JKAU: Eng. Sci., Vol. **27** No.1, pp: 3 - 17 (1437 A.H./ 2016 A.D.)

Doi: 10.4197/Eng. 27-1.1

Automatic Recognition of Artistic Arabic Calligraphy Types

S. R. Allaf and **R. Al-Hmouz**

Department of Electrical and Computer Engineering, King Abdulaziz University, P. O. Box 80204, Jeddah 21589, Saudi Arabia
allaf419@gmail.com, ralhmouz@kau.edu.sa

Abstract. In this paper, we propose a new approach to recognizing artistic Arabic calligraphy types, which are handwritten scripts written by special calligraphy pens. The nature of the structural composition of Arabic calligraphy makes it challenging to create such a recognition system: Difficulties include similarities among different types, overlap between letters, and letters that themselves assume different shapes. A new off-line technique of font recognition based on extracting distinctive features of each type is presented in this work. Features of selected types of artistic Arabic calligraphy are extracted to construct the features vector. The features vector is used in the classification process to recognize the type. A genetic algorithm is used to optimize the number of features and the image size that will be considered in the classification stage. In the classification stage, we used a neural network module. The approach is tested on two different datasets. One is a local dataset of three different Arabic handwritten calligraphy types, Thuluth, Reqaa, and Kufi. The other dataset is a public dataset of 10 different computer-generated fonts. The recognition error rate for the local and public datasets was 8.02% and 7.55%, respectively.

Keywords: Optical font recognition, Artistic Arabic calligraphy, Features extraction, Classification, Optimization, Neural network, Genetic algorithm.

1. Introduction

Arabic calligraphy has several famous styles (types), such as Thuluth, Naskh, Dewani, Reqa'a, Jali Dewani, Farsi, and Kufi. Each type has distinctive features that can be utilized in the identification process. Figure 1 shows some different artistic Arabic calligraphy types and their names. The distinctive properties of each type can be visually recognized. However, automating the process is challenging. Some of these challenges result from the similarity among different types, overlap between letters, and letters that themselves assume different shapes. Other challenges are related to the human nature of the calligrapher drawing the text. The calligraphers are writing the entire Arabic text with a pen, and the shape of the same letter-or even word-will have small variations from instance to instance. Moreover, Arabic calligraphy has structural characteristics, such as right to left direction, cursive script, dots, diacritics, and changes in letter shape, depending on letter location. Automatic recognition of artistic Arabic calligraphy types is critical to the development of engineering solutions (in both hardware and software fields) for the study and teaching of Arabic calligraphy.

Many letters of different Arabic calligraphy types are mostly the same just with small variations. Only a sensitive module can extract these tiny differences. For example, the letter alef (أ) has a semi-vertical shape among the different types. The orientation of the letter is different for each type and must be identified correctly to avoid incorrect identifications. Figure 2 shows different orientations for some Arabic calligraphy types for the letter alef (أ).

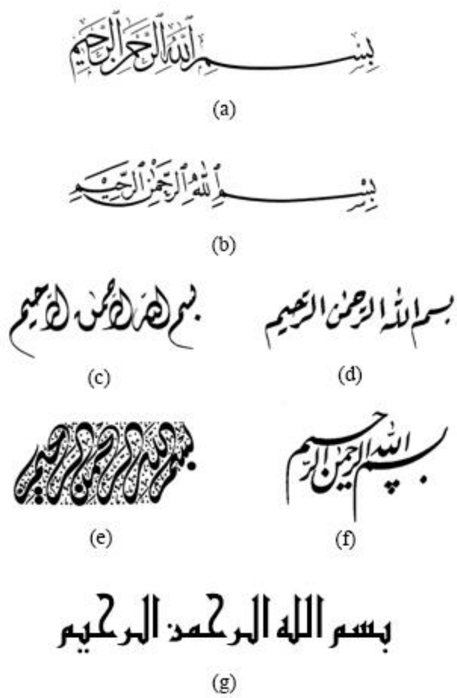


Fig. 1. Samples of different artistic Arabic calligraphy types:
(a) Thuluth, (b) Naskh, (c) Dewani, (d) Reqa'a, (e) Jali Dewani (f) Farsi (g) Kufi.

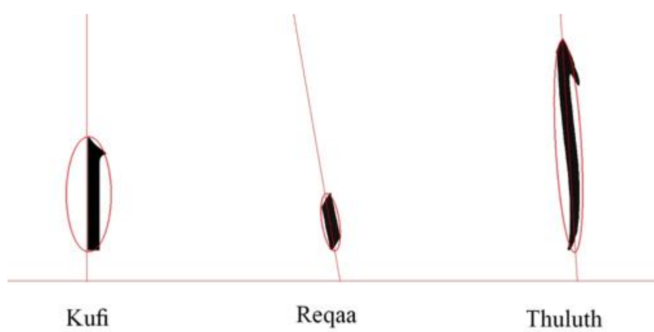


Fig. 2. Different orientations for the letter alef (أ) for three different Arabic calligraphy types.

Optical Font Recognition (OFR) is the process of identifying the font type from a text image or a document, and it is an application of Optical Character Recognition (OCR) [1]. OCR is one of the most important applications of digital image processing algorithms. Generally, font recognition can be divided into two groups, on-line and off-line recognition. On-line recognition is a way to recognize a text directly from an on-line device, such as drawing tables, optical pens, or even a gesture on touch screen. Off-line recognition addresses the pre-captured text images taken by a camera or scanner [2]. Most of the past attempts at OFR dealt with typographical letters printed using machines or printers. These letters have the same exact shape in all instances throughout the entire text image [2,3-4].

OFR is a pattern recognition problem based on the extraction of a set of features from document images. There are two ways to compute these features, locally from individual characters or globally from large text entities such as words, lines, or even paragraphs [5]. Generally, there are three main approaches to the OCR process: template matching methods [4], structural methods [5], and statistical methods [6]. In template matching methods a collection of character templates is maintained and used to identify patterns. The recognition consists of finding the closest matching template. Structural methods are based on the topological structure of the character. They model characters by structural features and their relationships. In statistical methods, the pattern recognition process is based on statistical decision theory; each pattern is considered as a single entity and is represented by a finite dimensional vector of pattern features [3].

The AprioriOptical Font Identification System (ApOFIS) uses global typographical features from the text image that are fed to a multivariate Bayesian classifier [1]. A global features approach was also presented in [6]; this method is based on statistical analysis of the relationships between the pixels at the edge of an image that contains black and white levels as an example. The proposed method has been implemented in an optical font recognition application to identify Arabic calligraphy font type. In Ref. [5], a texture-analysis-based approach to the recognition of Chinese and English fonts was presented. The document was considered to be an image that contains some specific textures, and then the texture was identified to recognize the font. The method is content-independent and involves no detailed local feature analysis.

Moreover, Abuhaiba [4] presented an algorithm for a priori Arabic optical Font Recognition (AFR) based on templates. Four attributes are used to identify a font type: typeface, size expressed in typographic points, slant, and weight. Meanwhile, Slimane et al. [7] have proposed a font and size identification method for ultra-low resolution Arabic word images using a stochastic approach. Their method treats a word image with a fixed length of overlapping sliding window. Each window is represented with 102 features, the distribution of which is captured by Gaussian Mixture Models (GMMs). These researchers present three systems: (1) A font recognition system, (2) A size recognition system, and (3) a font and size recognition system.

Recently in Ref. [8], the authors have presented a method for Arabic font recognition based on segmenting the diacritics from the input text image, then extracting a composite of central and ring projection (CCRP) features from these diacritics.

In this paper, statistical features will be extracted from a text image. The text image and the number of features in the classification stage are selected via an optimization process to improve the performance in regard to the classification error. The remaining parts of this paper can be divided as follows: In section 2, we describe the methodology and materials that are used to implement the proposed module including optimization and classification processes. Section 3 shows the experiment setups, followed by results and discussion in section 4. Finally, section 5 presents the conclusions .

2. Methodology and Materials

The proposed approach to Arabic calligraphy recognition consists of three stages: preprocessing of the image, features extraction, and classification. All stages will be explained in detail in subsequent subsections. The algorithm is visualized in flowchart form in Fig. 3.

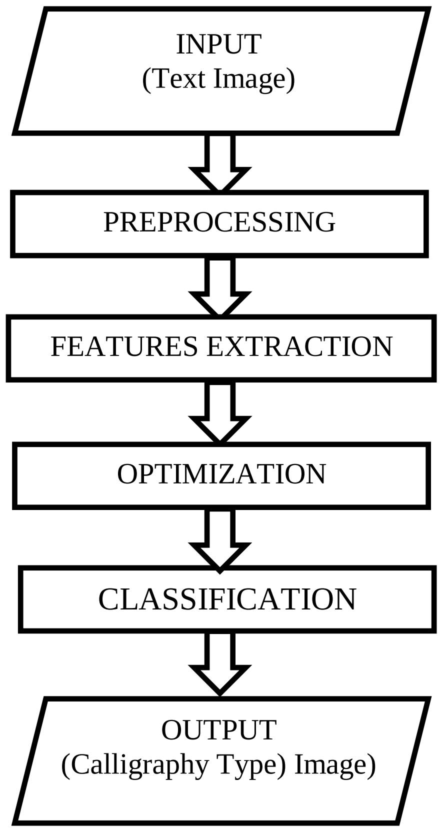


Fig. 3. Flowchart of recognition stages of the proposed module.

2.1 Preprocessing

As we are working on an off-line recognition system, scanner based text images will be the input to the system. Accordingly, the input text image must be prepared before further processing. First, the image is converted to binary format (black and white) which means that all the pixels of the image should have either 0 (black) or 1 (white) values. Hence, the image is represented in a matrix all the elements of which are 0 or 1. The scanning process might produce noise, which is represented by small black components in the text image. This noise is eliminated in this stage by a simple filter mechanism. Additionally, all unused (empty) rows and columns of the image matrix should be removed in order to extract some features (except the orientation case). Figure 4 shows examples of text images before and after preprocessing.

2.2 Features Extraction

Binary images can be represented by a matrix of zeroes and ones and each pixel of a binary image is an element of this matrix. All typical operations on matrices can be used to obtain information that can be used in the feature extracting stage. We form a vector from seven features; the number of features will be optimized in the later optimization stage. We primarily investigated the following features:

a) Reference line (RL).

b) The black/white ratio for the entire image.

c) The black/white ratio above the reference line.

d) The black/white ratio below the reference line.

e) Orientation of components in the image.

f) Density of components above the reference line.

g) Density of components below the reference line.

a) Reference line (RL)

We define a reference line (RL) as the row that contains the maximum number of pixels' letters. Having said that, RL is the index of the row in the matrix that has the maximum number of zeros. To locate the RL, we added all pixel values for each row (vertical histogram), and then we identify the index of the row that has a minimum value. Figure 5 shows the reference line for certain different types of Arabic calligraphy.

b) The black/white ratio for the entire image

The ratio of black and white pixels can be identified by dividing the number of black pixels by the number of white pixels. The number of black and white pixels can be easily obtained by adding up the number of black or white pixels in the image (see Fig. 6 c).

c) The black/white ratio above the reference line

In this feature and the next one, we calculate the ratio of black and white in the area above the reference line. We noticed that in some fonts most of the black pixels are located above the reference line. Thus, we focus only on the area above the reference line to calculate the ratio as depicted in Fig. 6 (a).

d) The black/white ratio below the reference line

The remaining part of the image is the area under the reference line. The ratio of black and white pixels is calculated for the lower part of the image below the reference line. Additionally, it was noticed in some fonts that most of the pixels in this area are white. Fig. 6. shows different ratios for a text image for the three fonts below the reference line. Figure 6 (b).

e) Orientation

Orientation is the angle between horizontal lines and the major axes of objects in the image. We specify the angle between the x-axis and the major axis of the ellipse of connected components in the binary image. Accordingly, there is a certain angle that most of the objects of calligraphy text form. Figure 7 shows the different angles (orientations) for three different Arabic calligraphy types. This feature is simply the mean of the angles formed by the objects.

f) Density of elements above the reference line

The density of elements can be obtained by counting the number of components above the reference line. It just compares the number of connected components above the reference line to the total number of connected components in the image (see Fig. 8 b).

g) Density of elements under the reference line

This is like the feature in (f), except instead of counting the number above the line, the density is computed for the number of connected components below the reference line (see Fig. 8 a).

Fig. 4. Examples of text images, before preprocessing (a), after removing the empty rows/columns around the text (b) and between the letters (c).

Fig. 5. Reference line position for three different Arabic Calligraphy types: (a) Thuluth (b) Kufi (c) Reqaa.

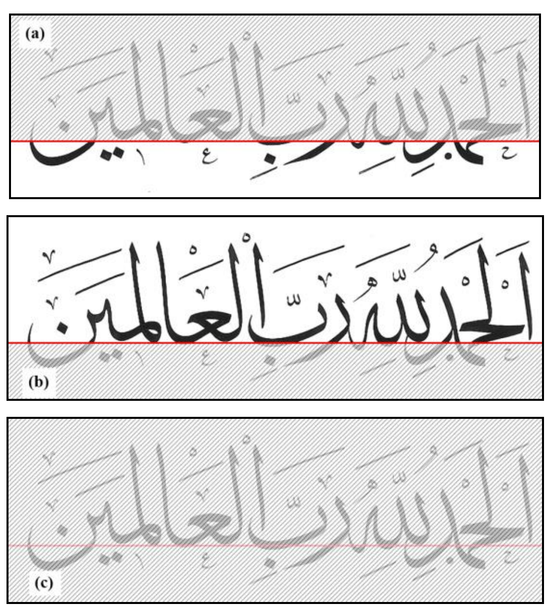
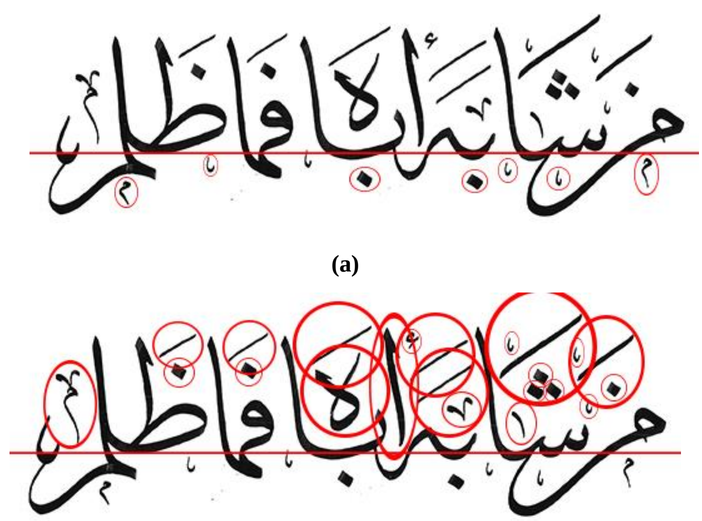


Fig. 6. Different ratio areas for a text image according the reference line:
- **(a) The area above the reference line.**
- **(b) The area below the reference line.**
- **(c) The total area of the image.**

(a) (b) (c)

Fig. 7. Orientations for different Arabic calligraphy types: (a) Reqaa (b) Kufi (c) Thuluth.



(a)

(b)

Fig. 8. Density of elements: (a) below reference line and (b) above reference line.

2.3 Optimization Scheme

The optimization block is presented in this work in order to improve the performance of the system. The genetic algorithm is a well-known optimization technique that is inspired by human genetics. The Genetic Algorithm (GA) is a random heuristic search algorithm based on natural selection and genetics.

The basic technique of GA is simulating the processes in natural systems that are needed for evolution. Evolution starts with a population of randomly generated individuals who form a generation. This process is repeated several times to create new generations. In each generation, every individual in the population is evaluated to check its fitness. The more fit individuals are randomly selected from the current population and each individual's genome is modified to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. The algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population [9].

Genetic algorithms are used to optimize:

1) Length and width of text images: Different sizes of image will result in classification error. Although the width and length can be determined in an iterative process, we used genetic algorithms to reduce the time needed to find the value of these parameters. The fitness function in this case is

$$w, l = arg_{i=1,2,..,m} \min r$$

where

w, l are the width and length, respectively, that have to be optimized; m is the number of artistic calligraphy used; and r is the classification error of the available set.

$$r = 1 - \frac{n}{N}$$

where

n:total number of images correctly classified.

N: total number of images in the dataset.

2) The number of features: Feature selection approach is considered in this part. It was noticed that the values of some features for the available fonts are almost the same. We used genetic algorithms to determine the most distinctive features of the fonts in order to consider them in the final stage. The fitness function in this case will be

$$f_{1,2,..7} = arg_{i=1,2,..,m} \min r$$

Where f is a binary vector each element of which is an 1 (informative feature) or a 0 (non informative feature).

2.4 Classification Scheme

Classification is the task of assigning objects to one of several predefined categories. Classification is a very widely studied problem that has many typical solutions. One of the most efficient methods of solution is a neural network (NN). A neural network is a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs [10].

A neural network has the following components:

a) Input Layer.

b) Hidden Layers.

c) Weighted Connections.

d) Output Layer.

Text images are presented to the network via the input layer, which communicates to the hidden layers where the processing is performed via the weighted connections. The hidden layers then link to an output layer, which shows the recognized type. Figure 9 shows a typical neural network diagram of 4x5x1 neurons in three layers.

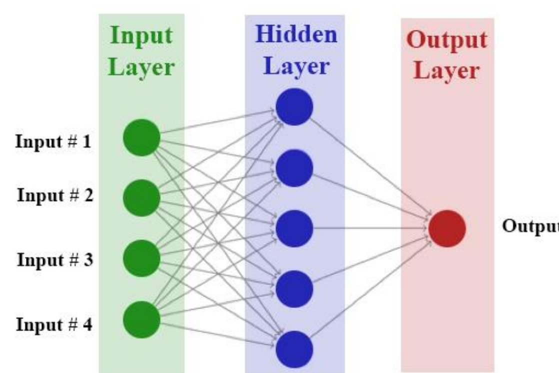


Fig. 9. A typical neural network diagram.

The classification process has three stages. In each stage we use a part of our dataset. Training, validation, and testing stages are used to build up a high efficiency classifier. The training stage is an operation to adjust the weights of the connections between the input layer and the hidden layers to increase the accuracy of the classification module. The validation stage can be considered as a part of the training stage, but actually we use a different part of the dataset to perform the validation stage. The goal of the validation process is tuning the parameters of the model to avoid overfitting. Additionally, we used the validation set in the optimization scheme. The testing stage is used to evaluate the performance of the model.

3. Experiment

The algorithm has been tested on two sets:

1- Local data set: 18 4- or 5-word sentences were written by a professional calligrapher, each in three fonts (Thuluth, Reqaa and Kufi). Next, they were scanned and saved in jpg format. Additionally, words were manually separated from sentence images, and some words were written individually because they could not be extracted from the original sentence. In total we had 18 sentences and 71 words for each font. Samples from the local set are displayed in Fig. 10.

2- Public set: A computer-generated font was also used [11]. In this set, there were 10 computer-based fonts in different sizes and formats. Only word images were available. Samples of the public set are shown in Fig. 11. The summary of both sets is presented in Table 1. We randomly combined 20 sentences, each with five words, for all fonts to evaluate the performance of the proposed work.

Fig. 10. Samples of the local dataset: (a) Thuluth (b) Kufi (c) Reqaa

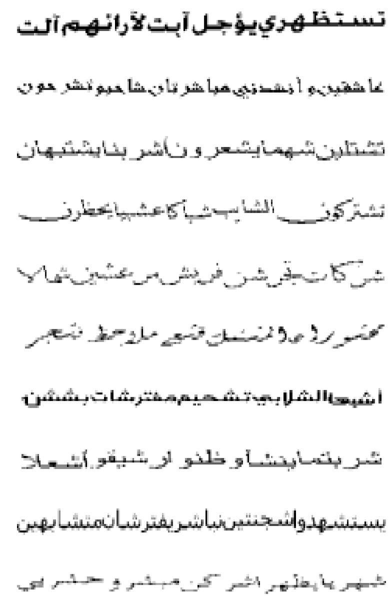


Fig. 11. Samples of the public dataset:(a) AdvertisingBold (b) Andalus (c) ArabicTransparent (d) DecoTypeNaskh (e) DecoTypeThuluth (f) DiwaniLetter (g) MUnicodeSara (h) SimplifiedArabic (i) Tahoma (k) TraditionalArabic.

Table 1. Summary of the local/public data sets used in the experiment.

Set	Calligraphy Type	No. of words/ type	No. of sentence s/type	Total no. of images
Local	Thuluth	71	18	89
	Reqaa	71	18	89
	Kufi	71	18	89
Public	AdvertisingBold	113284	0	113284
	Andalus	113284	0	113284
	ArabicTransparent	113284	0	113284
	DecoTypeNaskh	113284	0	113284
	DecoTypeThuluth	113284	0	113284
	DiwaniLetter	113284	0	113284
	MUnicodeSara	113284	0	113284
	SimplifiedArabic	113284	0	113284
	Tahoma	113284	0	113284
	TraditionalArabic	113284	0	113284

The data from both the public and local sets has been split into three subsets, the training set (70%), the validation set (10%), and the test set (20%). The results in each set were obtained by averaging the results for 10 iterations. In each iteration, 70% of the data are randomly selected for training the neural network, and the remaining 30% is randomly split between the validation set (10%) for optimization (size, feature selection) and the test set (20%). We use the performance of algorithms in the test set as the indicator of performance. The performance of the training and validation sets were also recorded.

As mentioned earlier, images come in different sizes. Increasing and decreasing the width and length can improve or degrade the classification rate. The genetic algorithm is considered here as an optimization tool for the image size as well as for the feature selection approach. The parameters of the genetic algorithms were set as follows: the number of generations is 100 and in each generation there are 20 populations. The mutation function is Gaussian with crossover of 0.8.

As for the neural network, the input layer was set to 7 neurons and the output layer was set to 3 in the local set and 10 in the public set. The hidden layer was set in an iterative process to achieve the minimum classification error. Figure 12 shows the performance of the validation set. The minimum classification error is achieved when setting the hidden layer to 7. Therefore, the structure of the neural network for both sets is shown in Fig. 13.

It should be noted that both optimization processes were applied on the validation set. In the size optimization stage of the local set, the width and length of the input font images were optimized. Fig. 14 shows the best classification error in each generation. This figure also shows that the process converges across generations to values of 823 and 2168 for width and length, respectively. The previous values were plugged into another optimization process in order to select the best combinations of the available seven features. Fig. 15. shows the best classification error in each generation. All features are presented at the end of the optimization except the feature of index 6 (density of elements under the reference line). Hence, the performance of the test set was recorded by ignoring the feature of density of elements under the reference line.

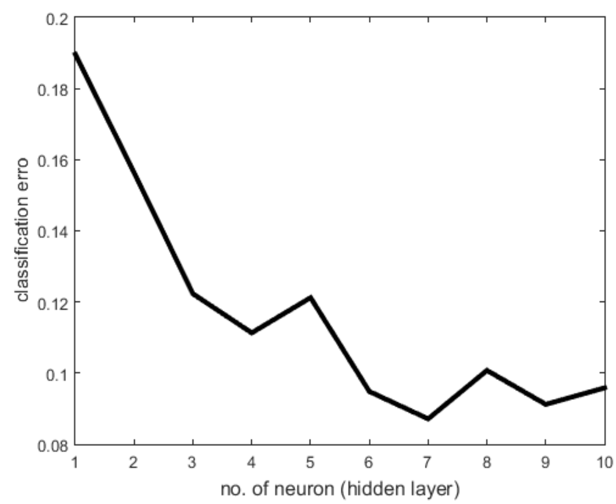
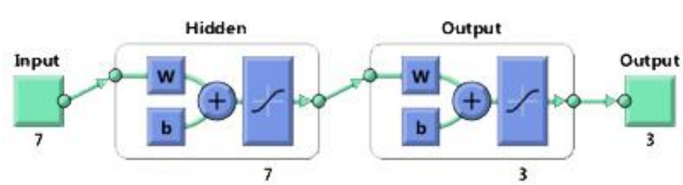
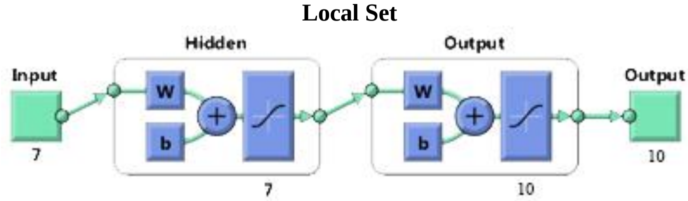


Fig. 12. The minimum classification error of the validation test when setting the hidden layer to 7.



Local Set



Public Set

Fig. 13. The structure of the neural network for (a) local set (b) public set.

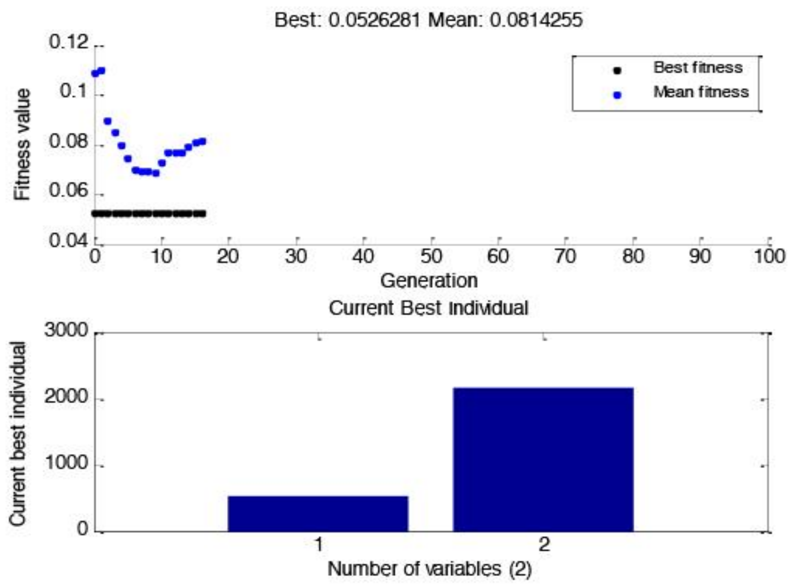


Fig. 14. The optimization process of image size using GA (local set).

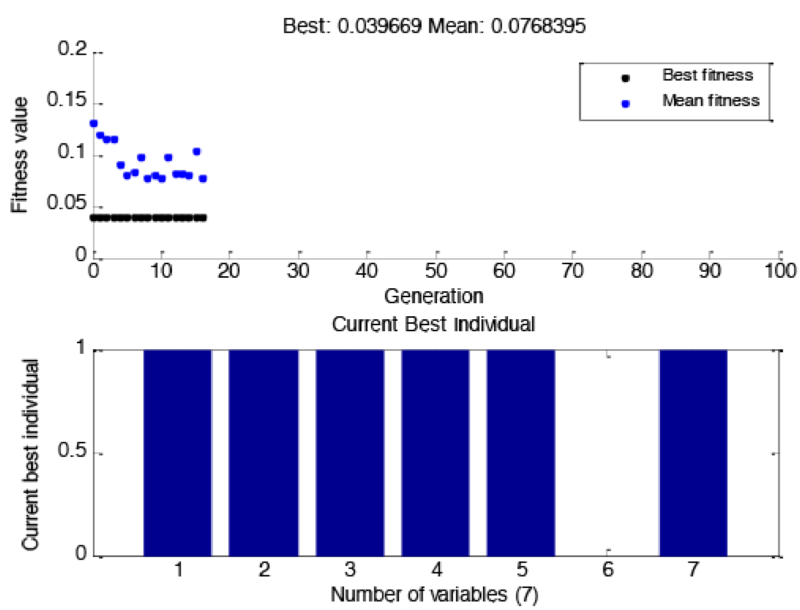


Fig. 15. The selection process of features using GA (local set).

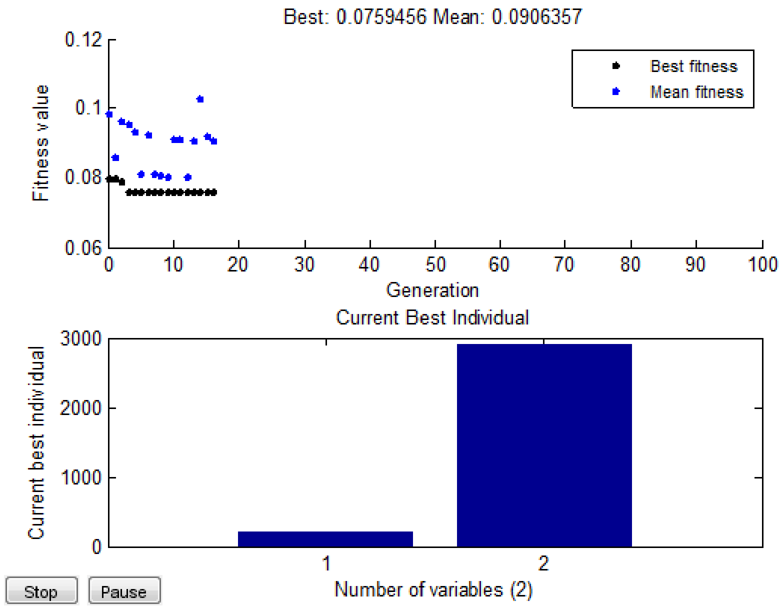


Fig. 16. The optimization process of image size using GA (public set).

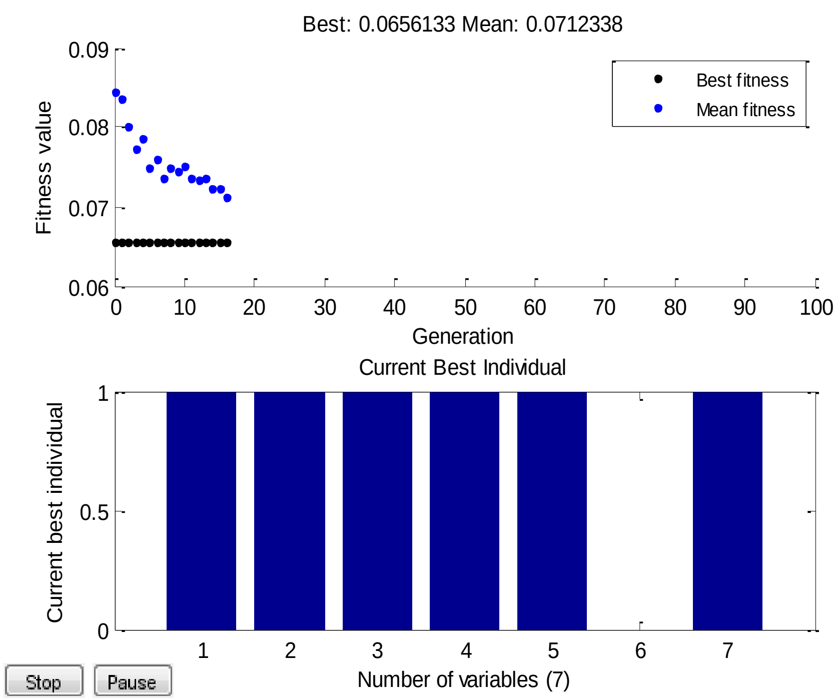


Fig. 17. The selection process of features using GA (public set).

For the pubic set the size converged to [219, 2919] (see Fig. 16) for width and length and the same feature (density of elements under the reference line) provide no extra information. On the contrary, it degrades the performance if it presents in the final stage (see Fig. 17) and is therefore ignored in recording the performance.

3. Results and Discussion

The classification errors for all subsets (training, validation, testing) are tabulated in Table 2. The performance was improved for all subsets. The best performance was achieved when using the size and features optimization scheme. The classification error for each font is also shown in Table 3. Kufi type has the minimum classification error due to the easily identifiable 90-degree angle it has between the vertical and horizontal parts of the letters. Table 4 shows the results of the public set with and without optimization. Almost the same performance was obtained for both sets, although the public set has 10 classes. Also, the individual font performance is shown in Table 5.

Table 2. The classification errors for all subsets (training, validation, testing) for the local set.

Criteria	Training	Validation	Testing
No optimization	9.16%	10.91%	10.18%
Size optimization	8.06%	8.01%	9.18%
Feature selection	6.37%	7.68%	8.02%

Table 3. The classification errors for all subsets (training, validation, testing) for the local set.

Calligraphy type	Testing set
Thuluth	6.67%
Kufi	0.6%
Reqaa	16.80%

Table 4. The classification errors for all subsets (training, validation, testing) for the public set.

criteria	Training	Validation	Testing
No optimization	9.87%	8.11%	8.18%
Size optimization	7.45%	7.59%	7.45%
Feature selection	7.55%	6.56%	7.55%

Table 5. The classification errors for all subsets (training, validation, testing) for the public set.

Calligraphy type	Testing set
AdvertisingBold	8.75%
Andalus	8.37%
ArabicTransparent	9.49%
DecoTypeNaskh	9.21%
DecoTypeThuluth	8.05%
DiwaniLetter	7.09%
MUnicodeSara	6.76%
SimplifiedArabic	8.76%
Tahoma	8.59%
TraditionalArabic	9.74%

In Table 6, we report the classification errors of other approaches in the literature. Although some approaches reported less classification error, the proposed algorithm is incomparable in terms of results because of the difference in the experimental setup (e.g., data set, training set, and testing set) and the nature of calligraphy data (printed and handwritten).

Table 6. Classification errors reported in the literature.

Publication	Classification Error Rate
Bataineh *et al*. [6]	4.66%
Abuhaiba [4]	22.6%
Al-Muhtaseb *et al*. [2]	0.06%
Slimane *et al*. [7]	5.5%
Ben Moussa *et al*. [12]	3.4%
Lutf *et al*. [8]	1.27%
Our work	8.02%

4. Conclusions

In this paper, we have presented a novel method for recognizing different types of Arabic calligraphy. We designed a method for recognizing handwritten (artistic) types. The model is based on extracting seven features from the text images to form a features vector that is used in the classification process. We used a neural network to recognize the different calligraphy types along with a genetic algorithm to select the best size of the text image and the best combination of the features to achieve a better classification rate. We tested our model on two different data sets. The results of the proposed approach showed a competitive recognition rate error. The problem of handwritten Arabic calligraphy is more complex than the computer based Arabic font according to the results obtained. For future work, scale invariant feature transform (SIFT) features will be examined in the presence of the feature selection approach, and more types of Arabic calligraphy will be tested.

References

[1] **Abdelwahab, Z.** and **Rolf, I.,** Optical Font Recognition Using Typographical Features, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **20**(8): 877-882 (1998).

[2] **Al-Muhtaseb, H. A., Mahmoud, S.A., Qahwaji, R.S.** and **Rami, S.**, Recognition of Off-line Printed Arabic Text Using Hidden Markov Models, *Signal Processing,* **88**: 2902–2912 (2008).

[3] **Abdelwahab, Z.,** Study of Optical Font Recognition Based on Global Typographical Features, *Doctoral dissertation,* The University of Fribourg, Switzerland, (1995)

[4] **Abuhaiba, I.,** Arabic Font Recognition Based on Templates, *The International Arab Journal of Information Technology*, **1** (0): 33-39 (2003).

[5] **Yong, Z., Tieniu, T** and **Yunhong, W.,** Font Recognition Based on Global Texture Analysis, Document Analysis and Recognition, 1999. ICDAR '99. *Proceedings of the Fifth International Conference on, Bangalore, Sep. 1999*, pp. 349 – 352.

[6] **Bataineh, B., Sheikh, A., Norul Huda, S.** and **Khairudin, O. A.,** *Statistical Global Feature Extraction Method for Optical Font Recognition*, N.T. Nguyen, C.-G. Kim, and A. Janiak (Eds.): ACIIDS 2011, LNAI 6591, pp. 257–267.

[7] **Fouad, S., Slim, K., Jean, H., Adel, A.** and **Rolf, I.**, A study on font-family and font-size recognition applied to Arabic word images at ultra-low resolution, *Pattern Recognition Letters*, **34**: 209-218 (2013).

[8] **Mohammed, L., Xinge, Y., Yiu-ming, C.** and **Philip, C.L.,** Arabic font recognition based on diacritics features, *Pattern Recognition*, **47**: 672-684 (2014).

[9] **Houck, C. R., Joines, J. A.** and **Kay, M. G.,** A Genetic Algorithm for function optimization: A Matlab Implementation, *NCSU-IE TR*, **95**(9): 523-532 (1995).

[10] **Yegnanarayana, B.,** *Artificial Neural Networks*, New Delhi, Prentice-Hall of India. (2006).

[11] Online available: https://diuf.unifr.ch/diva/APTI/.

[12] **Ben Moussa, S., Zahour, A., Benabdelhafi, A.** and **Alimi, A.,** New Features Using Fractal Multi-Dimensions for Generalized Arabic Font Recognition, *Pattern Recognition Letters*, **31** (5): 361-371 (2009).

التعرف الآلي على نوع الخط العربي الفني

سراج رضا علاف و **رامي الحموز**

قسم الهندسة الكهربائية وهندسة الحاسبات، جامعة الملك عبد العزيز، ص.ب.٨٠٢٠٤، جدة ٢١٥٨٩، المملكة العربية السعودية

allaf419@gmail.com, ralhmouz@kau.edu.sa

المستخلص. قمنا في هذه الدراسة بتقديم منهجية جديدة للتعرف على أنواع الخط العربي الفني المكتوبة بخط اليد، باستخدام قلم الخط العربي الخاص. تعتبر طبيعة التركيب البنائي للخط العربي مشكلة تحتاج إلى حل لإنشاء نظام تعرف آلي قادر على التغلب على المعوقات الموجودة، مثل التشابه بين الأنواع المختلفة، والتداخل بين الحروف، ووجود أكثر من شكل للحرف نفسه. في هذا العمل، تم تقديم نظام جديد غير لحظي للتعرف على نوع الخط، وذلك بالاعتماد على استخراج الملامح المميزة لكل نوع. تم استخراج الملامح المميزة لبعض الخطوط العربية الفنية المختارة، وذلك لبناء مصفوفة الملامح. كما تم استخدام مصفوفة الملامح في عملية التصنيف للتعرف على النوع. في مرحلة التصنيف، استخدمنا نموذج الشبكات العصبية. واستخدمت الخوارزمية الجينية لتحسين عدد الخصائص المستخدمة وحجم الصور المدخلة، والتي تم أخذها بعين الاعتبار في مرحلة التصنيف. وتم اختبار المنهجية على قاعدتي بيانات مختلفتين: قاعدة بينات خاصة مكونة من ثلاثة أنواع مختلفة من الخط العربي اليدوي (وهي الكوفي، الرقعة والثلث)، قاعدة البيانات الأخرى هي قاعدة بيانات منشورة، وتتكون من ١٠ أنواع مختلفة من الخطوط المولدة من خلال الحاسوب. وجد أن معدل الخطأ لقاعدة البيانات الخاصة والمنشورة كان ٨,٠٢٪ و ٧,٥٥٪ على التوالي.

كلمات مفتاحية: التعرف على الخط الفني العربي، استخراج وتصنيف خصائص الخط، الشبكة العصبية، الخوارزمية الجينية.