Journal of Computer Science

Original Research Paper

DAD: A Detailed Arabic Dataset for Online Text Recognition and Writer Identification, a New Type

Said S. Saloum

College of Computer and Information Sciences, Jouf University, Sakaka, Saudi Arabia

Article history
Received: 02-11-2020
Revised: 18-01-2021
Accepted: 18-01-2021

Email: sssaloum@ju.edu.sa
said.saloum@gmail.com

Abstract: This paper presents a novel Arabic dataset that considers the characteristics of the Arabic language filling some gaps not covered by existing datasets. Conventional datasets consider Arabic in a similar way to Latin languages. These datasets either delete diacritic and supplement marks, considering them as defects, or keep them without considering the actual meaning. More than half of all Arabic characters have diacritics above or below characters. In this context, this work presents the novel Detailed Arabic Dataset (DAD) for bridging these gaps. The additional marks included in this dataset are the single dot, two dots "-", three dots "^", Hamza and two supplement marks: The bar for Tah, or Zah and the complement bar for Kaf. A special application was built to generate a dataset for Arabic online recognition and writer identification (called OFMArabicDatasetBuilder). Totally the ground truth contains 93064 entries based on sub-word and letter parts (not on words or lines as other datasets). This dataset will provide researchers with a strong tool for online Arabic language text recognition especially in the segmentation phase and writer identification. This paper also presents benchmarking results of using k-nearest neighbours machine learning with DAD.

Keywords: Arabic Dataset, Arabic Benchmark, Arabic Recognition, Arabic Writer Identification, Diacritics Marks, Hamza, Supplement Marks, Tah, Zah

Introduction

In the literature, many papers that focus on Arabic text recognition and recognition. Most of these papers present offline recognition systems. The reason for this might be because databases for offline systems are easy to create and some benchmark databases for offline systems have become available over the last two decades (Al-Hashim and Mahmoud, 2010; Mezghani *et al*., 2012; Alamri *et al*., 2008; Mahmoud *et al*., 2012; Kharma *et al*., 1999).

Lately, several papers regarding online Arabic recognition have been published. However, many of them use their own databases (El Abed *et al*., 2009). To the best of my knowledge, only nine online text datasets and three online digits datasets have been published, which are the ones from the Online Arabic Handwriting Recognition in 2009 (El Abed *et al*., 2009), the On/Off (LMCA) Dual Arabic Handwriting Database (Kherallah *et al*., 2008), the online database of Quranic handwritten words (Abuzaraida *et al*., 2014), the one from the online Arabic handwriting recognition competition in 2011, the MAYASTROUN Multilanguage handwriting database (Njah *et al*., 2012), the OHASD online Arabic sentence handwritten on tablet PC database (Elanwar *et al*., 2010), the AltecOnDB large-vocabulary Arabic online handwriting recognition database (Abdelaziz and Abdou, 2014) and the one from the online Arabic handwriting digits recognition (Azeem *et al*., 2012). However, they do not properly handle all the features of Arabic Language. They simply provide databases similar to English language databases. They either delete the supplement marks or do not add pertinent information regarding these marks to the ground truth. Figure 1 illustrates the importance of these supplementary marks. In this figure, eight Arabic words are shown, all of which consist of the same sub-word (having three letters). However, the diacritics of each word completely change the meaning of the word. From this, one can understand the significance of including diacritics.

In this study, additional marks refer to both diacritics and supplement marks.

Fig. 1: Eight Arabic words with the same sub-word and different diacritics. The meaning of each word is different (two dots appear as "-"; three dots appear as "^")

When using the previous datasets, it is difficult to discern the cause of the recognition error. The error occurs in the body of the word (sub-word) or in one or more of the additional marks around it.

More than half of all Arabic characters have additional marks above or below the letters. However, no Arabic dataset contains information regarding these marks in the ground truth files. Alternatively, they are simply deleted as part of the preprocessing. Some of the online datasets provide the coordinates of pixels without referring to them in the ground truth. For example, any word in Fig. 1 has 1 entry in the ground truth, but in this dataset it will have 4 entries. If the writer had written two dots instead of "-", three dots instead of "^', or a mix of them, the ground truth will contain more entries. This information (style of writing) is very important in writer identification and very helpful in the segmentation phase.

This study is aimed at addressing this issue by presenting a dataset prepared using a tool designed specifically for the Arabic Language (OFM-ArabicDatasetBuilder). The ground truth files of this dataset contain information regarding sub-words, dots, Hamza, bar for Tah, Zah and complement for the letter Kaf. Some diacritics which are rarely used in handwritten texts are not considered in this version.

There are no criteria for a "good" dataset with regard to offline recognition; however, I do not believe this holds true for an online dataset. An online dataset, to be acceptable, must provide the researcher with the ability to rewrite any word in the dataset in the exact same manner that the original was written in (El Abed *et al*., 2009). To accomplish this, the database must contain all necessary information, including coordinates of all pixels, the time when the digital-pen/finger passed the pixel, the colour, the azimuth of the pen, the altitude of the pen, pressure and so on. This version of the OFM-ArabicDatasetBuilder is designed to manipulate the most important information required to rewrite a word as the original author had written the word, the coordinate and the time of every pixel. Other data (mainly pressure, azimuth and altitude of the pen) will be considered in the subsequent version.

The words in DAD were very carefully selected, such that they contain all Arabic letter shapes (initial, middle, last and isolated).

The paper is organized as follows. The next section introduces the most relevant related works. Section 3 indicates the main features of the Arabic language as the basis of this work. Section 4 presents the novel DAD. Section 5 describes the experimentation with this dataset and section 6 shows the results discussing the most relevant aspects. Finally, section 7 mentions the conclusions and future work.

Literature Review

Since the beginning of scientific research regarding optical and writer recognition, many researchers used datasets that they have created on their own. These datasets mostly included templates of tens of writers. Only a few datasets had templates exceeding 50 writers. Lately, benchmark databases have appeared. These databases include templates of hundreds of writers. Several even have 1000 or more writers. For example, “KHATT” database (Mahmoud *et al*., 2012) incorporated 1788 pages with a total of 165890 words. However, the oldest and most widespread database is IFN\ENIT, which includes texts written by 411 people with 26,459 words (Pechwitz *et al*., 2002; El Abed and Margner, 2007). Previous databases were for offline optical recognition, with more information about offline datasets as one can observe in (Parvez and Mahmoud, 2013).

Online Arabic handwriting databases are summarized in Table 1. The most well-known of these are QHW (Abuzaraida *et al*., 2014), LMCA (Kherallah *et al*., 2008) and ADAB (El Abed *et al*., 2011; Kherallah *et al*., 2011).

In the first work (QHW (Abuzaraida *et al*., 2014)), a special tool was used to record the coordinates of the dots over which the digital pen travels. Here, a platform was designed to collect handwritten information. A total of 120 words were written by 200 writers. Overall, 12000 samples with over 42000 characters and 23300 sub-words were included. However, no information regarding the time was considered.

Table 1: Online datasets

Dataset	Year	Words	Writers
LMCA (Kherallah *et al*., 2008)	2008	500	55
OHASD (Elanwar *et al*., 2010)	2010	3,825	48
ADAB (El Abed *et al*., 2011; Kherallah *et al*., 2011)	2011	33,164	166
MAYASTROUN (Njah *et al*., 2012)	2012	1,500	355
ALTECOnDb (Abdelaziz and Abdou, 2014)	2014	152,680	1000 (Not Free)
QHW (Abuzaraida *et al*., 2014)	2014	12,000	200
Online-KHATT (Mahmoud *et al*., 2018)	2018	80,000	623
AHWDB1,	2019	2,000	200
AHWDB2 (Al-Shamaileh *et al*., 2019)	2019	2,000	200

Table 2: Main Arabic characters found on the keyboard and their names in English letters

No.	Arabic letter	English name	No.	Arabic letter	English name
1	ا	A	20	ع	AIN
2	أ	AHU	21	غ	GAN
3	إ	AHD	22	ف	F
4	ب	B	23	ق	Q
5	ت	T	24	ك	K
6	ث	TH	25	ل	L
7	ج	G	26	م	M
8	ح	HH	27	ن	N
9	خ	KH	28	ه	H
10	د	D	29	و	W
11	ذ	THE	30	ي	YA
12	ر	R	31	ء	HAM
13	ز	Z	32	ئ	AMH
14	س	S	33	ؤ	WH
15	ش	SH	34	ة	TAM
16	ص	SAD	35	لا	LA
17	ض	DAD	36	لأ	LAHU
18	ط	TAA	37	لإ	LAHD
19	ظ	KTA	38	ى	AMK

The subsequent study (LMCA (Kherallah *et al*., 2008)) developed a special tool named "Handwriter", considering time. This study applied a sampling rate of 100 points/second. However, it disregarded the diacritics. Considering that 24 of 38 Arabic characters (listed in Table 2) contain marks above or below a letter, this database does not allow researchers to test all algorithms to recognize Arabic words. Hence, it is not particularly useful for writer identification. This database contains 30000 digits, 100000 Arabic letters and 500 Arabic words were written by 55 writers.

The third study (ADAB (El Abed *et al*., 2011; Kherallah *et al*., 2011)) is the most common among researchers (Elleuch *et al*., 2015; Potrus *et al*., 2014; Eraqi and Azeem, 2011; Chernodub and Nowicki, 2016; Hamdi *et al*., 2016; Ahmed and Azeem, 2011; Maalej *et al*., 2016; Abdelazeem and Eraqi, 2011). It consists of 19,575 Arabic words written by 166 different writers. This database is the only benchmark that has been widely recognized among researchers so far. However, it did not consider any information regarding diacritics or additional marks. The recent online version of KHATT dataset contains of 10,040 lines of Arabic text written by 623 writers. Part of the collected data is segmented into characters (separated dataset). It includes information about time and pressure. But it lacks detailed information about subwords, diacritics or additional marks (Mahmoud *et al*., 2018).

The last two datasets AHWDB1 and AHWDB2 appeared in 2019 and both of these just received one input, "Mohammad" and "Mohammad Abdallah" respectively. Each input written 10 times by 200 writers. The goal of these two datasets is to identify writers by their Arabic handwriting from one or two words only. The second group of DAD dataset will cover the shortage of this type of datasets. Detailed information about online text datasets can be found in (Al-Helali and Mahmoud, 2017; Al-Salman and Alyahya, 2017; Tagougui *et al*., 2013).

All datasets of the aforementioned studies build ground truth tables without considering Arabic language characteristics. Arabic words usually have multiple parts, referred to as sub-words in this study (see next section) and additional marks. In many cases, Arabic words consist of more than one sub-words. Each Arabic word will be divided into two or more sub-words if one of the non-connectable letters appears in the word.

Table 3: Arabic(Hindi) digits datasets

Dataset	Year	Digits	Writers
LMCA (Kherallah *et al*., 2008)	2008	30,000	55
AOD (Azeem *et al*., 2012)	2012	30,000	100
MAYASTROUN (Njah *et al*., 2012)	2012	6,500	355

There are datasets for online handwritten digits too. Many researchers had studied the recognition of Arabic (or Hindi) digits: Offline (de Sousa, 2018; Jaha, 2019; Abdleazeem and El-Sherif, 2008; El-Sawy *et al*., 2016; Abdelazeem, 2009; Almodfer *et al*., 2017; AlKhateeb and Alseid, 2014; Mahmoud, 2008) and online (Ahmad and Maen, 2008; Azeem *et al*., 2012). Researchers had used their own datasets or some benchmark datasets.

The current work resolves the Arabic characteristics problems appearing in previous studies, by designing a tool specialized for the Arabic alphabet.

Summarized Online Digit Datasets

The summarized online digit datasets contain isolated digits. In this new dataset DAD, the writers had asked to write the ten digits in one screen. In real life, native speakers usually write digits sequentially, as ID number or bank account. Table 3 indicates the features of the most relevant summarized datasets. These allow researchers to study the delay between every two digits. Notice the relevance of the connection from the last pixel in the first digit to the first pixel in the next digit in forensic sciences.

Characteristics of the Arabic Language

For simplicity and the benefit of speakers who are unfamiliar with Arabic, only the main characteristics that are required to build an Arabic dataset will be discussed. The Arabic alphabet is comprised of 28 letters. Some of these are similar in shape to that of the main body and are differentiated with dots placed above or below them (Al-Hashim and Mahmoud, 2010). The number of the dots is either one, two, or three. There is a symbol that is called “Hamza.” that appears above or below some letters (such as Alif, Waw, Yaa and Kaf). Sometimes this symbol is considered to be a different letter if it is written separately. The letters Tah, Zah and Kaf are sometimes written in two parts like the letter "t" in English. For this reason, the last two options (Bar to Tah and Zah and complement to Kaf) have been included as shown in Fig. 2. It is important to note, especially for non-Arabic readers, that the Arabic alphabet contains 28 letters, but because of the additional marks and connectivity, the Arabic keyboard contains additional characters (Table 1). Words written in Arabic differ from words written using the Latin alphabet, in the fact that the typed and hand-written words have almost the same shape in the latter. In other words, the printed Latin text is written using separate letters, whereas the letters of words written by hand are usually connected.

Fig. 2: The program divides the word into its sub-words and list them in the dropdown list

Additionally, there are no fixed rules for connecting letters to each other. Hence, the connections between letters can be considered as the writer's preference. For example, when someone writes the word "university," he/she can connect all the letters together, or he/she can write each letter separately. Arabic, however, has a rule that cannot be broken either for printed or hand-written texts. Understanding this rule explains the following:

- Why and how the program in Fig. 2 divides the word to sub-words
- The manner in which the truth file must reflect the structure of the word

The rule can be summarized as follows: Every letter must be connected to the letter after it, unless it is one of the letters indicated by the following numbers in Table 2: 1, 2, 3, 10, 11, 12, 13, 29, 31, 33, 34, 35, 36, 37 and 38. If one of these characters appears in a word, the word would be written as two sub-words. If two characters appear, the word would be written as three sub-words and so on a so forth. Figure 3 demonstrates how a five-letter word is written. Two letters are non-connectable with the letter after them; hence, the word appears to be written using three sub-words.

Fig. 3: Writing a five-letter word with two letters being nonconnectable letters. a-Isolated letters (not allowed in both printed and handwritten cases), b-printed form, c-handwritten form

Figure 3c shows the same word written by hand, which is somehow similar to the printed word. However, there are some differences between the printed and the handwritten word. For instance, two dots in the handwritten word are usually written as a small vertical bar and three dots are usually written with the symbol "^" (Fig. 1), also overlapping between sub-word, slant, skew, etc. Fig. 4 shows the importance of information about diacritics. It is a word written by three writers and the main difference is the diacritic of the first letter, so detailed information would help the researchers to develop new writer identification algorithms.

Detailed Arabic Dataset

The creation of DAD was performed with an iterative process with the three steps of data collection (see section 4.1), data extraction (introduced in section 4.2) and the incorporation of data in ground truth file (described in section 4.3).

Data Collection

The program instructs the volunteer to write 132 entries, in which10 entries are numbers 0-9 (the numbers must be written 10 times on 10 screens). Additionally, there are six words that must be entered 10 times and 62 words that must be entered just once. The information regarding these 132 entries is saved in a text file. In this study, 159 people voluntarily participated and they were recruited among instructors and students of the faculty.

Table 4 summarize important statistics about DAD, totally ground truth for DAD contains 93064 records in 18767 files, the number of records about five times more than the number of files, usually in other Arabic datasets the number of records is equal to the number of files, this is simply explained that the ground truth in DAD based on sub-words and letters parts not on words or lines. These details are important for segmentation and writer identification (Fig. 4).

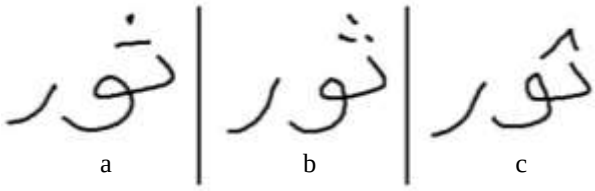

Fig. 4: Arabic word written by three writers (means ox), the main difference is the diacritics of the first letter. a-data set will contain two records: #1 and #2, b- dataset will contain three records of #1, c- dataset will contain one record of #3. In addition to two entries for the two

The file contains the following information about each writer: Writer name, strokes count, letters count, the word in Arabic letters, the word in English letters, the word in Unicode, the coordinates of all pixels detected by the hardware and the time in milliseconds for every pixel. This file contains a crude dataset called a library.

In fact, libraries can be used as datasets; however, some volunteers make grammatical or syntactic errors, or enter the words with significant slant and skew, or in some cases, it is simply difficult for both human and machine to read the word. Most importantly, it does not contain detailed information regarding additional symbols (diacritics and supplement marks). Hence, the library (crude dataset) must be treated, as explained in the next section. The dataset was collected using the “Wacom pro tablet”, which is a device that captures the writing when a user writes on it with a special digital pen.

Figure 5 shows an Arabic word written by two writers, Table 5 indicates the main information fined in truth files for each word. The second writer had written first and third sub-words using two strokes. Another valuable information that the first writer input the two additional marks immediately after the sub-word belongs to it, whereas the second writer input the three additional marks after writing all sub-words, this is valuable information for writer identification.

Data Extraction

The libraries generated in the previous section must be preprocessed before it can be used. This is done using the same application using the “edit Library mode”, which allows user to edit some Arabic samples. In this mode, the application simply redraws the first word in the library (crude dataset) and colours the first stroke in red. The user (author) can click "cancel this word" (Fig. 2) if the word is not legible or has any of the aforementioned issues. If the word is suitable for the dataset, the user must choose the type of stroke marked in red colour. If it is a sub-word, its name must be chosen from a drop-down list. If it is a diacritic mark, its type must be chosen then subword related to it from the drop-down list.

Table 4: Statistics about DAD

Entry	Group 1 62×1	Group 2 6×10	Total
Words	8959	8282	17241
Sub-words strokes	21795	24766	46561
Dot	8501	7109	15610
2 dots	5659	3921	9580
3 dots	930	0	930
Hamza	2777	0	2777
Bar for Kaf	136	0	136
Bar for Tah	1178	1231	2409
Numbers	14946 using strokes	15061	15061

Table 5: Comparison between to entries of the same word

No.	Truth table Rec.	No.	Truth table Rec.
a-1	#SArabic, طا #SEnglish,TAA,A	b-3	#SArabic, و #SEnglish,W
a-2	#T #MWArabic, طا #MWEnglish,TAA,A	b-4	#SArabic, لة #SEnglish,L,TAM
a-3	#SArabic, و #SEnglish,W	b-5	#SArabic, لة #SEnglish,L,TAM
a-4	#SArabic, لة #SEnglish,L,TAM	b-6	#2 #MWArabic, لة #MWEnglish,L,TAM
a-5	#2 #MWArabic, لة #MWEnglish,L,TAM	b-7	#2 #MWArabic, لة #MWEnglish,L,TAM
b-1	#SArabic, طا #SEnglish,TAA,A	b-8	#T #MWArabic, طا #MWEnglish,TAA,A
b-2	#SArabic, طا #SEnglish,TAA,A		

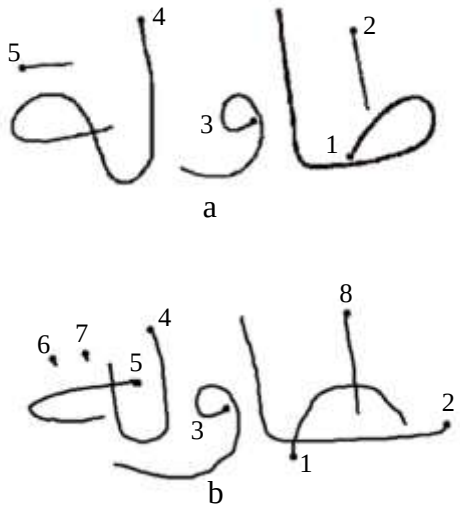

Fig. 5: Two words written by two writers, the numbers denote the order in which the strokes are written

For example the word in Fig. 6 contains three subwords, but it is written in five phases. The reason behind is that the first letter has a bar and the last letter has two dots above it (small horizontal line in the hand-written word). In order to build a ground truth file, the program instructs the user to enter the name of each stroke, as shown in Fig. 2, in the same order that the word was written using the digital pen by the writer. In this example, the names of the five strokes were asked to be entered as listed in

Table 6 to help the user, the program highlights the relevant stroke in red. The first, third and fourth strokes are sub-words without any additional marks. Their letters are listed in the dropdown list after the word is analysed. The second stroke is a bar for the “Taa” letter of the first sub-word "Bar to Tah or Zah." The final phase involves a line that replaces the two dots of the letter “Taa-Al-Marbouta.”

If the stroke is incorrect, the user can press the "Ignore this stroke" and then the program will delete it allowing the user to re-enter names. Another choice is that the user can name it "!" from the drop-down list, which implies that the end-user of the dataset must ignore this stroke.

In order to build the ground truth table for this word, some records are required to define the strokes of the word. In the next section, the information regarding this word and the required record is discussed.

The data are organized in a manner that makes processing easy and may be performed using any programming language, hence, every word has been stored in a separate text file, its format is original Comma-Separated Value (CSV) matlab file. The name of the file donates the writer ID, the Arabic word in Latin letters and a number. This number is generated by Windows operating system when the application attempt to store a file with a name that already exists. This occurs when the same writer inputs the same word more than once. In other words, the name of the file has the following Lowing structure: Writer ID, 1st letter, 2nd letter, …, nth letter(Number).txt.

Internally, the information in the file is stored in the records. The record has two items: Attributes and its data. The first six attributes contain information regarding the word (obtained from the library), Table 7. To make the information more readable, the attributes names start with the "#" character. Next, the attributes will be explained. #WriterName attribute: The data in this attribute is the writer ID, every volunteer must input his/her ID before he/she inputs words into the application. #StrokesCount attribute: The data in this attribute is a number that indicates the number of strokes used by the writer to write (draw) the word. The points of all strokes are detected using hardware starting with at the moment that the pen touches the screen and stopping at the moment the pen is raised from the screen. #LettersCount attribute: The data in this attribute is a number that indicates the number of letters in the Arabic word. This number is counted by the software with the knowledge that the word shown is the word that the volunteer must write in buttons bar.

Fig. 6: Arabic word entered by a volunteer

Table 6: Inserting necessary information about the word in Fig. 7 to ground truth

#Stroke	Sub-word	Dropdown list	Dot	Two dots	Three dots	Hamza	Bar to thaa	Complement to KAF
1	⦿	Choose 1st subword from the dropdown list						
2	⦿	Choose 1st subword from the dropdown list					⦿	
3	⦿	Choose 2nd subword from the dropdown list						
4	⦿	Choose 3rd subword from the dropdown list						
5	⦿	Choose 3rd subword from the dropdown list		⦿				

Table 7: Records used with letters' words

Attribute	Data
#WriterName	Writer identification.
#StrokesCount	Number of strokes.
#LettersCount	Number of letters in the Arabic word.
#WArabic	Word written in Arabic letters
#WEnglish	Letter of the Arabic word written in English letters, separated by commas.
#WUnicode	Arabic syllables in Unicode.

Table 8: Records used with letters in each word

Record	Data
#SArabic	Syllable in Arabic letters
#SEnglish	Letters of Arabic syllables written in Latin letters, separated by commas.
#SUnicode	Arabic syllable in Unicode.
#Dots	Number of points detected by hardware between when the pen touches the screen and when it is raised again. Subsequent line(s) list the coordinates of all points in the format x1, y1, x2, y2, x3, y3 and so on
#Tms	Number of time values. The subsequent line(s) list all times in millisecond in the format t1, t2, t3 and so on. t1 is the time between the first and second points, t2 is the time between the second and third points.

Table 9: Records used with supplements of letters

Records	Data
#1	Single dot above or below the Arabic letter.
#2	Two dots above or below the Arabic letter.
#3	Three dots above the Arabic letter (or under in some languages written in Arabic letters).
#H	Hamza above or below the Arabic letter.
#K	Bar for letter “Kaf” in the beginning or middle.
#T	Bar for “Tah” or “Zah”

#WArabic attribute: The data in this attribute is the Arabic word written in Arabic letters. This record may not correctly appear if the operating system does not support the Arabic language. For this reason, the two next attributes were added. #WEnglish attribute: The data in this attribute is the Arabic Word in English letters, depending on Table 2. #WUnicode attribute: The data of this attribute is the Arabic letters in unicode. The #StrokesCount attribute provides the number of strokes used to write (draw) the word. Hence, the succeeding records in the file provide information regarding every stroke. Five or six records are used per stroke (Table 8). If the stroke is a sub-word(s), then five records are used: #SArabic, #SEnglish, #SUnicode, #Dots and #Tms. If the file contains numbers only first record was used.

#Dots attribute: The data of this attribute is the number of points detected by the hardware between when the pen touched the screen and when it was raised back up. The lines(s) after this indicate the coordinates of all points in the format x1, y1, x2, y2, x3, y3 and so on.

#Tms: Number of time values. The next line(s) list all times in milliseconds in the format t1, t2, t3 and so on.

If the stroke is a dot(s) or Hamza or bar for Kaaf or Taa, then one of these records is added before the previous five records: #1, #2, #3, #H, #T, #K. for a single dot, two dots, three dots, Hamza, bar for Tha, or bar for Kaf respectively (Table 9).

Ground Truth File

The following example explains the structure of the ground truth file and the records mentioned in the previous section. Figure 7 illustrates the ground truth table for the word in Fig. 6. First, six records are used to indicate each of the following: Writer name, strokes count, letters count and the word in Arabic, Latin and Unicode. Since the strokes count is five, there are five sections (only three of them are shown in Fig. 7). The first section gives information regarding the first stroke and it consists of two letters (given in Arabic, Latin and Unicode). While writing these two letters, the volunteer passed the pen through 132 pixels.

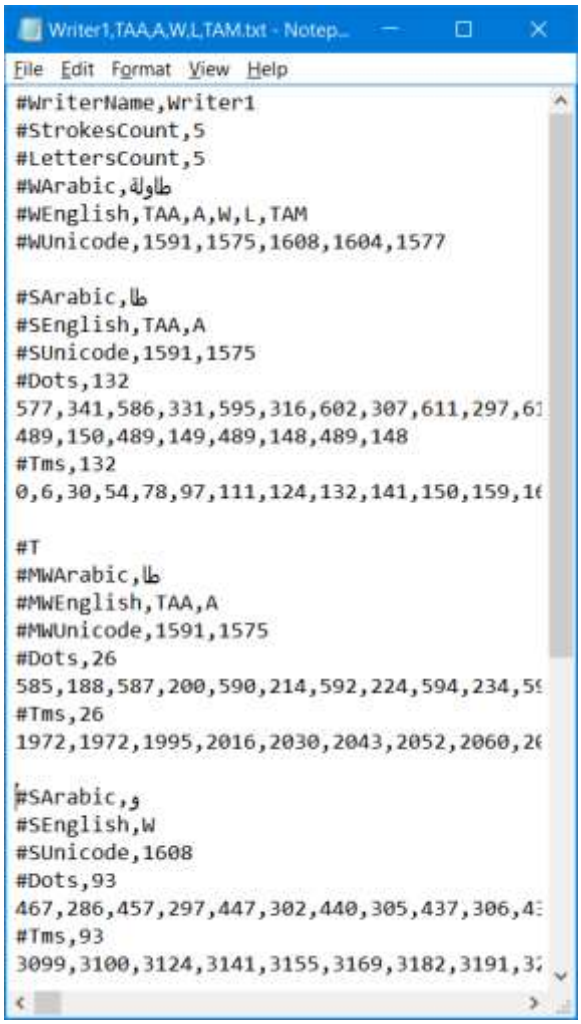

Fig. 7: Part of a truth Table for the Arabic word Tawila

The subsequent lines indicate the coordinates of these pixels in the previously mentioned format. The last record provides the time for each point (i.e., when the

hardware detects it). This is really helpful for writer identification. The next section contains six records. Since this section is not a sub-word, but rather a supplement mark (a bar for Tah), the first record is #T. The subsequent three records provide the name of the syllable the supplement mark belongs to. The next record (#Dots) provides the number of pixels used in writing the bar. The lines after this provide the coordinates of these pixels. The last record (#Tms) determines the time that each pixel was detected. Note that the writer starts to write the second syllable after approximately two seconds (1972 ms).

In other words, the writer uses approximately two seconds for writing the two letters in the previous subword, plus the delay between two sub-words. The delay is equal to the time of the first pixel in the second sub-word (1972 ms) minus the time of the last pixel in the first subword (1282 ms), which is not shown in Fig. 7. The delay time (time that the pen was raised from the tablet) is also a useful feature for writer identification.

The third section includes information regarding the third stroke. It consists of a letter as it appears from the first three records. The fourth record indicates the number of points and the coordinates of the points. The fifth and sixth records determine the number of time values and the next line lists the values themselves. The results of this study indicate that some writers usually wrote dots as a long line and consequently it appeared as two dots. Other writers sometimes wrote two dots as a small line and subsequently these were detected as a single dot. This behavior is important for identifying writers. Thus, in many cases, *x is added to the record. For example, if a dot appears as two dots, the record will be #1*2. Every group of DAD was randomly distributed into training, testing and verification sets, containing 70, 15 and 15% of entries of the dataset, respectively.

Writing Identification Using KNN and Nearest Neighbour Interpolation

To provide other researchers with a benchmark to compare their results, K Nearest Neighbour (KNN) classification algorithm will be used to test both groups. The KNN algorithm is a simple and very effective machine learning technique (Mohammad, 2019; Dhurandhar and Dobra, 2013), as a result it is a commonly used as classification algorithm among researchers. It is used in text recognition and categorization (Alotaibi *et al*., 2017; Wan *et al*., 2012; ALSaif and Alotaibi, 2019; Chen, 2018; La *et al*., 2012), writer identification, image annotation (Gu *et al*., 2017), digit recognition (Gu *et al*., 2017), Arabic language processing (Selamat *et al*., 2009; Boubaker *et al*., 2014; Hafiz and Bhat, 2016; Al-Tamimi *et al*., 2017; Assaleh *et al*., 2009), internet content filtering (Guo *et al*., 2018) and many more.

The KNN has several merits such simplicity and high accuracy but it is relatively computationally expensive.

Preprocessing

The preprocessing concerns the preparation of the writing recognition system when using KNN, since this preprocessing is related with how DAD can be used to achieve high accuracy levels.

Due to the variations in position and scale among different writers and even with the same writer, a preprocessing is an essential step to improving accuracy. Each word undergoes the following steps:

a- All strokes are combined into one stroke
b- Strokes from the previous step are resampled to be accommodated in a fixed number of points, equal for all strokes. For shorter strokes this means up-sampling, whereas it means down-sampling for long ones. Resampling is performed by Nearest Neighbour Interpolation (NNI) (Elglaly and Quek, 2011; Jiang *et al*., 2015; Jiang and Wang, 2015)
c- All stroke points are normalized to mean zero and standard deviation one. This normalization process is performed by subtracting the sample mean to all the values, so that the mean of the new values is zero. Then, all the values are divided by the standard deviation, so that the standard deviation of the new values is one. This normalization process allows one to properly apply KNN with an adequate balance between the different properties

KNN predictions are made using the training dataset directly. Predictions are made for a new data input by searching through the entire training set for the K most similar instances and then summarizing the output variable for K instances. For classification, the output is usually the most common class among the most similar cases. To determine which of the K instances in the training dataset are most similar to new input, a distance measure is used. The most popular distance metrics are Euclidean distance, Cosine distance, Minkowski distance, Mahalanobis distance, Chebychev distance, Hamming distance and Spearman distance.

In order to compare the accuracies of KNN with this dataset, several KNN model types were examined considering different parameter values, k and distance metrics. Table 10 summarizes these models and their importance.

The block diagram of Fig. 8 summarizes the whole work of building DAD and using it for experimentation. Firstly, an iterative process collected handwriting examples, extracted the relevant parts and included them in a ground truth file, conforming DAD. In the experimentation phase, DAD was used for training a recognition system with KNN. Later this recognition system was validated with different writing samples to assess its accuracy.

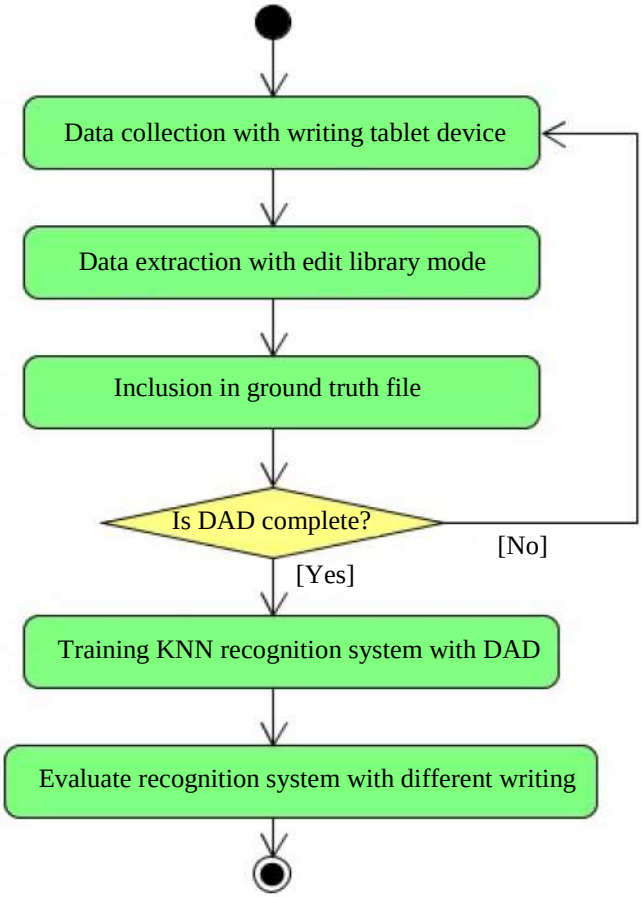

Fig. 8: Block diagram of creation and experimentation with DAD

Table 10: KNN models used in experiments

Model	k	Distance metric
Fine KNN	1	Euclidian
Medium KNN	10	Euclidian
Coarse KNN	100	Euclidian
Cosine KNN	10	Cosine
Cubic KNN	10	Minkowski
Weighted KNN	10	Euclidian

Results

The DAD dataset was evaluated by training a recognition system with it and measuring the accuracy of the trained system. In this research, both groups 1 and 2 are used for training and classification purposes.

Table 11, Fig. 9 and 10 show that the accuracy and prediction speed achieved using group 2 is higher than achieved using group 1. The accuracy and speed for the second group are very high except for the speed for the cubic model. For the first group the speed is very high except for the cubic model, as the first group, but the accuracy is not high as the second group.

Discussion

High level of accuracy is considered as an indicator of the utility of the dataset for being used in writing recognition systems.

The improvement of group 2 over group 1 is attributed to the fact that words in group 2 are repeated several times by every writer.

Using another preprocessing algorithm or another classifier as SVM or deep learning must improve the accuracy.

Regarding the different KNN models, the most effective one in terms of accuracy was Fine KNN. This

model used a K parameter of 1 (i.e., just considering one neighbour, i.e., the most similar one) and the Euclidian distance, which is essentially based on the distances for each dimension. This reveals how low amounts of neighbours can obtain appropriate results in this context of Arabic writing recognition.

Table 11: Accuracy and prediction speed

	First group		Second group	
Model	Accuracy %	Prediction speed obs/sec	Accuracy %	Prediction speed obs/sec
Fine KNN	86.4	~110	99.4	~110
Medium KNN	85.0	~110	98.9	~110
Coarse KNN	73.2	~120	96.0	~120
Cosine KNN	85.4	~120	98.9	~130
Cubic KNN	84.2	~3.8	98.7	~3.8
Weighted KNN	86.2	~130	99.1	~130

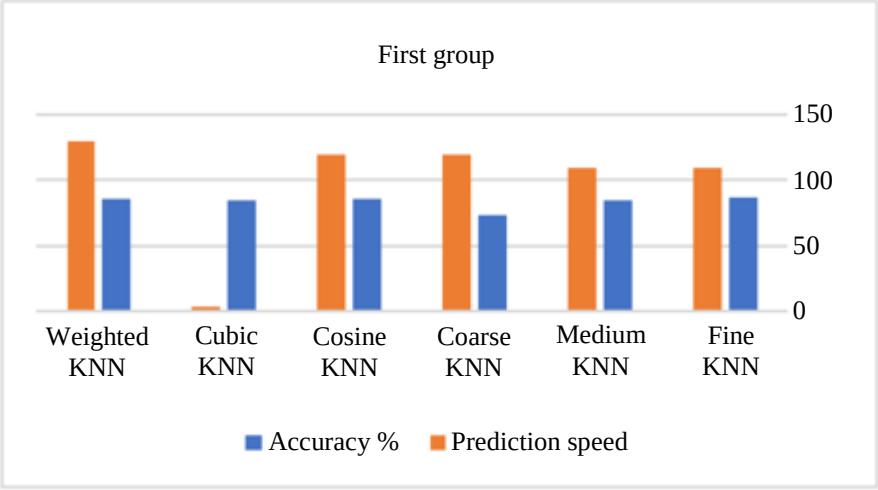

Fig. 9: Accuracy and prediction speed or first group

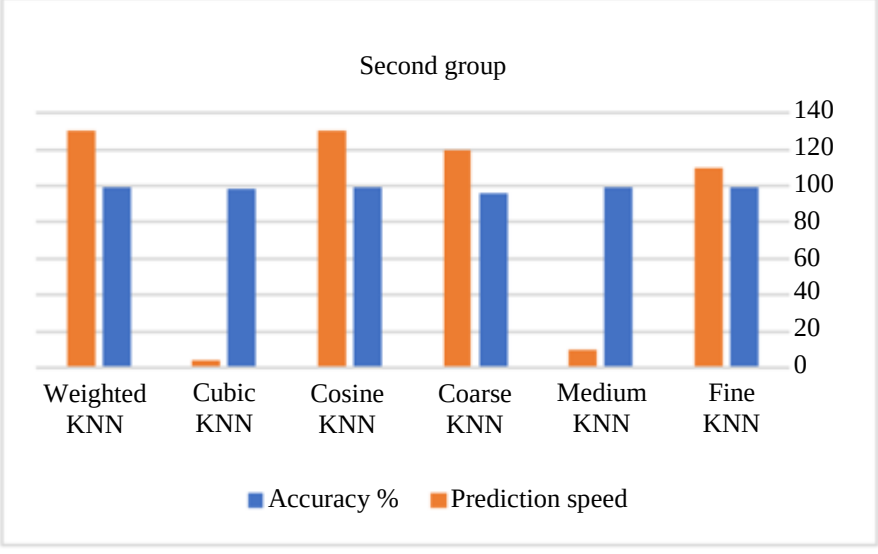

Fig. 10: Accuracy and prediction speed for second group

Conclusion and Future Work

This paper has introduced a new type of Arabic online handwritten word dataset. Its novelty relies on that it considers certain Arabic font details that are not taken into account by other similar datasets. Its ground truth files contain full details regarding sub-words and other diacritics: Single dot, two dots, three dots, Hamza and supplement marks (i.e., bar to "Tah" or "Zah" and Complement to "Kaf"). It contains 136 Bar for the letter "Kaf", 930 three dots, 2409 Bar for "Tah" or "Zah", 2777 Hamzas, 9580 two-dots, 15610 single dots and 46560 sub-words. It contains also 14946 Indian numbers.

Totally ground truth for DAD contains 93064 records in 18767 files. The number of records is about five times more than the number of files, usually in other Arabic datasets the number of records is equal to the number of files, this is explained because the ground truth in DAD is based on sub-words and letters parts instead of words or lines.

This dataset will provide researchers with a strong tool for online Arabic language text recognition especially in the segmentation phase and writer identification.

As future work, it is planned to make a new version of this dataset that can (a) consider rarely used diacritics such as "Madda" and (b) collect more data, such as pressure, altitude, azimuth and coordinates while the pen is raised. This information would be very helpful for writer identification. With the contribution of other researchers from other countries, it is also planned to increase the number of writers and words and to apply these methods for offline datasets.

For some methods, it is a good idea to start experiments with this dataset then going to other datasets with less information about the text, but more writers.

This dataset will be freely available for academic researchers worldwide with an interest in this study.

Acknowledgment

The author would like to acknowledge the support provided by Jouf University through project no. 37/386.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Abdelazeem, S. (2009, December). A novel domain-specific feature extraction scheme for arabic handwritten digits recognition. In 2009 International Conference on Machine Learning and Applications (pp. 247-252). IEEE.

Abdelazeem, S., & Eraqi, H. M. (2011, September). On-line Arabic handwritten personal names recognition system based on HMM. In 2011 International Conference on Document Analysis and Recognition (pp. 1304-1308). IEEE.

Abdelaziz, I., & Abdou, S. (2014). Altecondb: A large-vocabulary arabic online handwriting recognition database. arXiv preprint arXiv:1412.7626.

Abdleazeem, S., & El-Sherif, E. (2008). Arabic handwritten digit recognition. International Journal of Document Analysis and Recognition (IJDAR), 11(3), 127-141.

Abuzaraida, M. A., Zeki, A. M., & Zeki, A. M. (2014). Online database of Quranic handwritten words.

Ahmad, A. T., & Maen, H. (2008). Recognition of on-line handwritten Arabic digits using structural features and transition network. Informatica, 32(3).

Ahmed, H., & Azeem, S. A. (2011, September). On-line Arabic handwriting recognition system based on HMM. In 2011 International Conference on Document Analysis and Recognition (pp. 1324-1328). IEEE.

Alamri, H., Sadri, J., Suen, C. Y., & Nobile, N. (2008). A novel comprehensive database for Arabic off-line handwriting recognition. In Proceedings of 11th International Conference on Frontiers in Handwriting Recognition, ICFHR (Vol. 8, pp. 664-669).

Al-Hashim, A. G., & Mahmoud, S. A. (2010). Benchmark database and GUI environment for printed Arabic text recognition research. WSEAS Trans. Inf. Sci. Appl, 4(7), 587-597.

Al-Helali, B. M., & Mahmoud, S. A. (2017). Arabic Online Handwriting Recognition (AOHR) A Survey. ACM Computing Surveys (CSUR), 50(3), 1-35.

AlKhateeb, J. H., & Alseid, M. (2014, March). DBN-Based learning for Arabic handwritten digit recognition using DCT features. In 2014 6th international conference on Computer Science and Information Technology (CSIT) (pp. 222-226). IEEE.

Almodfer, R., Xiong, S., Mudhsh, M., & Duan, P. (2017, November). Very deep neural networks for hindi/arabic offline handwritten digit recognition. In International Conference on Neural Information Processing (pp. 450-459). Springer, Cham.

Alotaibi, F., Abdullah, M. T., Abdullah, R. B. H., Rahmat, R. W. B. O., Hashem, I. A. T., & Sangaiah, A. K. (2017). Optical character recognition for quranic image similarity matching. IEEE Access, 6, 554-562.

ALSaif, H., & Alotaibi, T. (2019). Arabic Text Classification using Feature-Reduction Techniques for Detecting Violence on Social Media. Work, 10(4).

Al-Salman, A., & Alyahya, H. (2017, October). Arabic online handwriting recognition: a survey. In Proceedings of the 1st International Conference on Internet of Things and Machine Learning (pp. 1-4).

Al-Shamaileh, M. Z., Hassanat, A. B., Tarawneh, A. S., Rahman, M. S., Celik, C., & Jawthari, M. (2019, June). New Online/Offline text-dependent arabic handwriting dataset for writer authentication and identification. In 2019 10th International Conference on Information and Communication Systems (ICICS) (pp. 116-121). IEEE.

Al-Tamimi, A. K., Shatnawi, A., & Bani-Issa, E. (2017, October). Arabic sentiment analysis of youtube comments. In 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT) (pp. 1-6). IEEE.

Assaleh, K., Shanableh, T., & Hajjaj, H. (2009). Recognition of handwritten Arabic alphabet via hand motion tracking. Journal of the Franklin Institute, 346(2), 175-189.

Azeem, S. A., El Meseery, M., & Ahmed, H. (2012). Online arabic handwritten digits recognition. In 2012 International Conference on Frontiers in Handwriting Recognition (pp. 135-140). IEEE.

Boubaker, H., Chaabouni, A., Halima, M. B., El Baati, A., & El Abed, H. (2014, August). Arabic diacritics detection and fuzzy representation for segmented handwriting graphemes modeling. In 2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR) (pp. 71-76). IEEE.

Chen, S. (2018, January). K-nearest neighbor algorithm optimization in text categorization. In IOP Conference Series: Earth and Environmental Science (Vol. 108, p. 052074).

Chernodub, A., & Nowicki, D. (2016). Orthogonal permutation linear unit activation function (OPLU). Lecture Notes in Computer Science (см. в книгах), 9887, 533-534.

de Sousa, I. P. (2018). Convolutional ensembles for Arabic handwritten character and digit recognition. PeerJ Computer Science, 4, e167.

Dhurandhar, A., & Dobra, A. (2013). Probabilistic characterization of nearest neighbor classifier. International journal of machine learning and cybernetics, 4(4), 259-272.

El Abed, H., & Margner, V. (2007, February). The IFN/ENIT-database-a tool to develop Arabic handwriting recognition systems. In 2007 9th International Symposium on Signal Processing and Its Applications (pp. 1-4). IEEE.

El Abed, H., Kherallah, M., Märgner, V., & Alimi, A. M. (2011). On-line Arabic handwriting recognition competition. International Journal on Document Analysis and Recognition (IJDAR), 14(1), 15-23.

El Abed, H., Märgner, V., Kherallah, M., & Alimi, A. M. (2009, July). Icdar 2009 online arabic handwriting recognition competition. In 2009 10th International Conference on Document Analysis and Recognition (pp. 1388-1392). IEEE.

Elanwar, R. I., Rashwan, M. A., & Mashali, S. A. (2010, December). OHASD: the first on-line Arabic sentence database handwritten on tablet PC. In Proceedings of World Academy of Science, Engineering and Technology (WASET), International conference on International Conference on Signal and Image Processing ICSIP (Vol. 69, pp. 910-915).

Elglaly, Y., & Quek, F. (2011). Isolated handwritten arabic character recognition using multilayer perceptron and k nearest neighbor classifiers.

Elleuch, M., Tagougui, N., & Kherallah, M. (2015, March). Arabic handwritten characters recognition using deep belief neural networks. In 2015 IEEE 12th International Multi-Conference on Systems, Signals & Devices (SSD15) (pp. 1-5). IEEE.

El-Sawy, A., Hazem, E. B., & Loey, M. (2016, October). CNN for handwritten arabic digits recognition based on LeNet-5. In International conference on advanced intelligent systems and informatics (pp. 566-575). Springer, Cham.

Eraqi, H. M., & Azeem, S. A. (2011, September). An on-line arabic handwriting recognition system: Based on a new on-line graphemes segmentation technique. In 2011 international conference on document analysis and recognition (pp. 409-413). IEEE.

Gu, Y., Xue, H., & Yang, J. (2017). Cross-modal saliency correlation for image annotation. Neural Processing Letters, 45(3), 777-789.

Guo, T., Wu, L., & Liu, J. (2018, April). Optimization of internet content filtering---Combined with KNN and OCAT algorithms. In AIP Conference Proceedings (Vol. 1955, No. 1, p. 040141). AIP Publishing LLC.

Hafiz, A. M., & Bhat, G. M. (2016). Arabic OCR using a novel hybrid classification scheme. J Pattern Recognit Res, 11(1), 55-60.

Hamdi, Y., Chaabouni, A., Boubaker, H., & Alimi, A. M. (2016, November). Hybrid neural network and genetic algorithm for off-lexicon online Arabic Handwriting Recognition. In International Conference on Hybrid Intelligent Systems (pp. 431-441). Springer, Cham.

Jaha, E. S. (2019). Efficient Gabor-based recognition for handwritten Arabic-Indic digits. International Journal of Advanced Computer Science and Applications, 10(1).

Jiang, N., & Wang, L. (2015). Quantum image scaling using nearest neighbor interpolation. Quantum Information Processing, 14(5), 1559-1571.

Jiang, N., Wang, J., & Mu, Y. (2015). Quantum image scaling up based on nearest-neighbor interpolation with integer scaling ratio. Quantum information processing, 14(11), 4001-4026.

Kharma, N., Ahmed, M., & Ward, R. (1999, May). A new comprehensive database of handwritten Arabic words, numbers and signatures used for OCR testing. In Engineering Solutions for the Next Millennium. 1999 IEEE Canadian Conference on Electrical and Computer Engineering (Cat. No. 99TH8411) (Vol. 2, pp. 766-768). IEEE.

Kherallah, M., Elbaati, A., Abed, H. E., & Alimi, A. M. (2008). The on/off (LMCA) dual Arabic handwriting database. In 11th International conference on frontiers in handwriting recognition (ICFHR).

Kherallah, M., Tagougui, N., Alimi, A. M., El Abed, H., & Margner, V. (2011, September). Online Arabic handwriting recognition competition. In 2011 International Conference on Document Analysis and Recognition (pp. 1454-1458). IEEE.

La, L., Guo, Q., Yang, D., & Cao, Q. (2012). Multiclass boosting with adaptive group-based kNN and its application in text categorization. Mathematical Problems in Engineering, 2012.

Maalej, R., Tagougui, N., & Kherallah, M. (2016, April). Online Arabic handwriting recognition with dropout applied in deep recurrent neural networks. In 2016 12th IAPR Workshop on Document Analysis Systems (DAS) (pp. 417-421). IEEE.

Mahmoud, S. A. (2008, December). Arabic (Indian) handwritten digits recognition using Gabor-based features. In 2008 International Conference on Innovations in Information Technology (pp. 683-687). IEEE.

Mahmoud, S. A., Ahmad, I., Alshayeb, M., Al-Khatib, W. G., Parvez, M. T., Fink, G. A., ... & El Abed, H. (2012, September). Khatt: Arabic offline handwritten text database. In 2012 International Conference on Frontiers in Handwriting Recognition (pp. 449-454). IEEE.

Mahmoud, S. A., Luqman, H., Al-Helali, B. M., BinMakhashen, G., & Parvez, M. T. (2018). Online-KHATT: An Open-Vocabulary Database for Arabic Online-Text Processing. The Open Cybernetics & Systemics Journal, 12(1).

Mezghani, A., Kanoun, S., Khemakhem, M., & El Abed, H. (2012, September). A database for arabic handwritten text image recognition and writer identification. In 2012 international conference on frontiers in handwriting recognition (pp. 399-402). IEEE.

Mohammad, A. H. (2019). Arabic text classification: A review. Modern Applied Science, 13(5).

Njah, S., Nouma, B. B., Bezine, H., & Alimi, A. M. (2012, September). Mayastroun: A multilanguage handwriting database. In 2012 International Conference on Frontiers in Handwriting Recognition (pp. 308-312). IEEE.

Parvez, M. T., & Mahmoud, S. A. (2013). Offline Arabic handwritten text recognition: a survey. ACM Computing Surveys (CSUR), 45(2), 1-35.

Pechwitz, M., Maddouri, S. S., Märgner, V., Ellouze, N., & Amiri, H. (2002, October). IFN/ENIT-database of handwritten Arabic words. In Proc. of CIFED (Vol. 2, pp. 127-136). Citeseer.

Potrus, M. Y., Ngah, U. K., & Ahmed, B. S. (2014). An evolutionary harmony search algorithm with dominant point detection for recognition-based segmentation of online Arabic text recognition. Ain Shams Engineering Journal, 5(4), 1129-1139.

Selamat, A., Subroto, I. M. I., & Ng, C. C. (2009). Arabic script web page language identification using hybrid-KNN method. International Journal of Computational Intelligence and Applications, 8(03), 315-343.

Tagougui, N., Kherallah, M., & Alimi, A. M. (2013). Online Arabic handwriting recognition: a survey. International Journal on Document Analysis and Recognition (IJDAR), 16(3), 209-226.

Wan, C. H., Lee, L. H., Rajkumar, R., & Isa, D. (2012). A hybrid text classification approach with low dependency on parameter by integrating K-nearest neighbor and support vector machine. Expert Systems with Applications, 39(15), 11880-11888.