad University, Jahrom, and Sirjan branches, Iran, in 2013 and 2016, respectively. She is a Ph.D. student in Electrical Engineering at Shiraz University, Shiraz, Iran. Her research interests include image processing and computer vision.

Hamed Agahi received B.Sc. M.Sc. and Ph.D. degrees in Electrical Engineering from University of Shiraz, Amirkabir University of Technology and University of Tehran, Iran, in 2005, 2008 and 2013, respectively. From 2009, he was with the Islamic Azad University, Shiraz Branch, Shiraz, Iran. His research interests include pattern recognition, image and signal processing, and fault detection and diagnosis applications.

Azar Mahmoodzadeh received B.Sc., M.Sc. and Ph.D. degrees in Electrical Engineering from University of Shiraz, University of Shahed and University of Yazd, Iran, in 2005, 2008 and 2013, respectively. From 2009, she was with the Islamic Azad University, Shiraz Branch, Shiraz, Iran. Her research interests include pattern recognition and image and signal processing.