

# Deep ensemble neural networks for recognizing isolated Arabic handwritten characters

Haifa Alyahya<sup>1\*</sup>, Mohamed Maher Ben Ismail<sup>2</sup> and AbdulMalik Al-Salman<sup>3</sup>

Lecturer, Department of Computer Science, King Saud University, Saudi Arabia<sup>1</sup>

Associate Professor, Department of Computer Science, King Saud University, Saudi Arabia<sup>2</sup>

Professor, Department of Computer Science, King Saud University, Saudi Arabia<sup>3</sup>

Received: 20-August-2020; Revised: 25-October-2020; Accepted: 05-November-2020

©2020 Haifa Alyahya et al. This is an open access article distributed under the Creative Commons Attribution (CC BY) License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## Abstract

*In recent years, handwritten character recognition has become an active research field. In particular, digitalization has triggered the interest of researchers from various computing disciplines to address several handwriting related challenges. Despite these efforts, there are still opportunities for the development and improvement of the recognition of the handwritten Arabic letters. In this paper, we designed and developed a deep ensemble architecture in which ResNet-18 architecture is exploited to model and classify character images. Specifically, we adapted ResNet-18 by adding a dropout layer after all convolutional layer and integrated it in multiple ensemble models to automatically recognize isolated handwritten Arabic characters. A standard Arabic Handwritten Character Dataset (AHCD) was used in the experiments to train and assess all the proposed models. Satisfactory results were obtained using all models. The best-attained accuracy was 98.30% using a typical ResNet-18 model. Similarly, 98.00% and 98.03% accuracies were obtained using an ensemble model with one fully connected layer (1 FC) and an ensemble with two fully connected layers (2 FC) coupled with a dropout layer, respectively.*

## Keywords

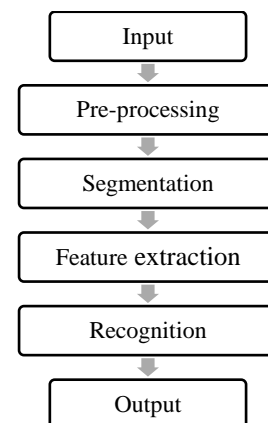
*Handwriting character recognition, Arabic, OCR, Online recognition, Handwritten recognition.*

## 1.Introduction

The widespread use of the Arabic language and the vast number of Arabic books and documents in several Arab and international countries has promoted researchers' interest in handwritten characters recognition [1]. In particular, these efforts were mainly intended to digitalize a large number of historical Arabic documents and ease the exploitation of the knowledge they enclose.

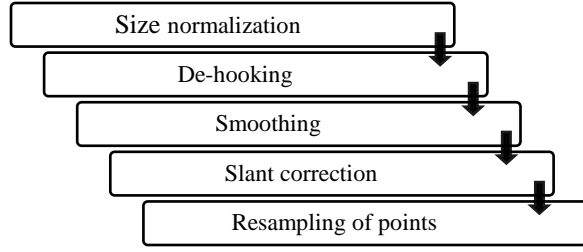
The Arabic Handwritten Character Recognition (AHCR) problem has been formulated as a typical pattern recognition problem. Specifically, as depicted in *Figure 1*, the earliest AHCR systems include components shared by various image-based pattern recognition applications such as medical image classification, cancer detection, biometric, etc. The main purpose of the pre-processing phase is to eliminate the effectiveness of noise, writing size, style differences, and duplicated points [2].

The five main steps of pre-processing are size normalization, de-hooking or interpolating missing points, smoothing, slant correction or duplicated point removal, and resampling of points, as shown in *Figure 2*.



**Figure 1** Main steps of a typical character recognition systems

\*Author for correspondence



**Figure 2** Preprocessing phase

The segmentation phase represents the process of partitioning words or sub-word into characters, strokes, or other units. It is divided into two main approaches; holistic and analytical approaches. One of the critical stages in handwriting recognition systems is the feature extraction which can be defined as the process of capturing the visual properties of the letter, such as shape attributes, pixels, and other information passed to the recognizer. In different languages, the written text

can be extracted using three types of features, statistical features, structural features and feature space transformation [3]. Finally, the recognition phase consists typically of shallow supervised machine learning models intended to map the extracted feature vectors into some pre-defined classes [4].

Despite the researchers' efforts, the recognition of Handwritten Arabic Script remains a challenging task due to the high variability of the Arabic characters. In fact, Arabic words are written from right to left. The letters are cursive. Moreover, some letters consist of more than one body (أ, إ, ك, ز, ئ). In addition, many letters have different numbers of dots with different positions in the Arabic alphabets, as shown in *Table 1*.

**Table 1** Dots number and position

Number of Dots	Letters	Position
Single dot	ب, ج, ح, ذ, ز, ض, ظ, ف, غ, ن	Above/middle/below
Double dots	ت, ق, ي	Above/below
Triple dots	ش, ث	Above

This made the Arabic Handwriting recognition one of the most active research areas. Especially, researchers from machine learning and image processing communities showed real interest in addressing the various problems faced by this application. One should note that the handwriting styles of different individuals made the problem more acute. Notably, the performance of typical HCR solutions is drastically affected by the significant variance of the letter's shapes, sizes and patterns [5].

The choice of the relevant features to encode the visual properties of the Arabic Handwritten Characters is a critical task due to the significant variation depending on the individuals writing styles. To overcome this challenge, an automated approach that can autonomously learn the visual descriptors and capture the letter relevant details might be deployed. Deep learning techniques can accomplish this by learning the most effective and classify them.

This led different researchers to use deep learning in their systems. In fact, different models and architectures have been used to recognize Handwritten Arabic Characters, digits and words [6]. Despite the researchers' efforts and the promising results, there is still room for improvement,

explicitly, after the emergence of successful deep learning architectures, such as ResNet[7], GoogleNet[8], VGGNet[9], AlexNet [10], ... etc.

The objectives of this research can be summarized as: (i) Survey the Arabic handwritten character recognition systems, (ii) Design convolutional neural networks (CNNs) [11] models to address the isolated Arabic handwritten character recognition problem, and (iii) Use a standard dataset and performance measures to validate and assess the proposed system.

The rest of the paper is organized as follows: the related works of the HCR and Arabic HCR that have used the Arabic Handwritten Character Dataset (AHCD) discussed in Section 2. Section 3 explains the proposed approach for our experiments. Then, the experimental results are presented in Section 4. Finally, Section 5 provided the conclusion and future works.

## 2.Related works

The recent progress in handwriting recognition exploited deep learning performance to recognize characters and digits in different languages. The authors in [12] addressed the Chinese handwriting character recognition problem using a deep

convolutional network. Their experiments were done with various filter numbers and several depths by using 200 and 3755 class datasets. They concluded that the best performance was reached using more filters and deeper network. The researchers in [13] adopted the ResNet-18 model [7] and used it to recognize Bangla handwritten characters. In particular, the original ResNet-18 yielded 94.52% classification accuracy, while the modified ResNet-18 achieved 95.10%. In [14], a new CNN based system was proposed for Amharic character recognition. The dataset used by the authors includes 132,500 Amharic characters. The accuracies achieved were 91.83% and 90.47% on the training set and validation set, respectively.

The researchers in [15] investigated the performance of some pre-trained models. Namely, they deployed DenseNet-201 [16], DenseNet-121 [16], Vgg-11 [9], Vgg-16 [9], Vgg-19 [9], AlexNet [10] and Inception V3 [15]. They trained these models using Devanagari Handwritten Character Dataset (DHCD) [17]. The Inception model overtook the other models and attained 99% as accuracy. AlexNet achieved 98% accuracy for approximately two minutes running time, while the inception model requires 16 minutes per epoch. The authors in [18] proposed a CNN-based model to recognize Arabic handwritten letters. They collected their own dataset, which contains 16800 images divided into two sets. The first one contains 13440 images for training, while the second set includes 3360 images for testing. The obtained accuracy on the test data attained 94.9%. In [19], two Arabic databases, AIA9K [20] and AHCD [18], were coupled with a convolutional neural network-based system for Arabic handwritten character classification. They trained their system using some new regularization, different learning rate and optimization techniques. It was noted that batch normalization and dropout techniques were used to enhance the overall system performance and reduce the overfitting. In addition, the two datasets were divided into three sets: training, validation and testing. The maximum accuracy reached for AHCD

dataset was 94.7%, while it attained 94.8% for AIA9K dataset. Finally, a data augmentation was performed for AHCD dataset, and the accuracy increased to 97.4%.

Similarly, a CNN based architecture consisting of four convolution layers, 4 Rectified Linear Units (ReLU), two max pooling layers and three fully connected layers was investigated in [21]. The average accuracy was 97.2% using the testing images set of AHCD. Recently, a typical CNN based architecture was implemented in [22] for Arabic Handwritten Character recognition. The authors selected 13,440 images randomly to train the provided CNN model while the remaining images were used for testing. However, in criticizing the model the experiments were not detailed enough to show all the steps. It was roughly reported that the accuracy exceeds 98%.

Another convolutional neural network-based model for handwriting recognition was outlined in [23]. It is an automatic model that can be used for offline recognition of Arabic Handwriting. This model was trained using Hijja [23] and AHCD [18] datasets, which were divided into three subsets with ratios 60% for training, 20% for validation and 20% for testing. The trained models achieved 88% accuracy using Hijja dataset [23] versus 97% using AHCD dataset [18].

Table 2 summarizes the relevant related works which used the standard AHCD [18] dataset along with deep learning-based solutions to address the Arabic Handwritten character recognition problem. As illustrated in [19], the testing accuracy was the lowest with 94.7%, but the results were improved to reach 97.4% by using data augmentation. The best accuracy, 98%, was attained using the CNN based architecture in [22]. However, in [22], the authors used training and testing sets with no validation set. This made their work prone to overfitting.

**Table 2** Summary of relevant related works that used the standard AHCD [18] dataset along with deep learning-based solutions

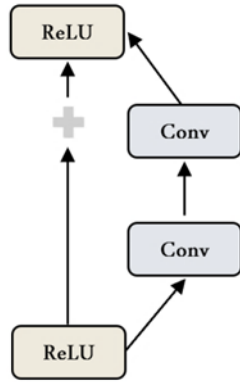
References & publication year	Data sets partitions	Methods	Testing accuracy
[18] 2017	13440 Training images 3360 Testing images	CNN	94.9%
[19] 2017	13440 Training images 3360 Testing images	CNN models with regularization parameters	94.7 % With data augmentation 97.4%

References & publication year	Data sets partitions	Methods	Testing accuracy
[21] 2019	13440 Training images 3360 Testing images	Modified CNN architecture	97.2%
[22] 2019	13440 Training images 3360 Testing images	Proposed CNN architecture.	98.0%
[23] 2020	60% Training 20% Validation 20% Testing	Deep neural network	97%

### 3. Proposed approach

#### 3.1 Residual Network (ResNet)

ResNet is a pre-trained deep learning neural network which can exhibit various sizes of deep layers. It is a traditional feedforward network that has several blocks of residual connection [24]. It was used for several pattern recognition applications and proved to be highly effective. In this work, we rely on ResNet-18 [7] to build our customized model. The proposed architecture is intended to enhance the accuracy of handwritten Arabic letters recognition. ResNet-18 consists of 18 deep convolutional neural network layers. As illustrated in *Figure 3*, the proposed ResNet model relies on a combination of Convolution and Rectified Linear Units (ReLU) layers.



**Figure 3** Basic block diagram of ResNet architecture

#### 3.2 Architecture

As Deep learning architecture, we propose an ensemble model based on the original ResNet-18 and modified ResNet-18 with different fully connected layer configuration. Specifically, for the modified module, a dropout layer with 0.2 rate was added after all convolutional layers.

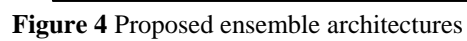
The two proposed architectures are: (i) An ensemble of a typical ResNet-18 and a modified ResNet-18 with one fully connected, and (ii) A similar ensemble where 2 FC are added to both, the original and the modified ResNet-18. *Figure 4* illustrates the detailed Architectures of the proposed ensemble models with 1 FC and an ensemble with 2 FC.

The rationale behind the proposed deep learning-based ensemble approach is the exploitation of the individual deep network's diversity for a more accurate decision. In other words, it is meant to fuse the features learned by both networks in order to improve the classification performance.

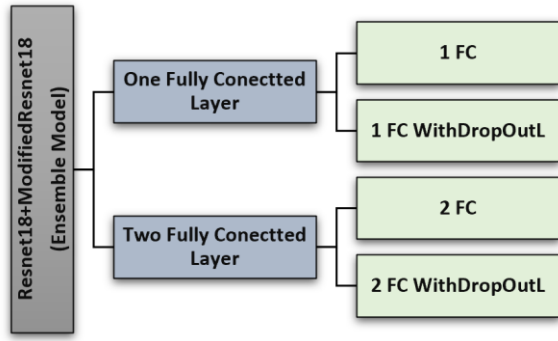
The main models also are divided into two main parts; with and without dropout layer before the fully connected layers, as shown in the categories of the proposed models depicted in *Figure 5*.

In the proposed model, first we divided the data into three sets: training set, validation set and testing set. The testing set which has not been seen by the system before is the most important set to evaluate the system, so that the results are guaranteed and accurate. After that, the images are improved and increased by the pre-processing step and data augmentation. Then, we started train the proposed model that shown in the previous *Figure 4*. Finally, the trained CNN model is saved and tested using validation set and testing set to evaluate the system. All the steps of the proposed recognition system are shown in *Figure 6*.

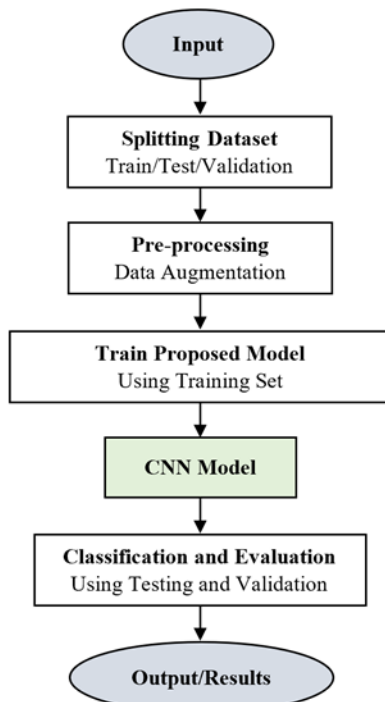
The images in the input layer have been resized to fit the size of the images in the ACHD. Then, we added the dropout layer after all conv layers to reduce overfitting. Finally, we trained the dataset with a different number of epochs and mini-batch sizes.







**Figure 5** Summary of the models proposed in this research



**Figure 6** The proposed recognition system

### 3.3 Data augmentation

In this work, we performed data augmentation as a strategy to extend the dataset and increase the number of images and their variability without the need to collect more data [15]. This data augmentation addresses the overfitting problem from the root (training phase) and improves the deep learning model generalization. Hence, it assumes that more information can be captured from the original image collection through simple operations.

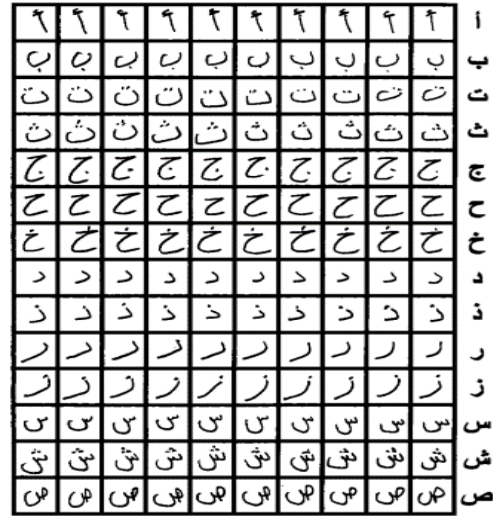
For the Arabic Handwritten Character Dataset (AHCD), we varied the character directions. The image probability was reflected with 50%

horizontally and vertically. We set the range of image rotation to be within the range  $[-30\ 30]$ .

## 4. Experiments

### 4.1 Dataset

For the study we used the Arabic Handwritten Character Dataset (AHCD) [18] to validate and assess the performance of the proposed system. This dataset has been commonly used by most handwriting recognition researchers to recognize isolated Arabic handwritten characters. *Figure 7* shows a sample of the Arabic letters from AHCD, which were collected by 60 participants aged 19 to 40. As can be seen in *Figure 7*, each row represents ten tentative writings of a given letter.



**Figure 7** Data collection for Arabic characters [18]

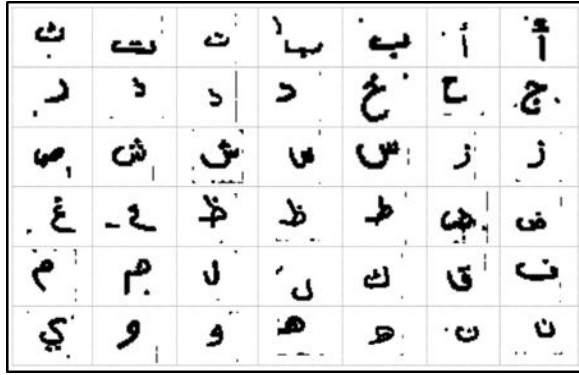
This dataset is composed of a training set and a testing set. The training set consists of 13440 images, where each letter is represented using 480 images. Similarly, the testing set includes 3,360 images, where each letter is represented using 120 images. This makes a total of 16800 images for all dataset characters.

Furthermore, in the context of the proposed deep learning models, we divided the dataset into three sets: training, validation, and testing sets. This would naturally avoid overfitting the data, especially when tuning the model parameters. In other words, the testing set was not used to train the model or tune its parameters. The partition adopted is reported in *Table 3*.

**Table 3** Data partition used in the experiments

Subset	Proportion	# Images
Training Set	60%	10080
Validation Set	20%	3360
Testing Set	20%	3360

The initial exploration of the dataset showed that 1023 letters that have similar stroke were wrongly labeled. For instance, the letter "Zai" was labeled as "Raa", and the letter "Taa" was labeled as "Thaa" and so on. Additionally, as preprocessing, we deployed some image enhancement operations to improve the quality of the letter images. *Figure 8* shows a sample of noisy images from the dataset prior to the preprocessing phase.

**Figure 8** Sample noisy images

Similarly, *Figure 9* illustrates the color inversion task conducted as a preprocessing step to all images.



(a)



(b)

**Figure 9** Sample of character images: (a) Before and (b) After colour inversion

#### 4.2 Models parameters

The proposed models have too many layers illustrated in *Table 4*. We trained the network using a different number of epochs and mini-batch sizes, as shown in *Table 4*, which significantly affected the results. After several tests, the best accuracies were obtained with 16 batch sizes and 100 epochs.

**Table 4** Models parameters

Models	1 FC	1 FC With DropOutL	2 FC	2 FC With DropOutL
Layers	160	161	161	163
# Epochs	256, 128, 64, 32, 16, and 8			
miniBatchSize	100, 40, 30, and 15			
Optimization	Gradient descent			
Momentum	0.9			
LearnRateDropFactor	0.1			

#### 4.3 Computing resources

Our experiments were conducted using a customized desktop computer with an intel core i9-7960x skylake-x processor, overclock 4100mhz, 32GB of RAM and 2x NVIDIA GeForce RTX 2080 Ti NVLink GPUs.

#### 4.4 Performance measures

Traditional evaluation and performance measures were used to evaluate our models such as accuracy,

error rate, precision, recall and specificity calculated using the following equations.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$\text{Error Rate} = \frac{FP}{\text{Total number of test images}} \quad (2)$$

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

$$\text{Recall} = \frac{TP}{TP+FN} \quad (4)$$

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (5)$$

Where TP, TN, FP and FN in the confusion matrix denote true positive, true negative, false positive, and false negative, respectively.

## 5. Results and discussion

As outlined above in this research, we investigated several ResNet-18 based deep models in order to address the Handwritten Arabic Character recognition problem. Namely, we implemented and deployed the typical Resnet-18, ensemble models based on 1 FC with/without a dropout layer, and ensemble models based on 2 FC with and without a dropout layer.

Table 5 reports the training performance measures obtained using the different models implemented and deployed to solve the Handwritten Arabic characters' recognition on the original and the pre-processed dataset. As expected, all models achieved considerably high accuracy and precision. This

proves that all models succeeded to fit both training datasets.

Similarly, Table 6 reports the results obtained using the considered models along with the validation sets extracted from the original and the pre-processed dataset. As one can see, the ensemble models based on 1 FC or 2 FC layers with a dropout layer outperform the other models slightly in terms of all performance measures. This proves that adding the dropout layer ensures better generalization of the model and prevent overfitting. This finding holds for both, the original and the pre-processed dataset.

The obtained results proved that the pre-processed dataset led to better results than the original dataset, especially with the training and validation set, as shown in Table 5 and Table 6. Further investigation revealed that the proposed models faced some difficulty in recognizing the character "Sheen", especially with the pre-processed dataset the average accuracies of the testing set was 69.6%. It was misclassified a considerable number of times, as reported in Table 8.

**Table 5** Training performance obtained using the original and the pre-processed datasets

Training set						
Model	Dataset	Acc	Error	Precision	Recall	Specificity
Resnet-18	Original	0.9972	0.0028	0.9972	0.0001	0.9999
	Modified	0.9984	0.0016	0.9985	0.0001	0.9999
1 FC	Original	0.9956	0.0044	0.9958	0.0002	0.9998
	Modified	0.9979	0.0021	0.9980	0.0001	0.9999
1 FC With DropOutL	Original	0.9943	0.0058	0.9943	0.0002	0.9998
	Modified	0.9978	0.0022	0.9979	0.0001	0.9999
2 FC	Original	0.9969	0.0031	0.9969	0.0001	0.9999
	Modified	0.9983	0.0017	0.9984	0.0001	0.9999
2 FC With DropOutL	Original	0.9961	0.0039	0.9962	0.0001	0.9999
	Modified	0.9977	0.0023	0.9978	0.0001	0.9999

**Table 6** Validation set performance obtained using the original and the pre-processed datasets

Validation set						
Model	Dataset	Acc	Error	Precision	Recall	Specificity
Resnet-18	Original	0.9571	0.0429	0.9584	0.0016	0.9984
	Modified	0.9580	0.0420	0.9593	0.0016	0.9985
1 FC	Original	0.9542	0.0458	0.9557	0.0017	0.9983
	Modified	0.9592	0.0408	0.9610	0.0015	0.9985
1 FC With DropOutL	Original	0.9622	0.0378	0.9633	0.0014	0.9986
	Modified	0.9631	0.0369	0.9642	0.0014	0.9986
2 FC	Original	0.9551	0.0449	0.9562	0.0017	0.9983
	Modified	0.9592	0.0408	0.9610	0.0015	0.9985
2 FC With DropOutL	Original	0.9616	0.0384	0.9626	0.0014	0.9986
	Modified	0.9610	0.0390	0.9626	0.0014	0.9986



**Table 7** Testing set performance obtained using the original and the pre-processed datasets

Testing set						
Model	Dataset	Acc	Error	Precision	Recall	Specificity
Resnet-18	Original	0.9792	0.0208	0.9792	0.0008	0.9992
	Modified	0.9708	0.0292	0.9720	0.0011	0.9989
1 FC	Original	0.9753	0.0247	0.9758	0.0009	0.9991
	Modified	0.9667	0.0333	0.9682	0.0012	0.9988
1 FC With DropOutL	Original	0.9744	0.0256	0.9748	0.0010	0.9991
	Modified	0.9729	0.0271	0.9743	0.0010	0.9990
2 FC	Original	0.9753	0.0247	0.9756	0.0009	0.9991
	Modified	0.9696	0.0304	0.9712	0.0011	0.9989
2 FC With DropOutL	Original	0.9768	0.0232	0.9770	0.0009	0.9991
	Modified	0.9688	0.0313	0.9702	0.0012	0.9988

**Table 8** Letters average accuracies obtained using pre-processed dataset

Letter	Validation set	Testing set	Letter	Validation set	Testing set
Alf	92.50%	95.2%	Dadd	97.84%	97.2%
Baa	98.00%	99.0%	Tah	97.00%	99.0%
Taa	93.32%	99.7%	Thah	93.84%	98.8%
Thaa	97.00%	96.3%	Ayn	97.00%	97.5%
Jeem	95.68%	98.3%	Ghyn	94.98%	98.9%
Haa	99.18%	98.7%	Faa	96.34%	98.0%
Khaa	91.82%	97.3%	Qaf	87.66%	95.8%
Dal	97.00%	98.8%	Kaf	97.80%	99.2%
Thal	93.16%	96.7%	Lam	97.18%	99.2%
Raa	97.84%	95.2%	Meem	99.02%	98.8%
Zai	93.30%	99.5%	Noon	96.34%	97.0%
Seen	97.84%	99.2%	Ha	99.18%	98.5%
Sheen	100%	69.6%	Waw	96.82%	97.5%
Sadd	97.18%	98.3%	Yaa	93.50%	98.2%

Therefore, we resorted to another way to improve the results by utilizing the trained models using the original data and test it with the modified testing set. As reported in *Table 9*, this yielded 98.30%, 98.00% and 98.03% as accuracies attained using the original ResNet-18, the ensemble with 1 FC and ensemble with 2 FC with dropout layer, respectively.

The chart in *Figure 10* reports the accuracies of all models that used the original training set to train the model and the modified testing set to test it. Resnet-18 and 2 FC with dropout layer gave the best results with accuracy higher than 98%. These results were impressive when we compare it with the other works that done using similar dataset. The model of the latest study [23] in 2020 achieved 97% accuracy. The

testing accuracy was the highest with 98% that attained using the CNN based architecture in [22]. But the authors achieved this result using training and testing sets with no validation set, so the overfitting not taken in considerable. The lowest accuracies were 95.3% in [25], 97% in [23] and 97.2% in [21].

*Table 10* shows the accuracy of our proposed ensemble models and typical Resnet-18 compared with the previous works. As shown in *Table 10*, the best model was a typical ResNet-18 which achieved 98.30% accuracy. Also, we obtained 98.01% and 98.03% accuracies using an ensemble model with 1 FC and an ensemble with 2 FC coupled with a dropout layer, respectively.

**Table 9** The performance of the considered models attained using the original training set and the pre-processed testing set

Trained model with original training set & modified testing set						
Model	Dataset	Accuracy	Error	Precision	Recall	Specificity
Resnet-18	Original Training	0.9972	0.0028	0.9972	0.0001	0.9999
	Modified Testing	0.9830	0.0170	0.9835	0.0006	0.9994

Trained model with original training set & modified testing set						
1 FC	Original Training	0.9956	0.0044	0.9958	0.0002	0.9998
	Modified Testing	0.9801	0.0199	0.9806	0.0007	0.9993
1 FC With DropOutL	Original Training	0.9943	0.0058	0.9943	0.0002	0.9998
	Modified Testing	0.9792	0.0208	0.9797	0.0008	0.9992
2 FC	Original Training	0.9969	0.0031	0.9969	0.0001	0.9999
	Modified Testing	0.9777	0.0223	0.9780	0.0008	0.9992
2 FC With DropOutL	Original Training	0.9961	0.0039	0.9962	0.0001	0.9999
	Modified Testing	0.9804	0.0196	0.9806	0.0007	0.9993

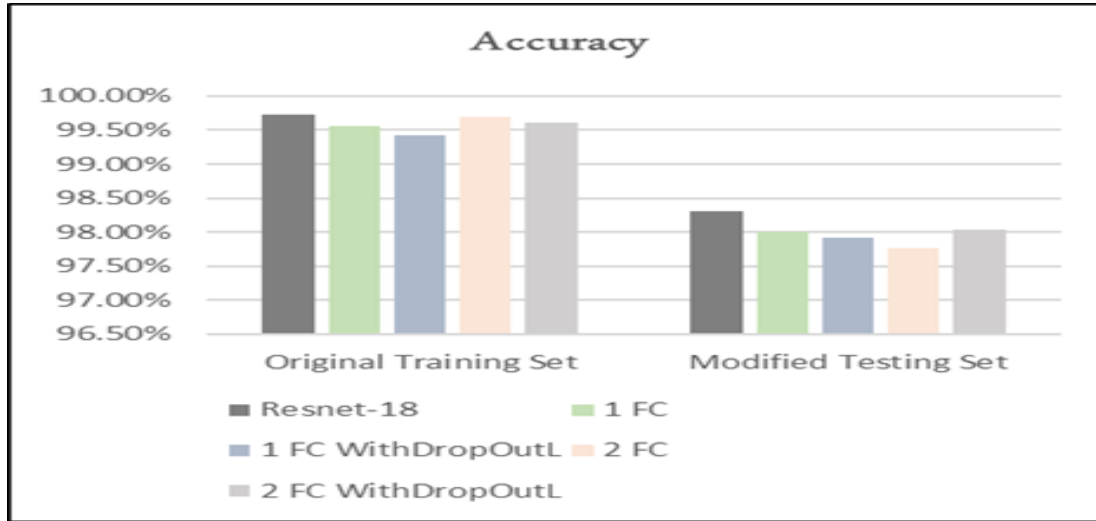


Figure 10 Accuracies attained by models implemented in this research

Table 10 Comparison of testing accuracy of our proposed models with the previous models

References & publication year	Data sets partitions	Testing accuracy	
[21] 2019	13440 Training images 3360 Testing images	97.2%	
[22] 2019	13440 Training images 3360 Testing images	98.0%	
[25] 2019	13440 Training images 3360 Testing images	95.3%	
[23] 2020	60% Training 20% Validation 20% Testing	97%	
Proposed Models	60% Training 20% Validation 20% Testing	<i>esnet-18</i>	98.30%
		<i>1 FC</i>	98.01%
		<i>2 FC With DropOutL</i>	98.04%

These results confirmed that the representation depth of the character image is valuable for their recognition performance, and that state-of-the-art results can be attained using ensemble models based on pre-trained models in which the architecture exhibits a substantially increased depth. Namely, ResNet based models is generalized well to the Handwritten Arabic Character recognition task and datasets. Moreover, they outperform more complex

recognition pipelines built around less deep character image representations. This confirms the importance of depth in visual representations. Besides, the dropout layers inserted in the proposed architectures ensure the model regularization, which addresses the overfitting challenge. Additionally, such dropout layers yield a more relevant and independent feature learning by avoiding the neurons co-adaptations.

## 6. Conclusion and future work

In this work, we addressed the feature extraction problem along with the choice of the most appropriate classifier to recognize Handwritten Arabic Character by using Deep Learning models. Typically, Convolutional Neural Networks are intended to learn the most relevant features over their numerous layers. This overcomes the challenge of choosing the best feature for Handwritten Arabic Character recognition. Particularly, we designed and developed a deep ensemble architecture in which ResNet-18 architecture is exploited to model and classify character images. AHCD was used in the experiments to train and assess all the proposed models. Satisfactory results were obtained using all models. The best-attained accuracy, 98.30%, was obtained using a typical ResNet-18 model. Similarly, 98.00% and 98.03% accuracies were obtained using an ensemble model with 1 FC and an ensemble with 2 FC coupled with a dropout layer, respectively.

To advance the research, other deep learning paradigms will be investigated for Arabic Handwriting recognition. In particular, the online recognition task will be tackled using Recurrent Neural Networks and appropriate time-series datasets.

## Acknowledgment

I extend my sincere thanks to my colleague Dr. Khaled Babbain for his support and assistance, as well as to my colleague Dhari Al-Abduh for allowing me to use his computer to train all the models.

## Conflicts of interest

The authors have no conflicts of interest to declare.

## References

- [1] Davis RH, Lyall J. Recognition of handwritten characters—a review. *Image and vision computing*. 1986; 4(4):208-18.
- [2] Inuganti S, Ramisetty RR. Preprocessing of online handwritten Telugu character recognition. *International Journal of Advanced and Applied Sciences*. 2017; 4(7):179-89.
- [3] Arica N, Yarman-Vural FT. An overview of character recognition focused on off-line handwriting. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*. 2001; 31(2):216-33.
- [4] Al-Salman A, Alyahya H. Arabic online handwriting recognition: a survey. In *proceedings of the 1st international conference on internet of things and machine learning 2017* (pp. 1-4).
- [5] Perwej Y, Chaturvedi A. Machine recognition of hand written characters using neural networks. *arXiv preprint arXiv:1205.3964*. 2012.

- [6] El-Sheikh TS, El-Taweel SG. Real-time Arabic handwritten character recognition. *Pattern Recognition*. 1990; 23(12):1323-32.
- [7] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In *proceedings of the IEEE conference on computer vision and pattern recognition 2016* (pp. 770-8). IEEE.
- [8] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In *proceedings of the IEEE conference on computer vision and pattern recognition 2015* (pp. 1-9).
- [9] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. 2014.
- [10] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In *advances in neural information processing systems 2012* (pp. 1097-105).
- [11] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. 1998; 86(11):2278-324.
- [12] Zhang Y. Deep convolutional network for handwritten Chinese character recognition. *Computer Science Department, Stanford University*. 2015.
- [13] Alif MA, Ahmed S, Hasan MA. Isolated Bangla handwritten character recognition with convolutional neural network. In *international conference of computer and information technology 2017* (pp. 1-6). IEEE.
- [14] Abdurahman F. Handwritten Amharic character recognition system using convolutional neural networks. *ICL 2020*.
- [15] Aneja N, Aneja S. Transfer learning using CNN for handwritten Devanagari character recognition. In *international conference on advances in information technology 2019* (pp. 293-6). IEEE.
- [16] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In *proceedings of the IEEE conference on computer vision and pattern recognition 2017* (pp. 4700-8).
- [17] Acharya S, Pant AK, Gyawali PK. Deep learning based large scale handwritten Devanagari character recognition. In *international conference on software, knowledge, information management and applications 2015* (pp. 1-6). IEEE.
- [18] El-Sawy A, Loey M, El-Bakry H. Arabic handwritten characters recognition using convolutional neural network. *WSEAS Transactions on Computer Research*. 2017; 5:11-9.
- [19] Younis KS. Arabic handwritten character recognition based on deep convolutional neural networks. *Jordanian Journal of Computers and Information Technology*. 2017; 3(3):186-200.
- [20] Torki M, Hussein ME, Elsallamy A, Fayyaz M, Yaser S. Window-based descriptors for arabic handwritten alphabet recognition: a comparative study on a novel dataset. *arXiv preprint arXiv:1411.3519*. 2014.
- [21] Najadat HM, Alshboul AA, Alabed AF. Arabic handwritten characters recognition using convolutional neural network. In *international*

conference on information and communication systems 2019 (pp. 147-51). IEEE.

- [22] Elmiad AK. Arab handwriting character recognition using deep learning. In international conference europe middle east & north africa information systems and technologies to support learning 2019 (pp. 410-5). Springer, Cham.
- [23] Altwaijry N, Al-Turaiki I. Arabic handwriting recognition system using convolutional neural network. Neural Computing and Applications. 2020:1-3.
- [24] Alom MZ, Taha TM, Yakopcic C, Westberg S, Sidike P, Nasrin MS, E et al. A state-of-the-art survey on deep learning theory and architectures. Electronics. 2019; 8(3):1-66.
- [25] Almansari OA, Hashim NN. Recognition of isolated handwritten arabic characters. In international conference on mechatronics engineering 2019 (pp. 1-5). IEEE.

**Haifa Alyahya** is a lecturer at Hail University. She obtained her Bsc. in Software Engineer from Hail University. and Msc. in Computer Science from King Saud University. Her research area covers Deep learning and Image Processing.  
Email: h.yahya@uoh.edu.sa



**Mohamed Maher Ben Ismail** is an associate professor at the Computer Science department of the College of Computer and Information Sciences at King Saud University. He received his PhD. degree in Computer Science from the University of Louisville in 2011. His research interests include Pattern Recognition, Machine Learning, Data Mining and Image Processing.



**Abdulmalik S. Al-Salman**, a full professor at Computer Science Department, King Saud University, Riyadh, Saudi Arabia. He received his Ph. D. from Computer and Information Sciences, Oklahoma State University, OK, USA in 1996; Master from Computer Science, Georgia University, GA, USA in 1992; BS from Computer Science, King Saud University, Saudi Arabia in 1988. He served as the head of the Computer Science Department, then the vice dean of the Computing College, then the General Secretary of the Scientific Council at King Saud University. He has more than 130 publications. His current research interests are Arabic Natural language Processing, Assistive technologies, and mobile applications.