

# Online Handwritten Arabic Digits (Indian) Recognition using Deep learning

**Khalid Mohammed Musa Yaagoup<sup>1,5</sup>, Abd Elhafeez Hamid<sup>2</sup>, Shazali Siddig Mohammed<sup>3</sup>,  
Mohamed Elhafiz Mustafa<sup>4,6</sup>**

Al Neelain University, Dept of Computer Science, Faculty of Computer Science and Information Technology, Sudan<sup>1</sup>

Kassala University, Dept of Computer Science, Faculty of Computer Science and Information Technology, Sudan<sup>2</sup>

Al Neelain University, Dept of Information Technology, Faculty of Computer Science and Information Technology,  
Sudan<sup>3</sup>

Faculty of Computer Science, Sudan University of Science and Technology, Khartoum, Sudan<sup>4</sup>

Faculty of Computer Science and Information Technology, University of the Holy Quran and Islamic Sciences, Sudan<sup>5</sup>

College of Computer and Information Sciences, Jouf University, Sakaka, Kingdom of Saudi Arabia<sup>6</sup>

**Abstract:** Recognition of handwritten digits has been an important area in recent years because of its uses in many fields. Arabic pattern digits, weak work is performed because Arabic digits (Indian) are more complicated than English patterns. This study focuses on the recognition component of the recognition of handwritten Arabic digits (Indian) that faces many obstacles, including the infinite variety of human handwriting and the broad public databases. The study presented a deep learning approach that can effectively be applied to the recognition of handwritten Arabic digits. Convolutional Neural Network (CNN) trained and tested MADBase database (Arabic handwritten digits images) with 60000 training and 10000 test images.. A contrast is made between the results, and it is seen at the end that the use of CNN has resulted in substantial improvements across various classification algorithms for machine learning, As a test accuracy with better results than other approaches using the same database, the test accuracy was improved to 99.25%.

**Keywords:** Recognition, handwritten, classification, Convolutional Neural Network.

## I. INTRODUCTION

Recently, Deep Convolutional Neural Networks (CNNs) has been one of the most attractive methods and has been a key factor in a number of recent achievements and challenging machine learning applications such as ImageNet challenge [2, 3, 4, 5, 6, 7], object detection [8, 9], image segmentation [10, 11], and face recognition [12, 13, 14].

The advantage of a convolutional neural networks (CNN) advantage is its applicability to digit recognition, such as OCR and hand-written digit recognition (HDR), for nearly any available language, i.e., English, Arabic, etc. Arabic Handwritten Digit Recognition (AHDR) research, however, lacks the application of deep learning models, particularly CNN and other common deep neural network models. The current study fills this research gap by suggesting an optimal model that is easy to train and yields the best test accuracy.

In this study, Arabic handwritten digit recognition using an optimal CNN model. CNN is a neural feed-forward multi-layer network that extracts features from input data. CNN trained with neural network back-propagation algorithm. CNN has the ability to learn complex high-dimensional, non-linear mappings from a very large number of data (images). In addition, CNN has excellent recognition rates for characters and digits[15, 16]. The benefit of CNN is that the salient characteristics that are invariant and a certain degree of shift and shake are automatically extracted[17].

This study's contributions are as follows. First for Arabic HDR, a convolutional neural network was developed that achieves improved data set accuracy under study. Second, a comparison of the proposed model using the MADBase dataset; the results showed that the proposed model had achieved high accuracy. In addition, the proposed model can be useful in automated educational applications for Arabic numerical learning and writing. The model can intelligently classify Arabic digits(Indian) written by children.

The study is organized as follows: Section. 2 offers an overview of some of the related work done in the area. The material and methods described in Section 3, we will discuss an objectives of study, dataset, convolutional neural network and experimental setup. Section 4 provides an overview of the results and discussion, and in section 5 we list our conclusions.

## II. RELATED WORK

Various methods have been suggested and high recognition rates for the recognition of handwritten English digits have been reported [18-20]. Niu and Suen[18] intended to classify handwritten digits using the Convolutional Neural Network (CNN) and the Support Vector Machine (SVM). Experiments have been performed on the database of MNIST digits. They obtain a recognition rate of 94.40 percent with a rejection rate of 5.60 percent. A supervised auto-encoder architecture based on extreme machine learning was implemented by Tissera and McDonnell[19] to classify Latin handwritten digits based on the MNIST dataset. Up to 99.19 percent can be properly classified by the proposed technique. To classify handwritten digits, Ali and Ghani introduced Discrete Cosine Transformation based on Hidden Markov models (HMM). They used MNIST as a dataset for training and testing. HMM was applied as a classifier to classify the dataset of handwritten digits. On average, the algorithm offers promising recognition results of 97.2 percent.

Many researchers have addressed the recognition of texts in recent years, including Arabic. In 2011, Ouafae et al.[21] proposed an improved method based on the Loci characteristic for recognizing Arabic digits. Their work is based on numeral recognition that is handwritten and printed. With multi-layer perceptron technique and K-nearest neighbor, the recognition is carried out. The algorithm on the dataset was trained to contain 600 Arabic digits with 200 test images and 400 training images. On small datasets, they were able to achieve a 99 percent recognition rate.

In 2008, using Gabor-based features and Support Vector Machines (SVMs), Mahmoud[22] proposed a technique for the automatic recognition of Arabic handwritten digits. A medium database was used with 21120 samples written by 44 authors. For testing, the dataset contains 30 percent and the remaining 70 percent of the data is used for training. Using three scales & five orientations and using four scales & six orientations, they obtained average recognition rates of 99.85 percent and 97.94 percent.

Selvi and Meyyappan[1] proposed a method for recognizing Arabic digits using the back propagation neural network in 2013. The final outcome shows that more than 96 percent accuracy of recognition for a small sample handwritten database is given by the proposed method.

Takruri et al.[23] introduced a three-level classifier for the classification of handwritten Arabic digits based on Support Vector Machine, Fuzzy C Means and Unique Pixels in 2014. On a public dataset, they tested the new algorithm. There are 3510 images in the dataset, 40 percent of which are used for testing and 60 percent of the images are used for training. The overall reported accuracy of testing is 88 percent.

In 2014, AlKhateeb and Alseid[24] proposed a method using the Dynamic Bayesian Network to classify Arabic handwritten digit recognition. They used discrete cosine transform features based on coefficients for classification. Their system has been trained and tested on a database of Arabic digits(Indian) (ADBase) containing 70,000 Arabic digits. On 10,000 test samples, they reported an average recognition accuracy of 85.26 percent.

In 2015, Salh et al.[25] and Babiker et al.[26] proposed method of linear correlation algorithms in two dimensions for the purpose of recognizing Arabic characters and numerals (Indian). Salh et al.[25] was used A private database with set of samples. They split the automatic reading process into five stages:(1) inputs stage, (2) Preprocessing, (3) Segmentation, (4) Calculating correlation value stage, (5) Classification. The system was evaluated on 10 images, and the result was analysed by statistical analysis software (SPSS) to calculate the false rejection rate, which is considered a good measure for testing the pattern recognition systems, Rate of true classification of the digits (0, 1, 2, 4, 5, and 7), (3), (6), (8), (9) are 100%, 80%, 70%, 60%, 50% respectively.

Elsawy et al.[27] proposed a method for classifying Arabic handwritten digit recognition using the CNN Based on LeNet-5 in 2017. They used two convolutional layers and two sub-sampling for automatic feature extraction, two fully connected layers as multi-layer perceptron hidden layers for nonlinear classification. Their system has been trained and tested on a database of Arabic digits(Indian) (MADBase) containing 70,000 Arabic digits. On 10,000 test samples, they reported an average recognition accuracy of 99 percent.

## III.MATERIAL AND METHODS

### A. Objectives

Recognizing Arabic digits(Indian) and different types of handwriting makes it important to find a new and advanced handwriting recognition method and work on it. A deep learning algorithm needs a huge amount of data to be able to make good decisions. The algorithms were applied to a small handwriting image in[1,21-23], the problem being the small training and testing database for images. In[27], a large Arabic handwriting digit database called (MADBase) was proposed with training and testing images. The proposed model will have the following objectives: (1) Improve the accuracy of the recognition of Arabic handwritten digits in the database under research, using deep learning techniques. (2) Different methods for extracting the features of handwritten Arabic digits(Indian) will be tested. (3) to develop an effective extraction technique that leads to a more precise method of handwritten Arabic digit recognition. (4) To measure the performance of the extracted features, various classifiers will be studied.

For training and testing MADBase dataset GPU-enabled TensorFlow with support for CUDA acceleration, we decided to use Keras and TensorFlow as the development environment.

The experiment was performed on a 2.2 GHz Core(TM) i7-2670QM CPU with 8G memory and GPU NVIDIA GeForce GT 525M running on windows, a neural network has improved due to the availability of GPU's. In order to train a proposed CNN, other cloud-based services such as the Google Cloud Web Services platform that offer resources were carried out.

### B. Dataset

An Arabic handwritten digital database (ADBase) and a modified version named (MADBase) [28] have been published by El-sherif and Abdleazeem. MADBase is a modified version of the ADBase benchmark in the same format as the MNIST[29] benchmark. The 70,000 digits written by 700 writers are composed of ADBase and MADBase.

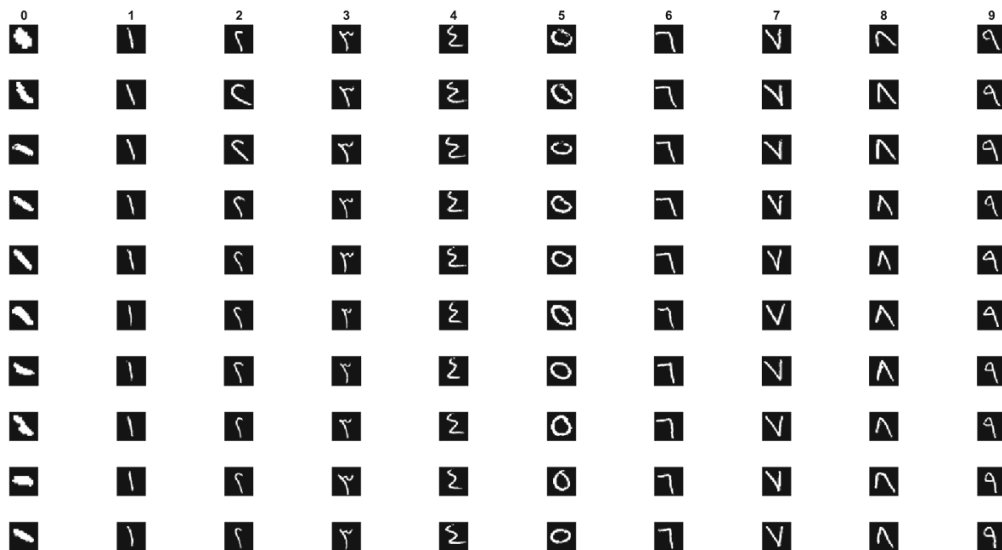


Fig. 1 MADBase benchmark training database sample

Every writer wrote ten times each digit (from 0 to 9). The database was collected from various institutions to ensure inclusion of various writing styles: Colleges of Engineering and Law, School of Medicine, Open University (whose students cover a wide variety of ages), a high school, and a government institution. The datasets are divided into two sets: a training set (60,000 digits per class of 6,000 images) and a test set (10,000 digits to 1,000 images per class). The ADBase and MADBase are open to researchers for free (<http://datacenter.aucegypt.edu/shazeem/>). Fig. 2 and Fig. 3 provide samples of the training and research images of the MADBase database.

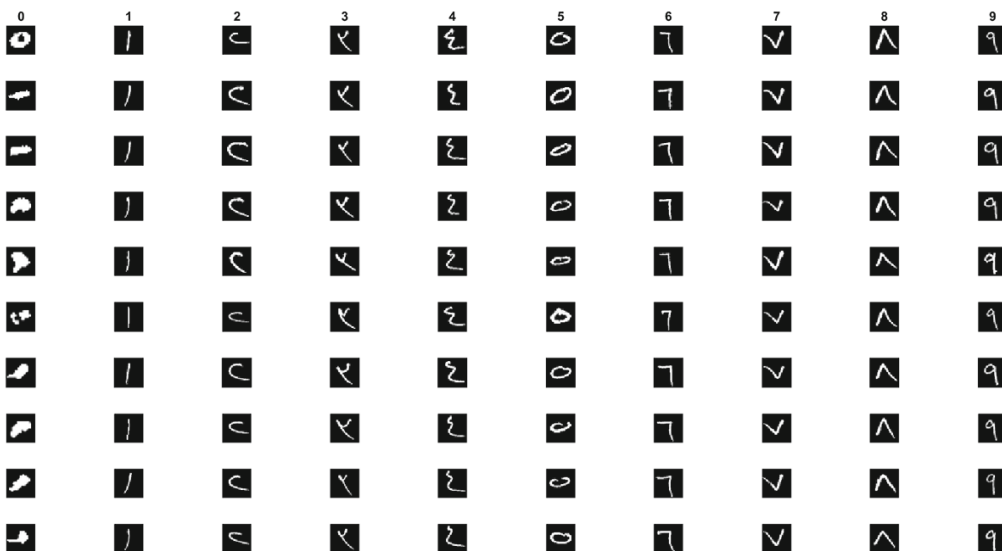


Fig. 4 MADBase benchmark testing database sample



### C. Convolutional Neural Network (CNN)

The CNN architecture differs from the typical neural architecture designed to emulate the human visual processing system, and has clearly enhanced 2D image processing structures. CNN's ability to recognize the classification and extraction of 2D features is effective. Furthermore by identifying shape variations, CNN's max-pooling layer shows its efficiency. The weights are related to a sparse CNN network that requires fewer parameters than networks of similar size that are completely connected. In addition, using a gradient-based learning algorithm that is successful in overcoming the decreasing gradient problem, the CNN is trainable. The gradient-based algorithm thus trains the entire network to decrease the criteria for error, resulting in better generalization and optimized weights.

The general CNN architecture consists of two main units: the classifier and the extractor feature. Inside the feature extraction unit, each layer of the network receives input from the output of the previous layer and feeds the current output as an input into the next layer. The output data with corresponding input data is projected by the classifier unit. There are two basic layers in feature extraction, as can be seen in Table 1: convolutional and pooling layers [30].

Each node applies convolutional layers operations on input nodes in convolutional layers to extract the features from input image data. The  $n$ th layer input is the  $n-1$ th layer output, whereas this input is passed through a series of filters or kernels followed by nonlinear activation of the features of the rectified linear unit (ReLU). For example, if  $f$  is the activation function of the ReLU,  $x_j^{n-1}$  is an input from the  $n-1$  layer,  $k_{i,j}^n$  is the filter of the  $n$ th layer, and biases from the  $n$ th layer are represented as  $b_{i,j}^n$ , then the convolution layer operates as:

$$x_j^n = f(x_i^{n-1} * k_{i,j}^n) + b_{i,j}^n \quad (1)$$

The pooling layer applies maximum or average operations to abstract features on the input nodes. For example, if a  $2 \times 2$  downsampling filter or kernel is applied to the pooling layer input, the output dimension will be reduced for all inputs to one-half of the related input dimension. The pooling operation can be expressed as follows:

$$x_j^n = \text{down } x_i^{n-1} \quad (2)$$

In contrast to neural networks, convolutional neural networks extract high to low-level features; however, CNN extracts features from low to high-level. These higher-level features are extracted from the propagation of lower-level inputs. This distribution decreases the size of the features depending on the size of the pooling mask and the convolution. However for the selection of optimized features of input images for higher classification accuracy, the amount of feature mapping escalates. The output of the last layer is used, as shown in Table 1, as an input to the fully connected layers that use the Softmax operation to classify inputs. For example, for a weight vector  $\omega$ , a sample input  $x$ , and linear function  $K$ , the Softmax operation is mathematically denoted as for the  $i$ th class.

$$P(y = i|x) = \frac{\exp(x^T \omega^i)}{\sum_{k=1}^K \exp(x^T \omega^k)} \quad (3)$$

Each layer consists of a set of neurons in conventional neural networks. Input to these networks is transfigured through a set of hidden layers that are connected to previous and previous layers by neurons. The CNN performance is higher than the CNN performance used for LeNet-5-based Handwritten Arabic Digits Recognition[27]. We were motivated to use this prominent and improved CNN performance feature on the proposed MADBase dataset.

Table 1. The architecture of the proposed model of the convolutional neural network (CNN)

Layers (type)	Output Shape	Parameters	Number of Parameters
Input	28,28,1	-	0
1 (Conv2D)	26, 26, 32	3×3 Conv, 32 ReLU	320
2 (MaxPooling)	13, 13, 32	2×2 max-pooling, stride 2	0
3 (Conv2D)	11, 11, 64	3×3 Conv, 64 ReLU	18496
4 (MaxPooling)	5, 5, 64	2×2 max-pooling, stride 2	0
5 (Conv2D)	3, 3, 64	3×3 Conv, 64 ReLU	36928
6 Flatten	576	-	0
7 Dense	32	ReLU	18464
8 Dense	10	Softmax	330
<b>Total parameter: 74,538</b>			
<b>Trainable parameters: 74,538</b>			
<b>Non-trainable parameters: 0</b>			

Table. 1 shows input images were fed to the CNN model from our MADBase dataset with a size of  $28 \times 28$  pixels per image. The first layer of our model was the 2D convolutional layer with a kernel size of  $3 \times 3$ . This layer utilizes every pixel of the input image. The outcome of this layer was implanted with a  $26 \times 26$  feature map. This feature map was embedded with the geometrical features in order to set up a feature vector. As an activation function, a ReLU (rectified linear unit)[31] was used to activate every output of the convolutional layer. Rather than the "Sigmoid" function[32], ReLU was used because of its ability to address the gradient vanishing problem. For stimulation that is similar to the

human brain mechanism, a ReLU uses the inherent threshold invariant. The first convolutional layer's input was fed into a max pool with a stride of 2 for nonlinear downsampling. This layer's output was  $13 \times 13$  feature vectors that were implanted into the second convolutional layer. A kernel mask with an input size of  $3 \times 3$  was applied to the input. The output of this layer for nonlinear downsampling with a stride of 2 was again fed to a max pool layer. This layer generated a size  $5 \times 5$  output that was fed into the third convolutional layer of 2D. The output of these layers for a 1D feature vector has been converted to flattening. To make use of a fully connected layer after the convolutional and max-pooling layers, this flattened layer was required. The two fully connected (dense) layers, which are artificial neural networks (ANNs), are used at the end of our model. The last fully-connected layer (Dense (10, activation = 'Softmax')) uses the probability distribution of each class to classify output. For adaptive learning optimizing, the Adam optimizer is used. Adam can be used as a combination of the algorithm for RMSprop and Stochastic Gradient Descent. Due to the use of square gradients to scale the learning rate, as used in RMSprop, the Adam optimizer was selected for our model; in addition, Adam takes advantage of momentum by moving the gradient average rather than the gradient itself, which is similar to Stochastic gradient descent SGD with momentum [33]. During this implementation, the default setting of the optimizer was altered to achieve optimum performance.

Various layers and learning rates, momentum, optimizers and size  $28 \times 28$  datasets were tested during the experimentation phase. However with the model as described in Table 1, the optimal results were generated. With fewer parameters, the model performed better and the computation and resource consumption were lower than that of other models.

#### D. Experimental setup

Using the following measures, we evaluate the efficiency of our proposed CNN model:

Recall (R): is the proportion of images correctly classified over the total number of images belonging to Class  $x$ :

$$R = \frac{TP}{TP+FN} \tag{4}$$

Precision (P): is the percentage of images that are classified correctly over the total number of classified images.:

$$P = \frac{TP}{TP+FP} \tag{5}$$

F1 measure: That's a measure that includes Recall and Precision:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{6}$$

TP (true positive) is the total number of images that can be correctly labeled as belonging to class  $x$ , FP (false positive) is the total number of images that were wrongly labeled as belonging to class  $x$ , FN (false negative) is the total number of images that were incorrectly labeled as not belonging to class  $x$ , and TN (true negative) is the total number of images that were incorrectly labeled as belonging to class  $x$ .

### IV. RESULTS AND DISCUSSION

This section incorporates the results obtained by the proposed model (CNN). To classify MADBase datasets, we will define the effects of applying the network. We will equate the findings obtained with those of other methods using the proposed method. Prediction accuracy is used as a measure of performance. The accuracy of the test accuracy (Correct Classification) obtained is 99.25%, the miss-classification rate in Table 2 is 0.75%.

Table. 2 Miss-Classification & correct-Classification rate & number of incorrect and correct recognition on testing data

Class	Digit in Arabic(Indian)	Digit Name	Miss Classification	# of Missed	Correct Classification	# of Correct
0	۰	sifr	1.8%	18	98.2%	982
1	۱	wahid	0.9%	9	99.1%	991
2	۲	ithnan	1.2%	12	98.8%	988
3	۳	thalaatha	0.6%	6	99.4%	994
4	۴	arbae	0.2%	2	99.8%	998
5	۵	khamsa	0.5%	5	99.5%	995
6	۶	sitta	0.4%	4	99.6%	996
7	۷	saba	0.3%	3	99.7%	997
8	۸	thamaaniya	0.5%	5	99.5%	995
9	۹	Tisa	1.1%	11	98.9%	989
Miss-Classification Rate= 0.75%						
Total Number of Miss-Classification = 75						

Class	Digit in Arabic(Indian)	Digit Name	Miss Classification	# of Missed	Correct Classification	# of Correct
Total Number of Correct Classification = 9925						

Table. 2 displays the missing classification rate and the number of missed classifications in the testing MADBase dataset. It can be seen that we achieve a very exciting average miss-classification rates of 0.2%, 0.3%, 0.4%, 0.5%, 0.5%, 0.6%, for classes 4, 7, 6, 5, 8, 3 respectively, and 0.9%, 1.1%, 1.2%, 1.8% for classes 1, 9, 2, 0 respectively. The total of miss-classification is 75. The Arabic digit (zero) have a highest number of wrong classification 18. The (zero) digit is easily confused by the two Arabic (five), (one) digits. This shows the great potential and recognition ability of Our CNN to classify Arabic handwritten digits. On the other hand, for digits with a high rate of miss-classification, we found that they have some common characteristics from confusion matrix of testing dataset(MADBase), which are summarized as:

- Some digits are very confusing with the similar strokes of digits shown in Table 3. The tiny distinction of stroke structure bring challenges for some similar character pairs, such as Arabic(zero) and Arabic(one), the difference between them are Arabic(zero) it is similar to a solid circle, Whereas, Arabic (five) similar to an open circle.
- Some digits are very confusable with the structure of other digits. The Arabic(six) digit is easily confused by the Arabic(nine), Arabic(two) digit is also confused by the Arabic (four).
- Some digit strokes are missing from the digit structure. Other digits which have the additional touching of the stroke that shown in Fig. 3.

Table 3: Some of arabic digits(Indian) stroke similarity from confusion matrix

Similar digits		Comments
۰	۰	Circle(open, solid)
۱	۱	
۳	۲	
۴	۲	
۹	۰	Circle shape
۹	۶	

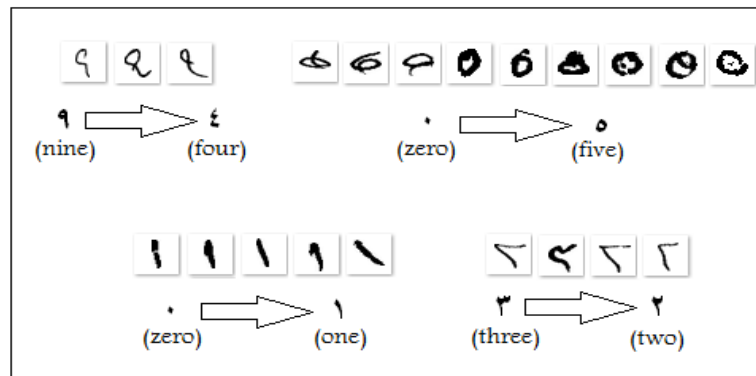


Fig. 3 Misclassified samples of digits

The misclassified samples are due to extra stroke, missing stroke, touching stroke. For a suggested plan to make a successful estimate, most of these samples are difficult.

We also evaluate the performance of our proposed CNN model using measurements (precision , recall, F1 score), a 99.25% accuracy, a 99.25% recall and a 99.25% F1 score. In Table 4, the detailed metrics are reported per digit for testing MADBase dataset, while Table. 5 shows detailed metrics are reported per digit for training MADBase dataset. Also Fig. 4, Fig. 5 shows Confusion matrix of testing and training MADBase datasets, respectively.

Table 4: Experimental results on the testing MADBase datasets(Classification Report)

Digit	Precision	Recall	F1-score	Support
۰	0.982	0.9869	0.9845	995
۱	0.991	0.991	0.991	1000
۲	0.988	0.991	0.9895	997

Digit	Precision	Recall	F1-score	Support
३	0.994	0.999	0.9965	995
६	0.998	0.9881	0.993	1010
०	0.995	0.9851	0.99	1010
१	0.996	0.992	0.994	1004
४	0.997	1	0.9985	997
८	0.995	0.998	0.9965	997
९	0.989	0.994	0.9915	995
<b>Accuracy</b>			<b>0.9925</b>	<b>10000</b>
<b>Macro avg</b>	<b>0.9925</b>	<b>0.9925</b>	<b>0.9925</b>	<b>10000</b>
<b>Weighted avg</b>	<b>0.9925</b>	<b>0.9925</b>	<b>0.9925</b>	<b>10000</b>

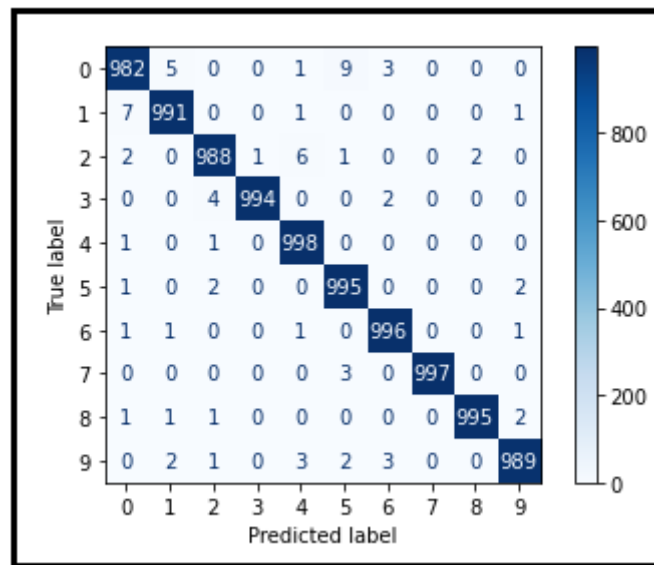


Fig. 4 Confusion Matrix for testing MADBase datasets

Table 5: Experimental results on the training MADBase datasets(Classification Report)

Digit	Precision	Recall	F1-score	Support
.	1	0.9998	0.9999	6001
१	0.9998	1	0.9999	5999
२	1	1	1	6000
३	1	1	1	6000
६	1	1	1	6000
०	1	1	1	6000
१	1	1	1	6000
४	1	1	1	6000
८	1	1	1	6000
९	1	1	1	6000
<b>Accuracy</b>			<b>0.99998</b>	<b>60000</b>
<b>Macro avg</b>	<b>0.99998</b>	<b>0.99998</b>	<b>0.99998</b>	<b>60000</b>
<b>Weighted avg</b>	<b>0.99998</b>	<b>0.99998</b>	<b>0.99998</b>	<b>60000</b>

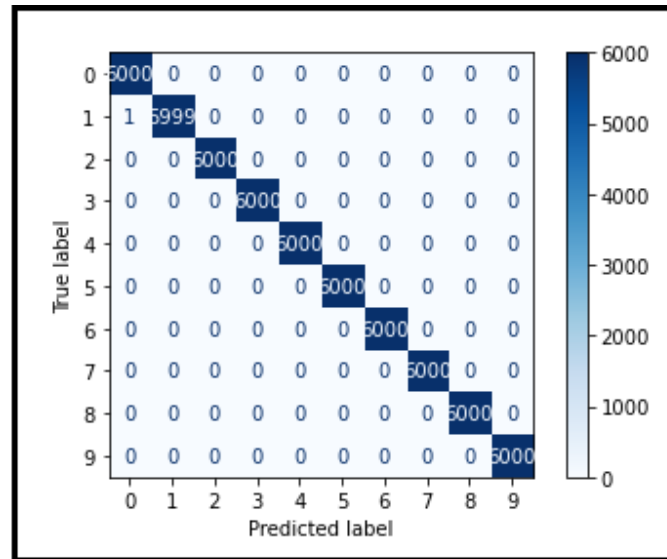


Fig. 5. Confusion Matrix for training MADBase datasets.

Table 6. Comparison of the proposed approach with the other approach.

Authors	Database	Training data Testing Data	Misclassification Error
Takruri et al. [23]	Public database	3510 images 60% training digits 40% testing digits	12%
AlKhateeb and Alseid. [24]	ADBase	60,000 training digits 10,000 testing digits	14.74%
Ouafae et al. [21]	Private databases	600 images 400 training digits 200 testing digits	1%
Selvi and Meyyappan. [1]	Private databases	Sample handwritten images are tested	4%
Mahmoud. [22]	Private databases	21120 images 70% training digits 30% testing digits	0.15% and 2.16%
Elsawy et al. [27]	MADBase	70000 images 60000 training digits 10000 testing digits	1% training 12% testing
<b>current approach</b>	<b>MADBase</b>	<b>70000 images</b> <b>60000 training digits</b> <b>10000 testing digits</b>	<b>0.00002% training</b> <b>0.75% testing</b>

The results obtained from CNN on the MADBase database are seen in Table 6. It can be seen from that the large database has the proposed solution and has the best error of miss-classification. The results are better than the results published in related work[1, 21-24, 27], but often it is difficult to compare because previous work has not been tested with a large database benchmark. On testing data, the proposed method obtained 99.25% accuracy on test data 0.75% miss-classification error.

## V. CONCLUSION

Handwritten Character Recognition for Arabic digits(Indian) is an active field of research that requires an improvement in accuracy at all times. We present a review of the recognition of Arabic character handwriting and its importance. To recognize handwritten letters, we suggested a deep learning model. Our model achieved excellent results on the MADBase dataset. As experimental results, the approach gives best accuracy in large database with 0.00002% training miss classification error rate and 0.75% testing miss classification error rate. Our future work will concentrate on improving the performance of handwriting recognition of Arabic digits (Indian) using other improved techniques for deep learning. Classification accuracy improved to 99.25% on testing MADBase dataset.



**ACKNOWLEDGMENT**

We are very thankful for her helpful advice and recommendations to our guide, **Mohamed Elhafiz Mustafa**, Asst. Professor, Faculty of Computer Science, Sudan University of Science and Technology. We also want to thank all our respected teachers and friends for their excellent contributions to the completion of this study and for their support.

**REFERENCES**

- [1]. P. Selvi and T. Meyyappan, "Recognition of arabic numerals with grouping and ungrouping using back propagation neural network," *Proceedings of the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering, PRIME 2013*, pp. 322–327, 02 2013.
- [2]. H. M. Albeahdili, H. A. Alwzawy, N. E. Islam et al., "Robust convolutional neural networks for image recognition," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 6, no. 11, pp. 105–111, 2015.
- [3]. F. Lauer, C. Y. Suen, and G. Bloch, "A trainable feature extractor for handwritten digit recognition," *Pattern Recognition*, vol. 40, no. 6, pp. 1816–1824, 2007.
- [4]. C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," pp. 562–570, 2015.
- [5]. M. A. Fischler and R. A. Eshelager, "The representation and matching of pictorial structures," *IEEE Transactions on computers*, vol. 100, no. 1, pp. 67–92, 1973.
- [6]. K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" pp. 2146–2153, 2009.
- [7]. 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, 2017.
- [8]. K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.
- [9]. X. Wang, M. Yang, S. Zhu, and Y. Lin, "Regionlets for generic object detection," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 17–24.
- [10]. C. Couprie, C. Farabet, L. Najman, and Y. LeCun, "Indoor semantic segmentation using depth information," *arXiv preprint arXiv:1301.3572*, 2013.
- [11]. R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
- [12]. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein et al., "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [13]. R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," vol. 2, pp. 1735–1742, 2006.
- [14]. O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," 2015.
- [15]. D. S. Maitra, U. Bhattacharya, and S. K. Parui, "Cnn based common approach to handwritten character recognition of multiple scripts," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2015, pp. 1021–1025.
- [16]. K. M. M. Yaagoup and M. E. Mustafa, "Online arabic handwriting characters recognition using deep learning," *International Journal of Advanced Research in Computer and Communication Engineering(IJARCCCE)*, vol. 9, no. 10, pp. 83–92, 2020.
- [17]. N. Yu, P. Jiao, and Y. Zheng, "Handwritten digits recognition base on improved lenet5," in *The 27th Chinese Control and Decision Conference (2015 CCDC)*. IEEE, 2015, pp. 4871–4875.
- [18]. X.-X. Niu and C. Y. Suen, "A novel hybrid cnn-svm classifier for recognizing handwritten digits," *Pattern Recognition*, vol. 45, no. 4, pp. 1318–1325, 2012.
- [19]. M. D. Tissera and M. D. McDonnell, "Deep extreme learning machines: supervised autoencoding architecture for classification," *Neurocomputing*, vol. 174, pp. 42–49, 2016. 6
- [20]. S. S. Ali and M. U. Ghani, "Handwritten digit recognition using dct and hmms," in *2014 12th International Conference on Frontiers of Information Technology*. IEEE, 2014, pp. 303–306.
- [21]. E. Ouafae, M. Hitmy, and F. Lekhal, "Arabic numerals recognition based on an improved version of the loci characteristic," *International Journal of Computer Applications*, vol. 24, pp. 36–41, 06 2011.
- [22]. S. A. Mahmoud, "Arabic (indian) handwritten digits recognition using gabor-based features," in *2008 International Conference on Innovations in Information Technology*. IEEE, 2008, pp. 683–687.
- [23]. M. Takruri, R. Al-Hmouz, and A. Al-Hmouz, "A three-level classifier: fuzzy c means, support vector machine and unique pixels for arabic handwritten digits," in *2014 World Symposium on Computer Applications & Research (WSCAR)*. IEEE, 2014, pp. 1–5.
- [24]. J. H. AlKhateeb and M. Alseid, "Dbn-based learning for arabic handwritten digit recognition using dct features," in *2014 6th international conference on Computer Science and Information Technology (CSIT)*. IEEE, 2014, pp. 222–226.
- [25]. G. Z. A. Salh, A. H. Mansour, and A. H. M. Babiker, "Arabic numerals recognition using linear correlation algorithm in two dimension," *International Journal of Computer Applications Technology and Research*, vol. 4, no. 1, pp. 1–6, 2015.
- [26]. A. H. M. Babiker, M. M. Musa, M. M. Osman, M. M. Alamiem, and M. M. Elsaied, "Isolated arabic handwritten character recognition using linear correlation," *International Journal of Computer Applications Technology and Research*, vol. 4, no. 12, pp. 928–932, 2015.
- [27]. A. Elsawy, H. El-Bakry, and M. Loey, *CNN for Handwritten Arabic Digits Recognition Based on LeNet-5*, 10 2017, pp. 566–575.
- [28]. A. Hafiz and G. Bhat, "Boosting ocr for some important mutations," *Second International Conference on Advances in Computing and Communication Engineering (ICACCE)*, pp. 128–132, 05 2015.
- [29]. A. Savita, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, "Improved handwritten digit recognition using convolutional neural networks (cnn)," *Sensors*, vol. 20, p. 3344, 06 2020.
- [30]. Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, 12 1998.
- [31]. V. Nair and G. Hinton, "Rectified linear units improve restricted boltzmann machines vinod nair," vol. 27, 06 2010, pp. 807–814.
- [32]. X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 649–657. [Online]. Available: <http://papers.nips.cc/paper/5782-character-level-convolutional-networks-for-text-classification.pdf>
- [33]. D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

**BIOGRAPHY**

**Khalid Mohammed Musa Yaagoup:** Lecturer in Faculty of Computer Science and Information Technology, University of the Holy Quran and Islamic Sciences, OCR, Deep learning.