DOI: 10.24996/ijs.2024.65.5.32

ISSN: 0067-2904

Optical Recognition System of Non-Dotted Arabic Letters

Ahlam M. Kadhim*, Huda M. Jawad, Farah Jawad Kadhum, Ali A. Al-Zuky
Department of Physics, College of Sciences, Mustansiriyah University, Baghdad, Iraq

Received: 31/8/2022 Accepted: 6/5/2023 Published: 30/5/2024

Abstract

Writing with Arabic characters is a very sensitive operation because some characters are vertically nested and written in multiple ways. The infinite variety of font styles and sizes poses many challenges when designing Arabic recognition systems. In this article, we used deep learning to build an Arabic recognition system that could recognize dotted letters. Our dataset contains 3000 color images for ten letters (م ،و ،ل ،ع ، ،ص ،س،ح ،و ،ط ،ر د). The training stage was achieved through 2250 images (10 classes) of this data, and 750 images were used for the validation stage. The classification accuracy of our proposed model is 98.67 percent. Experimental results show that Arabic font styles (Diwani, Hacent Dalal, Times New Roman, B Tanabe, and Kofi) have the best classification accuracy, so they are the preferred styles to use in Arabic writing texts. The font sizes 28, 36, 48, and 72 are the letters that have been classified with 100% rate accuracy using our trained model, so they are the font styles preferred to use in writing Arabic texts.

Keywords: Arabic character recognition system, Optical Character Recognition (OCR), Deep learning machine.

نظام التمييز البصري للحروف العربية الغير منقوطة

احلام مجيد كاظم*، هدى محمد جواد، فرح جواد كاظم، علي عبد داود
قسم الفيزياء، كلية العلوم، الجامعة المستنصرية، بغداد، العراق

الخلاصة

الكتابة بالأحرف العربية عملية حساسة للغاية ، بعض الأحرف متداخلة رأسياً وتكتب بطرق متعددة. ان التنوع اللامتناهي لأنماط وأحجام الخطوط انتج العديد من التحديات في تصميم أنظمة التعرف على الاحرف العربية. في هذه المقالة ، استعملنا التعلم العميق لبناء نظام التعرف على الاحرف العربية ويمكنه تمييز الحروف غير المنقطة. تحتوي مجموعة البيانات لدينا على 3000 صورة لعشرة احرف (م ، و ، ل ، ع ، ص ، س ، ح ، و ، ط ، ر د). تم تحقيق مرحلة التدريب من خلال 2250 صورة ، و 750 صورة لمرحلة التحقق من الصح موديل التصنيف. تبلغ دقة التصنيف للنموذج المقترح 98.67 بالمائة. تظهر النتائج التجريبية أن الخطوط العربية(Diwani, Hacent Dalal, Times New Roman, B Tanabe and Kofi) هي الأفضل دقة في التصنيف ، لذا فهي مفضلm لاستعمالها في الكتابة العربية. أحجام الخطوط 28 ، 36 ، 48 ، 72 هي الأحرف التي تم تصنيفها بدقة عالية باستعمال نموذجنا المدرب ، بحيث تكون أنماط الخطوط المفضلة لاستعمالها في كتابة النصوص العربية.

* Email : ahlammjeed@yahoo.com

1. Introduction

Character and symbol recognition is a process that enables computers to recognize written or printed characters, such as numbers or letters in digital images, and transform them into texts that a computer can recognize. One of the most important principles of natural language processing is OCR using image analysis techniques.

In today's world, digital mediums are more important than paper documents since they make it easier to scan and categorize them based on the content's subject. Automatic translation techniques and speech synthesis systems are also in wide use nowadays in the field of computer-aided document analysis [1].

Optical reading of symbols or characters is one of the simplest techniques for reading text from digital images. whereas OCR is an automatic conversion of text in digital images into text that is treated as text in computers using image scanning. The intelligent character recognition system can be considered an advanced OCR system. The use of intelligent recognition systems for handwritten and printed characters in digital images has greatly developed over recent decades, especially for languages whose lettering is separated. Recognition methods include intelligent word recognition (IWR), OCR, optical mark recognition (OMR), and intelligent character recognition (ICR), with OCR and deep learning for the CNN network being the most commonly used techniques. The core concept behind OCR is to convert scanned bitmaps of printed or handwritten text into machine-editable data files. There are many areas where this technique could be applied, like banks, health care, postal offices, education, the publishing business, finance, government organizations, and other areas [2].

In recent years, deep learning has gained a lot of traction in the pattern recognition field [3]. Deep learning uses neural networks for pattern recognition system design. The most critical component of an OCR system is the classifier, and neural networks are an outstanding solution for OCR classification because of their exceptional ability to extract meaning from complicated or imprecise data [4, 5]. When given a large amount of training data, they are found to be very efficient in solving computer vision problems and have state-of-the-art output. The benefit of CNN is that it automatically extracts the salient features of the input characters that are invariant and, to a degree, subject to change and distortion. Many researchers are interested in deep learning methods because they are the simplest way to deal with large amounts of data and automate the feature extraction process [6, 7, and 8].

Mohamed A. Radwan, Mahmoud Khalil, and Hazem Abbas present the Neural Networks Pipeline for Offline Machine-Printed Arabic OCR, an extended version of their paper from 2016, as a viable solution to the problem of Arabic optical character recognition based on a pipeline of three separate neural modules responsible for font size normalization and word sizing. The results of the dataset on Arabic characters are given and analyzed [9].

The use of a sliding window technique for segmentation is proposed by AlKhateeb et al. (2011) [10] and Radwan et al. (2018) [11]. The likelihood of a particular sliding window consisting of many segments being a real character is determined using a CNN. The portions that qualify as characters are then input into their character recognition model. When tested on many font sizes, this segmentation method showed very high performance on a single font but failed to keep up this high performance.

Hussein Osman et al. (2020) [12] Present a neural network model. The system has been tested on several open Arabic corpora datasets and has achieved outstanding results when compared to state-of-the-art Arabic OCR systems, with an average character segmentation accuracy of 98.06%, character recognition accuracy of 99.89%, and an overall system accuracy of 97.94%.

Arabic language processing requirements are not met by the use of a number of tools, packages, and natural language processing techniques. To create such tools, techniques and software are required to process and understand Arabic-language data. One of the things that must be taken into account is, first, that the Arabic language is written from right to left, and second, that it includes 28 different letters, in addition to the existence of several formulas for writing the same letter according to the position of the letter at the beginning, middle, or end of the word. And third and most important is the presence of diacritical marks above or below the letter, which are small letters that can be added either as a subscript or a superscript to give distinct spelling and different grammatical formulation, and diacritical marks are widely used in the Arabic language [13].

Previous research did not address the effect of classifying and distinguishing letters written in many Arabic font styles with different sizes. So in this study, a recognition system for non-dotted Arabic letters is proposed. The advantages of neural networks in deep learning are employed to produce a highly efficient classification model. The accuracy of the classification of Arabic letters reached 98.67% in most types of Arabic calligraphy currently in use and for different font sizes. In the training process, the neural network was trained on ten categories of Arabic letters written using ten known Arabic font styles of different sizes, then tested for the accuracy of discrimination and for colored text images written with font types different from what the model was trained on. Systems for classifying Arabic letters have been proposed by other researchers, but the type-changing effect of Arabic calligraphy style has not been studied yet, and none of the other researchers have studied the effect of changing the text size on the classification accuracy. So, in this work, the accuracy results of the classification model are analyzed by changing two factors: the first is font size, and the second is font type.

2. Methodology Optical Recognition System

CNN is a type of neural network capable of extracting very fine detail and features from images. This network's design consists of several main layers: convolution layers that use convolution operations to detect detail features; pooling (sub-sampling) layers that reduce data dimensions and transfer this reduction to the next layer. The final layer is the fully connected layer(s), which connect all neurons from the first layer to all neurons in the other layers. The first layers extract features from the raw image, and the final layers classify them. The first step is to decide on the layer order and number, followed by selecting the necessary functions and parameters. Reasonable hyperparameters should be chosen for each layer since they are critical to the network's overall effectiveness.

2.1 Data Initialization Step

For any recognition device, the pre-processing stage is critical. Converting an RGB image to a grayscale image, filtering, and smoothing are all part of the pre-processing level. While writing, letters have been written in various script and Arabic font styles [13]. The letters have been written within a Word document table, and then a cutting process has been done to separate each letter. The total number of images that entered the training stage was 3000 text-color images. Each one of them has been written and represented by the 10 different sizes for

each one of the ten Arabic fonts. To minimize the noise effect and enhance the readability of the input image, several pre-processing methods were used, such as:
1. Convert a color image to a grayscale image.
2. Convert a grayscale image to a digital binary image.
3. Apply morphological processing (dilated) to the binary image to identify and separate each letter.
4. Surround the letter's image with a colored box to get separated images for all letters within the word table.
5. Convert the grayscale image to a color image, then resize it to 30 x 30 and save it for each letter.

The program is built to segment each letter block and determine the coordinates of each block as separate color images of small sizes. To get a good result, any convolutional neural network needs a lot of training data. In our work, each class contains 300 color images of size 30*30*3, with 240 images of white background and black lettering. The rest of the images (60 images) have been processed by converting to binary and using logical conversions for image processing techniques to create a new image with a black background and a white letter color.

2.1 Create training and testing images

Ten non-dotted Arabic letters were studied and classified using a trained model with deep learning technology. The letters are Aeen (ع), Daal(د), Haa(ح), Laam(ل), Meem(م), Raa(ر),Seen(س), Saad(ص), Taa (ط) and Waaw(و) which are the classes of the training data. Each letter has been written in different sizes (16, 18, 20, 22, 24, and 26) and different fonts, as shown in Table 1. Ten styles of Arabic fonts were used in writing the letters to create the colored images used for the training and testing stages, as shown in Table 1. As can be noticed from Table 1, each letter has been written in ten forms according to type Arabic calligraphy ($font_i$) as mentioned in the Microsoft Word 2010 program, where (i = 1, 2,... 10) refers to Arabic script styles (Andalus, Diwani Letter, Hacen Dalal, Text AF_Al Hada, Times New Roman, Arabic Typeset, B Tanab, Courier New, Kufi, and Badiefont-Dima). All ten classes of letters have been written in ten Arabic scripts, so that each letter has ten forms that differ from each other.

Table 1: The non-semicolon Arabic letters for ten Arabic font styles

$\mathbf{font_i}$	**Font name**	$\mathbf{Class_1}$	$\mathbf{Class_2}$	$\mathbf{Class_3}$	$\mathbf{Class_4}$	$\mathbf{Class_5}$	$\mathbf{Class_6}$	$\mathbf{Class_7}$	$\mathbf{Class_8}$	$\mathbf{Class_9}$	$\mathbf{Class_{10}}$
$\mathbf{font_1}$	Andalus	ع	د	ح	ل	م	ر	س	ص	ط	و
$\mathbf{font_2}$	Diwani Letter	ع	د	ح	ل	م	ر	س	ص	ط	و
$\mathbf{font_3}$	Hacen Dalal Text	ع	د	ح	ل	م	ر	س	ص	ط	و
$\mathbf{font_4}$	AF_Al Hada	ع	د	ح	ل	م	ر	س	ص	ط	و
$\mathbf{font_5}$	Times New Roman	ع	د	ح	ل	م	ر	س	ص	ط	و
$\mathbf{font_6}$	Arabic Typeset	ع	د	ح	ل	م	ر	س	ص	ط	و
$\mathbf{font_7}$	B Tanab	ع	د	ح	ل	م	ر	س	ص	ط	و
$\mathbf{font_8}$	Courier New	ع	د	ح	ل	م	ر	س	ص	ط	و
$\mathbf{font_9}$	Kufi	ع	د	ح	ل	م	ر	س	ص	ط	و
$\mathbf{font_{10}}$	Badiefont-Dima	ع	د	ح	ل	م	ر	س	ص	ط	و

The program is built to segment each letter block and determine the coordinates of each block as separate color images of small sizes. To get a good result, any convolutional neural network needs a lot of training data. In our work, each class contains 300 color images of size 30*30*3, with 240 images of white background and black lettering. The rest of the images (60 images) have been processed by converting to binary and using logical conversions for image processing techniques to create a new image with a black background and a white letter color.

3. Training Stage

The CNN network is a type of deep learning neural network that has demonstrated unrivaled performance in resolving image-related problems. Some of these problems include image classification, image semantic segmentation, object detection in images, and many other applications of CNNs. Network layers are used to provide the network's final nonlinear function combinations. The reduction of an error on unseen samples is a major problem in machine learning. It is undesirable to have a model that performs well on the training stage but poorly on the unknown data test stage. Many recognition systems refer to techniques that are used to minimize these errors, so dropout and weight decay are popular regularization methods in learning networks [16].

In CNN networks, the input data is stored in a grid-like topology. A variety of layers receive this response grid: convolution layers, pooling layers, and fully connected layers, or dense layers, are the three basic layers for the CNN model, which are used in CNN as follows [17]:

Convolution layer: The convolution layer creates the features map, which will be used to discern specific input data images. These feature maps record the important features in an image, with minor changes in the resulting pixels placed in completely different feature maps [18].

Max Pooling Layer: Down sampling is one of the solutions to the problem that the features map can cause; it is achieved in the max-pooling layer, which scales the number of features produced.

Rectified linear unit (ReLU): The function ReLU is used as an activation function of a mathematical mechanism connected between convolution layers and the max pooling layer, which is employed using the following activation function as in Eq. (1) [19]:

$$\sigma(\mathrm{Y_j}) = (0, \mathrm{Y_j}) \qquad (1)$$

where j is the index of a pixel within the feature map.

Fully connected layer: It is used to classify the input images; usually it is the last layer in the CNN architecture. Figure 1 shows the multi-layer training model. Each layer will be used as an input for the next layer, and so will all other layers. In this research, the size of the input color images (30*30*3) was entered into the training stage, which contains four layers whose details and content are shown below:

1. Input the data set images of size 30*30*3.
2. first convolution layer using 8 filters of size 3*3. Output feature maps are processed and downsampled by the first Max Pooling layer of block size 2*2 and two strides.
3. second convolution layer using 16 filters of size 3*3. Output feature maps are processed and downsampled by the second Max Pooling layer of block size 2*2 and two strides.
4. third convolution layer using 32 filters of size 3*3. Output feature maps are processed and downsampled by the third Max Pooling layer of block size 2*2 and two strides.
5. fourth convolution layer using 64 filters of size 3*3. Output feature maps are processed and downsampled by the fourth Max Pooling layer of block size 2*2 and two strides.

6. The last layer is a fully contacted layer with 8*16*32*64 neurons for each band color image and for all numbers of an input data image. Using the soft max function, it is possible to obtain the best ten probabilities for the 10 letter classes.
7. The output of the training stage is the training classification model function (CMF) that is used for testing any image.

When the CMF is trained, it is very important to monitor training progress. From the plotting figures of different training metrics, the training progress method can be known.

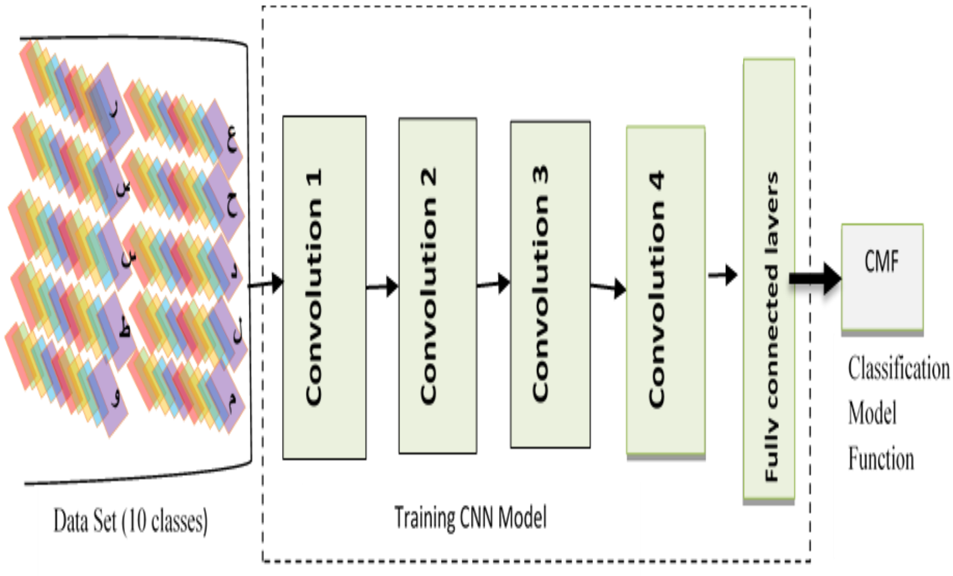

Figure 1: The CNN training stage for the classification model function (CMF)

It is important to determine whether the accuracy and speed of the neural network are improving and whether the network has increased classification accuracy for the training data. Figure 2 shows the training progress of the CMF model and the accuracy value of the training options. The training algorithm generates the number and displays the training metric graph for each iteration. A set of training model options has been created using the stochastic gradient descent momentum (sigmoid) function. The learning rate reduces by 0.2 every 5 epochs. The maximum training epoch number has been set to 10. The batch size is 64 samples per iteration to track the training progress. The training time is 20 seconds, as illustrated in Figure 2. The figure shows the validation measures each time the model is validated. The epoch is a full pass through the entered data set. Figure 2 illustrates that the missing classification rate is close to 0% when the training of data reaches epochs (8 to 10).

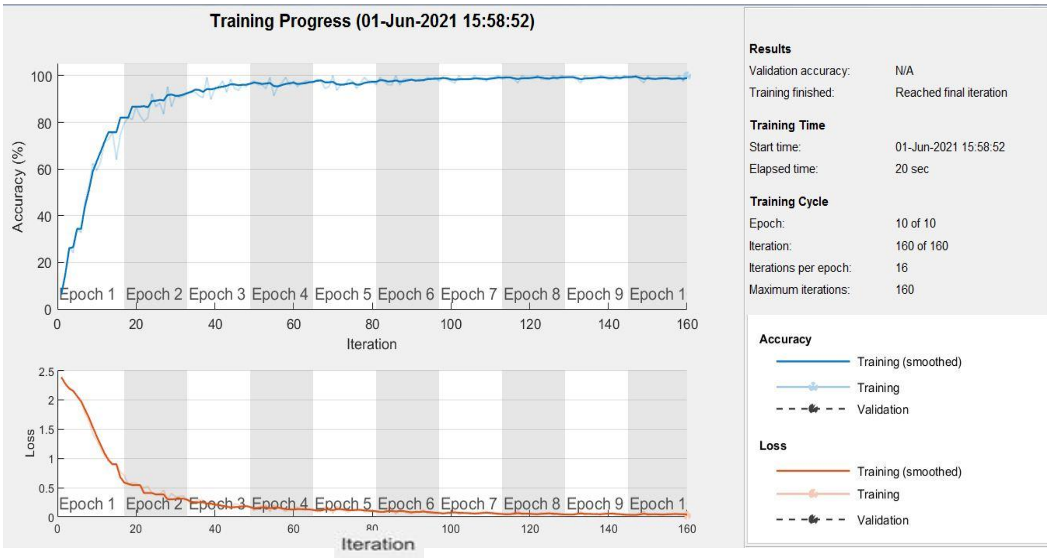

Figure 2: Validation accuracy and loss function calculation of training data

The epochs of the training model are equal to 10, but from epoch 8, the model shows a high accuracy classification with very low missing and errors. The figure plots of the missing errors function use the root mean square error instead of accuracy. The figure shows that the error rate begins with a gradual decrease for each epoch from epoch 1 to epoch 6 until it approaches zero at epochs 7, 8, and 9.

4. Testing Stage

Using the CMF model, the classification accuracy as a function of changing size and font style has been studied in the testing stage. The CMF model has been tested for the sizes 8, 9, 10, 11, 12, 14, 16, 18, 20, 22, 24, 28, 36, 48, and 72. The CMF model succeeded in accurate classification despite the challenges of different fonts and sizes in Arabic writing texts. Figure 3 shows the steps of the testing stage of a text image. The tested image is entered into the CMF model to get the probability matrix for the 10 classes. The maximum probability represents the right class. As an example, Figure 3 shows that the greatest probability is for the third class (Haa), which is the right class. The result of the soft max activation function for each unit's output ranges between 0 and 1. The soft-max classifier is used to get the probability value for each class; the soft-max function is applied to the last output units of CNN networks. The probability split matrix output so that the sum of all probabilities equals 1 could be represented as in Eq. (2) [19]:

$$\mathrm{SoftMax}(\mathrm{Z_j}) = \frac{e^{Z_j}}{\sum_{i=1}^{k}(e^{Z_i})} \qquad (2)$$

where Z is the input vector indexed by j, which ranges from 1 to k.

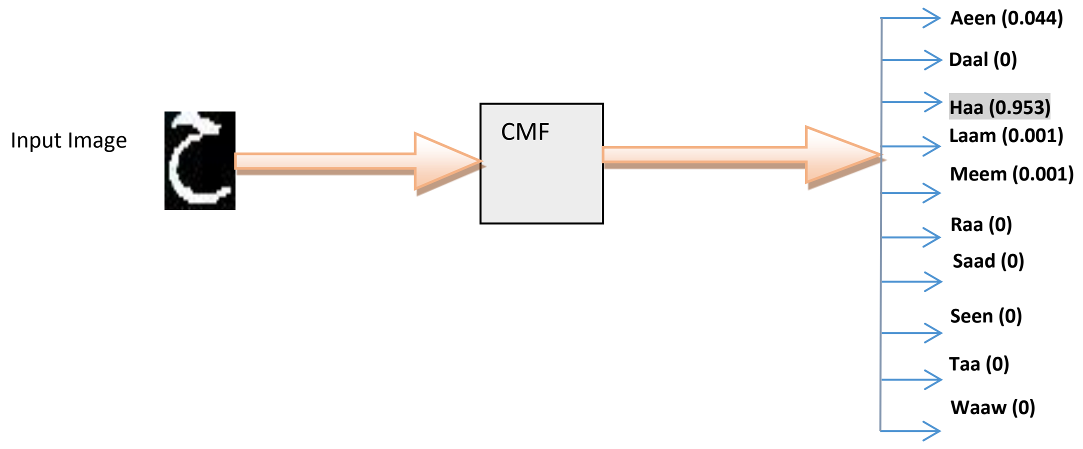

Figure 3: The testing stag for any text-colored image

Batch normalization is needed since the distribution of the input layers changes as earlier layer parameters change during the training processing. The CNN is needed to normalize the input of layers and thus accelerate learning [18]. At the last training, the fully connected layer was terminated by the applied soft-max function. The number of outputs is generated as the same class number within the classification step. There are ten probability values for each class, where the largest value identifies the right label of class for the input testing image.

5. Results and discussion

To evaluate CNN's model accuracy, we need some tools to understand how its work is performing. There are many measures introduced by researchers, and everyone considers certain aspects of algorithmic performance. The table shows high true cases to predict the right class for fonts 1, 2, and 3, while the false cases' classifications were obtained.

The font is 10 in the sizes 8, 9, 10, 11, and 12. In some cases, small sizes for the fonts 4, 5, 6, 7, 8, and 9 are not recognized. In Table 2, the true or false case classification is high for the letter (Aeen). In this article, general measures have been used to get valuable information about CMF accuracy evaluation. The important measure is accuracy (Acc), which is calculated according to Eq. 3. Accuracy is equal to the number of true positives (TP) produced divided by the sum of true positives and false positives (FP) [18, 19, and 20]:

$$\text{Acc} = \frac{\text{TP}}{\text{TP+FP}} \quad (3)$$

Table 2 illustrates right and false classification cases regardless of font sizes using the threshold value of score > 85 true (1), and the false classification represents the cases of score < 85 false (0). Figure 4 shows classification accuracy results for all font sizes, regardless of font styles. The classification accuracy is different for all classes of all sizes, some of which have 100% accuracy and some of which have less than that.

The Arabic letters Meem and Seen obtained the highest classification accuracy, reaching 100%, for the ten Arabic fonts in the sizes 8–72, while for the small font sizes (14–20), the classification accuracy rate reached 90 percent. The accuracy for the letters Aeen, Taa, Haa, Laam, and Waaw is perfect and has reached 100% to classify in sizes from 14 to 72 for all fonts. The small font sizes (8–14) and the accuracy range between 60–90% in all Arabic fonts Daal letter classification accuracy is in the range of 30–80% for sizes 8–14 and 90–100% for sizes 24–72. Raa letter classification accuracy is 20–70% for sizes (8–16) and 90% for sizes (16-72) to all Arabic fonts.

Table 2: True and false values for all Arabic fonts with different sizes

size	$font_1$	$font_2$	$font_3$	$font_4$	$font_5$	$font_6$	$font_7$	$font_8$	$font_9$	$font_{10}$
8	1	1	1	0	1	0	0	1	0	0
9	1	1	1	1	1	1	1	1	1	0
10	1	1	1	1	1	1	1	1	1	0
11	1	1	1	1	0	1	1	0	1	0
12	1	1	1	1	1	1	1	0	1	0
14	1	1	1	1	1	1	1	1	1	1
16	1	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	1
24	1	1	1	1	1	1	1	1	1	1
28	1	1	1	1	1	1	1	1	1	1
36	1	1	1	1	1	1	1	1	1	1
48	1	1	1	1	1	1	1	1	1	1
72	1	1	1	1	1	1	1	1	1	1

Saad letter classification accuracy is discrimination (40%–60%) for all font sizes. It is noticed from Figure 5 that the accuracy of discrimination is 40% for Saad writing using font 1, while for the classes 8–10 (Seen, Taa, and Waaw), it is 100%. At Font2, the accuracy of discrimination for all classes is about 90–100. At font 3, the accuracy of discrimination is about 90–100%, but for the Saad letter, it is zero.

At Font4, the accuracy of discrimination for classes 1 and 2 is about 80%, and then it becomes 100% for classes 3-6. The accuracy for the classes (7, 8, and 9) ranges from 80 to 90. For the font5 accuracy of discrimination for classes 1, 2, 3, 6, and 7, it was about 70–90%, then became 100% at classes 8, 9, and 10. It is noticed that, at font 6, the accuracy of discrimination classes 2, 6, 7, 8, and 10 ranges from 40 to 70%, then becomes 100% at classes 3, 4, 5, and
For the font7 accuracy of discrimination for all classes, it is about 90–100%, except for class 10, where it is less than 60%. The accuracy of discrimination for all classes is about 90–100%, but at class 7, the accuracy is 8%. It is found that discrimination accuracy for all classes is about 80–100%, but at class 7, the accuracy equals zero. The accuracy of

discrimination for classes 1–7 and 9–10 ranges from 60–100%, except for class 8, where it reaches 100%.

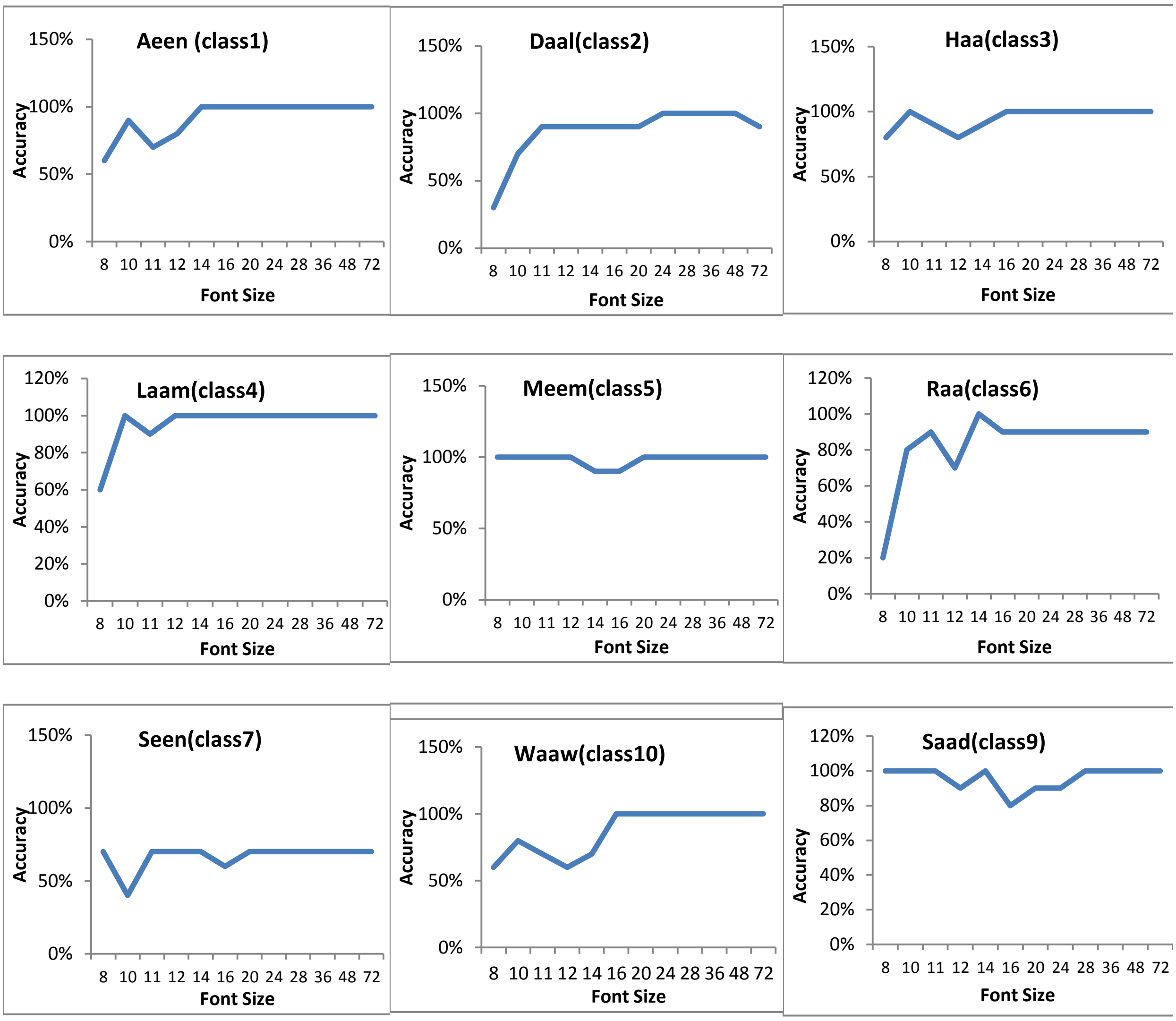

Figure 4: Accuracy as a function of font size for each class of non-semicolon Arabic letters

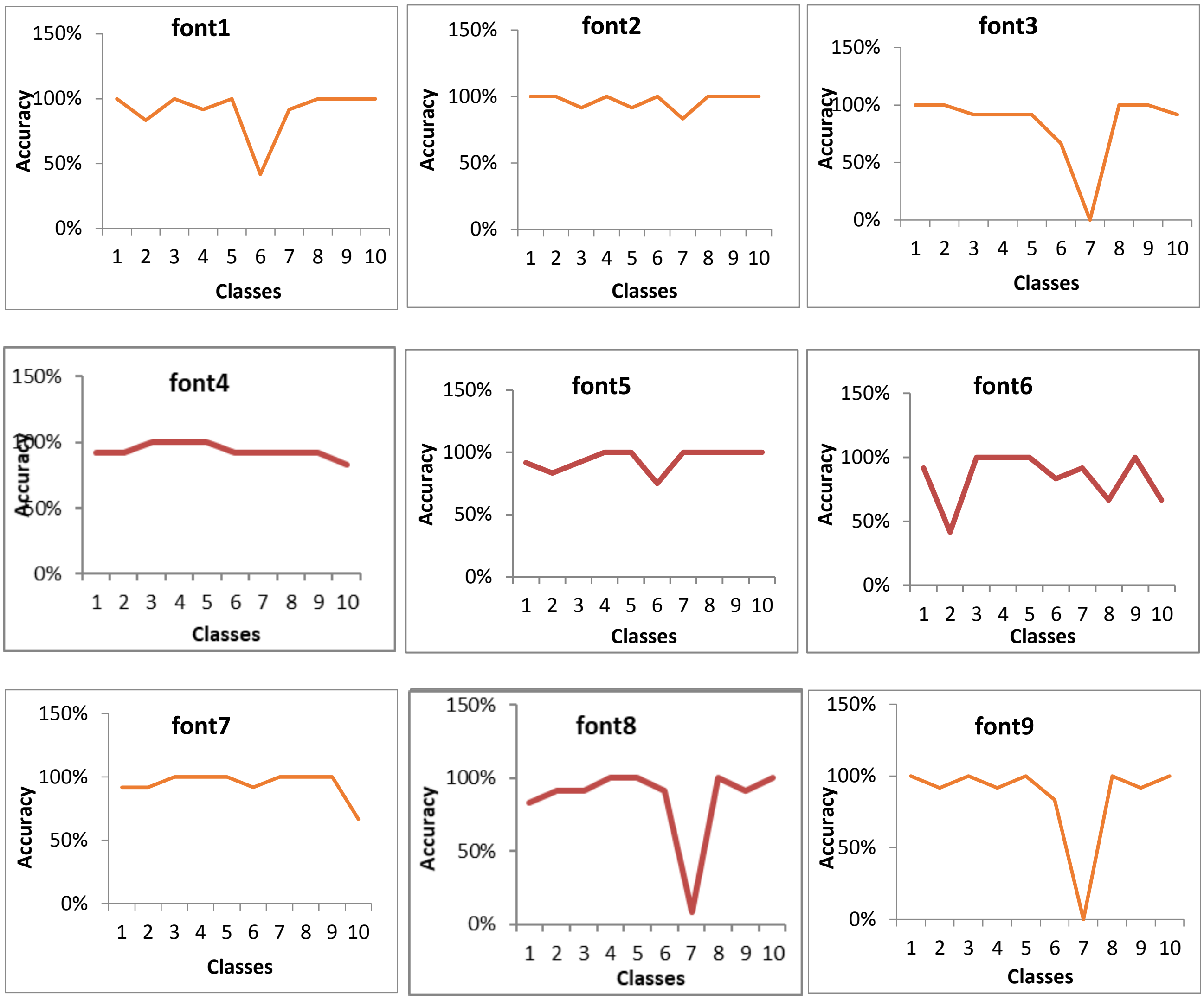

Figure 5: Accuracy of Arabic font style with all classes (letters)

6. Conclusion

The final results indicate some important conclusions about the use of Arabic fonts in writing texts. The fonts no. 2, 4, 5, and 7 (Diwani Letter, AF_Al Hada, Times New Roman, and B Tanab) are preferred for writing clear Arabic text because of the high classification accuracy for all letters. The fonts i(6,10) (Arabic Typeset, Badie Font-Dima) are not preferred in writing because of the low classification accuracy of letters. The letter shapes in these fonts are either similar in shape, very small, or unusually drawn. Classification accuracy for the letters Meem, Laam, and Seen is very high and stable. In all Arabic fonts, for some sizes of fonts, the classification accuracy is inaccurate and low for the letters Saad and Raa. It is preferable to use the font size (14–72) in Arabic writing since the classification accuracy is 100% for all Arabic fonts and letters under study. Avoid writing using sizes 8–12 because the classification accuracy is low and, in many cases, the letter shape is not distinguished or unusual. For future work, all Arabic language letters have been studied, and a classification model has been created to classify words in Arabic.

7. Acknowledgments

This work was supported by the Department of Physics, College of Science, Mustansiriyah University, Baghdad, Iraq, for its support in the present work.

References

[1] Lukasik, Edyta, Tomasz Zientarski, "Comparative analysis of selected programs for optical text recognition," *Journal of Computer Sciences Institute*, vol. 7, no. 1 pp. 191-194, 2018.

[2] Anil, R., et al. "Convolutional neural networks for the recognition of Malayalam characters," *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and*

Applications FICTA, Springer, Cham, 2015.

[3] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *nature, vol.* 521, no. 7553 pp. 436-444, 2015.

[4] Gajoui, and Mohammed Oumsis, "Diacritical Language OCR based on neural network: Case of Amazigh language," *Procedia computer science, vol* 73, pp. 298-305, 2015.

[5] Yang, Sixiong, Bin Wu, and Wenzhe Wang, "Improved NMS Filter of Similar Categories for Road Damage Detection," *Proceedings of the 3rd International Conference on Vision, Image and Signal Processing*, 2019.

[6] Rawat, Waseem, and Zenghui Wang, "Deep convolutional neural networks for image classification: A comprehensive review, "*neural computation*, vol. 29, no. 9, pp. 2352-2449, 2017.

[7] Shin, Hoo-Chang, et al, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning, "*IEEE transactions on medical imaging, vol.* 35, no. 5, pp. 1285-1298, 2016.

[8] Radwan, Mohamed A., and Hazem M. Abbas, "Predictive segmentation using multichannel neural networks in Arabic OCR system," *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*. Springer, Cham, 2016.

[9] Slimane, Fouad, et al, "Database and evaluation protocols for Arabic printed text recognition," *DIUF-University of Fribourg-Switzerland* 1, 2009.

[10] AlKhateeb, Jawad H., et al, "Offline handwritten Arabic cursive text recognition using Hidden Markov Models and re-ranking," *Pattern Recognition Letters* , vol. 32, no. 8, pp. 1081-1088, 2011.

[11] Radwan, Mohamed A., Mahmoud I. Khalil, and Hazem M. Abbas, "Neural networks pipeline for offline machine printed Arabic OCR," *Neural Processing Letters, vol.* 48, no. 2, pp. 769-787, 2018.

[12] Osman, Hussein, et al, "An Efficient Language-Independent Multi-Font OCR for Arabic Script," *arXiv preprint arXiv, 2009-09115*, 2020.

[13] F. A. Abdulghani and N. A. Abdullah, "A Survey on Arabic Text Classification Using Deep and Machine Learning Algorithms," *Iraqi Journal of Science*, vol. 63, no. 1, pp. 409–419, Jan. 2022.

[14] R. L. Priya and S. V. Jinny, "Elderly Healthcare System for Chronic Ailments using Machine Learning Techniques – a Review," *Iraqi Journal of Science*, vol. 62, no. 9, pp. 3138–3151, Sep. 2021.

[15] Olawoyin, Anifat M., Carson K. Leung, and Alfredo Cuzzocrea, "Open Data Lake to Support Machine Learning on Arctic Big Data," *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021.

[16] Sarkhel, Ritesh, et al, "A multi-objective approach towards cost effective isolated handwritten Bangla character and digit recognition," *Pattern Recognition, vol.* 58, pp. 172-189, 2016.

[17] Wu, J, "Introduction to convolutional neural networks, National Key Lab for Novel Software Technology, "*Nanjing Univ, China*, vol. 5, pp. 23, 2017.

[18] GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron: *Deep learning*, MIT press, 2016.

[19] Ioffe, Sergey, and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *International conference on machine learning*, PMLR, 2015.

[20] El-Sawy, Ahmed, Mohamed Loey, and Hazem El-Bakry, "Arabic handwritten characters recognition using convolutional neural network," *WSEAS Transactions on Computer Research,* vol. 5, no. 1, pp. 11-19, 2017.

[21] Vakili, Meysam, Mohammad Ghamsari, and Masoumeh Rezaei, "Performance analysis and comparison of machine and deep learning algorithms for IoT data classification," *arXiv preprint arXiv:2001.09636,* 2020.

[22] KHOSA, Charanjit K., et al, "Convolutional neural networks for direct detection of dark matter," *Journal of Physics G: Nuclear and Particle Physics*, vol. 47, no. 9, pp. 095201, 2020.

[23] Gallus, Michal, and Alberto Nannarelli, "Handwritten digit classification using 8-bit floating point based convolutional neural networks," *DTU Compute Technical Report* , vol. l, no. 1, 2018.