IJDAR (2015) 18:263–270
DOI 10.1007/s10032-015-0248-9

International Journal on Document Analysis and Recognition (Springer)

ORIGINAL PAPER

An efficient algorithm for Arabic optical font recognition using scale-invariant detector

Mahmoud A. A. Mousa[1,2] · **Mohammed S. Sayed**[3,4] · **Mahmoud I. Abdalla**[3]

Received: 6 May 2014 / Revised: 20 April 2015 / Accepted: 10 June 2015 / Published online: 2 July 2015

Abstract This paper proposes a new algorithm for Arabic optical font recognition (AOFR) as the first stage for Arabic optical character recognition. The proposed algorithm uses scale-invariant detector, gradient-based descriptor, and *k*-means clustering. The scale-invariant detector is used to find key points that identify the font of an image of printed Arabic text. The work in this paper compares between several scale-invariant detectors and selects the best one for AOFR. A gradient-based descriptor similar to the one in the famous scale-invariant feature transform algorithm is used to describe the detected key points. In addition, *k*-means clustering is used for font classification. In this paper, the mean recognition rate is used to evaluate the performance of the proposed algorithm. The proposed algorithm shows superior performance when compared with recently published algorithms for AOFR.

Keywords Arabic optical font recognition (AOFR) · Arabic optical character recognition (OCR) · Scale-invariant detectors

1 Introduction

Font recognition is a step which is very useful in multi-fonts optical character recognition (OCR) system. Arabic language has characteristics that represent challenges for OCR. The first characteristic is the cursive nature of the Arabic text which is written from right to left. The second and the most important characteristic is that Arabic characters may have different shapes for the same character, and this depends on the location of the character in the word [1]. Also, the character shape changes from font to another. Hence, optical font recognition is an essential step in Arabic OCR systems [1,8]. Font identification is used in OCR systems to detect the characters dataset in which the classification process will take place. Figure 1 shows different Arabic fonts used in this paper. Some fonts are similar in most characters' shapes, and this presents a big challenge.

This paper presents a new algorithm for Arabic optical font recognition (AOFR) as the first stage for Arabic optical character recognition (AOCR). The proposed algorithm uses scale-invariant detector, gradient-based descriptor, and *k*-means clustering to identify the used Arabic font. The rest of the paper is organized as follows. Section 2 reviews several optical font recognition algorithms. Section 3 gives a brief overview on scale-invariant detectors and descriptors. Section 4 presents the proposed algorithm for AOFR. Section 5 demonstrates the results and performance analysis. Section 6 concludes this paper.

✉ Mohammed S. Sayed
mohammed.sayed@ejust.edu.eg

Mahmoud A. A. Mousa
mamosa@zu.edu.eg

Mahmoud I. Abdalla
mabdalla@zu.edu.eg

[1] Department of Computer Science, University of Liverpool, Liverpool, UK

[2] Department of Computers and Systems Engineering, Zagazig University, Zagazig, Egypt

[3] Department of Electronics and Communications Engineering, Zagazig University, Zagazig, Egypt

[4] Department of Electronics and Communications Engineering, Egypt-Japan University of Science and Technology, New Borg El-Arab, Egypt

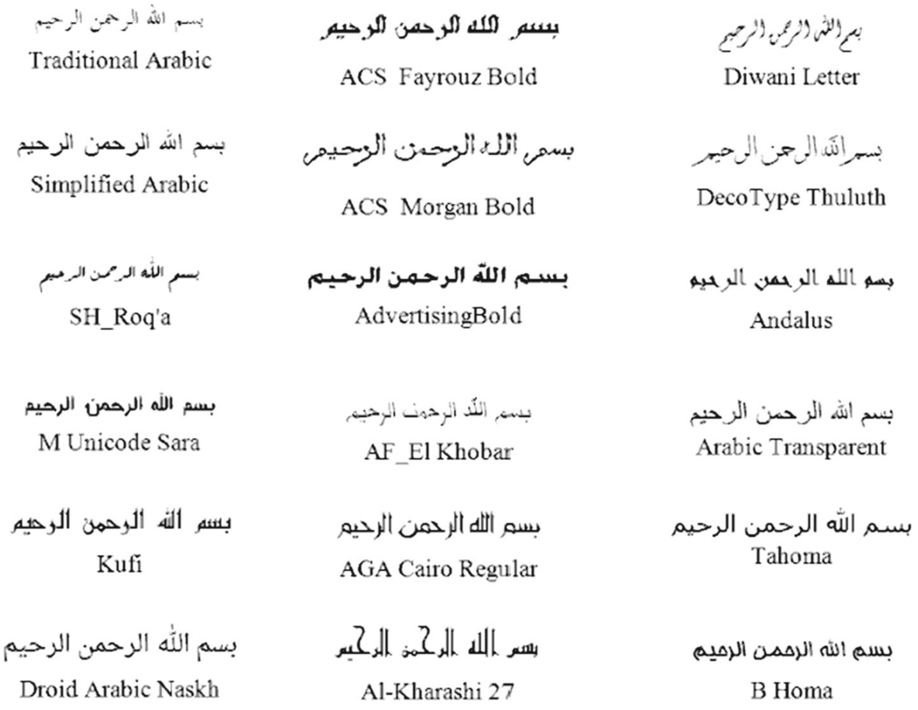

Fig. 1 Different forms for the same sentence according to the font type used

2 Optical font recognition

There are several font recognition approaches. These approaches differ mainly in the method used for features' extraction. Feature extraction stage is important to find a set of measured values that accurately identify the input font. Two common approaches exist in this field.

1. Recognition based on typographical features which analyzes character weights, space width, and various types of projections of the text images.
2. Recognition based on textural features which uses, for example, Gabor filter or wavelet transform to describe the text image.

In [2], a sliding window with a fixed length or width is used to extract a vector of features that is used to identify the font using Gaussian mixture models. The fixed sliding window is used in both the horizontal and vertical directions. For horizontal window with width eight pixels, it uses a window height of 45 pixels. In addition, the vertical window with height eight pixels uses a window width of 20 pixels. Those windows are applied in a sequential manner. For each iteration, the window moves by 1 pixel from right to left or from top to bottom. This algorithm was applied on images with text lines level so character segmentation is not required to be done. Training and test stages were done on 10 different fonts. Each font has 10 different sizes in the training image databases. This algorithm gives 93.2 % recognition rate for those 10 fonts. The output of the binarization for dots and broken stocks is used as features in [3]. It uses the size of the bonding box that contains the binarized points as a feature to recognize a specific font. This algorithm was applied on seven different fonts with seven different sizes each and gives a recognition rate of 95.7 %.

An algorithm based on the statistical analysis of edge pixel relationships was presented in [4]. A preprocessing operation is done on the text image in order to remove the spaces between words, subwords, and lines. The output of the preprocessing stage is inputted to a Laplacian filter. Features are extracted by applying an eight neighboring kernel matrix which relates the scoped pixel to their neighboring pixels. This paper uses 18 features by calculating homogeneity, pixel regularity, weights, edge direction, and edge regularity. It gives a maximum recognition rate of 97.85 %. The algorithm in [5] proposed wavelet-based feature extraction in Arabic OFR approach to produce a feature vector which was used with neural network in the recognition stage. This algorithm gives recognition rate of 99 % in single font multi-size system. It also gives 96.5 % recognition rate when applied on 10 fonts with five different sizes each.

In [6,7], features from the holes of the letters and the horizontal profile projection of the text lines are used. These algorithms work at the text line images level to identify the font type. They depend on observation that the shape of the holes which found in letters is varied according to the font used, and this can be used to extract features distinct for font classes. The algorithm in [6] uses some features from the horizontal profile of the text line image such as the height of the text line, the distance between top of the text line and the base line, the distance between bottom of text line and the base line, and the location of both the 2nd and 3rd maximums with respect to baseline. The algorithm was used to recognize seven different fonts with seven different sizes each. It gives a recognition rate of 93.7 %. Slimane et. al. presented in [8] a stochastic-based approach to tackle the problem of font and size recognition. Their method treats a word image with a fixed-length overlapping sliding window where each window is represented with a 102 features whose distribution is captured by Gaussian mixture models (GMMs).

3 Scale-invariant detectors and descriptors

3.1 Detectors

The scale-invariant detectors such as Hessian, difference of Gaussian (DoG), and Harris-Laplace were designed such that they find a set of key points that are not affected by the variations in image scale or orientation. Those key points such as corners exhibit signal changes in two directions. Those detectors [9–12] will be discussed briefly as follows:

- *Harris-Laplace detector* computes the multi-scale representation for the Harris interest point detector. Harris detector uses the second moment matrix shown in Eq. (1) which describes the gradient distribution in the local neighborhood of a point in the image. The extracted points are those for which curvatures are significant such as corners and junctions and achieve Eq. (2). The interest points are selected at which the Laplacian local measure is maximal over scales.

$$\begin{aligned}\mu\left(x,\sigma_I,\ \sigma_D\right) &= \begin{bmatrix} \mu_{11} & \mu_{12} \\ \mu_{21} & \mu_{22} \end{bmatrix} \\ &= \sigma_D^2\, g\left(\sigma_I\right) * \begin{bmatrix} L_x^2\left(x,\sigma_D\right) & L_x L_y\left(x,\sigma_D\right) \\ L_x L_y\left(x,\sigma_D\right) & L_y^2\left(x,\sigma_D\right) \end{bmatrix}\end{aligned} \tag{1}$$

where:

σ_D is the differentiation scale, σ_I is the integration scale
L_x and L_y are the first-order derivation in *x* and *y*

$$\begin{aligned}\text{Differentiation scale} &= 0.7 * \text{Integration Scale} \\ \text{Cornerness} &= \det\left(\mu\left(x,\sigma_I,\sigma_D\right)\right) \\ &\quad -\alpha\, \text{trace}^2\left(\mu\left(x,\sigma_I,\sigma_D\right)\right) > \text{Threshold}\end{aligned} \tag{2}$$

- *Hessian detector* uses Harris corner measure on the second moment matrix to rely on interest points which are detected at multiple scales. It uses a multiple-scale iterative algorithm to localize and select scale and affine invariant points. Choosing the interest points is done by studying Hessian matrix shown in Eq. (3) at each individual scale. The determinant and trace of Hessian matrix are calculated at each point. Interest points are those which achieve local maximum of both the trace and the computed determinant.

$$H = \begin{bmatrix} I_{\text{xx}}\left(x,\sigma_D\right) & I_{\text{xy}}\left(x,\sigma_D\right) \\ I_{\text{xy}}\left(x,\sigma_D\right) & I_{\text{yy}}\left(x,\sigma_D\right) \end{bmatrix} \tag{3}$$

where: I(x) is the image intensity function
I_{xx}, I_{yy}, I_{xy} is the second-order Gaussian smoothed image derivatives

- *Hessian Laplace detector* localizes the interest points of the Hessian matrix using an iterative search based on the Laplacian of Gaussian.
- *Difference of Gaussian (DoG) detector* is done by subtracting blurred version of the image from another less burred one. The blurred image is obtained by a convolution between the original grayscale image and Gaussian kernels with different standard of deviations. The DoG operates as a band-pass filter that discards all the high frequencies that often include noise and keeps spatial frequencies that belongs to the original image. The DoG is used to increase the visibility of edges in image and also used for blob detection.
- *Scale-Invariant Feature Transform (SIFT) detector* consists of several stages. It constructs the scale space representation of the image and studies the image at different scales to find the characteristic scale range in which the interest points are located correctly with the variation in the viewpoints. The DoG is then calculated for the image. Key points are localized which are maxima and minima in the DoG image. A filtering operation is done to remove edges and low-contrast regions. An orientation is calculated for each point, and any further calculations are done relative to this orientation which makes the image rotation invariant besides the scale-invariant property obtained from the first stage.
- *Dense SIFT (D_SIFT) detector* is another version of SIFT detector, but it finds larger number of key points than SIFT.
- Multi-scale approaches such as *multi-scale Harris and multi-scale Hessian* find the features at different image sizes. This will form a very large number of detected points for the same image. On the other hand, many points are detected with the same characteristics but exist at different locations. This increases the mismatching probability at the matching stage.

3.2 Gradient (SIFT) descriptor

The SIFT descriptor is suitable for all the above interest point detectors and achieves good results with them. It encodes the image information in a localized set of gradient orientation histograms. It operates on the regions around the key points extracted from the detectors by sampling these regions using the region scale to select the level of the Gaussian blur [12, 13].

4 The proposed algorithm

Figure 2 shows a text image with the interest points located on it using Hessian detector as an example. The figure shows the number of matched interest points between (a) two different text images with the same font, (b) two images with the same text in different fonts, (c) two images with different texts in different fonts. It is clear from the figure that for two images with the same font, even with different texts, the number of matched key points is very larger (3094 matches in Fig. 2a). However, for two images with different fonts, even with the same text, the number of matched key points is very small (66 matches in Fig. 2b). In addition, when the text and font differ, the number of matched key points is still very small (58 matches in Fig. 2c) that means the number of matched key points does not go high except when the fonts match. Therefore, number of matched key points can be used as an efficient differentiation feature between Arabic fonts.

The proposed algorithm consists of a scale-invariant detector which is used to extract key points. Next, the SIFT gradient local descriptor is used to describe those extracted key points. The features vector outputted from the descriptor is entered to the *K*-means clustering algorithm to make a classification decision [11,14]. Since the recognition rate of our algorithm depends on the selected scale-invariant detector, we studied several scale-invariant detectors to select the most suitable one for AOFR.

As shown in Fig. 1, each font has its own characters shape that must be correctly identified in order to describe the font characteristics. A complete study was done on Arabic characters in order to provide a proof of concept to select the best key point detectors for AOFR. The key point detectors which are suitable for Arabic characters should satisfy the following conditions:

1. Identifies the start and the end curvatures of the characters.
2. Identifies the dash (–) which connects the beginning curvature to the ending curvature.
3. Identifies the Hamza and the points in the characters.

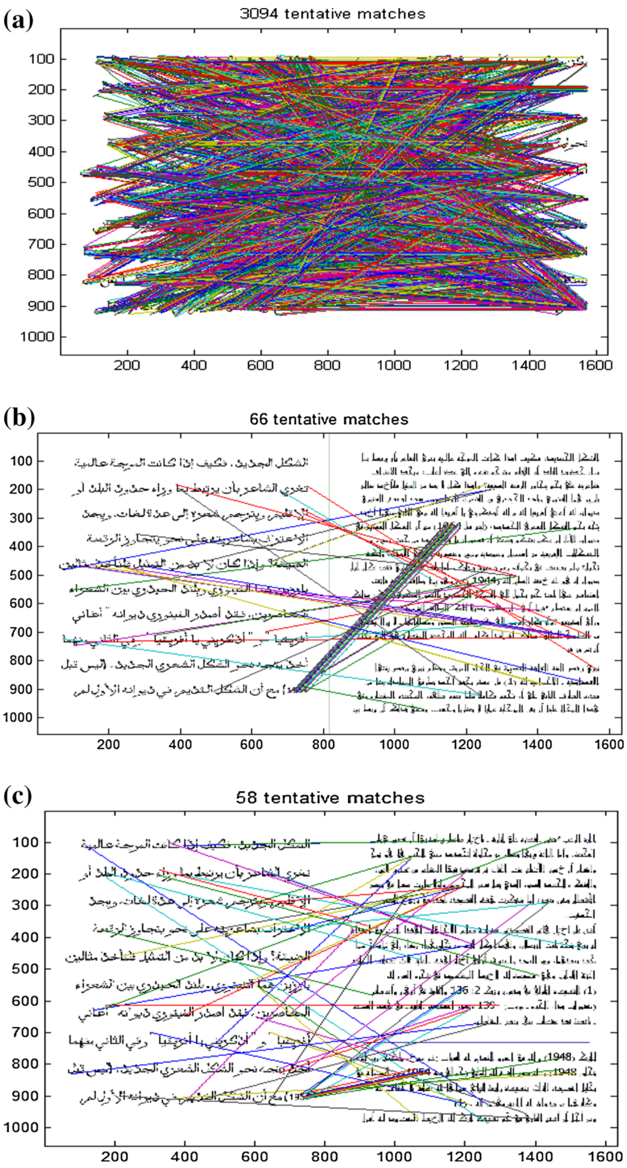

Fig. 2 Matched key points between two text images. **a** Different text and same font. **b** Same text and different fonts. **c** Different text and different fonts

These conditions should satisfy together to achieve good performance. Figures 3, 4 and 5 show the detected key points for characters Alif (أ), Geem (ج), and Kaaf (ك) using *Times New Roman (Headings CS)* font using different detectors. These detectors are Hessian, Hessian Laplace, Harris-Laplace, DoG, multi-scale Harris, multi-scale Hessian, SIFT, and D_SIFT.

Figure 3 shows the key points detected for the character Alif (أ) using different key point detectors. Hessian detector detects key points that describe the Hamza, the beginning, and the end of the character but does not describe the vertical dash which connects the beginning and the end of the character. Hessian Laplace and multi-scale Hessian detectors detect

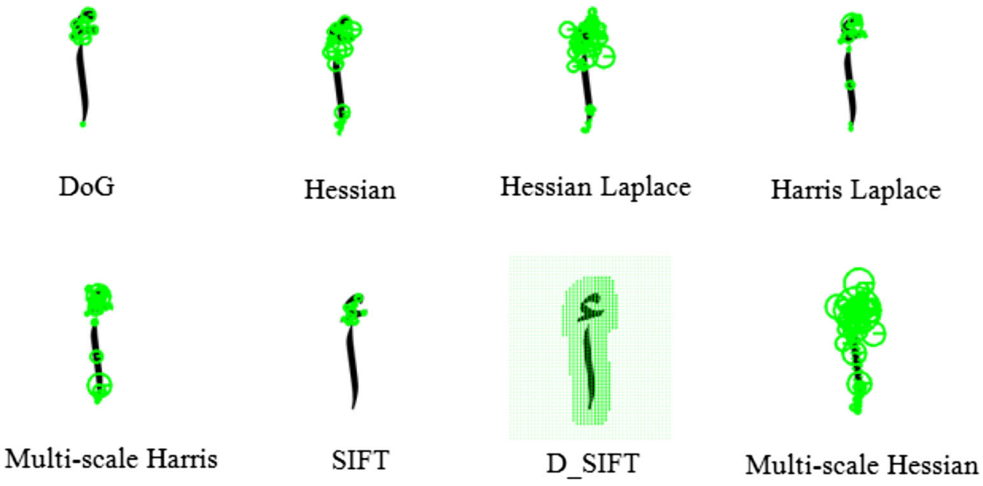

Fig. 3 Key points detected for the character Alif (أ) using various scale-invariant key point detectors

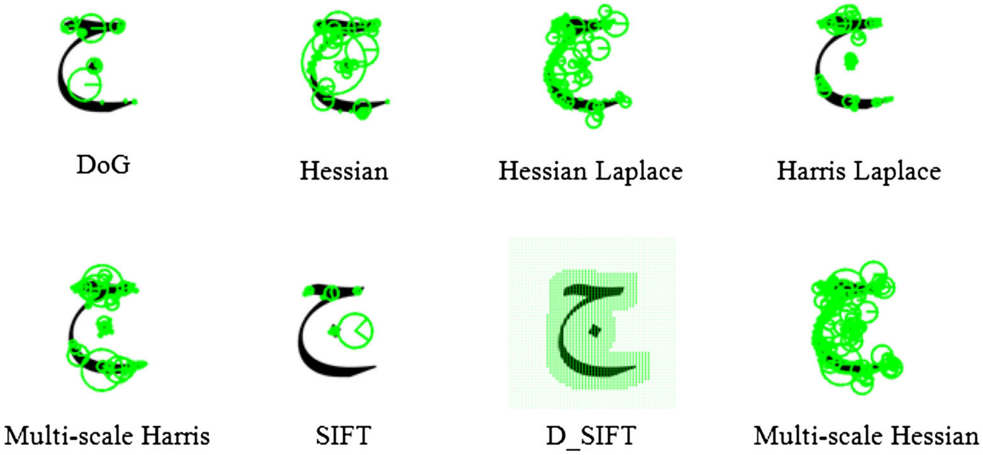

Fig. 4 Key points detected for the character Geem (ج) using various scale-invariant key point detectors

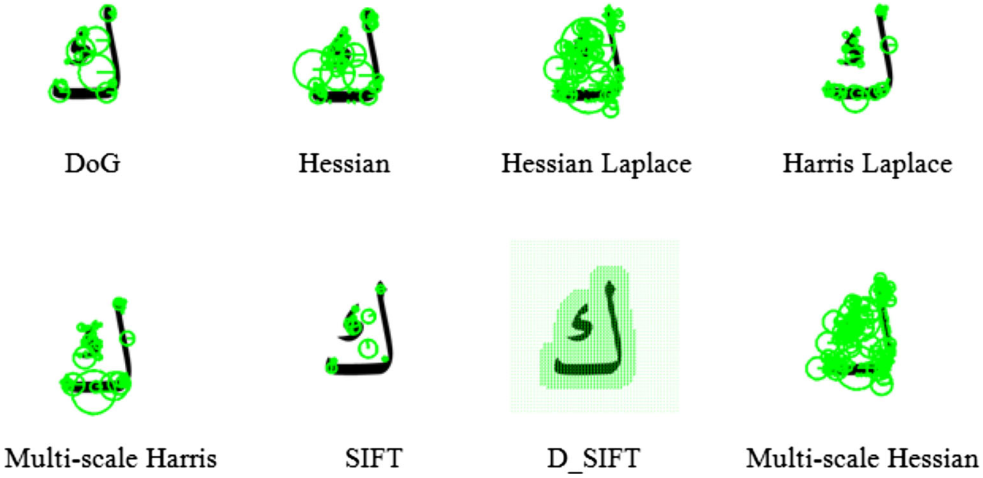

Fig. 5 Key points detected for the character Kaaf (ك) using various scale-invariant key point detectors

key points for Hamza, beginning, end, and the dash, but there are key points detected in the plank area around the Hamza which may cause errors in the recognition. D_SIFT detector detects large number of key points that does not help us in the recognition of the character and decreases the recognition rate. The DoG detector detects key points that describe only the Hamza and the end of the character. The SIFT detector detects only the key points that cover the Hamza. All the previously discussed key points are not suitable for the character Alif (أ). Harris-Laplace and multi-scale Harris detectors are the best ones that cover the character important parts. From Figs. 4 and 5, it can be seen that Harris-Laplace detector is the best one which describes the characters Geem (ج) and Kaaf (ك) with few redundant key points less than other detectors.

After studying all Arabic characters and the key points detected for them, it was concluded that the key points detected with Harris Laplace detector represent characters and their features (i.e., start and end curvatures, dash, Hamza, and points) better than the other detectors. In addition, all the above-mentioned detectors were examined on Arabic text images typed in different fonts to select the best detector as explained the results section. From the testing results, it was concluded that using Hessian or Harris-Laplace detectors results in the best performance for AOFR. Although Harris Laplace detector is slower than Hessian detector, it was selected in the proposed algorithm because it is more robust.

5 Results and performance analysis

The datasets used in our paper is divided into two parts. The first part is to train the system which consists of 20 pages for each font. The second part is to test the performance of the system which consists of 100 pages for each font. Each page consists of Arabic printed text using only one font with font sizes varying from 10 to 48.

The performance analysis is divided into three phases. In the first phase, different scale-invariant detectors are examined along with the gradient descriptor and *k*-means clustering, and the mean recognition rate is estimated for each of them. This phase aims to find the best key point detectors that are suitable for describing the nature of Arabic fonts and can be used in AOFR. The scale-invariant detectors considered in this paper are (Harris-Laplace, Hessian, Hessian Laplace, DoG, SIFT, D_SIFT, multi-scale Harris, multi-scale Hessian).

Figure 6 shows the mean recognition rate (RR) of the AOFR systems with each of the scale-invariant detectors mentioned above with respect to the number of fonts used. This comparison was carried out using (ACS Fayrouz Bold, ACS Morgan Bold, AF_El Khobar, AGA Cairo Regular, Al-Kharashi 27, Andalus, Arabic Transparent, B Homa, Droid Arabic Naskh, Kufi, B Zar, DecoType Thuluth, M Unicode Sara, SH_Roq'a, Traditional Arabic) Arabic fonts, where the first five fonts were used for the five font results in Fig. 6 and the first 10 fonts were used for the 10 font results. It can be noted from the test results of the different scale-invariant detectors that Hessian and Harris-Laplace detectors have the best performance for AOFR.

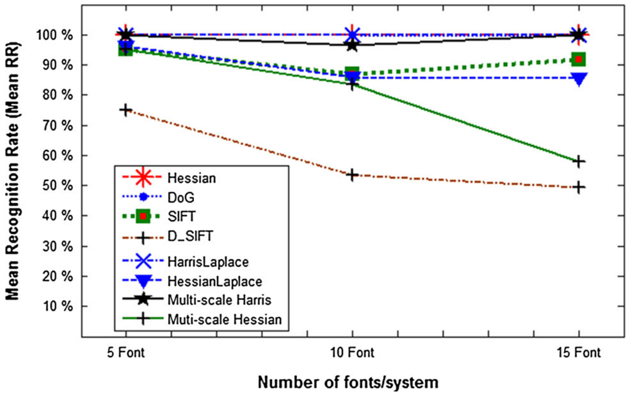

Fig. 6 Mean RR of each scale-invariant detector in five font, 10 font, and 15 font systems

The second test phase is to compare the performance of the proposed algorithm with that of the state-of-the-art algorithms in the field of AOFR. Table 1 shows the mean recognition rate for the proposed algorithm and the algorithms in [2–8]. The results of the proposed algorithm in each row of Table 1 were obtained using the same set of fonts used in the paper that it is compared with. The fonts used for each comparison are shown also in Table 1. From the mean recognition rate in Table 1, it is clear that the proposed algorithm outperforms the other algorithms in [2–8]. The miss classification in our algorithm occurs when trying to find the font type of a page that has very few written words. With a few written words, the located key points are insufficient to take the correct decision especially with fonts that have high similarity such as Traditional Arabic, Simplified Arabic, and Arabic Transparent which can be seen in Fig. 1.

The third test phase is to find the critical resolution in which the tested image's font can be identified correctly with the same mean recognition rate shown in Table 1. Figure 7 shows the average mean recognition rate over 15 fonts for different test images' resolution reductions with respect to the training images resolution. The horizontal axis represents the attenuation ratio between the test image resolution and the training image resolution, and the vertical axis represents the average mean recognition rate for the reduced resolution. It can be concluded that the accepted attenuation factor in the test images' resolution is 38 % of the training images' resolution.

The proposed algorithm succeeded in detecting the font of images containing only one line. So, for multi-font text, paragraphs can be separated with any well-known segmen-

Table 1 A comparison between the proposed algorithm and other previous algorithms

	Proposed algorithm	Slimane et. al. [2]	Pourasad et. al. [3]	Bataineh et. al. [4]	Essoukri et. al. [5]	Pourasad et. al [6,7]	Slimane et. al. [8]
Fonts (10 fonts): AdvertisingBold, Andalus, Arabic Transparent, DecoTypeThuluth, Diwani Letter, Droid Arabic Naskh, M Unicode Sara, Tahoma, Traditional Arabic, Simplified Arabic	99.2 %	93.2 %	–	–		–	
Fonts (7 fonts): Andalus, Thuluth, Diwani, Naskh, Kufi, Roqaa, persian	99.5 %	–	95.7 %	97.85 %	–	93.7 %	–
Fonts (10 fonts): Andalus, Arabic transparent, Advertising bold, Diwani letter, DecoTypeThuluth, Simplified Arabic, Tahoma, Traditional Arabic, DecoTypeNaskh, M Unicode Sara	99.2 %	–	–	–	96.5 %	–	94.5 %

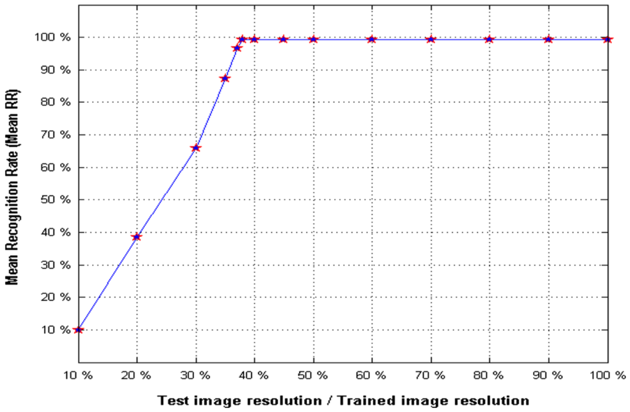

Fig. 7 Average mean recognition rate over 15 fonts at different resolution reductions with respect to the training image resolution

tation approach and apply the algorithm to detect the font for every paragraph. The paragraph size can be as small as one line; however, it has to have only one font.

6 Conclusion

Font recognition is an essential step in multi-fonts optical character recognition (OCR) systems. Arabic language has challenging characteristics for OCR that elevates the need for Arabic optical font recognition (AOFR). A new algorithm for AOFR is proposed in this paper. The proposed algorithm uses scale-invariant detector, gradient-based descriptor, and *k*-means clustering to recognize the Arabic font in text image. The proposed algorithm shows a promising performance, and it produces a mean recognition rate of 99.2–99.5 % and outperforms the algorithms in [2–8] that are used for AOFR.

References

1. Al-Badr, B., Mahmoud, S.A.: Survey and bibliography of Arabic optical text recognition. ELSEVIER Signal Process. **41**, 49–77 (1995)
2. Slimane, F., Kanoun, S., Alimi, A.M., Ingold, R., Hennebert, J.: "Gaussian mixture models for Arabic font recognition", In: Proceedings of international conference on pattern recognition, pp. 2174 – 2177, (2010)
3. Pourasad, Y., Ghorbani, A., ghouparanloo, S.: Farsi font and font size recognition based on analyzing binarization effect on small components of document images. J. Basic Appl., pp. 9563–9568 (2012)
4. Bataineh, B., Norul, S., Abdullah, H., Omar, K.: A statistical global feature extraction method for optical font recognition. In: Proceedings of the third international conference on intelligent information and database systems, Daegu, Korea, pp. 257–267, April 2011
5. Essoukri, N., Amara, B., Gazzah, S.: Une approche d'identification des fontes arabes. CIFED **8**, 21–25 (2004)
6. Pourasad, Y., Hassibi, H., Ghorbani, A.: Farsi font recognition in document images using PPH features. Int. J. Nat. Eng. Sci. **5**(3), 17–20 (2011)
7. Pourasad, Y., Hassibi, H., Ghorbani, A.: Farsi font recognition using holes of letters and horizontal projection profile, In: Proceed-

ings of the first international conference on innovative computing technology, Tehran, Iran, pp. 235–243 (2011)

8. Slimane, F., Kanoun, S., Hennebert, J., Alimi, A.M., Ingold, Rolf: A study on font-family and font-size recognition applied to Arabic word images at ultra-low resolution. Pattern Recognit. Lett. **34**, 209–218 (2013)
9. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. Int. J. Comput. Vis. **60**(1), 63–86 (2004)
10. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. Springer Sci. Int. J. Comput. Vis. **65**(1/2), 43–72 (2006)
11. Szeliski, R.: Computer Vision: Algorithms and Applications. Texts in Computer Sciences. Springer, London (2010)
12. Lowe, David G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
13. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **27**(10), 31–37 (2005)
14. Mackay, D.J.C.: Chapter 20. An example inference task: Clustering. In: Information theory, inference, and learning algorithms, Cambridge Univ. Press, Cambridge (2003)