Pattern Recognition 47 (2014) 2231–2240

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

KAFD Arabic font database

Hamzah Luqman [a], Sabri A. Mahmoud [a,*], Sameh Awaida [b]

[a] King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia
[b] Qassim University, Qassim 51452, Saudi Arabia

ARTICLE INFO

Article history:
Received 24 July 2013
Received in revised form
11 December 2013
Accepted 22 December 2013
Available online 3 January 2014

Keywords:
Arabic font database
Arabic font recognition
Log-Gabor filters
Classification
Feature extraction

ABSTRACT

Font recognition is useful for improving optical text recognition systems' accuracy and time, and to restore the documents' original formats. This paper addresses a need for Arabic font recognition research by introducing an Arabic font recognition database consisting of 40 fonts, 10 sizes (ranging from 8 to 24 points) and 4 styles (viz. normal, bold, italic, and bold–italic). The database is split into three sets (viz. training, validation, and testing). The database is freely available to researchers.[1] Moreover, we introduce a baseline font recognition system for benchmarking purposes, and report identification rates on our KAFD database and the Arabic Printed Text Image (APTI) database with 20 and 10 fonts, respectively. The best recognition rates are achieved using log-Gabor filters.

1. Introduction

Over the last years, a considerable improvement has been achieved in the area of Arabic text recognition (ATR) [3,24,29], whereas Arabic font recognition (AFR) has not been studied as extensively as ATR despite its importance in improving the recognition accuracy [27].

Optical font recognition (OFR) is the process of recognizing the font of a given text image [27]. Font recognition is useful in improving the text recognition phase in terms of recognition accuracy and time. Recognizing the font before using optical character recognition (OCR) helps in using Mono-font recognition (which assumes one known font) that results in better recognition rates compared to Omni-font recognition (which assumes a number of predetermined fonts) and requires less recognition time. In addition, recognizing the text font enables the system to reproduce not only the text but also the font and style of the examined document; resulting in more savings in time compared to manual human editing where the writer needs to recover the text font and style.

Researchers of AFR used different feature types, various numbers of fonts, and different databases. These differences, especially in the used data, make it inappropriate to compare the accuracy results of the different techniques. The different data is justified by the lack of a benchmarking database for font recognition.

Benchmarking databases are very important for AFR research. They are an essential requirement for the development, evaluation, and comparison of different AFR techniques. The lack of a benchmarking database for AFR systems resulted in researchers developing their own data. These datasets are limited in the number of fonts, styles, and scanning resolutions. Such limitations in the datasets resulted in the limitations of the outcomes of the research. Examples of two widely used databases in the field of AFR are the Arabic Printed Text Image (APTI) and ALPH-REGIM databases. The APTI database is a synthesized multi-font, multi-size, and multi-style database created by Slimane et al. [23]. It is a word level database where each text image consists of only one word. The APTI database contains a lexicon of 113,284 Arabic words. It consists of 10 fonts (viz. Deco Type Thuluth, Andalus, Deco Type Naskh, Simplified Arabic, Arabic Transparent, M Unicode Sara, Diwani Letter, Advertising Bold, Traditional Arabic, and Tahoma), 10 sizes (6, 7, 8, 9, 10, 12, 14, 16, 18, and 24 points), and 4 font styles. Their images are of low resolution (72 dot/in.) and contain 45,313,600 word images. The dataset consists of six sets; five of them are available for researchers. The text images of this database are synthesized which is a limitation. In addition, the text is identical for each font and some of the used font sizes (6 and 7) are rarely used in Arabic documents. The ALPH-REGIM database is a paragraph level database created by Ben Moussa et al. [17]. It consists of more than 5000 text images of 14 Arabic fonts with a resolution of 200 dpi, containing both printed and handwritten scripts for Arabic and Latin languages. Fourteen fonts were used with Arabic printed texts and eight with Latin texts. The 14 Arabic fonts are Arabic Transparent, Hada, Naskh, Ahsa, Badr, Kharj, Andalus, Dammam, Buryidah,

* Corresponding author.
E-mail addresses: g201002600@kfupm.edu.sa (H. Luqman), smasaad@kfupm.edu.sa (S.A. Mahmoud), s.awaida@qu.edu.sa (S. Awaida).
[1] Corresponding author may be contacted to get access to the database.

0031-3203/$ - see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.12.012

Hijaz, Khoubar, Thuluth, Diwani, and Koufi. In contrast to APTI database, some of the used fonts in this database are not commonly used in Arabic documents like Ahsa and Dammam. In addition, this database lacks the ground truth of the images and it is available in only one font size, one style, and one low resolution.

Other researchers used less-prevalent font databases. These databases have some limitations like low number of fonts, single resolution, synthesized text, etc. To our knowledge, no database with a large number of Arabic fonts, sizes, and different scanning resolutions is publicly available for researchers.

In this paper, we present the design and implementation of our KAFD database (**K**ing Fahd University **A**rabic **F**ont **D**atabase). KAFD addresses the limitations in the surveyed databases by introducing a multi-font, multi-size, multi-style, and multi-resolution database. It is a freely available and comprehensive database, containing four resolutions (100, 200, 300, and 600 dpi) and two forms (page and line). KAFD consists of 40 Arabic fonts. Each font in this database consists of its unique text. For each font, 10 font sizes (8, 9, 10, 11, 12, 14, 16, 18, 20, and 24 points) are prepared. For each font size, four font styles are prepared. KAFD database is organized into three sets (viz. training, testing, and validation). In addition, we introduce a successful Arabic font recognition prototype using log-Gabor features. These features are tolerant to noise and are extracted from the images without segmentation.

This paper is organized as follows: Section 2 presents the literature review of Arabic/Farsi font recognition research; overview of KAFD and its construction process are presented in Sections 3 and 4, respectively. Section 5 presents our proposed AFR technique, and Section 6 details the experimental results. Finally, conclusions are presented in Section 7.

2. Literature review

Different feature types with various numbers of fonts and different databases are used by researchers for Arabic/Farsi font recognition. This section contains a concise literature review in the field of AFR while concentrating on the used databases. For more thorough and complete literature review, readers are referred to [13].

To our knowledge, Zramdini and Ingold were one of the earliest researches who proposed a statistical approach for Latin font recognition at the text line level based on global typographical features [28]. They tested their system on a database that contains 10 fonts, 7 sizes, and 4 styles. For each font, 100 text lines were printed and then scanned again at 300 dpi. The authors reported a recognition rate of almost 97%. However, their approach needs more than one line of text to achieve high recognition rate and seems unsuitable for cursive languages like Arabic. Abuhaiba [1] classified the font of a word into three fonts by matching its symbols (usually less than character width) to the templates of fonts. The fonts of the best matching templates are retained and the most frequent one is taken as the word font. Using a dataset of 185,839 words, Abuhaiba reported a 77.4% recognition rate. In [2] he used decision tree classifier to classify the samples into one of

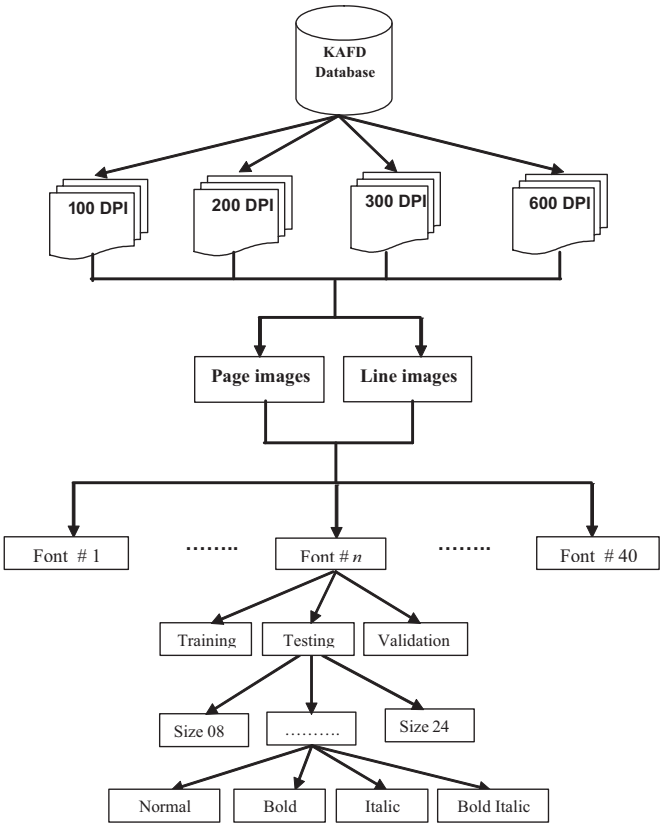

Fig. 1. KAFD structure.

Table 1
Surveyed techniques.

Paper	# of Fonts	# of Sizes	# of Styles	Training samples	Testing samples	Recognition rate (%)	Classifier
Zramdini and Ingold [28]	10	7	4	28,000 line images	30,000 line images	97	Multivariate Bayesian
Abuhaiba [1]	3	3	4	185,839 word images		77.4	*K*-nearest neighbor
Abuhaiba [2]	3	3	3	72,000 word image	36,000 word images	90.8	Decision tree
Borji et al. [8]	7		4			82.0 and 85.0	SVM & WED
Chaker et al. [9]	10				360 characters	100.0	–
Khosravi et al. [12]	10	6	2	15,000 line images	5000 line images	94.2	MLP
Slimane et al. [25,26]	10	10	1	100,000 word images	100,000 word images	99.1 99.6	Gaussian mixture model
Ben Moussa et al. [18]	10	4		500 block images	500 block images	96.6	*K*-nearest neighbor
Pourasad et al. [19,20]	7	7		245 images	600 images	93.7	
Pourasad et al. [21]	20					94.0	Euclidean Distance
Bataineh et al. [7]	7				14 images	43.7	BPNN
Bataineh et al. [5,7]	7			420 block images 448 block images	280 block images 252 block images	97.9 and 98.0	Decision tree
Zahedi et al. [27]	20			20 paragraph images	75 and 1400 block images	100.0	*K*-nearest neighbor
Imani et al. [11]	10			4500 block images	500 block images	≈ 95.0	SVM, RBFNN, KNN
Senobari and Khosravi [22]	10			15,000 line images	5000 line images	95.6	MLP
Lutf et al. [14]	10	4	4	320 paragraph images	960 page images 7680 line images	98.7 and 95.4	Normalized cross correlation

the three fonts. Using 48 features with 72,000 samples for training and 36,000 samples for testing, the reported recognition rate was 90.8%. Borji and Hamidi [8] used global texture analysis and Gabor filters with weighted Euclidean distance and SVM classifiers. Using a dataset of seven fonts and four styles, average recognition rates of 85.0% with weighted Euclidean distance and 82.0% with SVM were reported. Chaker et al. [9] recognized the font of isolated characters based on the dissimilarity index calculated on the polygonal approximation of the character. The characterized character was classified to one of the 10 fonts using the minimum dissimilarity measure. Using three testing sets of 120 characters each, a 100.0% recognition rate is reported. The data and the number of fonts are limited, which may justify the very high recognition rates.

Slimane et al. [25] proposed a technique to recognize Arabic fonts using 102 features using Gaussian mixture models (GMM). This approach uses a fixed-length sliding window to extract the features, so there is no need for prior segmentation of Arabic words into characters. Using a dataset of 100,000 word samples for training and 100,000 word samples for testing, a recognition rate of 99.1% is reported. Slimane et al. [26] used the same features to recognize Arabic fonts at ultra-resolution word images. Using a dataset of 10 fonts and 10 sizes of APTI database, a recognition rate of 94.5% is reported. To improve the accuracy of the system, they considered Arabic transparent and simplified Arabic fonts as one font and obtained a 99.6% recognition rate. These results are on par with their previous reported recognition rates. Ben Moussa et al. [18] proposed a method for Arabic and Latin font recognition using fractal geometry on global textures. Using *K*-nearest neighbor for classification and a dataset of 1000 block images of 10 fonts and four sizes, a recognition rate of 96.6% is reported. Khosravi et al. [12] proposed an approach based on Sobel–Roberts features (SRF) to recognize 10 Farsi fonts with sizes of 11–16 points and 2 styles. The new features are based on grouping of Roberts and Sobel gradients in 16 directions. Multi-layer perceptron (MLP) with 40 hidden neurons is used as a classifier. Using a database of 500 document images (20,000 line images), a recognition rate of 94.2% is reported. The same features (SRF) are combined with wavelet transform in [22] and a recognition rate of 95.6% is reported using a MLP classifier on a dataset of 10 Farsi fonts, 15,000 samples for training, and 5000 for testing. Pourasad et al. [21] extracted the boundary points of the query letter to recognize its font. Then the spatial matching between these points and boundary points of dataset symbols is done to recognize the font of the letter. To evaluate this technique, a dataset consisting of 20 fonts is used and a 94.0% recognition rate was reported. However, this technique is time consuming. The horizontal projection profile and holes of letters are used to recognize seven fonts and seven sizes [19,20]. Two datasets of 245 and 600 images were used for training and testing, respectively, with a reported recognition rate of 93.7%.

Zahedi and Eslami [27] recognized the Farsi fonts using scale invariant feature transform (SIFT) method. They recognized the fonts based on the similarity between the objects in the tested images and the extracted key points. They evaluated their technique over a dataset consisting of 1400 block images and 20 font types. A 100.0% recognition rate is reported. Computation time especially for large datasets is a drawback of this technique. Imani et al. [11] used three classifiers (viz. SVM, RBFNN, and KNN) in a majority voting approach to classify the image into one of the 10 fonts. More than 95.0% recognition rate was reported on a dataset of 5000 images. Bataineh et al. [6] used 22 statistical features to identify one of the seven Arabic calligraphy types using back-propagation neural network (BPNN). A dataset of 14 Arabic degraded document images were used and an accuracy of 43.7% is reported which is too low for practical applications. The problem with the proposed method is the need for prior window size setting. Bataineh et al. [6] proposed a technique to classify the Arabic calligraphies using weights, homogeneity, pixel regularity, edge regularity, and edge direction features. They conducted their experiments on a dataset of seven fonts consisting of 700 samples (100 per font). This dataset was split into 60% for training and 40% for testing. A 97.9% recognition rate using decision tree classifier is reported. In Bataineh et al. [5] they split the dataset into 64% for training and 36% for testing and obtained a recognition rate of 98.0%. Lutf et al. [14] proposed an approach for AFR based on recognizing the font of the diacritics segmented from the text images. To identify the font of the diacritics, a composite of central and ring projection features is extracted from each diacritic. Using a dataset of 10 fonts, 4 sizes, and 4 styles, recognition rates of 98.7% and 95.4% are obtained at the page and line levels, respectively. The authors used the same texts for all the fonts. Furthermore, basing font recognition on diacritics is not practical since Arabic diacritics are not mandatory in Arabic writing and are only limited to few text sources (e.g. religious documents, legal documents). Moreover, when the authors used a real database, they obtained a recognition rate lower than synthesized database since diacritics are small objects and would be susceptible to noise.

As stated earlier, it is inappropriate to compare the different techniques' recognition rates given that they are using different data with different numbers of fonts, sizes, etc. Table 1 shows the reported recognition rates, used datasets, and classifiers.

3. KAFD overview

KAFD is a multi-font, multi-size, multi-style, and multi-resolution Arabic text database. The used fonts are based on the most frequent used fonts in Arabic books, magazines, letters, theses, etc. The texts of KAFD are collected from different subjects like religious, medicine, science, history, etc. Each font in this database contains unique text.

KAFD includes the most commonly used 40 Arabic fonts in printed documents for its text images. Four of the fonts (Arabic Transparent, Times New Roman, Simplified Arabic, and Arial) share the same Arabic font face with slight/no variations in letter spacing. Previous researchers used two of these fonts in their database, and had to group them into one class in order to improve their font-classification results [25]. Although these four fonts are similar they have been added as these fonts are very common in printed documents. In addition, they may be combined and used for Arabic text recognition.

This database is available in four resolutions (100, 200, 300, and 600 dpi) and in two forms (page and line). It consists of 40 fonts as listed in Fig. 4. For each font, 10 font sizes are prepared (8, 9, 10, 11, 12, 14, 16, 18, 20, and 24 points). For each font size, four font styles are prepared (viz. normal, bold, italic, and bold–italic). KAFD is organized into three sets: training, testing, and validation. The structure of KAFD database is shown in Fig. 1.

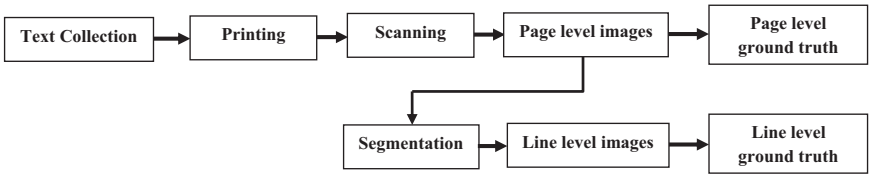

Fig. 2. KAFD high level implementation process.

Table 2
Number of KAFD images.

S. no.	Font	Page level database						Line level database					
		Training set	Testing set	Validation set	Total	Resolutions (dpi)	Total	Training set	Testing set	Validation set	Total	Resolutions (dpi)	Total
1	Advertising Bold	430	152	144	726	100, 200, 300, 600	2904	9718	3342	2982	16,042	100, 200, 300, 600	64,168
2	AGA Granada Regular	425	144	136	705	100, 200, 300, 600	2820	9265	3196	2738	15,199	100, 200, 300, 600	60,796
3	AGA Kaleelah Regular	440	152	150	742	100, 200, 300, 600	2968	9958	3244	3218	16,420	100, 200, 300, 600	65,680
4	Akhbar	431	147	150	728	100, 200, 300, 600	2912	8432	2715	2982	14,129	100, 200, 300, 600	56,516
5	Al-Mohand	425	144	152	721	100, 200, 300, 600	2884	9134	2813	3161	15,108	100, 200, 300, 600	60,432
6	AL-Qairwan	428	152	144	724	100, 200, 300, 600	2896	9152	3135	2928	15,215	100, 200, 300, 600	60,860
7	Andalus	424	145	144	713	100, 200, 300, 600	2852	8806	2920	2968	14,694	100, 200, 300, 600	58,776
8	Arabic Transparent	430	150	148	728	100, 200, 300, 600	2912	12,591	4228	4176	20,995	100, 200, 300, 600	83,980
9	Arabic Typesetting	428	144	144	716	100, 200, 300, 600	2864	11,482	3832	3676	18,990	100, 200, 300, 600	75,960
10	Arabswell	434	148	148	730	100, 200, 300, 600	2920	7948	2578	2582	13,108	100, 200, 300, 600	52,432
11	Arial	419	140	144	703	100, 200, 300, 600	2812	11,181	3560	3756	18,497	100, 200, 300, 600	73,988
12	Arial Unicode MS	438	145	152	735	100, 200, 300, 600	2940	10,847	3534	3876	18,257	100, 200, 300, 600	73,028
13	Courier New	412	144	148	704	100, 200, 300, 600	2816	11,499	3868	3997	19,364	100, 200, 300, 600	77,456
14	Deco Type Naskh	437	150	149	736	100, 200, 300, 600	2944	6960	2230	2302	11,492	100, 200, 300, 600	45,968
15	Deco Type Thuluth	428	144	152	724	100, 200, 300, 600	2896	6950	2218	2392	11,560	100, 200, 300, 600	46,240
16	Diwani Letter	428	148	144	720	100, 200, 300, 600	2880	7078	2388	2208	11,674	100, 200, 300, 600	46,696
17	Freehand	432	148	148	728	100, 200, 300, 600	2912	9554	3136	3094	15,784	100, 200, 300, 600	63,136
18	Hadeel	436	132	136	704	100, 200, 300, 600	2816	9650	2542	2816	15,008	100, 200, 300, 600	60,032
19	Lotus Linotype	436	148	152	736	100, 200, 300, 600	2944	7580	2392	2568	12,540	100, 200, 300, 600	50,160
20	M Unicode Sara	434	144	152	730	100, 200, 300, 600	2920	9534	3024	3300	15,858	100, 200, 300, 600	63,432
21	Maghribi Assile	432	151	147	730	100, 200, 300, 600	2920	12,209	4079	3785	20,073	100, 200, 300, 600	80,292
22	Microsoft Sans Serif	412	145	136	693	100, 200, 300, 600	2772	11,868	4020	3784	19,672	100, 200, 300, 600	78,688
23	Microsoft Uighur	447	140	139	726	100, 200, 300, 600	2904	13,613	4028	5046	22,687	100, 200, 300, 600	90,748
24	Midan	428	138	144	710	100, 200, 300, 600	2840	11,930	3608	3836	19,374	100, 200, 300, 600	77,496
25	Motken Unicode Hor	428	144	152	724	100, 200, 300, 600	2896	9208	2976	3131	15,315	100, 200, 300, 600	61,260
26	Nawel	416	148	144	708	100, 200, 300, 600	2832	9038	3104	2982	15,124	100, 200, 300, 600	60,496
27	Pen Kufi	423	148	149	720	100, 200, 300, 600	2880	8129	2818	2836	13,783	100, 200, 300, 600	55,132
28	Quran2	425	144	148	717	100, 200, 300, 600	2868	6900	2140	2262	11,302	100, 200, 300, 600	45,208
29	Rateb	422	147	148	717	100, 200, 300, 600	2868	8646	2956	2968	14,570	100, 200, 300, 600	58,280
30	Rekaa	424	136	146	706	100, 200, 300, 600	2824	7296	2180	2484	11,960	100, 200, 300, 600	47,840
31	SC Dubai	432	152	152	736	100, 200, 300, 600	2944	10,900	3620	3708	18,228	100, 200, 300, 600	72,912
32	SC Gulf	436	144	132	712	100, 200, 300, 600	2848	7922	2456	2186	12,564	100, 200, 300, 600	50,256
33	Sc-Alyermook	424	136	136	696	100, 200, 300, 600	2784	10,900	3516	3547	17,963	100, 200, 300, 600	71,852
34	Sc-ouhod	416	148	146	710	100, 200, 300, 600	2840	15,344	5086	4946	25,376	100, 200, 300, 600	101,504
35	Segoe UI	419	142	147	708	100, 200, 300, 600	2832	10,117	3320	3474	16,911	100, 200, 300, 600	67,644
36	Simplified Arabic	430	154	154	738	100, 200, 300, 600	2952	8213	2772	2768	13,753	100, 200, 300, 600	55,012
37	Tahoma	436	150	150	736	100, 200, 300, 600	2944	11,594	3852	3803	19,249	100, 200, 300, 600	76,996
38	Times New Roman	430	144	144	718	100, 200, 300, 600	2872	11,563	3679	3714	18,956	100, 200, 300, 600	75,824
39	Traditional Arabic	432	149	140	721	100, 200, 300, 600	2884	8827	2928	2716	14,471	100, 200, 300, 600	57,884
40	Zarnew	424	135	129	688	100, 200, 300, 600	2752	8200	2290	2251	12,741	100, 200, 300, 600	50,964
Total		**17,131**	**5816**	**5820**	**28,767**		**115,068**	**389,736**	**126,323**	**127,947**	**644,006**		**2,576,024**

4. Database construction process

Fig. 2 shows the construction process of KAFD. As shown in the figure, the process starts by collecting the texts, and then the constructed pages are printed and scanned in different resolutions. In addition, the text images are segmented into lines and their ground truth files are added.

We collected Arabic texts from different subjects like religious, medicine, science, history, and other sources. The used texts contain names, places, cities, numbers, etc. Each Arabic text used for each font in this database is different (unique) from the texts used for other fonts. After collecting the texts, we constructed the 40 fonts as follows:

- Each font consists of 10 sizes (8, 9, 10, 11, 12, 14, 16, 18, 20, and 24 points). The text for each size is identical to other sizes for the same font and is different from other fonts.
- For each size, four font styles are used (viz. normal, bold, italic, and bold–italic). These styles cover the different writing styles in Arabic documents.
- For each font style, three disjoint sets are constructed (60% for training, 20% for testing, and validation each).
- The number of pages in each set starts by 10 pages for 8 points size (6 training, 2 testing, and 2 validation). This number of pages increases as the font sizes increase. The first 20 pages (12 training, 4 testing, and 4 validation) are used for sizes larger than 12 points. The total number of printed pages is 28,767 for each resolution.

KAFD database is printed using HP Laser jet 600-M601 printer with a print resolution of 1200 × 1200 dpi. The printed pages of KAFD database are scanned using Ricoh IS760D scanner. Pages are scanned in grayscale, and in 100, 200, 300 and 600 dpi resolutions. This process resulted in 115,068 page level images for all resolutions (28,767 page images per resolution) as shown in Table 2 which shows the number of page images for each font in the four resolutions (100, 200, 300, and 600 dpi). Each text image is saved in "tiff" file type with a name that reflects the image font type, size, style, set, resolution, and page number (and line number for line level database) as shown in Fig. 3.

All text images are segmented into lines and ground truth files for each page and line are built. Segmentation enables the researchers to use the database at the page and line levels. This phase resulted in 2,576,024 line images (644,006 line images per resolution) as shown in Table 2. The truth-values of the page and line images are kept in text files. Similar names are used for the page and line images and their truth-values.

Table 2 shows the number of page and line images of each set (training, testing, and validation) of each font in the four resolutions (100, 200, 300, and 600 dpi).

KAFD database has undergone three phases of verification. In the first phase, the scanned images are checked for the quality of scanning to ensure the absence of cropped portions from the page images. In addition, the scanned images are checked for skew and noise. In the second phase, scanned images are verified at the line level to ensure that there are no errors in the line segmentation. This phase of verification includes checking that the number of the segmented line images of the page image matches the number of lines in the page image. The third phase verifies the correctness of the ground truth of the scanned page and line level images by ensuring that the ground truth matches the printed text at the page and line levels. Fig. 4 shows line samples of the 40 fonts of KAFD database.

Table 3 shows a comparison between KAFD and the main Arabic font databases that are used in AFR (viz. APTI and ALPH-REGIM databases). Other font recognition databases are less used and limited in several aspects like the number of fonts, sizes, styles, etc. and hence were not included in the comparison. Several attributes are used in the comparison. As shown in Table 3, KAFD outperforms all other databases in almost all attributes. KAFD has the highest number of fonts (40 fonts) whereas the largest number of fonts of other databases is 14 of ALPH-REGIM database. KAFD and APTI databases have the largest number of sizes and styles. KAFD database is available in two forms (viz. page and line levels). KAFD database is available in four resolutions, while others are available in only one resolution. KAFD database has the largest number of page and line images. The number of images of KAFD and APTI databases cannot be compared because APTI is a word level database while KAFD is page and line levels database. KAFD has scanned printed text using a scanner machine while the APTI database is synthetically generated.

(a) *Font_Size_Style_Set_Resolution*_Pg*ddddd*.tif

(b) *Font_Size_Style_Set_Resolution*_Pg*ddddd*_ln*ddddd*.tif

Font: Font name
Size: Font size (2 digits)
Style: Normal (N), Bold (B), Italic (I), or Bold-Italic (BI)
Set: Training (Tr), Testing (Ts), or Validation (V)
Resolution: 200dpi, 300dpi, or 600dpi
ddddd: Serial number (5 digits)

Fig. 3. Image names format. (a) Page level images. (b) Line level images.

5. Arabic font recognition using log–Gabor filters

In order to design the best performing Arabic font recognition system, we tested 10 different types of features on the KAFD database. These features include a modified version of the curvature features (concave and convex features), direction features, and direction length features of [15]; box counting dimension (BCD) features [18]; center of gravity features, number of vertical and horizontal extrema features, number of black and white components features, smallest black component features, and log baseline position features of [25]; and log-Gabor filter features. The best results were achieved using log-Gabor filters with eight orientations and four scales. Furthermore, log-Gabor features are tolerant to noise and can be extracted from line images without word or character segmentation.

A two-dimensional log-Gabor filter is defined in the log-polar coordinates of the Fourier domain as Gaussians shifted from the origin [10]

$$G_{m,n}(\rho,\theta)=\exp\left(-\frac{1}{2}\left(\frac{\rho-\rho_m}{\sigma_\rho}\right)^2\right)\exp\left(-\frac{1}{2}\left(\frac{\Theta-\Theta_n}{\sigma_\Theta}\right)^2\right)$$

where (ρ, θ) are the log-polar coordinate (in $\log_2$ scale); m is the number of scales, $m=(1,..,4)$; n is the number of orientations, $n=(0,..,7)$; (ρ_m, θ_n) are the coordinates of the center of the filter, and $(\sigma_\rho, \sigma_\theta)$ are the band widths in ρ and θ.

The filter orientations are calculated using the following equation:

$$\theta_k=\frac{2\pi k}{n},\quad k=\{0,1,\ldots,n-1\} \tag{2}$$

To speed up the computation, we implemented the filter in the frequency domain as in [16]. The Fourier transforms of the image (i) and filter (g) are computed using fast Fourier transform (FFT) and multiplied to give the Fourier transform of the filtered image (o). This is followed by taking the inverse Fourier transform as

Fig. 4. Line samples of KAFD 40 fonts.

shown in (3) to compute the filtered image (o)

$$o_{mn} = FFT^{-1}(FFT(i) \times FFT(g_{mn})) \quad (3)$$

where o_{mn} is the filtered image with a filter with scale m and orientation n, i is the image, g_{mn} is the Gabor filter with scale m and orientation n. o_{mn} is a complex function with a real part $\Re\{o_{mn}\}$ and imaginary part $\Im\{o_{mn}\}$. The magnitude response $\|o_{mn}\|$ is espressed as

$$||o_{mn}|| = \sqrt{^2\{o_{mn}\} + ^2\{o_{mn}\}} \quad (4)$$

The filtered image of Eq. (4) is split into 3×3 segments. The mean and variance of each segement of $\|o_{mn}\|$ are taken as features.

The log-Gabor filter is used as the DC-component of the log-Gabor is zero (unlike Gabor filter) and the overall number of required filters can be reduced with respect to Gabor filters [4]. Several scales and orientations were experimented with and best results are achieved with four scales and eight orientations using Eq. (2).

Table 3
Databases comparison.

Evaluation criteria	ALPH-REGIM	APTI	KAFD
Number of fonts	14	10	40
Number of sizes	1	10	10
Number of styles	1	4	4
Resolutions (dpi)	200	72	100 200 300 600
Text levels	Paragraph	Word	Page–line
Number of page images			115,068
Number of paragraph images	More than 5000		
Number of line images			2,576,024
Number of word images		45,313,600	
Scanning method	Scanner	Synthesized	Scanner

6. Experimental results

In our experiments, we tested several orientations and scales of the log-Gabor filter. Eight orientations (0, 22.5, 45, 67.5, 90, 112.5, 135, and 157.5) and four scales are experimentally found to give the best accuracy. Each image in this experiment is segmented into nine segments (three horizontal and three vertical) which resulted

Table 4
Confusion matrix using 20 fonts, 10 sizes, and 4 styles of KAFD database.

Font	Courier New	Deco Type Naskh	M Unicode Sara	Segoe UI	Traditional Arabic	AGA Kaleelah Regular	Arial Unicode MS	Diwani Letter	Al-Mohand	AL-Qairwan	Arabic Typesetting
Courier New	**7856**	0	1	8	7	0	0	0	0	0	3
Deco Type Naskh	0	**4468**	0	1	3	1	0	1	2	0	131
M Unicode Sara	0	0	**6317**	0	0	1	0	0	3	0	0
Segoe UI	10	0	2	**6518**	2	0	108	1	24	0	0
Traditional Arabic	5	8	5	31	**5126**	0	0	0	19	0	87
AGA Kaleelah Regular	0	0	1	0	0	**4968**	0	0	0	0	2
Arial Unicode MS	3	4	1	9	0	0	**7278**	1	3	1	0
Diwani Letter	1	0	0	1	0	1	0	**4591**	15	0	0
Al-Mohand	10	24	3	7	4	19	6	0	**6353**	0	0
AL-Qairwan	0	0	0	0	0	0	0	0	3	**6031**	8
Arabic Typesetting	1	70	6	24	98	1	4	0	7	0	**7111**
Arabswell	0	0	8	6	8	2	0	0	40	0	1
Microsoft Uighur	20	3	0	0	4	0	60	5	34	0	78
Motken Unicode Hor	2	3	7	3	4	11	1	0	0	0	1
Freehand	0	0	0	1	0	10	14	0	0	1	0
Tahoma	1	0	1	14	0	0	255	1	0	0	0
Arabic Transparent	0	0	0	8	4	0	1	0	39	0	0
Lotus Linotype	0	0	7	12	103	0	0	0	68	0	28
Nawel	1	0	19	3	0	0	0	0	0	0	0
Zarnew	1	15	6	30	103	0	5	0	18	0	74
Average											

Font	Arabswell	Microsoft Uighur	Motken Unicode Hor	Freehand	Tahoma	Arabic Transparent	Lotus Linotype	Nawel	Zarnew	Recognition rate (%)
Courier New	0	0	1	0	0	0	0	0	0	**99.8**
Deco Type Naskh	3	64	2	1	1	7	3	0	2	**95.2**
M Unicode Sara	0	0	1	0	1	0	1	0	0	**99.9**
Segoe UI	3	5	0	0	34	91	0	0	1	**95.9**
Traditional Arabic	6	53	1	0	1	22	320	1	83	**88.9**
AGA Kaleelah Regular	0	0	0	0	4	1	0	0	0	**99.8**
Arial Unicode MS	0	25	1	0	131	3	0	0	0	**97.6**
Diwani Letter	0	0	2	0	0	0	0	0	0	**99.6**
Al-Mohand	7	33	0	0	12	61	4	0	0	**97.1**
AL-Qairwan	0	8	0	4	1	8	0	0	0	**99.5**
Arabic Typesetting	4	65	10	1	3	16	64	0	23	**94.7**
Arabswell	**5048**	2	10	0	0	5	3	0	0	**98.3**
Microsoft Uighur	0	**8558**	0	0	1	60	3	0	0	**97.0**
Motken Unicode Hor	4	1	**6075**	5	1	0	0	0	0	**99.3**
Freehand	0	1	1	**6188**	13	1	0	0	0	**99.3**
Tahoma	6	21	0	1	**7349**	2	2	0	2	**96.0**
Arabic Transparent	0	34	0	1	1	**8310**	2	0	4	**98.9**
Lotus Linotype	5	38	0	0	3	8	**4309**	0	379	**86.9**
Nawel	3	0	2	0	3	0	0	**6054**	1	**99.4**
Zarnew	11	130	1	0	0	3	550	1	**3554**	**78.9**
Average										**96.1**

in 576 features (eight orientations, four scales, nine segments, and variance and mean features ($8 \times 4 \times 3 \times 3 \times 2 = 576$)). To evaluate these features, several experiments are conducted using the APTI and KAFD databases.

6.1. Experimental results using APTI

In these experiments we used the APTI with 10 fonts, 2 font sizes (6 and 24), and 1 font style with 188,838 and 188,776 word samples for training and testing, respectively. In this experiment, a recognition rate of 99.84% is obtained. Another experiment is conducted using 10 fonts, 10 font sizes, (6, 7, 8, 9, 10, 12, 14, 16, 18 and 24) and 1 font style with (100,000) and (100,000) word samples for training and testing, respectively. In this experiment, a recognition rate of 91.0% is obtained. The error rate in this experiment resulted from the misclassification between Arabic Transparent and Simplified Arabic fonts. This misclassification is due to the similarity between them. Grouping these two fonts as one class increased the recognition rate to 98.1%.

6.2. Experimental results using KAFD

In our experiments with KAFD, 10 and 20 fonts of this database are used. We started by 10 fonts (viz. Courier New, Deco Type Naskh, M Unicode Sara, Segoe UI, AGA Kaleelah Regular, Diwani Letter, Al-Mohand, AL-Qairwan, Arabswell, and Freehand). Using a dataset of 10 fonts, 10 sizes, 4 styles, 89,607 line images for training, and 59,245 line images for testing, a recognition rate of 99.5% is obtained. In other experiments, we added 10 more fonts (viz. Traditional Arabic, Arial Unicode MS, Arabic Typesetting, Microsoft Uighur, Motken Unicode Hor, Tahoma, Arabic Transparent, Lotus Linotype, Nawel, and Zarnew). Using a dataset of 20 fonts, 10 sizes, 4 styles, 182,016 line images for training, and 126,532 line images for testing we obtained a recognition rate of 96.1%. Table 4 shows the confusion matrix of the results. It is clear from the table that the misclassifications resulted mainly between the following fonts:

1. Tahoma and Arial Unicode.
2. Traditional Arabic, Lotus Linotype, and Zarnew.

Fig. 5 shows the first group of the similar fonts. It is clear from the figure that the dissimilarity between these fonts cannot be easily distinguished by humans.

Based on the confusion matrix of Table 4, we grouped similar fonts into two font groups. The first group consists of Tahoma and Arial Unicode fonts; and the second group consists of Traditional Arabic, Lotus Linotype, and Zarnew fonts. Using these classes, we performed a set of experiments on 17 font classes (20 fonts), with 10 sizes and 4 styles. In these experiments, a significant improvement in the recognition rate is achieved, leading to an average recognition rate of 98.1%. Table 5 shows the confusion matrix of this experiment. We notice from this confusion matrix that some fonts like M Unicode Sara and Courier New have high recognition rates, whereas other fonts like Arabic Typesetting and Segoe UI have lower recognition rates due to the similarity of these fonts to group 1 and group 2 fonts, respectively.

Our approach is compared with the published work using different numbers of fonts, sizes, styles, and dataset size as shown

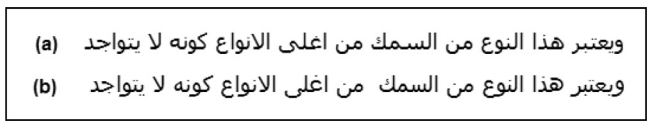

Fig. 5. Similarity between the first group of fonts. (a) Tahoma. (b) Arial Unicode.

Table 5
Confusion matrix using KAFD database with 20 fonts, 10 sizes and 4 styles after grouping similar fonts.

Font	Courier New	Deco Type Naskh	M Unicode Sara	Segoe UI	Group 1 *	AGA Kaleelah Regular	Group 2 +	Diwani Letter	Al-Mohand	AL-Qairwan	Arabic Typesetting	Arabswell	Microsoft Uighur	Motken Unicode Hor	Freehand	Arabic Transparent	Nawel	Recognition rate (%)
Courier New	**7856**	0	1	8	7	0	0	0	0	0	3	0	0	1	0	0	0	**99.7**
Deco Type Naskh	0	**4468**	0	1	8	1	1	1	2	0	131	3	64	2	1	7	0	**95.3**
M Unicode Sara	0	0	**6317**	0	1	1	1	0	3	0	0	0	0	1	0	0	0	**99.9**
Segoe UI	10	0	2	**6518**	3	0	142	1	24	0	0	3	5	0	0	91	0	**95.9**
Group 1 *	6	23	18	73	**14,527**	0	9	0	105	0	189	22	221	2	0	33	2	**95.4**
AGA Kaleelah Regular	0	0	1	0	0	**4968**	4	0	0	0	2	0	0	0	0	1	0	**99.8**
Group 2 +	4	4	2	23	4	0	**15,013**	2	3	1	0	6	46	1	1	5	0	**99.3**
Diwani Letter	1	0	0	1	0	1	0	**4591**	15	0	0	0	0	2	0	0	0	**99.6**
Al-Mohand	10	24	3	7	8	19	18	0	**6353**	0	0	7	33	0	0	61	0	**97.1**
AL-Qairwan	0	0	0	0	0	0	1	0	3	**6031**	8	0	8	0	4	8	0	**99.5**
Arabic Typesetting	1	70	6	24	185	1	7	0	7	0	**7111**	4	65	10	1	16	0	**94.7**
Arabswell	0	0	8	6	11	2	0	0	40	0	1	**5048**	2	10	0	5	0	**98.3**
Microsoft Uighur	20	3	0	0	7	0	61	5	34	0	78	0	**8558**	0	0	60	0	**97.0**
Motken Unicode Hor	2	3	7	3	4	11	2	0	0	0	1	4	1	**6075**	5	0	0	**99.3**
Freehand	0	0	0	1	0	10	27	0	0	1	0	0	1	1	**6188**	1	0	**99.3**
Arabic Transparent	0	0	0	8	10	0	2	0	39	0	0	0	34	0	1	**8310**	0	**98.9**
Nawel	1	0	19	3	1	0	3	0	0	0	0	3	0	2	0	0	**6054**	**99.5**
Average																		**98.1**

*Group 1 **: Tahoma and Arial Unicode fonts.
Group 2 +: Traditional Arabic, Lotus Linotype, and Zarnew fonts.

Table 6
Comparison of Arabic font recognition techniques.

Authors	Fonts	Sizes	Styles	Database name	Database level	Training dataset	Testing dataset	Accuracy (%)	Statistical significance interval
Bataineh et al. [7]	7	–	–	Authors data	Block	420	280	**97.9**	± **1.93**
Zahedi et al. [27]	20, 10	–	–	Authors data	Paragraph,	20	75	**100.0**	± **3.48**
				[12]	line		1400	**100.0**	± **0.19**
Bataineh et al. [5]	7	–	–	Authors data	Block	448	252	**98.0**	± **2.04**
Slimane et al. [25]	10	10	1	APTI	Word	100,000	100,000	**99.1**	± **0.05**
Ben Moussa et al. [18]	10	–	–	ALPH-REGIM	Paragraph	500	500	**99.7**	± **0.75**
Lutf et al. [14]	10	4	4	Authors data	Page, line	320	960	**98.7**	± **0.75**
							7680	**95.4**	± **0.41**
Our approach	10	10	1	APTI	Word	100,00	100,000	**98.1**	± **0.07**
	20	10	4	KAFD	Line	182,016	126,532	**98.1**	± **0.06**

in Table 6. Bataineh et al. [6,5], Zahedi and Eslami [27], Ben Moussa et al. [18], Slimane et al. [25], and Lutf et al. [14] used limited datasets. In contrast, our approach used 20 fonts, 10 sizes, and 4 styles.

7. Conclusions

In this paper we presented our KAFD Arabic font database. It consists of the most commonly used 40 Arabic fonts, 10 sizes, and 4 styles. These fonts are scanned using four scanning resolutions (100, 200, 300, and 600 dpi). Moreover, it is available in two text forms (page and line). A total of 115,068 and 2,576,024 is the number of KAFD page and line images, respectively. In addition, KAFD database is made freely available to researchers. We included the ground truth at the page and line levels, hence it may be also used for multi-font ATR.

We used KAFD database for Arabic fonts recognition using several types of features. The best results are obtained using log-Gabor filters. These features are tolerant to noise and are extracted from the images without segmentation. 576 features are extracted using log-Gabor with eight orientations and four scales. We evaluted our features using two different databases (viz. APTI and KAFD). Using the APTI database, a recognition rate of 98.1% is obtained on 10 fonts, 10 sizes, and 1 style whereas 98.1% recognition rate is obtained on KAFD of 20 fonts, 10 sizes, and 4 styles.

Conflict of interest

None declared.

Acknowlegement

The authors would like to acknowledge the support provided by King Abdul-Aziz City for Science and Technology (KACST) for funding this work under Project no. AT-30–53 through King Fahd University of Petroleum & Minerals (KFUPM). The third author would like to thank Qassim University for supporting this research and providing the computing facilities. The authors would also like to thank the anonymous reviewers whose comments helped in improving this paper.

References

[1] Ibrahim Abuhaiba, Arabic font recognition based on templates, Int. Arab J. Inf. Technol. 1 (0) (2003) 33–39.
[2] Ibrahim S. Abuhaiba, Arabic font recognition using decision trees built from common words, J. Comput. Inf. Technol. 13 (3) (2005) 211–224.
[3] N.B. Amor, N.E.B. Amara. A hybrid approach for multifont Arabic characters recognition, in: Fifth WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED'06) Madrid, Spain, 2006.
[4] J. Arrospide, L. Salgado, Log-Gabor filters for image-based vehicle verification, IEEE Trans. Image Process.: Publ. IEEE Signal Process. Soc. XXX (February) (2013) 1–10.
[5] Bilal Bataineh, Siti Norul Huda Sheikh Abdullah, Khairuddin Omar, A novel statistical feature extraction method for textual images: optical font recognition, Expert Syst. Appl. 39 (5) (2012) 5470–5477.
[6] Bilal Bataineh, Siti Norul Huda Sheikh Abdullah, Khairudin Omar, A statistical global feature extraction method for optical font recognition, in: Proceedings of the Third International Conference, ACIIDS 2011 on Intelligent Information and Database Systems, Daegu, Koria, 2011, pp. 257–267.
[7] Bataineh, Bilal, Siti Norul, Huda Sheikh, Khairudin Omar. Arabic Calligraphy Recognition Based on Binarization Methods and Degraded Images, 3 June 2011.
[8] A. Borji, M. Hamidi, Support vector machine for Persian font recognition, Eng. Technol. 2 (3) (2007) 10–13.
[9] Ilham Chaker, Mostafa Harti, Hassan Qjidah, Rachid Benslimane, Recognition of Arabic characters and fonts, Int. J. Eng. Sci. 2 (10) (2010) 5959–5969.
[10] Sylvain Fischer, Filip Šroubek, Laurent Perrinet, Rafael Redondo, Gabriel Cristóbal, Self-invertible 2D log-Gabor wavelets, Int. J. Comput. Vis. 75 (2) (2007) 231–246.
[11] Maryam Bahojb Imani, Mohamad Reza Keyvanpour, Reza Azmi, Semi-supervised Persian font recognition, Proc. Comput. Sci. 3 (2011) 336–342.
[12] Hossein Khosravi, Ehsanollah Kabir, Farsi font recognition based on Sobel–Roberts features, Pattern Recognit. Lett. 31 (1) (2010) 75–82.
[13] Hamza Luqman, Arabic Font Recognition (M.Sc. thesis), King Fahd University of Petroleum and Minerals, Saudi Arabia, 2013.
[14] Mohammed Lutf, Xinge You, Yiu-ming Cheung, C.L. Philip Chen, Arabic font recognition based on diacritics features, Pattern Recognit. (2013) 1–13.
[15] Sabri A. Mahmoud, Arabic character recognition using fourier descriptors and character contour encoding, Pattern Recognit. 27 (6) (1994) 815–824.
[16] Sabri A. Mahmoud, Wasfi G. Al-Khatib, Recognition of Arabic (Indian) bank check digits using log-Gabor filters, Appl. Intell. (2010) 1–12.
[17] Sami Ben Moussa, Abderrazak Zahour, Abdellatif Benabdelhafid, Adel M. Alimi, New features using fractal multi-dimensions for generalized Arabic font recognition, Pattern Recognit. Lett. 31 (5) (2010) 361–371.
[18] Sami Ben Moussa, Abderrazak Zahour, Abdellatif Benabdelhafid, Adel M. Alimi, New features using fractal multi-dimensions for generalized Arabic font recognition, Pattern Recognit. Lett. 31 (5) (2010) 361–371.
[19] Yaghoub Pourasad, Houshang Hassibi, Azam Ghorbani, Farsi font recognition in document images using PPH features, Int. J. Natural Eng. Sci. 5 (3) (2011) 17–20.
[20] Yaghoub Pourasad, Houshang Hassibi, Azam Ghorbani, Farsi font recognition using holes of letters and horizontal projection profile, Innovative Comput. Technol. (2011) 235–243.
[21] Pourasad, Yaghoub, Hushang Hassibi, Majid Banaeyan. Farsi font recognition based on spatial matching, in: 18th International Conference on Systems, Signals and Image Processing (IWSSIP), 2011, pp. 1–4.
[22] Senobari, Ehsan Mortazavi, Hossein Khosravi. Farsi font recognition based on combination of wavelet transform and Sobel–Robert operator features, in: 2011 Second International eConference on Computer and Knowledge Engineering (ICCKE), 2012, pp. 29–33.
[23] F. Slimane, R. Ingold, S. Kanoun, A. Alimi, J. Hennebert, A new Arabic printed text image database and evaluation protocols, in: 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, 2009, pp. 946–50.
[24] Slimane, Fouad et al. ICDAR 2011 – Arabic recognition competition: multi-font multi-size digitally represented text, in: 2011 International Conference on Document Analysis and Recognition, 2011, pp. 1449–1453.
[25] Slimane, Fouad, Slim Kanoun, Adel M. Alimi, Rolf Ingold, Jean Hennebert, Gaussian mixture models for arabic font recognition, in: 2010 20th International Conference on Pattern Recognition, 2010, pp. 2174–2177.
[26] Fouad Slimane, Slim Kanoun, Jean Hennebert, Adel M. Alimi, Rolf Ingold, A study on font-family and font-size recognition applied to Arabic word images at ultra-low resolution, Pattern Recognit. Lett. 34 (2) (2013) 209–218.

[27] Morteza Zahedi, Saeideh Eslami, Farsi/Arabic optical font recognition using SIFT features, Proc. Comput. Sci. 3 (2011) 1055–1059.
[28] A. Zramdini, R. Ingold, Optical font recognition using typographical features, IEEE Trans. Pattern Anal. Mach. Intell. 20 (8) (1998) 877–882.
[29] Mohammad Tanvir Parvez, Sabri A Mahmoud, Off-line Arabic Handwritten Text Recognition: A Survey, ACM Comput. Surv. (CSUR) 45 (2) (2013) 23.

Hamzah Luqman received his B.S. from Ahgaff University, Yemen, in 2006 and received his M.S. in computer science from King Fahd University for Petroleum and Minerals, Saudi Arabia, in 2013. His research interests include distributed computing, image processing, document analysis and understanding, pattern recognition, artificial intelligence. He has publications in Arabic font recognition and distributed computing.

Sabri A. Mahmoud is a Professor of computer Science in the Information and Computer Science Department, King Fahd University of Petroleum and Minerals. He received his B.S. in electrical engineering from Sind University, Pakistan, in 1972, received his M.S. in Computer Sciences from Stevens Institute of Technology, USA, in 1980, and his Ph. D. degree in Information Systems Engineering from the University of Bradford, UK, in 1987. His research interests include pattern recognition, Arabic document analysis and recognition (including Arabic text recognition and writer identification), Arabic natural language processing, image analysis and application of pattern recognition in software engineering, medical imaging, etc. He is a senior member of IEEE. He published over 80 papers in refereed journals and conference proceedings in his research areas of interest.

Sameh Awaida is an Assistant Professor in the Computer Engineering Department in Qassim University, Saudi Arabia. He obtained his Ph.D. in Computer Science and Engineering (CSE) from KFUPM in 2011. He got his B.Sc. and M.Sc. in Electrical Engineering from University of Hartford (CT, USA) in 2003 and 2005, respectively. Previously, he worked as a Lecturer in KFUPM and PSUT. His research interests include pattern recognition, image processing and embedded systems. Along with two patents, he has published more than 10 research papers in international journals and conferences.