S.A. Ahmed
Chemical Engineering Department, University of Technology,Baghdad, Iraq.
80155@uotechnology.edu.iq

Comprehensive collection for Arabic characters and numbers written by hand

Abstract- *An Optical Character Recognition system for Arabic language should recognize Arabic handwritten words. However, it is difficult to find a freely accessible and comprehensive database of all Arabic words that can be employed for this purpose. Therefore, it is more efficient to divide the Arabic words into sub-words or characters. As there is no comprehensive Arabic handwritten character database that is accessible free of charge, interested researchers can utilize the database developed as a part of this work in recognition system training and output testing.In the present paper, a database is presented containing scanned images of 700 Arabic handwritten characters, Hindi numbers used in Arabic countries, and some special characters utilized in Arabic alphabet, along with their different positions (e.g., standalone, initial, medial and terminal), different sizes, styles and font colors. The aim is to provide sufficient samples for all character shapes for software training, resulting in greater accuracy in the recognition phase.These forms were filled by students of the Applied Sciences College, University of Technology, Baghdad, Iraq and were scanned at the 200, 300, and 600 dpi resolution. A graphical user interface (GUI) software environment is employed to make the manipulation of the created database easier, and provide many image processing functions that are allowed to be built the database easier.*

Received on: 25/05/2017
Accepted on: 12/10/2017

Keywords- *Arabic text recognition, Arabic characters, Hindi numbers, Image Database.*
.

How to cite this article: : S. A. Ahmed, “Comprehensive collection for Arabic characters and numbers written by hand,” *Engineering and Technology Journal*, Vol. 35, Part B, No. 2, pp. 204-210, 2017.

1. Introduction

Arabic language is important in the culture of many people, due to the large user base of about 300 million individuals worldwide. Most efforts to recognize Arabic text that have been made in the 21st century have focused on the recognition of scanned off-line printed documents [1, 2, 3, and 4]. A number of methods were proposed by researchers, who are interesting in recognize Arabic handwritten, to process documents written in Arabic in order to recognize them [5]. Consequently, developing Optical Character Recognition systems is challenging, as there is no free public database.

In 2001, Dehghan and colleagues [6] developed a database with more than 17820 names of residents of 198 Iranian cities. In 2002, IFN/ENIT database, which contains Arabic names of individuals living in different cities, is available for use in Arabic handwriting recognition free of charge [7]. In 2004, Al-Ma'adeed et al. [8] presented database of Arabic handwriting AHDB—a database containing handwriting of 100 individuals, which included the most common Arabic words used in writing cheques and some handwritten pages.

In 2009, The few Arabic databases that are available for research in Arabic text recognition system (ATRS) have limited content, domain and application range, as they contain data pertaining to postal addresses and numerals, with no more than 4800 words [9].

In 2010, Hashim and Mahmoud [10] presented Printed Arabic Text Database (PADB)database containing 6945 scanned pages of printed Arabic text. However, none of the aforementioned databases included handwritten Arabic characters and Hindi numbers with several handwritten types and font sizes, written by different individuals.

The aim of the present paper is to address this shortcoming and provide a database of Arabic characters and Hindi numbers that are used in Arabic countries for recognition research, especially for recognition systems focusing on handwritten texts. Unfortunately, the currently available databases that are freely available for researchers of Arabic handwritten texts are insufficient for this purpose. The objective of this paper was to build a database containing Arabic characters and Hindi numbers. Matlab 2008a has been used to implement the proposed study and image processing algorithms.

https://doi.org/10.30684/etj.2017.138672
2412-0758/University of Technology-Iraq, Baghdad, Iraq

2. Some key Arabic language features

While there are many features in Arabic language, those pertinent for this research are summarized below:

I. Arabic alphabet consists of 28 letters. The Arabic script is cursive and is written from right to left. The baseline of the word is connected the letters for the word [11].

II. There is no distinction between capital and lower-case letters due to having only one case [12].

III. The letter width is variable (for example ق and ا) [13].

IV. Each letter in Arabic can take three different shapes, according to its position within the word [13], i.e., start, middle and end position. However, six letters (أ د ذ ر ز و) have no start or middle location shape. The letters that come after these six letters must be written in their start location shape, while the previous six letters are joined from the right side only. All Arabic letters are Dual Joining, except for, this is according to the joining type defined by the Unicode Standard [12].

V. These three letters only (ع غ هـ) have four different glyphs according to their location in the word, while the rest of the Arabic letters have two different glyphs in different locations inside the word [12].

3. The Proposed Database

This section describes how the database was built. Building proposed database required two steps: first, is designing form and collecting the data. Second, illustrate the steps of the algorithm to process the forms and building the database.

I. The Design of the Proposed Database

Due to the lack of freely accessible databases for the Arabic language, developing such a database was the key objective of this work. The database presented in this paper will be freely accessible for research purposes. The database contains 700 forms, which are saved as images. Each form consists of 10 columns and 18 rows, as shown in Figure (1). It is designed to contain 28 samples of Arabic letters in all shapes (start, middle and end), depending on the position in the word. It also comprises of ten Hindi numbers (0−9) used in Arabic countries, along with three commonly used special characters (/, *and -). The sample of the form was completed by students of Applied Sciences College, University of Technology, Baghdad, Iraq as shown in Figure (2). Each student filled the form in his/her own handwriting, thus providing individual samples for all letters, numbers, and special characters in a specific place in the form. As a result, 700 styles for each letter case were entered into the database.

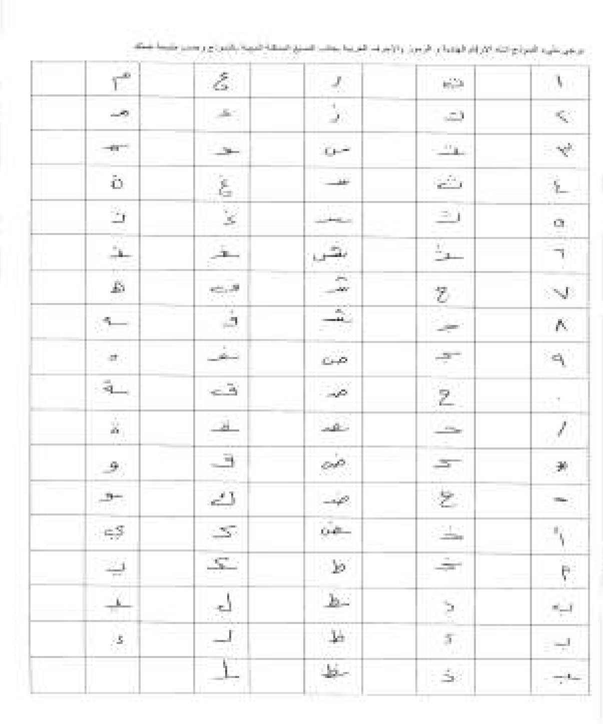

Figure 1: Sample of form before filling

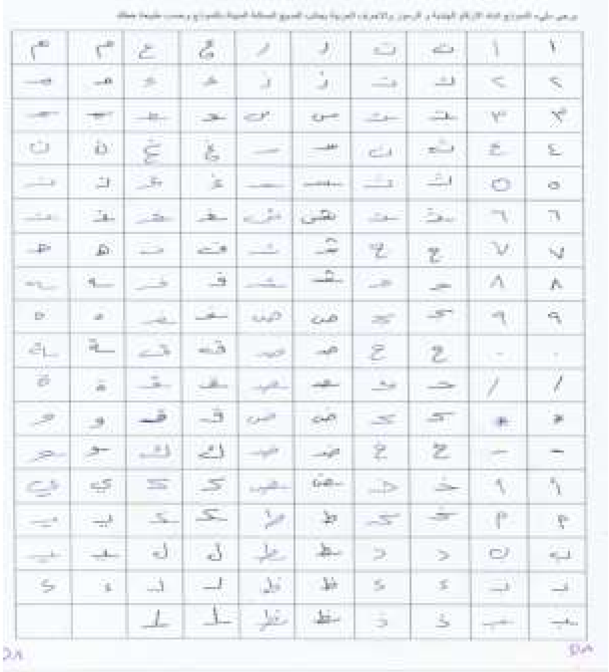

Figure 2: Sample of form after filling

II. The Steps of work

The algorithm utilized in developing the database is described below.

Step 1: Collecting Data in order to build a database that contains characters in all possible shapes, along with numbers and special characters, handwriting samples is needed. As described above, 700 forms were collected to be used for this purpose. Each of the students taking part in the data collection filled the forms using pen and pencil, as shown in Figure (2)

Step 2: Scanning Forms all completed forms were scanned in order to provide a digital format for further processing. The forms were scanned with 200 dpi, 300 dpi and 600 dpi resolution, using Canon Lide110 scanner. Image sizes were not defined and differed from one image to another because the handwritten texts are written in different handwriting styles.

Step 3: Preprocessing

Stage 3.1: Noise Removal Using the median filter, which is a nonlinear operation, the "salt and pepper" noise was removed from the image.

Stage 3.2: Image Resize The output pixel value is a weighted average of pixels in the nearest 4×4 neighborhood.

Stage 3.3: Enhance Image Images were enhanced by modifying the brightness and contrast using intensity adjustment.

Stage 3.4: Enlarge Image Images were enlarged via the bicubic interpolation method in order to determine the values for the additional pixels in the output image.

Step 4: Detecting the Coordinates, Cropping and Saving

4. Experimental Results and Discussion

After collecting and scanning the forms, in order to build the proposed database, each form was processed before cropping the handwritten letters or numbers to enhance the images and obtain the best results. The preprocessing steps consist of processing input data to produce output data to be used as input in another stage.

In the first stage of preprocessing, the noise is removed using the median filter, which is a nonlinear operation that removes the "salt and pepper" noise from the image. Many filters for noise removal exist, but using the median filter is the most effective since it reduces the noise and preserves the edges. The colored input image is converted to a grayscale image, which is subsequently processed. Figure (3) shows a scanned form with noise, while the scanned form after noise removal is shown in Figure (4).

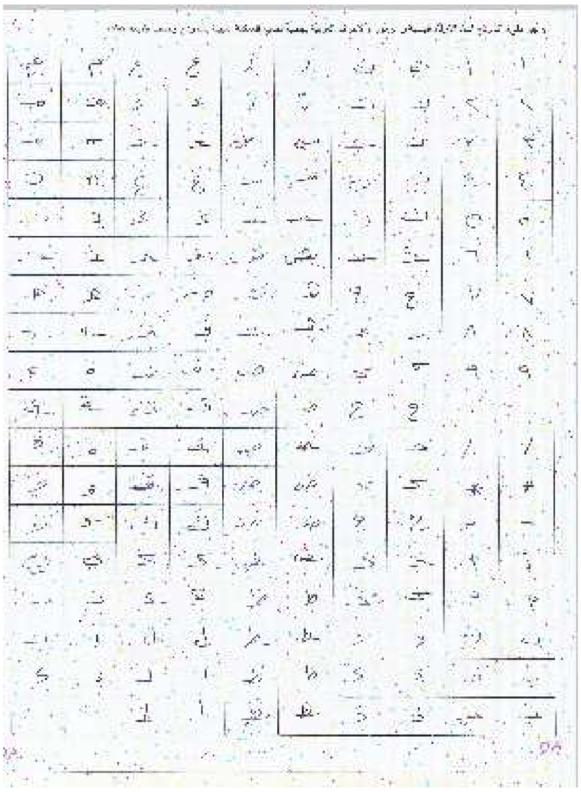

Figure 3: Scanned sample filled form with noise

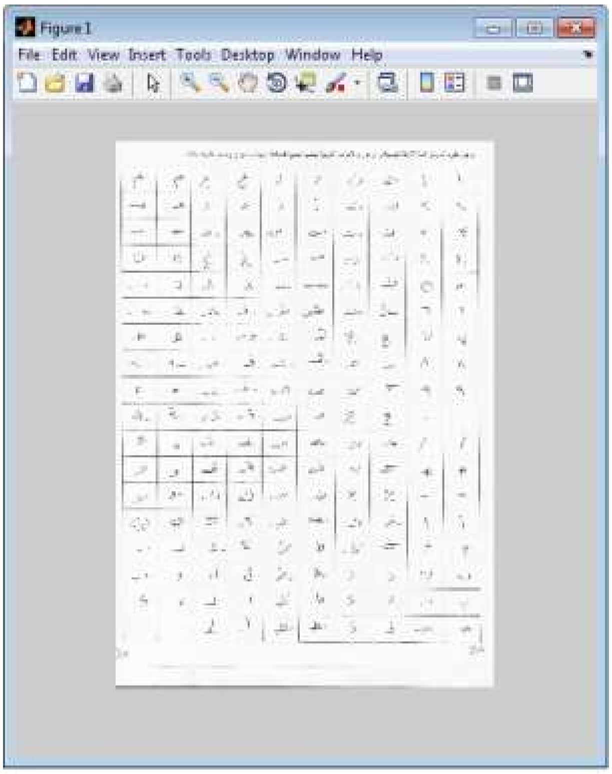

Figure 4: Scanned sample filled form after removing noise.

After removing noise from the image, the image should be resized by setting the output pixel value equal to the weighted average of pixels in the nearest 4×4 neighborhood. The interpolation method was used in this step to prevent losing some features of the image shape, while losing the pixels when resizing the image. All image sizes were fixed to 500 after experiments and their aspect ratios were preserved by scaling their height, as this was the best size to show the image without losing too many pixels. Figure (5) shows a scanned form after resizing.

After resizing the image, it should be enhanced to make the handwritten letters and numbers clearer for detecting the coordinates by the mouse of the computer, cropping and saving the image of letters or numbers digit into the database. The enhancing phase also includes adjusting brightness and contrast, which are modified using intensity adjustment. This technique changes the image's intensity values to a new range, thereby increasing the contrast in the output image. Figure (6) shows a scanned form after enhancing. In the last preprocessing step, the images are enlarged. This stage is necessary, as the two preceding steps resulted in the output image appearing in a "stair-step" pattern. The image loses some of the original pixels due to resizing and increasing contrast. Consequently, the image should be enlarged to ensure that the output image contains a greater number of pixels than the original one. The image is enlarged using the bicubic interpolation method in order to determine the values for the additional pixels in the output image. The interpolation method determines the value of an interpolated pixel by finding the point in the input image that corresponds to a pixel in the output image. Then it calculates the value of the output pixel by computing a weighted average of some set of pixels in the neighborhood of the point. The weighting processes are based on the distance between each pixel and the reference point. Figure (7) shows the scanned form after enlargement.

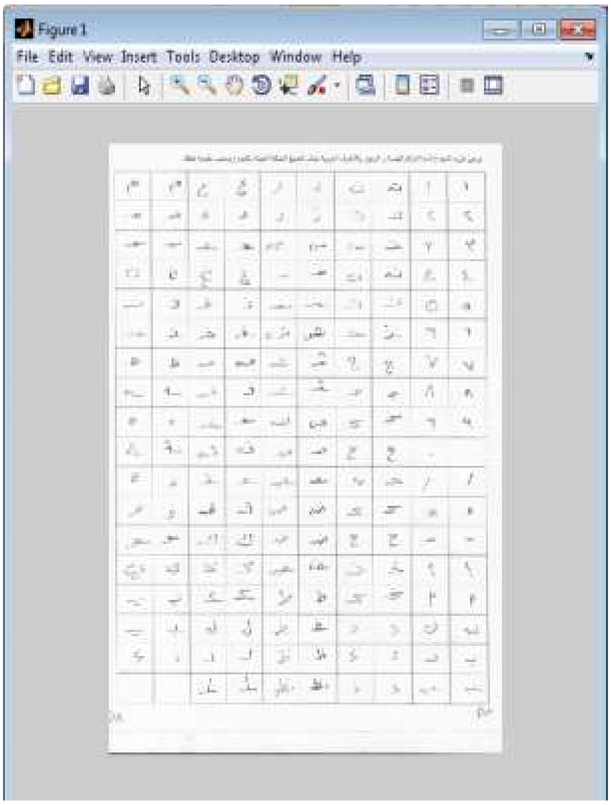

Figure 5: Scanned sample form after resizing.

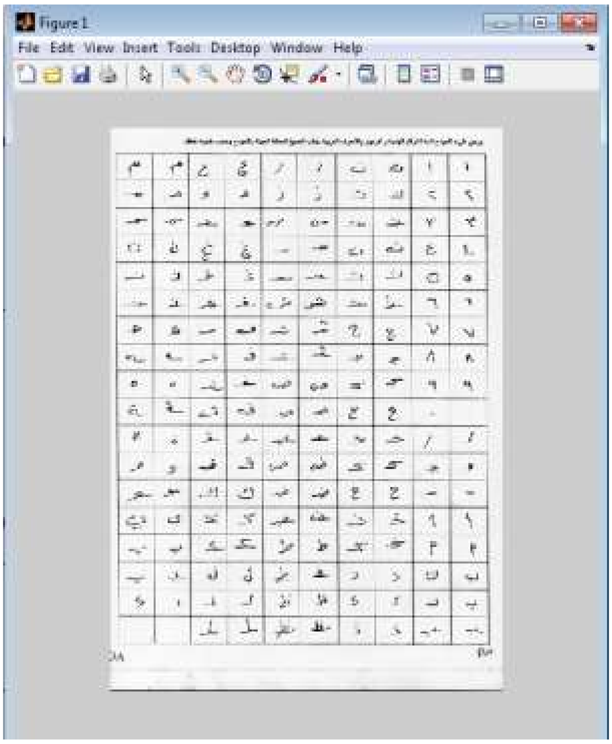

Figure 6: Scanned sample form after enhancing.

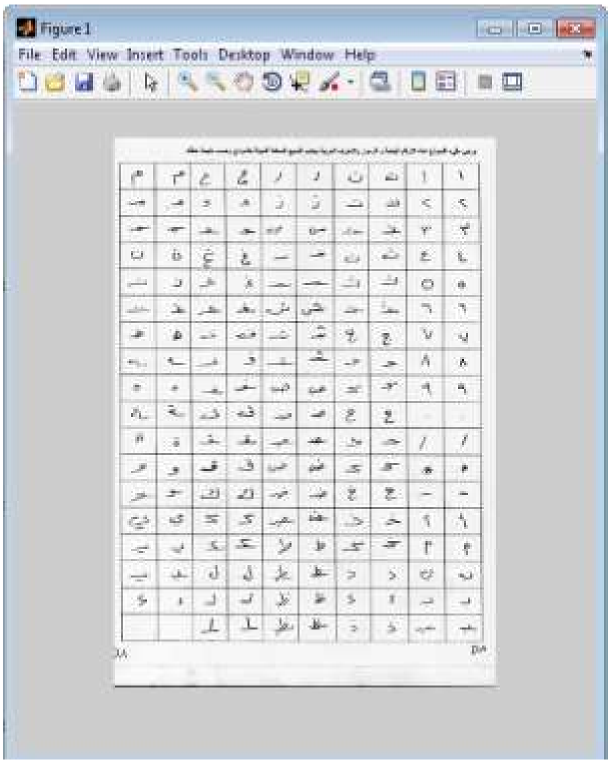

Figure 7: Scanned sample form after enlarge.

The results of performing these preprocessing steps are shown in Figure (8) and Figure (9).

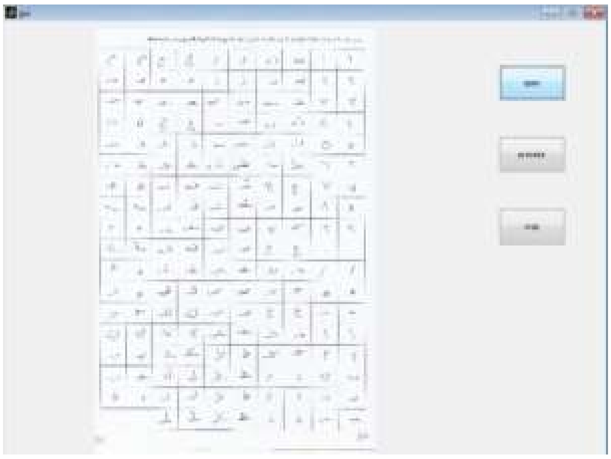

Figure 8: form before preprocessing

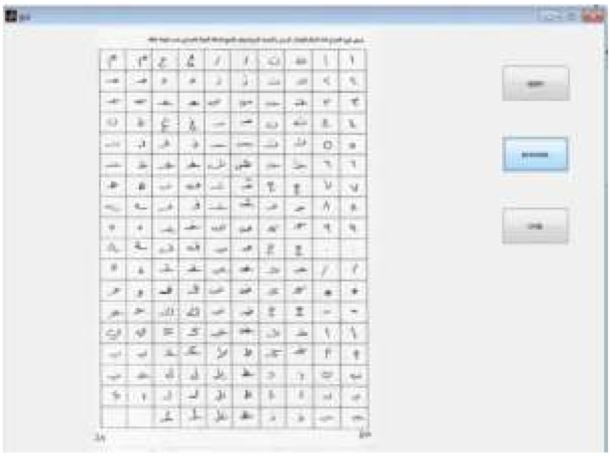

Figure 9: form after preprocessing

Now, the processed images are more explicit and clearer than the originals and can be used for detecting the coordinates of each handwritten character, as shown in Figure (10). Individual characters are cropped from the main image as shown in Figure (11). To save the image of each cropped character, the user should select the “Save As” option from the “File” menu, as shown in Figure (12), and then select the proper folder name of that character, as shown in Figure (13). Finally, the folder that states the character shape depending on its position in the word should be selected, as shown in Figure (14). The steps described above were performed in MATLAB using the “crop image” tool, which was used to save the cropped images. The “Save As” option from the “Image Tool” File menu was used to store the modified data in a file or to use the Export to Workspace option to save the modified data in the workspace variable. Finally, the database of Arabic characters, Hindi numbers and some special characters was created, as shown in Figure (15). Figure (16) shows the folders that contain the character shapes depending on the position in the word.

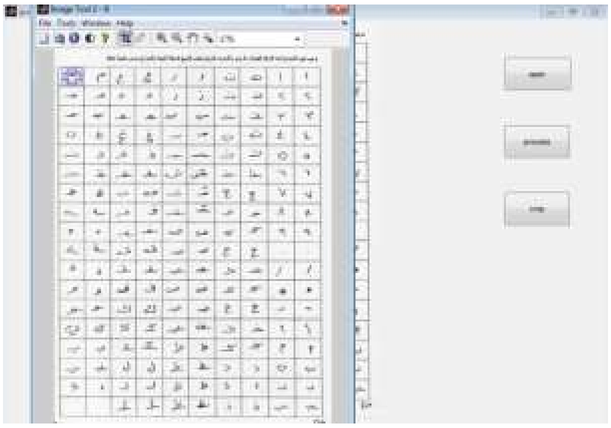

Figure 10: detect coordinate of each characters

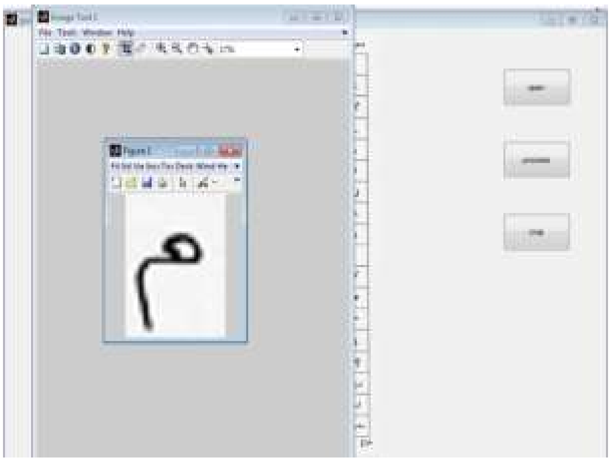

Figure 11: new image from a piece of the main image after cropping

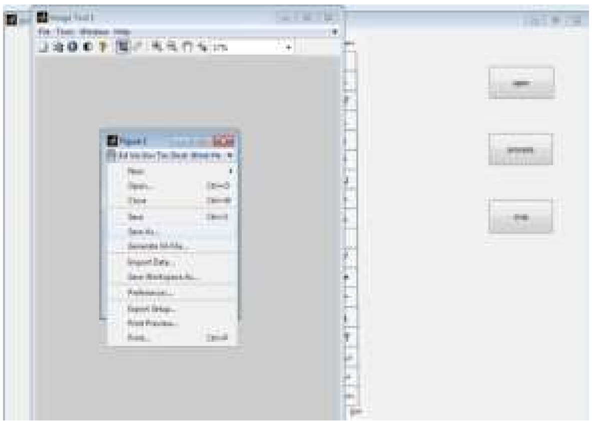

Figure 12: select save as option from the file menu

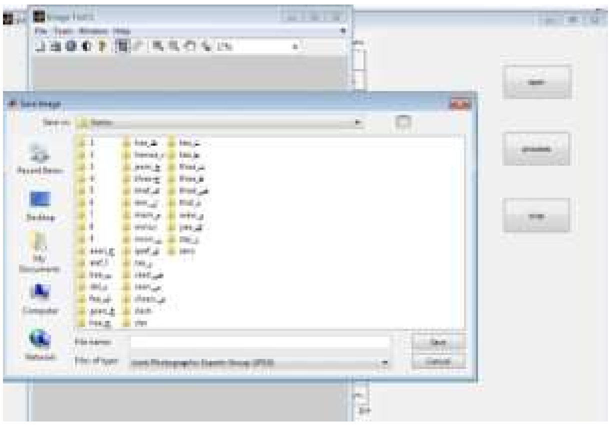

Figure 13: select the proper folder name of character

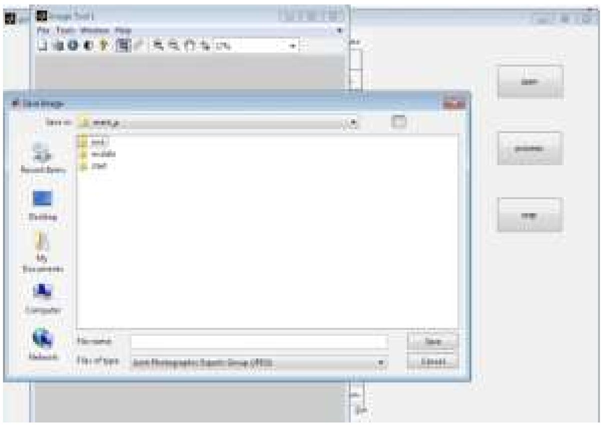

Figure 14: select the folder that state the shape of character depending on its position in the word

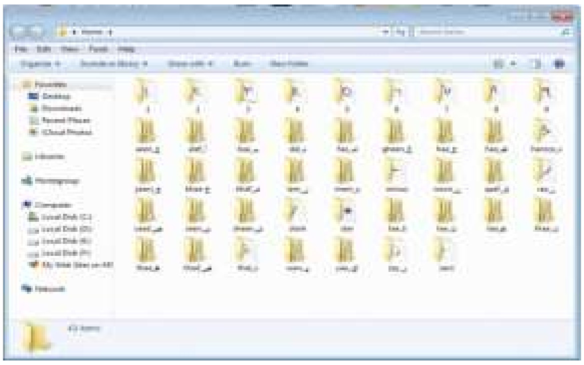

Figure 15: the database of Arabic characters, Hindi numbers and some special characters

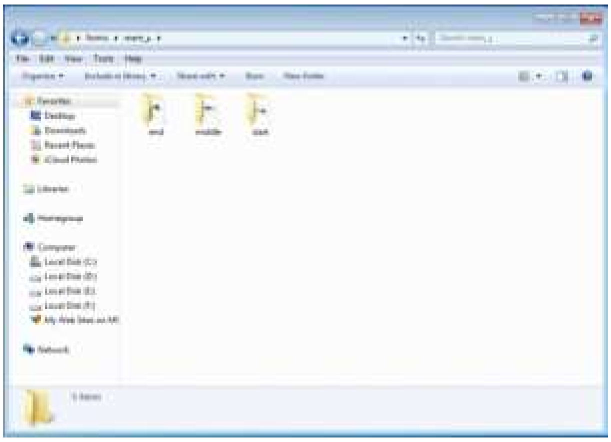

Figure 16: folders that contain the shapes of character depending on its position in the word.

٥. The Collected Database

The database presented in this paper comprises of scanned images of 700 forms that are scanned using 200 dpi, 300 dpi and 600 dpi resolutions. The database also contains 43 folders of images—28 Arabic characters from Alef (أ) to Yaa (ي) addition to Hamza (ء) and Taa (ة), 10 Hindi numbers (from (0 to 9), and three special characters (*, /, -). Each folder represents one character and contains subfolders whose number is equal to all possible shapes the character can take (at the start, in the middle, and at the end of a word). All characters, numbers and special characters were cropped from the main form.

6. Conclusion

The database developed as a part of this research addressed the lack of a suitable database of Arabic handwritten text. This database can be used by researchers and developers working in the field of automatic Arabic Text Recognition to collaborate remotely and compare algorithms and results. This database will soon be made freely available for researchers and developers, including the probabilities of each Arabic letter shape as one of the input features. It can thus be used for the training stage in the machine learning algorithms aimed at recognizing handwritten Arabic characters. The database proposed in this paper was built using Matlab 2008a and image processing manipulation.

References

[1] Khorsheed M. S., “Offline recognition of omnifont Arabic text using the HMM ToolKit (HTK)”. Pattern Recognition Letters, Vol 28, pp. 1563 - 1571, 2007.

[2] Shaaban Z., “A New Recognition Scheme for Machine- Printed Arabic Texts based on Neural Networks”. Proceedings of World Academy of Science, Engineering and Technology,Vienna, Austria, Vol. 31, pp. 706 – 709, 2008.

[3] Slimane F., Ingold R., Alimi M. A. and Hennebert J., “Duration Models for Arabic Text Recognition using Hidden Markov Models”. CIMCA Vienne, Austria, pp. 838 – 843, 2008.

[4] Jalil Luma Fayeq, Mohammed Modhar Mohsen, "A Modified Back Propagation Algorithm for Assyrian Optical Character Recognition Based on Moments". Eng. &Tech.Journal, Vol.34,Part (B), No.2, pp. 255-268, 2016.

[5] Abdul Hassan Alia Karim, Kadhm Mustafa Salam, "An Efficient Image Thresholding Method for Arabic Handwriting Recognition System". Eng. &Tech.Journal, Vol.34,Part (B), No.1, pp. 26-34,2016.

[6] Dehghan, M., Faez, K., Ahmadi, M., and Shridhar, M. "Handwritten farsi (arabic) word recognition: a holistic approach using discrete HMM". Pattern Recognition, vol.34, No. 5, pp.1057–1065, 2001.

[7] M. Pechwitz, S. S. Maddouri, V. Maergner, N. Ellouze, and H. Amiri, “IFN/ENIT - database of handwritten Arabic words”. In Proc. of CIFED 2002, Hammamet, Tunisia, October 21-23 2002,pp. 129–136.

[8] Al-Ma’adeed, S., Elliman, D., Higgins, C.A., "A data base for Arabic handwritten text recognition research". The International Arab Journal of Information Technology, Vol. 1, No. 1, PP. 117 – 12, January 2004.

[9] Al-Muhtaseb H, Mahmoud S, and Qahwahi R. "A novel minimal script for Arabic text recognition databases and benchmarks". International Journal of circuits, systems and signal processing, Vol. 3, pp. 145 – 153, 2009.

[10] Hashim A.G., Mahmoud S.A." Printed Arabic text database (PATDB) for research and benchmarking". In Proceedings of the 9th WSEAS international conference on Applications of computer engineering, ACE’10, pp. 62–68, Stevens Point,Wisconsin, USA, March 2010.

[11] L. Lorigo, V. Govindaraju, “Offline Arabic Handwriting Recognition: A Survey”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 28, no. 5, pp. 712-724, 2006.

[12] AbdelRaouf A., A Higgins C., and Khalil M., “A Database for Arabic Printed Character Recognition”. ICIAR, LNCS 5112, pp. 567 – 578, 2008.

[13] Slimane, F., Ingold, R., Kanoun, S., Alimi, M.A., Hennebert, J.:"A new Arabic printed text image database and evaluation protocols". In: Proceedings of 10th IEEE International Conference on Document Analysis and Recognition (ICDAR 2009), pp. 946– 950. Barcelona (Spain) (2009).

Author(s) biography

Safa Amin Ahmed, M.Sc. in computer scince, Al-Balqaa' Applied University, Salt, Jordan. Her research interest are text recognition, image processing. Mohameed is currently Assistant lecturer in Chemical Engineering Department,University of Technology, Baghdad, Iraq.