Aolah Databases for New Arabic Online Handwriting Recognition Algorithm

Samia Heshmat(✉) and Mohamed Abdelnafea

Faculty of Engineering, Aswan University, Aswan 81542, Egypt
samia.heshmat@aswu.edu.eg

Abstract. Developing an online handwriting recognition system for Arabic script used in pen-based devices plays an important role in making these devices available and usable for Arabic society. This paper is carried out for Arabic script to overcome the difficulties presented in the Arabic language in cursive, overlapping, handwriting variability, different writing styles, delayed strokes, and other challenges. An algorithm for recognizing Arabic strokes written by hand is proposed; since there are some troubles in distinguishing the written stroke for similar characters. The uniqueness of the recommended algorithm is dealing with every stroke in the character separately. Furthermore, in the current research, two novel databases for Arabic characters and Arabic characters' strokes are generated. The two databases are presented, one for Arabic characters by different writers for the 28 Arabic characters, the other database is extracted from the previous database by taking only the Arabic character strokes. The algorithm used for data collection is distinguished by the ability to deal with each stroke in the written characters separately. The code acts as a simulation of a stylus pen and a touch screen. Stroke capturing is achieved by collecting data points along the path of an input device (stylus pen or mouse) same time those characters are written.

Keywords: Arabic online database · Data collection and preprocessing · Machine learning · Handwriting recognition · Artificial Intelligence

1 Introduction

Arabic script is an alphabet written from right to left which contains two types of symbols for writing words: letters and diacritics. Letters consist of two parts: letter form and letter mark. The letter form is an essential component in each letter with a total of 19 letter forms. The letter marks may be dots, short Kaf, or Hamza letter mark. Hamza is used for both the letter form and the letter mark, which appears with other letter forms. The Madda letter mark is a Hamza variant; Fig. 1 indicates the Arabic script. Diacritics, the second symbol in writing Arabic words which is not essential in writing like the main letter. Three types of diacritics are there: Vowel which are Fatha ◌َ, Damma ◌ُ, Kasra ◌ِ, or Sukun ◌ْmeans no vowel, Nunation which is a doubled version of their corresponding short vowels are two Fathas, two Dammas, two Kasras, and Shadda which is a consonant doubling diacritic. Figure 2 shows types of Arabic diacritics [1–3].

E. H. Barney Smith and U. Pal (Eds.): ICDAR 2021 Workshops, LNCS 12916, pp. 294–311, 2021.
https://doi.org/10.1007/978-3-030-86198-8_21

There are many challenges do exist when the Arabic script is written by hand and that is due to its unique nature, it is cursive and overlapping occurs between Arabic letters, each character has more than one shape, and other challenges [4]. To build a recognition system for Arabic handwriting words researchers need a real and substantial database [5, 6]; hence, this work is a contribution in producing an Arabic database to help researchers in this field and to overcome challenges existed in previous databases. The databases are developed based on an algorithm that uses stroke capturing to facilitate recognition of Arabic characters [7]. The proposed databases (AOLAH) are typical formats of online handwritten data which is a sequence of coordinate points of the moving pen point. Connected parts of the pen trace, in which the pen point is touching the writing surface, are called strokes [8–10].

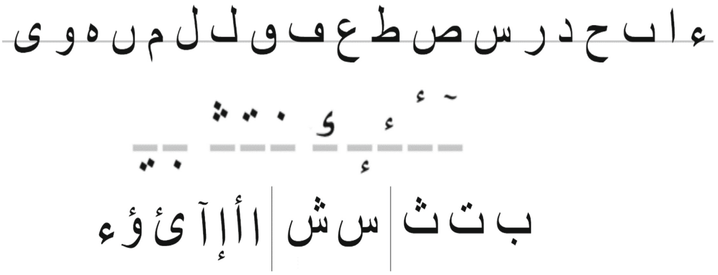


Fig. 1. Arabic script.

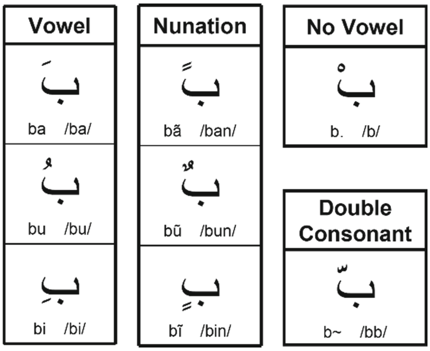


Fig. 2. Arabic diacritics types.

The remainder of this paper is organized as follows: a discussion about existing Arabic databases is shown in Sect. 2. Section 3 presents the proposed AOLAH databases for Arabic online handwriting letters and strokes. While Sects. 4 and 5 describe the proposed Arabic online handwriting recognition algorithm with showing the optimum proposed recognition model Conclusions are given in Sect. 6.

2 Present Databases for Handwritten Arabic Letters

This section describes the main databases used in online Arabic handwriting recognition researches. Table 1 shows a summary of these databases.

2.1 LMCA (2008) [11, 12]

The On/Off (LMCA) dual Arabic handwriting database; this abbreviation is from the French sentence which is Lettres, Mots et Chiffres Arabe. This database contains 30,000 digits, 100,000 Arabic letters and 500 Arabic words; there were 55 participants invited to contribute. This database was developed by REGIM laboratory which is abbreviated for REsearch Group on Intelligent Machines. Both on/off line handwritten characters and words were considered. LMCA database is limited to a small set of words, and the letters are collected separately which means not segmented from cursive text.

2.2 OHASD (2010) [13]

This database is considered as first online Arabic sentence database handwritten on tablet PC. The final version of this dataset is composed of 154 paragraphs, selected from public daily news, written by 48 writers, having a total of 3,825 words and 19,467 characters, after excluding erratic/illegible handwritings. This database has a limited lexicon, limited data, and a limited number of writers.

Table 1. Main present online databases.

Database	Chars	Words	Writers	Main Drawbacks
LMCA (2008)	100000	500	55	– Limited to a small set of words – Letters collected separately
OHASD (2010)	19467	3825	48	– Limited lexicon – Limited data – Limited number of writers
ADAB (2011)	174690	33164	166	– Isolated word samples "not a natural Arabic online handwriting" – No segmentation of words into letters
ALTEC (2014)	106433	152680	1001	– Data collected not in a natural Arabic online handwriting way
QHW (2014)	42800	12000	200	– Closed vocabulary database.- Limited number of words
Online-KHATT (2018)	801421	80931	623	– No dealing with characters on the base of its strokes

2.3 ADAB (2011) [14, 15]

This database was developed by the institut fuer Nachrichtentechnik and the research group on intelligent machines (REGIM). It contains online samples of 937 Tunisian city names that consist of 33,164 Arabic words which are 174,690 characters written by approximately 166 writers. It is used in competitions. The data are available in isolated word samples which are not a natural Arabic online handwriting, and no segmentation of the words into letters is provided.

2.4 ALTEC (2014) [16]

This database is produced by the Arabic language technology center (ALTEC) for online Arabic text with a large lexicon. It consists of 152,680 samples of 39,945 unique words, including 325,477 samples of 14,740 unique parts of a word, the database is collected from approximately 1,000 writers where samples are complete sentences that include digits and punctuation marks and the collected data is available on sentence, word and character levels. The main drawback of this database is that the data are collected by using a device digitally captures and stores everything written or drawn with ink on ordinary paper.

2.5 QHW (2014) [17]

The Quranic handwritten words database is the most commonly used words in the holy Quran. Handwritten words were chosen as the most common words repeated in the holy Quran. The initial version of QHW database includes 120 handwritten words and divided equally into two sets written by 200 writers in total. The QHW database contains 12,000 sample including more than 42,800 characters and 23,300 sub words. This database is a closed vocabulary database and has samples of a limited number of words.

2.6 Online-KHATT (2018) [18]

The Online-KHATT database contains more than 80,000 Arabic words written by 623 writers with approximation 801,421 characters using a source text that covers several domains to ensure a wide range of topics. Online-KHATT database may be considered as the largest Arabic online text database in terms of the number of lines written with electronic pens using natural Arabic text; however it ignored dealing with characters on the base of its strokes.

3 Proposed AOLAH Databases for Arabic Online Handwriting Letters and Strokes

Due to the drawbacks presented in the previous databases there is an essential need for databases overcomes those drawbacks. This work tries to seed a seed in this field. The proposed Arabic online handwriting recognition algorithm that is used in collecting databases mainly depends on the idea of collecting strokes as a separate unit as the stroke is the first base of any word. To do the process of stroke capturing we had developed an algorithm that was written by MATLAB. This algorithm provides a GUI to display the collected data from pen movements, theses pen movements were simulated by mouse where pen down is simulated by mouse left click, pen movement is simulated by holding the mouse left click while writing, and pen up is simulated by releasing mouse left click. The input pen movements are collected as a sequence of points and further are stored in a text file. The text file storage is required to retain original pen movements that are required at later stages in recognition beginning with preprocessing [19, 20]. Furthermore, those text files may also be used to verify the input stroke shape by the help of any application that may visualize data like Microsoft excel. Figure 3 indicates the graphical user interface of the developed application to collect the databases with the Arabic character zha which is written in three strokes and the screenshot of the data stored in the text file for this character is shown in Fig. 4, where the beginning and end of each stroke is clarified in the table.

The Proposed AOLAH databases are contributions from Faculty of Engineering, Aswan University to help researchers in the field of online handwriting recognition to build a powerful system to recognize Arabic handwritten script. AOLAH stands for Aswan On-Line Arabic Handwritten where Aswan is a small beautiful city located at the south of Egypt. Word On-Line in database's name means that the databases are collected the same time as they are written. While, Arabic word is used because these databases are just collected for Arabic characters; and Handwritten word since these databases are written by the natural human hand.

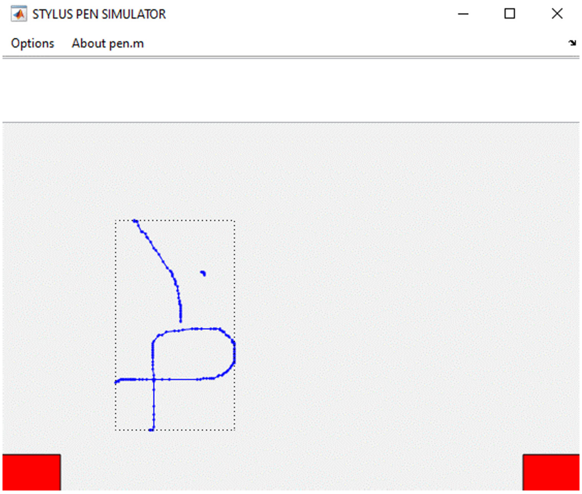
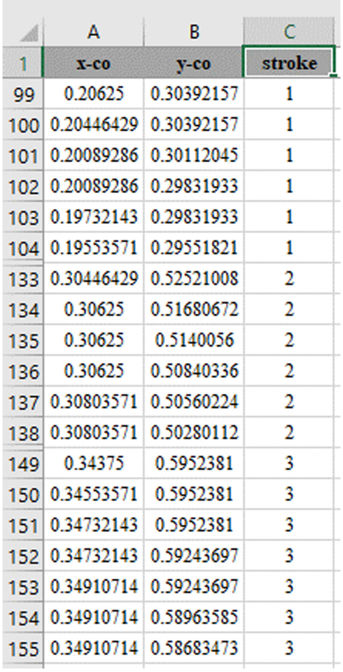


Fig. 3. GUI for the proposed data collection.



	A	B	C
1	x-co	y-co	stroke
99	0.20625	0.30392157	1
100	0.20446429	0.30392157	1
101	0.20089286	0.30112045	1
102	0.20089286	0.29831933	1
103	0.19732143	0.29831933	1
104	0.19553571	0.29551821	1
133	0.30446429	0.52521008	2
134	0.30625	0.51680672	2
135	0.30625	0.5140056	2
136	0.30625	0.50840336	2
137	0.30803571	0.50560224	2
138	0.30803571	0.50280112	2
149	0.34375	0.5952381	3
150	0.34553571	0.5952381	3
151	0.34732143	0.5952381	3
152	0.34732143	0.59243697	3
153	0.34910714	0.59243697	3
154	0.34910714	0.58963585	3
155	0.34910714	0.58683473	3

Fig. 4. Sample of data collected in csv file.

In order to collect data, we had used a help from volunteers students of Faculty of Engineering, Aswan University with ages from 18 to 20 years old.

To facilitate the procedure of collecting data to the volunteers we had prepared a collecting form with all steps needed to be done by students and we did not mention any constraints on the writing style. The indications include creating a folder for each volunteer and writing the 28 characters of Arabic script using the GUI. A total of 97 volunteers were participated All these files are reviewed to guarantee the accepted files for the database. A total of 2,520 files are accepted from the 97 volunteers, representing 90 files for each character after excluding unaccepted files. A second database is extracted from the previous accepted database by extracting strokes from characters. 17 strokes are separated from 28 characters and a database of 1,710 files representing strokes was created, strokes shapes selected with their IDs are shown in Table 2. We have demanded from Aswan University, that we had used their resources to collect the databases, to make these databases available for free.

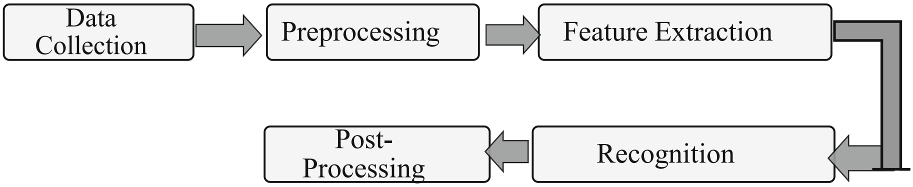


Fig. 5. Stages of online recognition system.

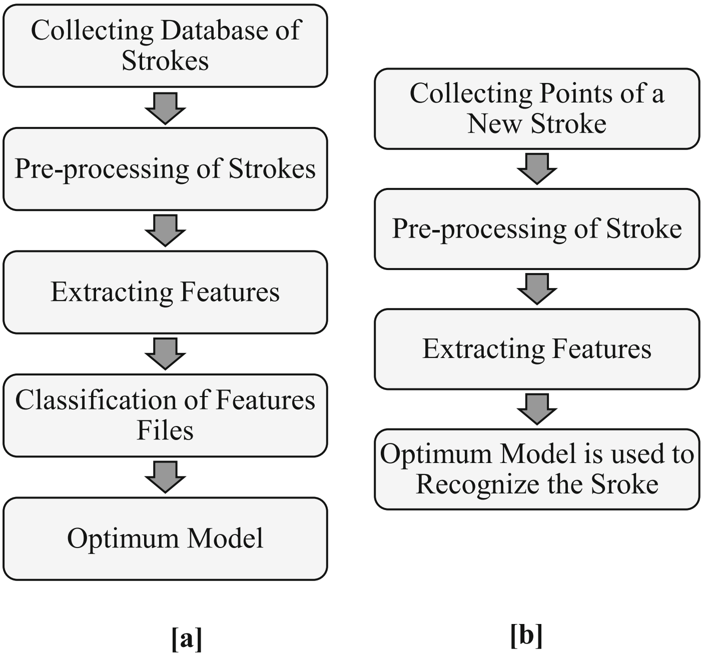

Collecting Database of Strokes
Pre-processing of Strokes
Extracting Features
Classification of Features Files
Optimum Model
[a]
Collecting Points of a New Stroke
Pre-processing of Stroke
Extracting Features
Optimum Model is used to Recognize the Sroke
[b]

Fig. 6. System used in verifying the databases **[a]** Training and validation phase, **[b]** Testing phase.

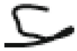
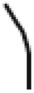
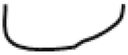
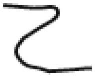
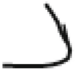
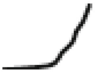
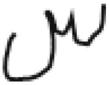
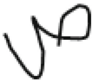
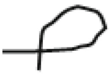
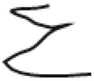
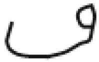
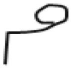
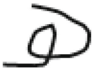
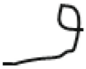
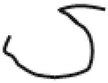
In order to verify our collected databases, we should use them in building a recognition system. Most of the online recognition systems follow typical structure of pattern recognition systems; which basically consist of five major stages, data collection, preprocessing, feature extraction, recognition, and postprocessing as illustrated in Fig. 5 [21–24]. Other researchers claim that the recognition system typically comprises of two stages, training and test stages. In the training stage, data are refined, their remarkable features are extracted, similar symbols are merged (clustering) and their features' representatives are stored as training samples, while during the test stage matching takes place for identifying similar features with test features in classification. This recognition system will be used in verifying our databases and its block diagram is shown in Fig. 6 [25].

4 Proposed Arabic Online Handwriting Recognition Algorithm

4.1 Preprocessing Stage

According to the model, the first stage in the proposed recognition system is preprocessing. The preprocessing algorithm is needed to remove variations present in the stroke captured by tablet or smart phone. These variations are mainly present in the form of size, slant, unwanted sharp edges and missing points, etc., so there is a persistent need for preprocessing stage after data collection. The five preprocessing phases in proposed algorithm are used in sequential order after the process of data collection; which

Table 2. Arabic characters strokes with IDs.

Stroke ID	Stroke Shape	Stroke ID	Stroke Shape
10		11	
12		13	
14		15	
16		17	
18		19	
20		21	
22		23	
24		25	
26			

are as following: resizing and centering, interpolating missing points, smoothing, slant correction and resampling of points [26–28].

4.1.1 Resizing and Centering

Resizing and centering phase of stroke is a necessary process that should be performed in order to recognize the stroke. This can be done by assuming a certain frame with a fixed size then moving the stroke to the assumed center point of the frame.

4.1.2 Interpolation

The interpolation phase is used since the stroke may have been written with high speed, so that missing points in the stroke will be found. These missing points can be calculated using various interpolation techniques such as Bezier and B-Spline. We have opted piecewise Bezier interpolation in our procedure because it helps to interpolate points among fixed number of points. In piecewise interpolation technique, a set of consecutive four points is considered for obtaining the Bezier curve. The next set of four points gives the next Bezier curve [29]. The pseudocode of interpolation phase is shown in Fig. 7.

4.1.3 Smoothing

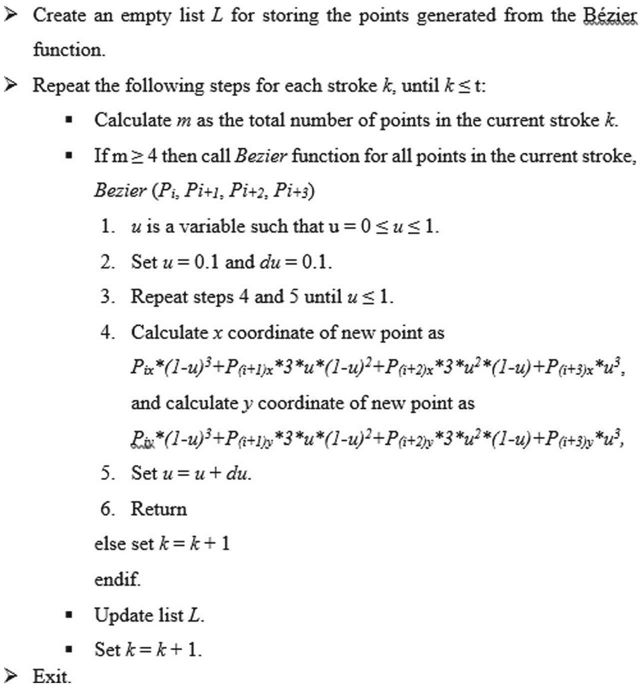
Flickers do exist in handwriting because of individual handwriting style and the hardware used. These flickers can be removed by modifying each point of the list with mean value of k-neighbors and the angle subtended at position from each end, this phase is called smoothing phase.

4.1.4 Slant Correction

Slant correction is required to correct the shape of input handwritten character as most of the writers handwriting is bend to left or right directions. Slant correction for a stroke becomes complex as no baseline can be assumed. In case of single stroke, no bottom-line marks can be made. As such the chain code estimation method by Yimei [30] has been applied for slant correction in Arabic strokes.

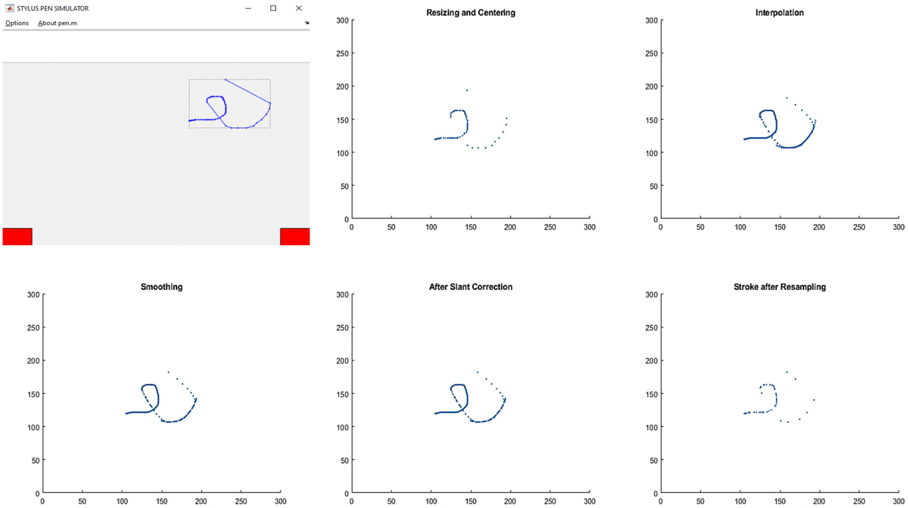
4.1.5 Resampling

Due to variations in writing speed, the acquired points are not distributed evenly along the stroke trajectory. Resampling is used to get a sequence of points which is almost equidistant. Besides the removal of variations, this step is essentially because it reduces the number of points in a stroke to a certain value. After resampling, the data is significantly reduced and the irregularly placed data points that create jitter on the trajectory of the stroke are removed. This makes the resampling step very useful in noise elimination as well as data reduction. In this phase new data points are calculated on the basis of the original points of list. After this phase, only 64 equidistant points will be present in the stroke, those 64 points is of great importance in the next step in recognition system, feature extraction. Figure 8 clarifies the five phases of preprocessing after data collection.



```
➢ Create an empty list $L$ for storing the points generated from the Bézier function.
➢ Repeat the following steps for each stroke $k$, until $k \leq t$:
    ▪ Calculate $m$ as the total number of points in the current stroke $k$.
    ▪ If $m \geq 4$ then call Bezier function for all points in the current stroke,
      Bezier ($P_i$, $Pi_{+1}$, $Pi_{+2}$, $Pi_{+3}$)
        1. $u$ is a variable such that $u = 0 \leq u \leq 1$.
        2. Set $u = 0.1$ and $du = 0.1$.
        3. Repeat steps 4 and 5 until $u \leq 1$.
        4. Calculate $x$ coordinate of new point as
           $P_{ix}*(1-u)^3+P_{(i+1)x}*3*u*(1-u)^2+P_{(i+2)x}*3*u^2*(1-u)+P_{(i+3)x}*u^3$,
           and calculate $y$ coordinate of new point as
           $P_{ix}*(1-u)^3+P_{(i+1)y}*3*u*(1-u)^2+P_{(i+2)y}*3*u^2*(1-u)+P_{(i+3)y}*u^3$,
        5. Set $u = u + du$.
        6. Return
      else set $k = k + 1$
      endif.
    ▪ Update list $L$.
    ▪ Set $k = k + 1$.
➢ Exit.
```

Fig. 7. Algorithm for interpolation.



Resizing and Centering

Interpolation

Smoothing

After Slant Correction

Stroke after Resampling

Fig. 8. Phases of preprocessing of an Arabic stroke.

4.2 Feature Extraction Stage

Feature extraction stage is one of the important stages in online handwritten character recognition, and selection of a feature extraction technique is an important task as efficiency of any online handwriting recognition system highly relies on the features which are considered as input to a classifier. There is no standard strategy for extracting features. Features that provide good results for one script may not provide good results for other scripts [31–33].

In the present study, we have presented two different techniques for feature extraction, one by just rearranging the preprocessed points without applying any transformation as shown in Fig. 9, while the second feature extraction technique is by applying Two-Dimensional Discrete Fourier Transform (2D-DFT) on the rearranged preprocessed points of the input stroke, Eq. 1 [34].

$$F[k,l] = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f\left[x,y\right].e^{-2\pi j(\frac{kx}{M}+\frac{ly}{N})} \tag{1}$$

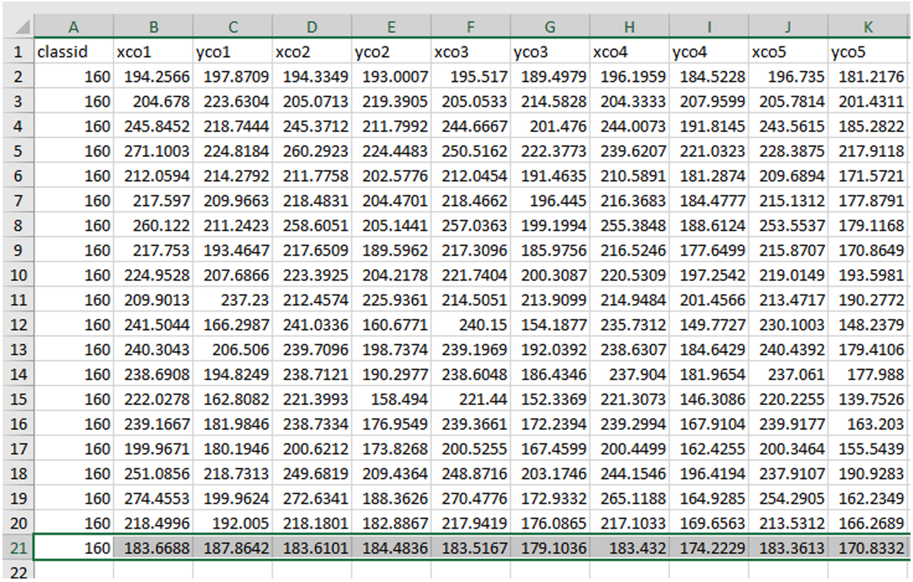
To reduce operations and computations we had used Fast Fourier Transform (FFT) instead of DFT, and after applying 2D-DFT, we got complex numbers as output. We had used experiments for both real part coefficients and imaginary part coefficients of these complex numbers as features and stored in a file, called feature file and this feature file is taken as input to the classifier.

4.3 Classification Stage

In machine learning and statistics, classification is the problem of identifying to which of a set of categories a new observation belongs, on the basis of a training set of data containing observations whose category membership is known [35, 36].

To evaluate the model after classification k-fold cross-validation is used, where the training data is divided into k parts; out of k parts, k-1 parts are used for training and remaining one part is used for testing. Each observation in the data sample is assigned to an individual group and stays in that group for the duration of the procedure. This means that each sample is given the opportunity to be used in the hold out set one time and used to train the model k-1 times [37, 38].

MATLAB Classification Learner application was used to train models to classify data, where we had used this application to perform automated training to search for the best classification model type, including decision trees, support vector machines, nearest neighbors, and ensemble classification. We had performed supervised machine learning by supplying a known set of input data which is our collected database and known responses to the data which is character stroke IDs. We had used the data to train a model that generates predictions for the response to new data [39–41].



	A	B	C	D	E	F	G	H	I	J	K
1	classid	xco1	yco1	xco2	yco2	xco3	yco3	xco4	yco4	xco5	yco5
2	160	194.2566	197.8709	194.3349	193.0007	195.517	189.4979	196.1959	184.5228	196.735	181.2176
3	160	204.678	223.6304	205.0713	219.3905	205.0533	214.5828	204.3333	207.9599	205.7814	201.4311
4	160	245.8452	218.7444	245.3712	211.7992	244.6667	201.476	244.0073	191.8145	243.5615	185.2822
5	160	271.1003	224.8184	260.2923	224.4483	250.5162	222.3773	239.6207	221.0323	228.3875	217.9118
6	160	212.0594	214.2792	211.7758	202.5776	212.0454	191.4635	210.5891	181.2874	209.6894	171.5721
7	160	217.597	209.9663	218.4831	204.4701	218.4662	196.445	216.3683	184.4777	215.1312	177.8791
8	160	260.122	211.2423	258.6051	205.1441	257.0363	199.1994	255.3848	188.6124	253.5537	179.1168
9	160	217.753	193.4647	217.6509	189.5962	217.3096	185.9756	216.5246	177.6499	215.8707	170.8649
10	160	224.9528	207.6866	223.3925	204.2178	221.7404	200.3087	220.5309	197.2542	219.0149	193.5981
11	160	209.9013	237.23	212.4574	225.9361	214.5051	213.9099	214.9484	201.4566	213.4717	190.2772
12	160	241.5044	166.2987	241.0336	160.6771	240.15	154.1877	235.7312	149.7727	230.1003	148.2379
13	160	240.3043	206.506	239.7096	198.7374	239.1969	192.0392	238.6307	184.6429	240.4392	179.4106
14	160	238.6908	194.8249	238.7121	190.2977	238.6048	186.4346	237.904	181.9654	237.061	177.988
15	160	222.0278	162.8082	221.3993	158.494	221.44	152.3369	221.3073	146.3086	220.2255	139.7526
16	160	239.1667	181.9846	238.7334	176.9549	239.3661	172.2394	239.2994	167.9104	239.9177	163.203
17	160	199.9671	180.1946	200.6212	173.8268	200.5255	167.4599	200.4499	162.4255	200.3464	155.5439
18	160	251.0856	218.7313	249.6819	209.4364	248.8716	203.1746	244.1546	196.4194	237.9107	190.9283
19	160	274.4553	199.9624	272.6341	188.3626	270.4776	172.9332	265.1188	164.9285	254.2905	162.2349
20	160	218.4996	192.005	218.1801	182.8867	217.9419	176.0865	217.1033	169.6563	213.5312	166.2689
21	160	183.6688	187.8642	183.6101	184.4836	183.5167	179.1036	183.432	174.2229	183.3613	170.8332
22											

Fig. 9. Preprocessed points rearranged in a single row.

Seven experiments were held to find the optimum accuracy, training time, and prediction speed:

- Without Applying FFT.
- Real part coefficients that was obtained after applying FFT is used as features and the feature file is taken as input to MATLAB classification learner app.
- Imaginary Part Coefficients as features.
- Real Part Coefficients normalized to 15.
- Imaginary Part Coefficients normalized to 15.
- Real Part Coefficients normalized to 100.
- Imaginary Part Coefficients normalized to 100.

5 Optimum Proposed Recognition Model

Here we had held a comparison between all experiments that were achieved in to decide which model we will use in our recognition system. The comparison was held in terms of accuracy and prediction speed because they are the parameters that are needed in our recognition system, training time is not so important because the training is done just one time and is not needed then in recognition. First, a comparison with the six experiments that had applied FFT will be held as they are common in applying the same transformation on the preprocessed points, then the best of those will be parts of the next comparison against the remaining experiment. The first comparison indicates that for all experiments tree classifiers give high prediction speeds with lower accuracies, SVM classifiers give lower accuracies with medium prediction speeds, KNN classifiers give low accuracies with medium prediction speed and Ensemble classifiers give higher accuracies with medium or low prediction speeds.

The best results were achieved almost from experiment 2 “using real part coefficients of FFT”, also it is obvious that the best classifier learner among all classifiers of experiment 2 is SVM classifiers and Ensemble classifiers, Ensemble (Subspace KNN) classifier gives the highest accuracy (75.6%) but the prediction speed is so low (360 obs/sec), however Quadratic SVM gives a near accuracy of (74.4%) but with a better prediction speed of (1900 obs/sec). The other comparison that was held between experiment 2 using real part coefficients of FFT and experiment 1 without applying FFT is shown in Table 3. This comparison indicates that experiment 2 has better prediction speeds for almost all the classifiers, however experiment 1 gives more better accuracy. The highest accuracy from experiment 1 is for the Quadratic SVM classifier (86.4%) with a prediction speed of (1600 obs/sec). Notice that if we use another PC device in our experiments, the accuracy of models will still the same but the prediction speed and training time will differ according to the PC specifications. For example, when we used a PC with AMD A8–3870 CPU 3.00 GHz and 12.0 GB installed memory (RAM), the Quadratic SVM classifier gave an accuracy of 86.6% with a prediction speed 530 obs/sec. According to the previous comparisons it is clear that the optimum recognition model is the model from experiment 1 Quadratic SVM classifier with the accuracy (86.4%) in our recognition model.

Table 3. Comparison between FFT based feature extraction and without applying FFT.

Classification learner	Experiment 1		Experiment 2	
	Accuracy	Prediction speed obs/sec	Accuracy	Prediction speed obs/sec
Tree “Fine Tree”	72.1%	~13000	60.0%	~35000
Tree “Medium Tree”	52.4%	~12000	40.2%	~40000
Tree “Coarse Tree”	25.6%	~13000	19.4%	~34000
SVM “Linear SVM”	84.5%	~2400	71.6%	~3500
SVM “Quadratic SVM”	**86.4%**	**~1600**	74.4%	~1900
SVM “Cubic SVM”	85.8%	~1600	73.8%	~1800
SVM “Fine Gaussian SVM”	63.5%	~740	50.8%	~850

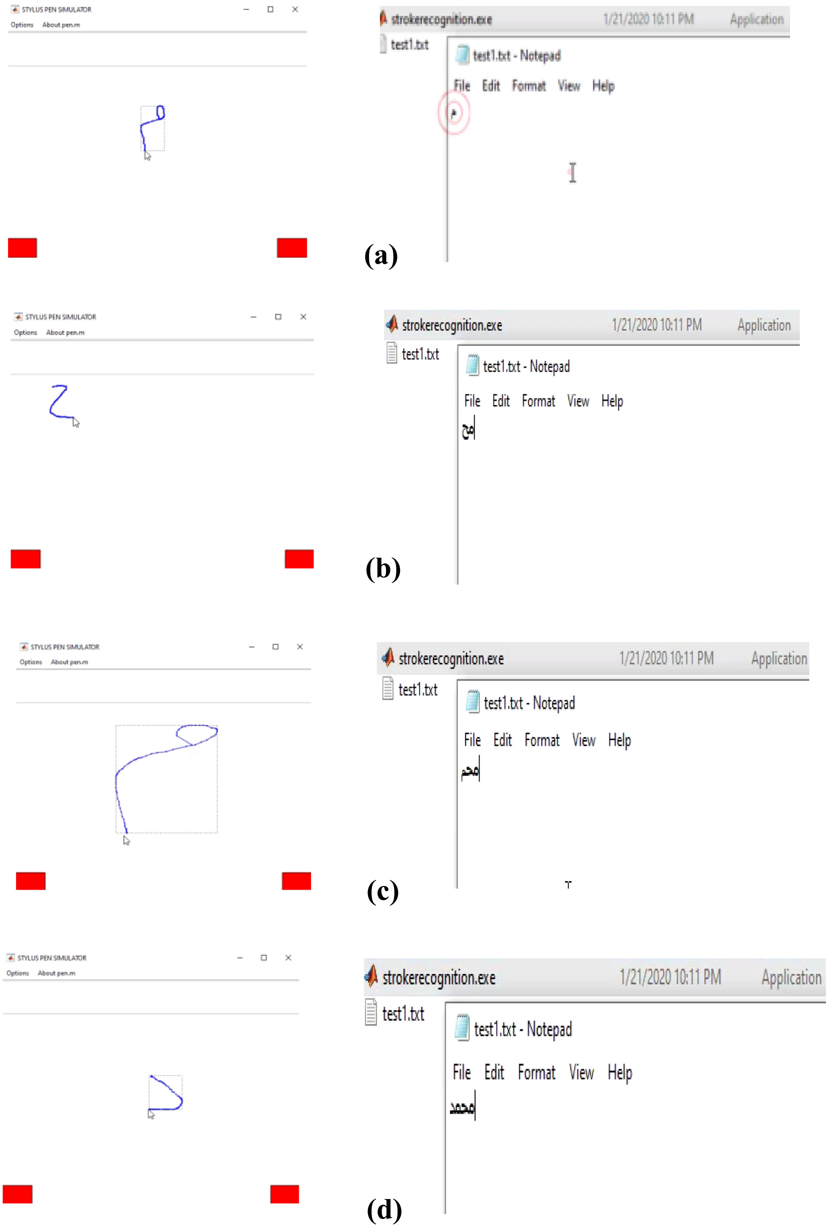
(*continued*)

Table 3. (*continued*)

Classification learner	Experiment 1		Experiment 2	
	Accuracy	Prediction speed obs/sec	Accuracy	Prediction speed obs/sec
SVM "Medium Gaussian SVM"	84.7%	~1300	73.8%	~1300
SVM "Coarse Gaussian SVM"	73.4%	~1200	46.3%	~1100
KNN "Fine KNN"	81.8%	~3700	60.1%	~4500
KNN "Medium KNN"	79.5%	~4000	52.6%	~4600
KNN "Coarse KNN"	61.6%	~3300	32.9%	~4000
KNN "Cosine KNN"	80.2%	~3600	62.5%	~4100
KNN "Cubic KNN"	79.5%	~160	59.4%	~160
KNN "Weighted KNN"	81.7%	~4000	55.1%	~4300
Ensemble "Boosted Trees"	69.9%	~6700	62.6%	~7600
Ensemble "Bagged Trees"	82.2%	~4700	71.5%	~5300
Ensemble "Subspace Discriminant"	80.4%	~2000	69.5%	~2000
Ensemble "Subspace KNN"	82.2%	~350	75.6%	~360
Ensemble "RUSBoosted Trees"	52.4%	~7500	36.4%	~8100

5.1 Testing of the Optimum Model

After creating classification models interactively in Classification Learner, we can export our optimum model to the workspace or make a standalone application. We can then use the produced application to make predictions using new data. The application will follow the stages that were used in the training phase by collecting data using the GUI, preprocessing points collected and outputs a total of 64 points, extracting features by just rearrange the preprocessed points to a single row with 128 features representing the entered stroke, then with the help of the trained model structure the app will predict ID of the entered stroke. Figure 10 shows the recognition of an Arabic handwritten cursive word (محمد) pronounced "Mohamad" after writing it with online handwriting letter by letter and predicts its characters by the proposed recognition model. Word Mohamad in



(a) م

(b) مح

(c) محم

(d) محمد

Fig. 10. The recognition of Arabic handwriting cursive word (محمد) "Mohamad" by the proposed model: From (a) to (d) at left-side GUI used to enter a stroke, at right-side predicted character by the proposed recognition model and written in text file

Arabic consists of four letters. In the stylus pen simulator, an Arabic stroke is written by hand (م) as shown in left-side in Fig. 10(a); the output of the code using Quadratic SVM model was ID22 with a prediction time of 0.896426 s. We export the output ID to a text file by first convert it to an Arabic character with identical shape. Therefore, the stroke ID22 will be converted to “meem” character (م) as shown in right-side in Fig. 10(a). After that, by repeated the same sequence for the three remaining letters of word Mohamed shown from Fig. 10(b) to Fig. 10(d). These indicate the prediction of the three Arabic strokes “hha” with ID13, “meem” with ID22, and “dal” with ID14 respectively.

6 Conclusion

This paper presented novel Arabic handwritten characters and strokes databases. These databases are focused only on Arabic handwritten characters with Naskh style. A lot of work is needed from researchers to supply Arabic society with this kind of strokes databases; Ruqaa,

Thuluth, Diwani are some styles of Arabic language that are needed to be part of the future databases. Furthermore, collecting databases for shapes of Arabic characters depending on their locations in the word is quite needed. Moreover, databases of diacritics will be also of great importance for more advanced character recognition. More volunteers from different ages are needed to make a powerful database.

Our study was based on mentioned machine learning technique using supervised learning as the database collected was with known stroke IDs and that is the cause of using classification, each stroke was given an ID and database was collected according to these IDs. The workflow for recognition was by collecting data, preprocess the data, derive features using preprocessed data, train models using features derived, iterate to find the best model, and then integrate the optimum-trained model into the recognition system.

Acknowledgments. The authors thank all participants’ contribution to (AOLAH) databases formulation. They sincerely appreciated Dr. Omar Abdel-Reheem and Eng. Fatma Gamal, from Aswan faculty of engineering, Aswan University, for their help to facilitate the data collecting process.

References

1. Habash, N.Y.: Introduction to Arabic Natural Language Processing. Morgan & Claypool, San Rafael (2010)
2. AlMuallim, H., Yamaguchi, S.: A method of recognition of Arabic cursive handwriting. IEEE Trans. Pattern Anal. Mach. Intell. **PAMI-9**(5), 715–722 (1987)
3. Elbaati, A., Kherallah, M., Ennaji, A., Alimi, A.M.: Temporal order recovery of the scanned handwriting. In: 2009 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, 26–29 July 2009
4. Alimi, A.: A neuro-fuzzy approach to recognize Arabic handwritten characters. In: Proceedings of International Conference on Neural Networks (ICNN 1997), Houston, TX, USA, 12–12 June 1997

5. Abuzaraida, M.A., Zeki, A.M., Zeki, A.M.: Recognition techniques for online Arabic handwriting recognition systems. In: 2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT), Kuala Lumpur, Malaysia (2012)
6. Al-Helali, B.M., Mahmoud, S.A.: Arabic online handwriting recognition (AOHR): a survey. ACM Comput. Surv. **50**(3), 1–35 (2017)
7. AbdElNafea, M., Heshmat, S.: Efficient preprocessing algorithm for online handwritten Arabic strokes. In: 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), Aswan, Egypt, 2–4 February 2019
8. Sharma, A.: Online Handwritten Gurmukhi Character Recognition “thesis”. Patiala, Punjab, India: School of Mathematics and Computer Applications, Thapar University, February 2009
9. Harouni, M., Mohamad, D., Rasouli, A.: Deductive method for recognition of on-line handwritten Persian/Arabic characters. In: 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), Singapore, Singapore, 26–28 February 2010
10. Haraty, R., Ghaddar, C.: Arabic text recognition. Int. Arab J. Inf. Technol. **1**, 156–163 (2004)
11. Kherallah, M., Elbaati, A., El Abed, H., Alimi, A.M.: The On/Off (LMCA) Dual Arabic Handwriting Database. In: REGIM: Research Group on Intelligent Machines, University of Sfax (2008)
12. Kherallah, M., Tagougui, N., Alimi, A.M., El Abed, H., Margner, V.: Online Arabic handwriting recognition competition. In: 2011 International Conference on Document Analysis and Recognition, Beijing, China (2011)
13. Elanwar, R.I.M., Rashwan, M.A., Mashali, S.A.: OHASD: the first on-line Arabic sentence database handwritten on tablet PC. Int. J. Comput. Inf. Eng. **4**(12), 1907–1912 (2010)
14. El Abed, H., Kherallah, M., Märgner, V., Alimi, A.M.: On-line Arabic handwriting recognition competition ‘ADAB database and participating systems.’ Int. J. Doc. Anal. Recogn. (IJDAR) **14**, 15–23 (2011)
15. Azeem, S.A., Ahmed, H.: Recognition of segmented online Arabic handwritten characters of the ADAB database. In: 2011 10th International Conference on Machine Learning and Applications and Workshops, Honolulu, HI, USA, 18–21 December 2011
16. Abdelaziz, I., Abdou, S.: AltecOnDB: a large-vocabulary arabic online handwriting recognition database. arXiv, 24 December 2014
17. Abuzaraida, M.A., Zeki, A.M., Zeki, A.M.: Online database of Quranic handwritten words. J. Theor. Appl. Inf. Technol. **62**(2), 485–492 (2014)
18. Mahmoud, S.A., Luqman, H., Al-Helali, B.M., BinMakhashen, G., Parvez, M.T.: Online-KHATT: an open-vocabulary database for Arabic online-text processing. Open Cybern. Syst. J. **12**(1), 42–59 (2018)
19. Mahajan, L., Kulkarni, G.A.: Digital pen for handwritten digit and gesture recognition using trajectory recognition algorithm based on triaxial accelerometer. IOSR J. Electron. Commun. Eng. (IOSR-JECE) **10**(1), 24–31 (2015)
20. Nakkach, H., Hichri, S., Haboubi, S., Amiri, H.: A segmentation-free approach to strokes extraction from online isolated Arabic handwritten character. In: 2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Monastir, Tunisia, 21–23 March 2016
21. Plamondon, R., Srihari , S.N.: On-line and off-line handwriting recognition: a comprehensive survey. IEEE Trans. Pattern Anal. Mach. Intell. **22**(1), 63–84 (2000)
22. Sharma, A., Kumar, R., Sharma, R.K.: Online handwritten Gurmukhi Character recognition using elastic matching. In: 2008 Congress on Image and Signal Processing, Sanya, Hainan, China, 27–30 May 2008
23. Priya, A., Mishra, S., Raj, S., Mandal, S., Datta, S.: Online and offline character recognition: a survey. In: 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, India, 6–8 April 2016

24. Mezghani, N., Mitiche, A., Cheriet, M.: On-line recognition of handwritten Arabic characters using a Kohonen neural network. In: Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition, Niagara on the Lake, Ontario, Canada, 6–8 August 2002
25. Santosh, K.C., Nattee, C.: A comprehensive survey on on-line handwriting recognition technology and its real application to the Nepalese natural handwriting. Kathmandu University J. Sci. Eng. Technol. **5**(1), 31–55 (2009)
26. Abuzaraida, M.A., Zeki, A.M., Zeki, A.M.: Problems of writing on digital surfaces in online handwriting recognition systems. In: 2013 5th International Conference on Information and Communication Technology for the Muslim World (ICT4M), Rabat, Morocco, 26–27 March 2013
27. El-Wakil, M.S., Shoukry, A.A.: On-line recognition of handwritten isolated Arabic characters. Pattern Recogn. **22**(2), 97–105 (1989)
28. Al-Emami, S., Usher, M.: On-line recognition of handwritten Arabic characters. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7), 704–710 (1990)
29. Mortenson, M.E.: Mathematics for Computer Graphics Applications . Industrial Press Inc., South Norwalk (1999)
30. Ding, Y., Kimura, F., Miyake, Y., Shridhar, M.: Accuracy improvement of slant estimation for handwritten words. In: Proceedings 15th International Conference on Pattern Recognition. ICPR-2000, Barcelona, Spain, 3–7 September 2000
31. Ramzi, A., Zahary, A.: Online Arabic handwritten character recognition using online-offline feature extraction and back-propagation neural network. In: 2014 1st International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Sousse, Tunisia, 17–19 March 2014
32. Al-Habian, G., Assaleh, K.: Online Arabic handwriting recognition using continuous gaussian mixture HMMS. In: 2007 International Conference on Intelligent and Advanced Systems, Kuala Lumpur, Malaysia, 25–28 November 2007
33. Boubaker, H., El Baati, A., Kherallah, M., Alimi, A.M., Elabed, H.: Online Arabic handwriting modeling system based on the graphemes segmentation. In: 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010
34. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2nd edn. Prentice Hall, Upper Saddle River (2002)
35. White, R.L.: Methods for Classification, 16 August 1996. http://first.astro.columbia.edu/rick/SCMA/node2.html
36. W. Contributors, Training, Validation, and Test Sets, Wikipedia, The Free Encyclopedia, 14 December 2019. https://en.wikipedia.org/w/index.php?title=Training,_validation,_and_test_sets&oldid=930696087
37. Brownlee, J.: A Gentle Introduction to k-fold Cross-Validation, 23 May 2018. https://machinelearningmastery.com/k-fold-cross-validation/
38. Shalev-Shwartz, S., Ben-David, S.: Understanding Machine Learning: From Theory to Algorithms . Cambridge University Press, New York (2014)
39. MATHWORKS. Classification Learner App (2019). https://www.mathworks.com/help/stats/classificationlearner-app.html?s_tid=srchtitle
40. MATHWORKS. Choose Classifier Options (2019). https://www.mathworks.com/help/stats/choose-a-classifier.html
41. MathWorks. Machine Learning in MATLAB (2019). https://www.mathworks.com/help/stats/machine-learning-in-matlab.html