RECOGNITION OF ARABIC HANDWRITTEN CHARACTERS USING RESIDUAL NEURAL NETWORKS

Ahmad T. Al-Taani* and Sadeem T. Ahmad

(Received: 8-Mar.-2021, Revised: 8-May-2021, Accepted: 22-May-2021)

ABSTRACT

This study proposes the use of Residual Neural Networks (ResNets) to recognize Arabic offline isolated handwritten characters including Arabic digits. ResNets is a deep learning approach which showed effectiveness in many applications more than conventional machine learning approaches. The proposed approach consists of three main phases: pre-processing phase, training the ResNet on the training set and testing the trained ResNet on the datasets. The evaluation of the proposed approach is performed on three available datasets: MADBase, AIA9K and AHCD. The proposed approach achieved accuracies of 99.8%, 99.05% and 99.55% on these datasets, respectively. It also achieved a validation accuracy of 98.9% on the constructed dataset based on the three datasets.

KEYWORDS

Residual networks, Deep learning, Deep neural networks, Arabic handwritten characters, Characters recognition.

1. INTRODUCTION

Optical Character Recognition (OCR) is an electronic conversion of images of printed/handwritten text into computer-encoded text. Handwritten OCR system is divided into online and offline recognizers based on the input method. Online data is made through devices, such as tablets, computer mouse or electronic pen, while offline data is collected from scanned images of typed/handwritten documents. The recent OCR approaches are mainly applying conventional machine learning or deep learning approaches. Conventional machine learning approaches, such as Multi-layer Perceptron (MLP) and Support Vector Machine (SVM) approaches require expert engineers and specialists.

OCR is a multidisciplinary area of research in artificial intelligence, computer vision and pattern recognition. OCR is frequently used as a day of data entry from printed papers, like invoices, passport documents, mails, …etc. OCR is also a common method of digitizing printed texts, so that they can be electronically edited, searched, …etc. OCR is used in many applications, like machine translation, text mining and cognitive computing.

Processing Arabic language has lately gained attention from scholars with the increase in Arabic scholars. It became necessary all over the world and is almost digitizing massive amounts of information that is daily being processed. An example of this information comprises medical records, license plate recognition, checks to verify and old documents for digital libraries. Isolated characters have more complicated features to detect and different shapes for each letter based on the context of the character. The total number of expanded characters reaches 84 shapes that are made of the 28 basic letters.

Several studies that used deep neural networks for Arabic character recognition (Convolutional Neural Networks (CNNs), Deep Belief Networks (DBNs), Recurrent Neural Networks (RNNs)) gave promising results. Convolutional approaches automatically extract features from raw images. CNN-based architecture provides an end-to-end solution without the need to have a handcrafted-feature extraction or data representation transformation in contrast to many different conventional approaches [1]. CNN OCR systems must include four essential components: convolutional layer, pooling layer, fully connected layer and loss function that is added in the last layer. Such systems provide an effective performance by applying a drop-out of layers and control the size of the CNN.

One of the first CNN architectures was the LeNet-5 architecture which was introduced by Lecun et al.[2] and is primarily implemented for the OCR system of handwritten zip code digits by the U.S. postal

A. T. Al-Taani* and S. T. Ahmad are with the Department of Computer Science, Yarmouk University, Irbid, Jordan. Emails: ahmadta@yu.edu.jo and ahmadta@yu.edu.jo
*A. T. Al-Tanni is currently with the Department of Data Science and Artificial Intelligence, Princess Sumaya University for Technology, Amman, Jordan. Email: a.altaani@psut.edu.jo

services data and is also applied in face recognition systems. It consists of five layers, including two convolutional layers, two pooling layers and finally, a fully connected layer. Another popular architecture called AlexNet was constructed by Krizhevsky et al. [3]. It is considered one of the first networks that comprise a sort of depth. It is composed of 5 convolutional layers, 3 pooling layers and the last convolutional layer is followed by three fully connected layers.

Wu et al. [4] proposed a model called the Directly Connected Convolutional Neural Networks (DCCNNs) model. The obtained results pertaining to their proposed model are compared to the results related to the previous models, where it is proved to take a less computational time for recognition and training large images/datasets. The comparison consists of five different datasets, including the MNIST dataset for isolated handwritten digits. By using the MNIST dataset, an accuracy of 98.96% is slightly improved and is 1.3 to 1.4 times faster than the conventional CNNs.

Ashiquzzaman and Tushar [5] proposed an offline Arabic numeral recognizer by applying CNNs. They also modified the Multi-layer Perception (MLP) by applying a dropout regularization for solving the overfitting problem. The proposed approach is trained and tested on the CMATERDB 3.3.1 Arabic handwritten digit dataset. The proposed approach achieved an accuracy of 97.4%, while the modified MLP approach achieved an accuracy of 93.8%.

Mudhsh & Almodfer [6] proposed the VGGnet architecture for Arabic OCR handwritten alphanumeric characters. Their model was applied to the MADBase database with an accuracy of 99.66% and to the HACDB database with an accuracy of 97.32%.

Younis & Alkhateeb [7], Tomimori et al. [8] and Eladel et al. [9] introduced a handwritten digital recognition model by using CNNs. They modified the architecture of the network based on their own experiments and trials. Younis & Alkhateeb [7] created a simple CNN architecture for Arabic handwritten digit recognition and face recognition models. To train their model, they used MNIST dataset, which achieved an accuracy of 98.11%.

Eladel et al. [9] proposed an approach that has an impact on improving the classification accuracy and testing speed. Their results for the MNIST achieved an accuracy of 95.7% and the CIFAR-10 achieved an accuracy of 99.71%.

Younis [10] presented a deep neural network-based handwritten Arabic character recognition system. ResNet-18 architecture was applied with batch normalization for regularization and dropout to prevent overfitting. He obtained recognition accuracies of 94.8% with AHCD database and 97.6% with AIA9k.

Elleuch et al. [11] investigated the Deep Belief Neural Networks (DBNNs) approach for Arabic handwritten character/word recognition. The DBNN approach is trained and tested on the HACDB and IFN/ENIT databases. The obtained results of the two experiments showed that a 2.1% error rate resulted for characters, but for words, the error rate exceeded 40% concluding that the proposed DBNN approach is still unready to deal with high-level dimensional data.

Tagougui & Kherallah [12] proposed a model that consisted of the DBN approach and BottleNeck feature classifier for the Arabic handwritten character OCR. The LMCA database was used for training and testing. The experimental results showed that the proposed approach outperformed some previous approaches.

Karthik and Srikanta [13] proposed the Deep Belief Network (DBN) approach to recognize different handwritten Kannada characters. The experiment achieved an average accuracy of 95% by using raw pixels and an accuracy of 96.41% tested on a dataset consisting of 18,800 samples.

Recently, Mustafa and Elbashir [14] used the CNN for the recognition of Arabic names. The dataset (SUST-ARG) containing 8028 Arabic names was used for training and testing the proposed approach. Experimental results showed that the proposed approach has achieved an accuracy of 99.14%.

Another recent work on Arabic handwriting recognition was proposed by Ghanim et al. [15]. They proposed a multistage cascading approach for Arabic offline handwritten character recognition. The Hierarchical Agglomerative Clustering (HAC) algorithm was used to cluster and rank the dataset. Then, six different deep CNN approaches were used in the recognition process. The IFN/ENIT Arabic dataset was used to evaluate and compare the six deep CNN approaches. The proposed approach achieved promising results in terms of computation time and complexity, as well as classification results.

Most recent research on Arabic handwritten recognition is done by Altwaijry and Al-Turaiki [16]. In this study, they presented a novel dataset consisting of 47,434 Arabic characters written by children aged from 7 to 12 years. This dataset was used to train and test the proposed approach for the recognition of Arabic handwritten characters using CNN. The proposed approach achieved an accuracy of 88% on the AHCD and the Hijja datasets, respectively.

Akouaydi et al. [17] proposed the use of CNN based on Beta-elliptic parameters and fuzzy elementary perceptual codes for the recognition of Arabic online characters. An accuracy of 98.90% was achieved using two Arabic datasets; LMCA and MAYASTROUN.

According to the previous studies, it is confirmed that convolutional approaches perform more effectively when applying image recognition through different related approaches. In the case of offline/online character recognition, it is created in the form of 2D-vector, which is more suitable. The DBNs require extra efforts for transforming the data, followed by exceeding computational complexity. Results motivate to use one of their improved architectures, which are anticipated to be effective and appropriate for this research problem.

Table 1 gives a summary of the key literature analyzed in this research. This Table shows the relationships between the key studies on Arabic OCR as well as their limitations. The scope of these studies is the recognition of Arabic offline handwritten characters. It can be seen that [6] has achieved the highest accuracy on the MADBase dataset among other approaches, while [1] and [5] achieved the next best results. Approaches in [6] and [16] obtained lower efficiency compared to other approaches.

Table 1. A summary of key literature.

Ref.	Approach	Scope	Limitations	Results
[1]	CNN	Offline Arabic handwritten character recognition	Depends on the hyper-parameters' tuning and the size of the dataset.	97.32% on OIHAC dataset
[5]	CNN	Identifying offline handwritten numbers based on conducted experiments	The model is not enhanced more than the limit already enhanced.	97.4% on CMATERDB dataset
[6]	CNNs / VGGnet	Arabic handwritten alphanumeric character recognition	The system is simple and generic and does not perform effectively for words.	99.66% on MADBase dataset and 97.32% on HACDB dataset
[10]	ResNet-18	Offline Arabic handwritten character recognition	Low efficiency compared to other approaches.	94.8% on AIA9k dataset and 97.6% on AHCD dataset
[16]	CNN	Arabic handwritten character recognition	Small dataset and low efficiency compared to other approaches.	88% on a small artificial dataset

Arabic handwritten text recognition is one of the hot topics and challenging areas of the fields of pattern recognition and image processing. Deep neural networks as based on CNNs and DBNs show promising results in Arabic character/numeral recognition process in terms of accuracy and speed. Different machine learning approaches have been proposed for the recognition of Arabic characters. Most of these studies are realized to efficiently perform for smaller datasets. However, the critical issue that is encountered is that such approaches require an extensive effort from a domain expert engineer to design a feature extractor that transforms the raw data into an appropriate representation or feature vector from which a classifier could possibly recognize the input pattern [28]. In this study, we propose the use of ResNets for the recognition of Arabic handwritten characters using large standard datasets of isolated Arabic handwritten characters. ResNet is used in this paper, since it is one of the recent architectures of deep learning and not used yet for Arabic OCR. The significance of this paper is that it fills the gaps in recent related research and improves the classification rates of previous approaches. The implemented model deals with a higher-level data when being embedded to real applications which identify words rather than isolated characters only. It also could be used in the field of computer vision, such as handwriting recognition and natural object classification.

The rest of the paper is organized as follows: Section 2 presents the proposed methodology, experimental results are presented and discussed in Section 3 and conclusions drawn from this study and directions for future work and presented in Section 4.

2. METHODOLOGY

2.1 Overall Research Design

The proposed handwritten Arabic OCR approach using the ResNet architecture is presented. The first step is to conduct the dataset preparation and pre-processing that includes resizing, colour binarization, noise reduction and finally, classification into separate classes. The second step is to pass the dataset through to the deep ResNet and to train the network on the 76 determined classes including characters and digits. The final step is to test the trained network and evaluate its overall performance based on accuracy, precision and recall. Figure 1 shows the overall architecture of the proposed approach.

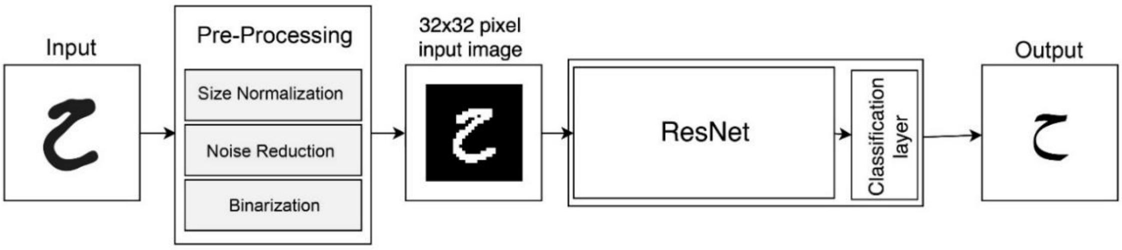

Figure 1. The overall architecture of the proposed approach.

Image pre-processing is performed for all characters in the dataset. This step includes noise reduction, color binarization, scaling and cropping the handwritten characters. To eliminate noise, thin features must be preserved as much as possible. Character edge pixels are identified as target pixels for the smoothing process, where such a process is only executed for the color difference. After that, the size of the target pixels is calculated for brightness and contrast operations. The final two steps are resizing the images to be of the size 32×32 pixels and the color depth of the images must be unified and should be black or white.

The Residual Neural Network (ResNet) is defined as a modularized architecture. An example of a generic ResNet, which stacks units with a similar connection shape, is proposed by He et al. [18]. ResNets won the 1st place in the ILSVRC 2015 ImageNet classification competition with an error rate of 3.57%. It consists of a set of residual blocks that are expressed in Equations 1 and 2.

$$y_1 = h(x_I) + F(x_I, W_I) \quad (1)$$

$$x_{I+1} = f(y_I) \quad (2)$$

where x_I and x_{I+1} denote the input and output of the ith unit which represents the feature value, F denotes the residual function, $h(x_I)$ represents the identity mapping and f denotes the activation function. Convolutional layers are combined with a down-sampling layer, then with an activation function. The Rectified Linear Unit (ReLU) is defined as a popular used activation function and is formulated in Equation 3.

$$f(x) = \max(0, x) \quad (3)$$

Pre-processed input character image 32×32 pixels are considered to form a 2D-vector and by keeping the spatial order of pixels, they can pass through to the pre-trained ResNet units. The maps of features are generated by convolutions with feature extraction kernels of 3×3.

After each stack of residual blocks, a pooling layer is attached. L2-pooling is a way of summarizing information from the convolutional layer by taking the square root of the squares' summation of the activations in a 3×3 region. It is used after each few stacks of residual blocks in order to reduce complexity [19].

Residual block is a special feature and identity mapping of the ResNet. It helps with training the network more effectively and faster, as it is also anticipated to simplify learning. Additionally, it solves the optimization problem by avoiding zero mapping and model closer to identity [20].

It is used only with input/output layers of the same dimension. Rather than expecting stacked layers to approximate $H(x)$, by Equation 4, layers are approximated by applying residual functions.

$$F(x) = H(x) - x \tag{4}$$

The residual block basically consists of a shortcut connection and a sequence of layers including convolutional layers. There exist a variety of forms of residual blocks, including two specific types: the standard block and the bottleneck block. Based on the implementation performed by He et al. [21], after each convolutional layer, a Batch Normalization (BN) layer is attached. The BN is adopted so that dropout is eliminated. Ioffe & Szegedy [22] studied the effect of the BN, which acts as a regulator speeding up training and reducing the over-fitting problem.

When building a residual block, the effect of depth and width must be considered. To acquire fewer parameters and decrease depth, bottleneck blocks are used to produce a thinner network. They consist of three convolutional layers; a 1×1 layer for down-sampling channel dimension, a 3×3 layer and a 1×1 layer for up-sampling the channel dimension.

2.2 Convolutional Layer

Convolution is defined as a mathematical operation that does an integral part of the product of two signals with one of them being flipped. The input of the convolutional layer is divided into equal regions of pixels (neurons), where each region is attached to a neuron of a hidden-layer and the convolved output is assigned. For example, in Figure 2, with 3×3 kernel slides (assuming Stride = 1) all over the input, the output represents the dot-product of a selected 3×3 image-input and flipped kernel. Additionally, it creates a convolved feature map (hidden layer) and so on for the rest of the kernels in which more hidden layers based on learned features are created.

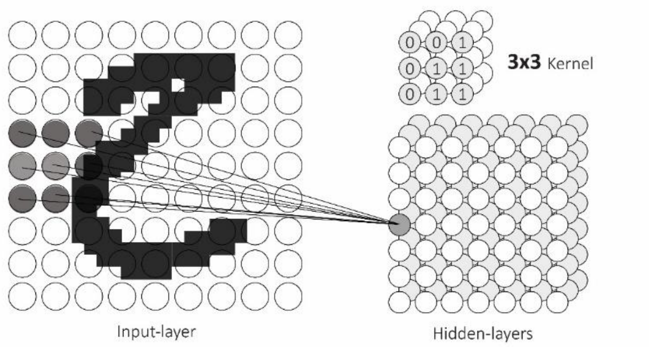

Figure 2. The convolutional layer.

Activation function is defined as a non-linear transformation and it basically decides whether a neuron should be activated or not. The Rectified Linear Unit (ReLU) is widely used in computer vision-related problems where deep networks are used, since it converges faster than the previous activations, such as sigmoid and is cheaper to be computed. As a result, it is leading to encounter a faster training time. However, it has a downside of causing dead neuron, once the neuron is always negative.

According to a study conducted by Xu et al. [23], different types of rectified activation functions in convolutional network are tested and compared in terms of accuracy and error rate. The activations comprise the standard ReLU, Leaky (LReLU), Parametric (PReLU) and Randomised Leaky (RReLU). The aim is to prove the effect of a non-zero slope for negative parts in rectified activation units on improving results. Based on their experimental results, the RReLU can overcome the other activations, but the PReLU is suspected not to function with smaller datasets. In this research, the LReLU activation is used, as shown in Equation 5 (Maas et al. [24]), where, $a_i$denotes the small, positive defined number and denotes a set that is highly based on experiments.

$$f(x_i) = \begin{cases} x_i & x \geq 0 \\ \frac{x_i}{a_i} & x < 0 \end{cases} \quad (5)$$

where: $a_i \epsilon (1, +\infty)$.

Batch normalization layer adjusts and scales the activations. It produces activations with a stable distribution throughout training, by enforcing the values of each layer to represent the same distribution. It solves the problem of internal covariance shift, which represents the amount by which the hidden unit values shift around. It is applied before non-linear layers.

2.3 Classification Layer

The classification layer basically consists of three parts. The first part represents the flattening layer that takes the output of the last residual block after applying the activation function and the average pooling. The flattening layer transforms the output into a 1-D vector to be used in the subsequent layer, which represents a fully connected layer. It consists of 1000-feature maps made up by the global average pooling, where each neuron is connected to the entire neurons within the next layer. Finally, the softmax activation function is responsible for predicting the final output (see Equation 6). It squashes the outputs of the layer beforehand along towards the range between zero and one for each neuron and the entire assigned values after applying Softmax must have a summation of one. A normal distribution of the values simplifies scattered predictions. There exist 76 classes (output units) in the thesis's case that represents Arabic characters and digits.

$$\sigma(z)_i = \frac{e^{z_j}}{\sum_{K=1}^{K} e^{K_j}} \quad (6)$$

where *Z* denotes the vector of the inputs related to the output layer and *j* indexes the output units *j*=1, 2, … k.

3. EXPERIMENTAL RESULTS

In this section, the results of the proposed approach based on the Arabic handwritten dataset are presented. The accuracy and validation pertaining to the proposed approach are found for 76 classes of many different handwritten characters and digits. The dataset used in this study is also presented. Data analysis and interpretation are presented as well.

3.1 Dataset

A new constructed dataset from some datasets used by other researchers is used to evaluate the proposed approach. The new dataset consists of letters and digits from 0 to 9 for covering the shortage that is found through existing ones. Modified letters as Al-Hamza (ء) and tā' marbūṭah (ة) are not included in most datasets. The collected samples are re-categorized to 76 classes, including the entire contextual cases of letters.

The digits' dataset is extracted from the MADBase proposed by El-Sawy et al. [25] and which represents the largest found dataset. The characters are extracted from the AHCD proposed by El-Sawy et al. [26], the DBAHCL approach proposed by Lamghari and Raghay [27], the OIHAC approach proposed by Boufenar et al. [28] and the AIA9k approach proposed by Torki et al. [29].

Images are initially pre-processed and segmented and hence, the dimensions are unified into an image size of 32×32 pixels and the colours are converted into black and white, as shown in Figure 3. Finally, a random subset is labelled and separated into the 76 classes. The total quantity of the dataset reaches 10340 images, which are split into 80% training (8320 images) and 20% validation (2100 images). Table 2 summarizes the characteristics of the datasets used in the construction of the new used dataset.

3.2 Data Analysis and Interpretation

The results are evaluated by measuring the standard deviation of the layer responses, where the obtained results reveal the strengths and weaknesses of residual functions among the involved layers. Responses represent each layer's output after providing nonlinear functions, such as the ReLU and Addition (see Equation 7).

$$y_i = h(x_I) + F(x_I, W_I) \tag{7}$$

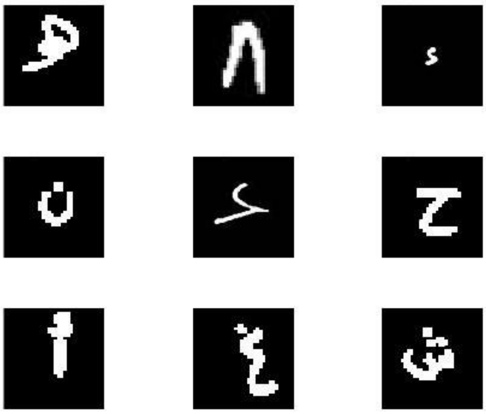

Figure 3. Randomly selected images.

Table 2. Datasets characteristics.

Dataset	Characteristics
MADBase [25]	The number of samples: 70000 Classes: 10 Dimensions: 32×32
AHCD [26]	The number of samples: 16800 Classes: 28 Dimensions: 32×32 Colour space: Grey
DBAHCL [27]	The number of samples: 5400 Classes: 54 Dimensions: 80×80 Colour space: RGB
OIHAC [28]	The number of samples: 5600 Classes: 28 Dimensions: 128×128 Colour space: Grey
AIA9k [29]	The number of samples: 8737 Classes: 28 Dimensions: 32×32 Colour space: Grey

However, residual responses are generally closer to zero, particularly for deeper networks where each layer tends to modify the signal loss [4]. Basically, the evaluation and analysis pertaining to the results apply the Cross Entropy (Equation 8) as a loss function that finds the distance between the predicted probability and the real one.

$$H(p,q) = -\sum_x p(x) \times \log q(x) \tag{8}$$

where $p(x)$ denotes the desired probability and $q(x)$ denotes the actual probability. Additionally, the performance of the produced method is evaluated based on the Recall and Precision parameters (Sokolova & Lapalme [30].

Precision (Equation 9) represents the average per-class agreement pertaining to the data class labels including those containing a classifier. The values used to describe Recall, shown in Equation 10, describe the average per-class effectiveness of a classifier to identify class labels.

$$Precision = \frac{\sum_i^l \frac{tp_i}{tp_i + fp_i}}{l} \tag{9}$$

$$Recall = \frac{\sum_i^l \frac{tp_i}{tp_i + fn_i}}{l} \tag{10}$$

3.3 Experiments

In this research, a 117-layer residual network is constructed with a network width of 15 units. Deep learning networks are deeply requiring big data to model the training datasets. Unfortunately, many applications do not have big datasets, such as OCR. Data augmentation comprises a group of techniques that enhance the size and quality of training datasets to build better deep learning models. Data augmentation can improve the performance of deep learning models and expand limited datasets to take advantage of the capabilities of big datasets [31]. In this paper, data augmentation is done by rotating the training examples horizontally and vertically using different rotation angles resulting in increasing the dataset to about 31,260 images divided into 24960 for training and 6300 for testing. For the subsequent experiments, the constructed network is used where the network is trained on publicly available datasets to prove the efficiency related to this method.

In terms of the training parameters' setting, the network is initially trained for 80 epochs and after that, for 90 epochs in the following experiment and finally, for 200 epochs. An epoch represents the number of passes through the training set before convergence. The learning rate is set proportional to the mini-batch size as it drops after the 60th epoch, where the validation accuracy drops in a few iterations before improving through the subsequent iterations. The learning rate is initially set to 0.1. The learning rate and batch size can implicitly influence the noise that is derived from performing the Stochastic Gradient Decent. The mini-batch size is set to 20, the proportional to the training set size related to each character. Table 3 summarizes the training parameters that are previously described.

Table 3. The tuning of training options.

Option	Value
Initial Learning Rate	0.1
Max Epochs	80, 90, 200
Mini Batch Size	20
Learning Rate Drop Factor (Changed after 60 epochs)	0.01

The experiments are performed on a computer with an Intel core i5, 2.8 GHz processor and 8 GB of RAM. Table 4 shows the trained network results through training. The lowest training error (loss) was 0.025% and the validation accuracy reached 98.9% achieved in the fourth quarter of the experiment after 200 training epochs.

Table 4. Trained network results.

Epochs	Accuracy	Loss	Validation Error
50	90.1%	0.23%	9.9%
100	97.8%	0.14%	2.2%
150	98.7%	0.05%	1.3%
200	98.91%	0.025%	1.09%

The training process required around 13 hours to terminate the entire determined iterations of 80800 iterations per epoch. The reason behind this process is the complexity of the network and the computations that are needed to train each layer which is considered a deep network with over 100 layers and the expanded number of samples after performing the data augmentation process.

3.4 Results

The results are summarized based on validation error, validation accuracy and loss. The experiment was sectioned into 4 phases throughout the 200 epochs. Table 5 shows a validation error of 9.9% for 50 epochs, 2.2% for 100 epochs, 1.3% for 150 epochs and 1.09% for 200 epochs. The lowest training error (loss) was 0.025% and the validation accuracy reached 98.9% achieved in the fourth quarter of the experiment after 200 training epochs. The overall progress of the experiment is shown in Figure 4, that displays the validation accuracy progress through the training process and Figure 5 shows the loss where it converges to 0.025% through the training process.

Table 5. A summary of results.

Epochs	Accuracy	Loss	Validation Error
50	90.1%	0.23%	9.9%
100	97.8%	0.14%	2.2%
150	98.7%	0.05%	1.3%
200	98.91%	0.025%	1.09%

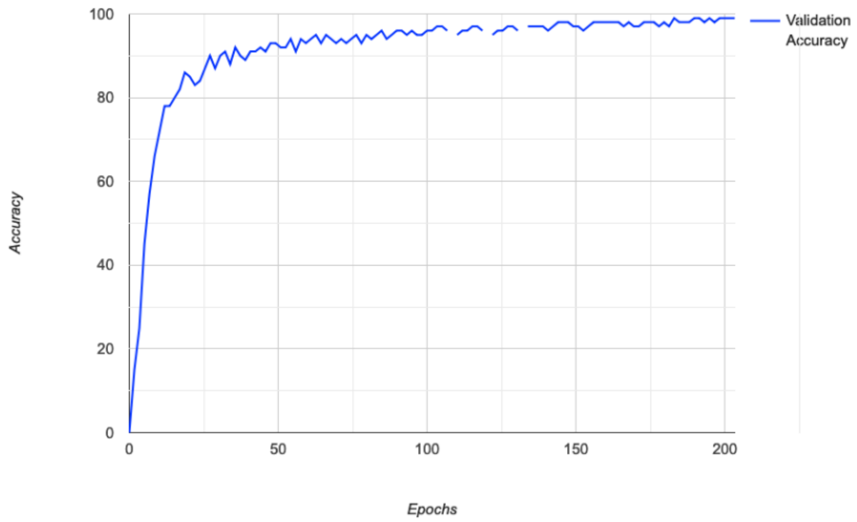

Figure 4. The experiment validation accuracy.

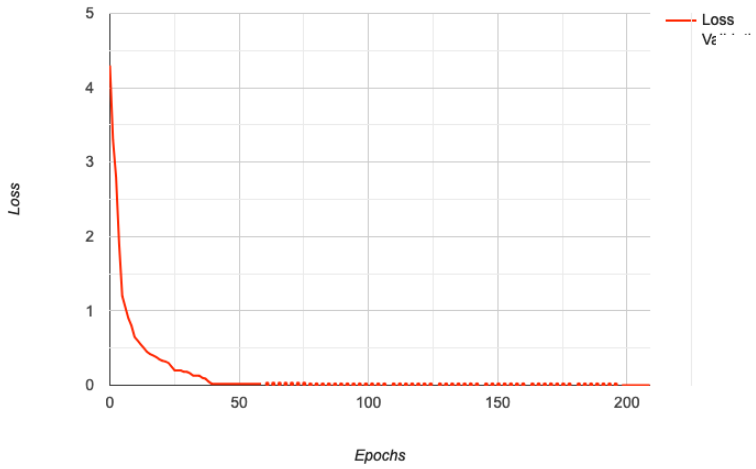

Figure 5. The experiment loss mean.

3.5 Comparison with Other Approaches

Experiments are performed on available datasets and the results of the proposed approach are compared with those of previous works using the same datasets (Table 6).

Comparison of the proposed approach with other approaches is shown in Figure 6. The proposed approach obtained an accuracy of 99.55% on the AHCD dataset, whereas Younis [10] obtained an accuracy of 94.8% on the same dataset using deep CNN and batch normalization approaches. On the AIA9k dataset, the proposed approach achieved an accuracy of 99.05%, while Younis [10] obtained an accuracy of 97.6%. Using the MADBase dataset, the proposed approach achieved an accuracy of 99.80%, while other researchers obtained less accuracy; Mudhsh & Almodfer [6] obtained an accuracy of 99.66% and Younis [10] achieved an accuracy of 97.6%. It can be inferred from the results that the proposed approach achieved more accurate results compared to other approaches.

Table 6. Comparisons with other approaches.

Dataset	Reference	Approach	Epochs	Classification Accuracy
AHCD	[10]	Deep CNN	18	94.8%
	Proposed Approach	ResNets	50	99.55%
AIA9K	[10]	Deep CNN	18	97.6%
	Proposed Approach	ResNets	50	99.05%
MADBase	[10]	Deep CNN	18	97.6%
	[6]	CNN/ VGGnet	N.A.	99.66%
	[32]	Deep CNN	N.A.	99.30%
	Proposed Approach	ResNets	50	99.80%

4. Discussion

It can be inferred form the obtained results that they seem extremely promising as shown previously in terms of the validation accuracy and loss. Precision and recall of each misclassified character are displayed in Table 7. It is seen that characters share several and similar features that are most often being misclassified. For example, the character (ـف) is mistaken with characters (ـمـ) and (ـهـ). Accordingly, it is previously mentioned that a character could be mistaken with another character by a dot in cases of a variety of writing styles, which represents one of the major challenges that is related to the language itself. Additionally, it is assumed in this thesis that the reason behind this mistake refers to gaining a small training/testing dataset for characters that are written within the context, where contextual characters only possess 80 training samples and 20 samples for validation.

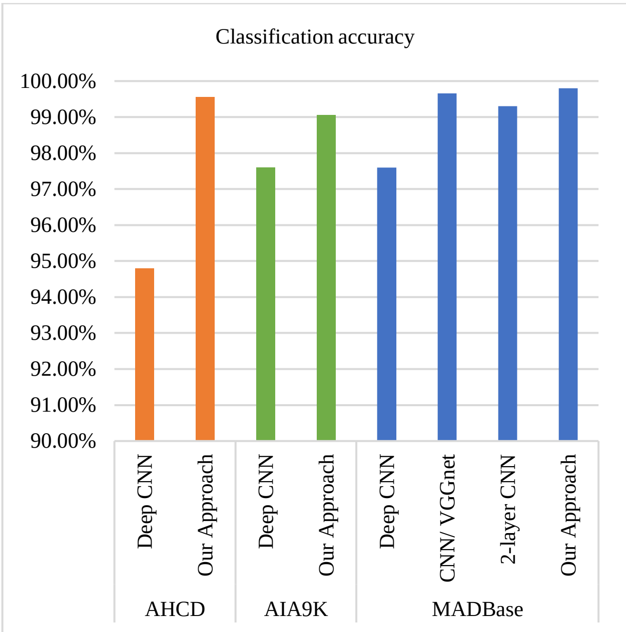

Figure 6. Results' comparison.

On the other hand, the overall recall error is seen to be satisfying. However, two characters are classified twice with wrong labels per character and the rest of the characters. As seen in Table 7, a summary of misclassified characters is shown with the mean average of precision and recall. Also, a summary of the error mean of precision and recall is shown in Figures 7 and 8.

As seen in the comparison with other approaches related to deep learning networks, it is shown that the proposed approach achieved more accurate results. It is previously claimed that the addition of residual blocks that have a special skip connection (or identity mapping) can affect the emerging results positively. Nonetheless, stacking convolutional layers to a definite number performs efficiently while the network is still deepened. The ResNet architecture shows that the network proceeds deeper with more than 150 layers if there is a large training dataset available. Additionally, by using the batch normalisation as a regularisation method for limiting the over-fitting problem and replacing the ReLU activation function with the LReLU function makes several neurons be as active as possible, which we tested on both activations and observed a noticeable improvement in the results.

Table 7. Misclassified labels, precision and recall percentages.

Character	Precision %	Character	Recall %
حـ	95.2	حـ	95
ـحـ	95.2	ـع	95
ـد	95.2	ظ	95
ث	97.6	ـسـ	95
ص	95.2	ـس	95
ـص	94.7	ـص	90
ط	97.6	ـطـ	95
ـعـ	95.2	ـفـ	95
فـ	90.9	ـمـ	95
ـك	95.2	ـم	95
مـ	95	ـن	95
ـن	90.5	ن	97.5
ـو	95.2	هـ	95
ي	95	ـه	95
٣	97.6	و	95
٧	97.6	ي	95
ء	95	ـي	95
٤	97		

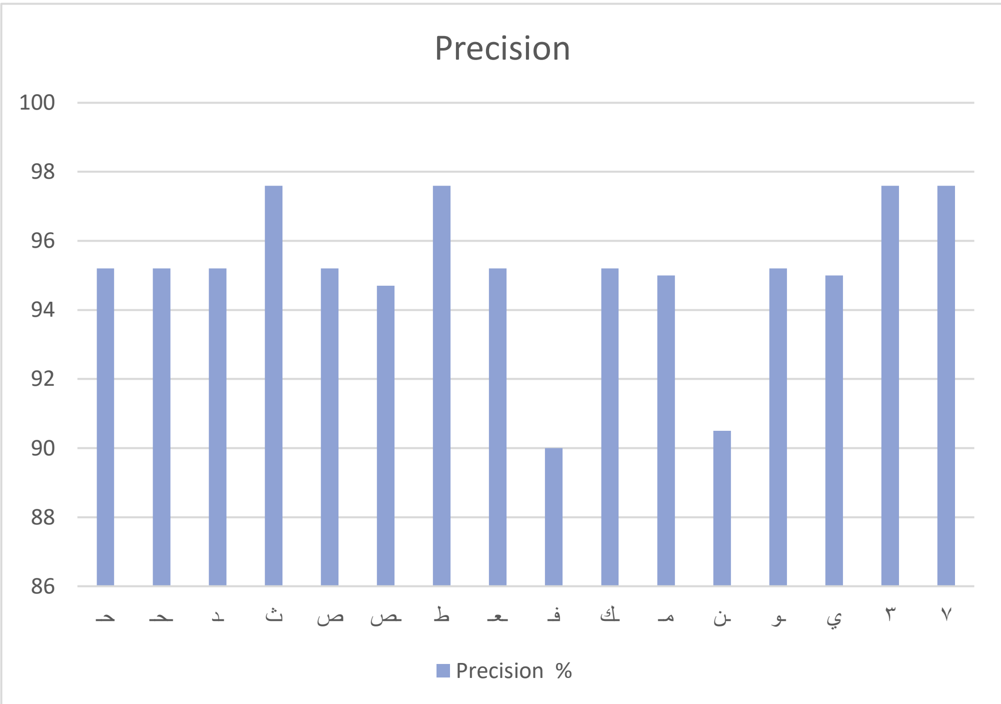

Figure 7. Precision percentage of misclassified characters.

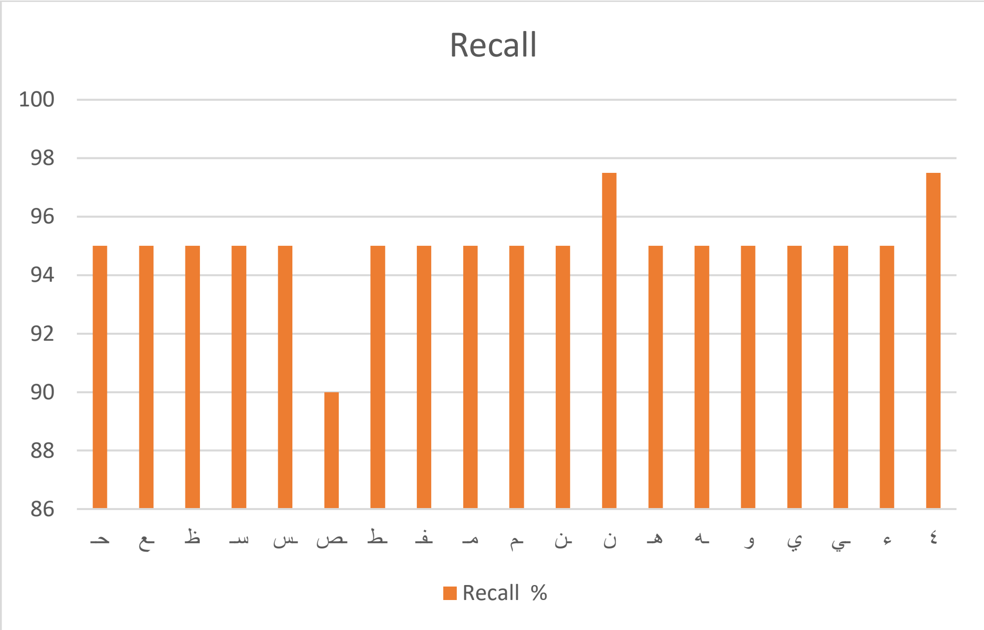

Figure 8. Recall percentage of misclassified characters.

5. CONCLUSIONS

In this research, a ResNet approach is constructed to recognize isolated Arabic handwritten characters in different contextual cases and digits. The existing architecture is modified as required in terms of several layers, network parameters, regularization techniques and activations being regularly used. The network is trained on a dataset that is collected from previously published datasets in order to obtain an inclusive dataset. The cursive nature of Arabic scripts, the variety of writing styles of each person and the excessive need of enriching Arabic language processing resources all represent robust motivations to start carrying out this study.

Previous researchers discussed different approaches for solving this problem. The focus of this study is on researchers who incubate deep learning methods. Starting from famous convolutional networks, such as the LeCun’s LeNet and VGGNets networks, the experiments show that deeper networks could work more efficiently with the chosen input size and problems, such as vanishing gradient and over-fitting problems.

Previous researchers suggested solutions, such as eliminating the use of a dropout layer to deal with the over-fitting problem by using the right activation function, such as the ReLU function and BN for regularization to solve such problems (e.g. the vanishing gradient problem). Moreover, the addition of a shortcut connection into the core building block produces an obvious improvement compared to previous approaches. The main findings and objectives of this study can be summarized as follows:

- In this research, we discussed the problem of recognizing handwritten Arabic characters and eastern Arabic digits. The results of this approach outperformed previous scholars’ approaches’ results.

- Deeper networks must be handled correctly to avoid problems such as over-fitting. In this research, it was proved that the addition of a shortcut connection could handle deeper networks and improve accuracy. Deeper networks can learn features in different levels of abstraction. However, wider, shallower networks perform good in memorization but not in generalization.

- The experiments with the constructed dataset that included both characters and digits proved that the approach used succeeded for this case as seen in the results’ section. Moreover, there are many similar features between eastern Arabic digits and characters, which implies the accuracy of the approach.

Directions for future work include working on a segmentation-free technique that is capable of recognizing the sequence of characters (words) with the addition of long-short term memory network specifications.

REFERENCES

[1] C. Boufenar and M. Batouche, "Investigation on Deep Learning for Off-line Handwritten Arabic Character Recognition Using Theano Research Platform," Intelligent Systems and Computer Vision (ISCV), pp. 1-6, DOI: 10.1109/ISACV.2017.8054902, Fez, Morocco, 2017.

[2] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based Learning Applied to Document Recognition," Proc. of the IEEE, vol. 86, no. 11, pp. 2278-2324, DOI: 10.1109/5.726791, 1998.

[3] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural networks," Communications of the ACM, vol. 60, no. 6, pp. 84–90, DOI: 10.1145/3065386, 2017.

[4] S. Wu, S. Zhong and Y. Liu, "Deep Residual Learning for Image Steganalysis," Multimedia Tools and Applications, vol. 77, pp. 10437-10453, 2018.

[5] A. Shiquzzaman and A. K. Tushar, "Handwritten Arabic Numeral Recognition Using Deep Learning Neural Networks," Proc. of the IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR), pp. 1-4, DOI: 10.1109/ICIVPR.2017.7890866, Dhaka, Bangladesh, 2017.

[6] M. Mudhsh and R. Almodfer, "Arabic Handwritten Alphanumeric Character Recognition Using Very Deep Neural Network," Information, vol. 8, no. 3, pp. 105-118, DOI: 10.3390/info8030105, 2017.

[7] K. Younis and A. N. Alkhateeb, "A New Implementation of Deep Neural Networks for Optical Character Recognition and Face Recognition," Proc. of the International Conference on New Trends in Information Technology (NTIT-2017), pp. 1-6, Amman, Jordan, April 2017.

[8] H. Tomimori, K. T. Chen and T. Baba, "A Convolutional Neural Network with Incremental Learning," Journal of Signal Processing, vol. 21, no. 4, pp. 155-158, 2017.

[9] A. Eladel, R. Ejbali, M. Zaied and C. Ben Amar, "Fast Deep Neural Network Based on Intelligent Dropout and Layer Skipping," Proc. of the International Joint Conference on Neural Networks, [Online], Available: https://doi.org/10.1109/IJCNN.2017.7965947, 2017.

[10] K. S. Younis, "Arabic Handwritten Character Recognition Based on Deep Convolutional Neural Networks," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 3, no. 3, pp. 186-200, 2017.

[11] M. Elleuch, R. Maalej and M. Kherallah, "A New Design Based-SVM of the CNN Classifier Architecture with Dropout for Offline Arabic Handwritten Recognition," Procedia Computer Science, vol. 80, pp. 1712-1723, DOI: 10.1016/j.procs.2016.05.512, 2016.

[12] N. Tagougui and M. Kherallah, "Recognizing Online Arabic Handwritten Characters Using a Deep Architecture," Proc. of the 9th International Conference on Machine Vision (ICMV 2016), vol. 10341, DOI: 10.1117/12.2268419, Nice, France, 2017.

[13] S. Karthik and K. Srikanta Murthy, "Deep Belief Network Based Approach to Recognize Handwritten Kannada Characters Using Histogram of Oriented Gradients and Raw Pixel Features," International Journal of Applied Engineering Research, vol. 11, no. 5, pp. 3553-3557, 2016.

[14] M. E. Mustafa and M. K. Elbashir, "A Deep Learning Approach for Handwritten Arabic Names Recognition," International Journal of Advanced Computer Science and Applications, vol. 11, no. 1, pp. 678-682, DOI: 10.14569/IJACSA.2020.0110183, 2020.

[15] T. M. Ghanim, M. I. Khalil and H. M. Abbas, "Comparative Study on Deep Convolution Neural Networks DCNN-based Offline Arabic Handwriting Recognition," IEEE Access, vol. 8, pp. 95465-95482, 2020.

[16] N. Altwaijry and Isra Al-Turaiki, "Arabic Handwriting Recognition System Using Convolutional Neural Network," Neural Computing and Applications, vol. 33, pp. 2249-2261, 2021.

[17] H. Akouaydi, S. Njah, W. Ouarda, A. Samet, M. Zaied and A. M. Alimi, "Convolutional Neural Networks for Online Arabic Characters Recognition with Beta-elliptic Knowledge Domain," Proc. of the IEEE International Conference on Document Analysis and Recognition Workshops (ICDARW), pp. 41-46, DOI: 10.1109/ICDARW.2019.50114, Sydney, Australia, 2019.

[18] K. He, X. Zhang, S. Ren and J. Sun, "Identity Mappings in Deep Residual Networks," Proc. of the European Conference on Computer Vision (ECCV 2016), Part of the Lecture Notes in Computer Science, vol. 9908, DOI: 10.1007/978-3-319-46493-0_38, Springer, Cham, 2016.

[19] P. Sermanet, S. Chintala and Y. LeCun, "Convolutional Neural Networks Applied to House Numbers Digit Classification," Proc. of the 21st International Conference on Pattern Recognition (ICPR2012), pp. 3288-3291, Tsukuba, Japan, 2012.

[20] M. Längkvist, L. Karlsson and A. Loutfi, "A Review of Unsupervised Feature Learning and Deep Learning for Time-series Modeling," Pattern Recognition Letters, vol. 42, pp. 11-24, 2014.

[21] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, DOI: 10.1109/ICDARW.2019.50114, Las Vegas, USA, 2016.

[22] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," Proc. of the 32nd International Conference on International Conference on Machine Learning (CML'15), vol. 37, pp. 448–456, 2015.

[23] B. Xu, N. Wang, T. Chen and M. Li, "Empirical Evaluation of Rectified Activations in Convolutional Network," Machine Learning (cs.LG); Computer Vision and Pattern Recognition (cs.CV); Machine Learning (stat.ML), [Online], available: http://arxiv.org/abs/1505.00853, 2015.

[24] A. L. Maas, A. Y. Hannun and A. Y. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models, " Proc. of the 30th International Conference on Machine Learning, JMLR: W&CP, vol. 28, Atlanta, Georgia, USA, 2013.

[25] A. El-Sawy, M. Loey and E. B. Hazem, "Arabic Handwritten Characters Recognition Using Convolutional Neural Network, " WSEAS Transactions on Computer Research, vol. 5, pp.11-19, 2017.

[26] A. El-Sawy, H. EL-Bakry and M. Loey, "CNN for Handwritten Arabic Digits Recognition Based on LeNet-5," Proc. of the International Conference on Advanced Intelligent Systems and Informatics (AISI 2016), Part of the Advances in Intelligent Systems and Computing, vol. 533, DOI: 10.1007/978-3-319-48308-5_54, Springer, Cham, 2017.

[27] N. Lamghari and S. Raghay, "DBAHCL: Database for Arabic Handwritten Characters and Ligatures," International Journal of Multimedia Information Retrieval," vol. 6, pp. 263-269, 2017.

[28] C. Boufenar, M. Batouche and M. Schoenauer, "An Artificial Immune System for Offline Isolated Handwritten Arabic Character Recognition," Evolving Systems, vol. 9, pp. 25-41, 2018.

[29] M. Torki, M. E. Hussein, A. Elsallamy, M. Fayyaz and S. Yaser, "Window-based Descriptors for Arabic Handwritten Alphabet Recognition: A Comparative Study on a Novel Dataset," arXiv:1411.3519v2, [Online], Available: https://arxiv.org/pdf/1411.3519.pdf , 2014.

[30] M. Sokolova and G. Lapalme. "A Systematic Analysis of Performance Measures for Classification Tasks," Information Processing and Management, vol. 45, no. 4, pp. 427-437, 2009.

[31] C. Shorten and T. M. Khoshgoftaar, "A Survey on Image Data Augmentation for Deep Learning," Journal of Big Data, vol. 6, no. 60, DOI: 10.1186/s40537-019-0197-0, 2019.

[32] G. Latif, J. Alghazo, L. Alzubaidi, M. M. Naseer and Y. Alghazo, "Deep Convolutional Neural Network for Recognition of Unified Multi-language Handwritten Numerals," Proc. of the 2nd IEEE International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR), pp. 90-95, DOI: 10.1109/ASAR.2018.8480289, London, UK, 2018.

ملخص البحث:

تقترح هذه الدراسة استخدام الشّبكات العصبيّة المتبقية (ResNets) للتعرف على الأحرف العربية المكتوبة بخط اليد، بما في ذلك الأرقام العربية. وتمثل شبكات ResNets نهجاً من التعلّم العميق الذي أظهر فعالية في العديد من التطبيقات أكثر من مناهج التعلم الآلي التقليدية. يتكون النّهج المقترح من ثلاث مراحل رئيسية هي: مرحلة ما قبل المعالجة، وتدريب الشبكة على مجموعة التدريب، واختبار الشبكة المدربة على مجموعات البيانات. تم إجراء تقييم للنّهج المقترح على ثلاث مجموعات بيانات متاحة هي: MADBase و AIA9k وAHCD. وقد حقق النّهج المقترح دقة بلغت 99.8٪ و 99.05٪ و 99.55٪ على مجموعات البيانات المذكورة على التوالي. كما حققت دقّة تحقق بلغت 98.9٪ على مجموعة البيانات المُنشأة بناءً على مجموعات البيانات الثلاث.