

ARABIC HANDWRITTEN CHARACTER CLASSIFICATION AND RECOGNITION USING CNN AND TRANSFER LEARNING APPROACH

OMAR ALI BORAİK¹, M. RAVIKUMAR² and CHANNABASAVA CHOLA³

¹Department of Computer Science, Seiyun University, Hadramout, Yemen.

^{1,2}Department of Computer Science and MCA, Kuvempu University, Shankaraghatta, Shimoga, Karnataka, India.

³Department of Computer Science and Engineering, Indian Institute of Information Technology, Kottayam, India.

Email: ¹omaralib2010@gmail.com, ²ravi2142@yahoo.com, ³channabasavac7@gmail.com

Abstract

Arabic Handwritten character recognition is still a complex task. Lack of Arabic handwritten database is considered a huge problem in improving a new model for Arabic character recognition task or even proposing a new one. The remarkable similarity between Arabic letters' shapes, especially the handwritten scripts lead to the misclassification of Arabic characters. Though, there are a lot of efforts that have been made in Object Character Recognition (OCR) systems for Arabic text. Recently, these efforts have only reached partial use of classification and recognition. Progress in recognizing Arabic handwritten scripts is still unsatisfactory. Newly, deep learning methods are utilized, which make significant progress in identifying Arabic texts. This proposed study, deep learning-based models namely the ResNet50, DenseNet 121, VGG16 and CNN models were used. Also, a transfer-learning approach was applied for classifying and recognizing Arabic handwritten characters. The current study aims to get benefit from the pre-trained transfer-learning approaches on the ImageNet dataset. Then, it re-trains the approaches on a collected dataset, which includes 131,000 images of Arabic handwritten characters. The results of the models of character recognition with Inception V3 and ResNet50 are improved. So, it is easy to generalize to similar problems of classifying handwritten characters. Therefore, the proposed models on the collected dataset in terms of overall accuracy, recall, precision, and F1-score by 98.94%, 100%, 93.75%, and 96.43%, respectively. Overall classification accuracy of 98.95% is achieved. Therefore, the proposed models on the collected dataset in terms of overall accuracy, recall, precision, and F1-score by 98.94%, 100%, 93.75%, and 96.43%, respectively. Permanently, the overall classification accuracy of 98.95% is achieved.

Keywords: Arabic character Recognition; Deep learning; Character classification; Transfer learning; CNN.

1. INTRODUCTION

Arabic language spoken by more than 447 million in the middle east region, also known as official language in various sectors like education, media and government offices [1]. Arabic handwritten characters recognition is one of the most challenging and complex tasks in computer vision. The task of Arabic handwritten characters recognition can be generalized to system will to the banking system for check recognition, postal code recognition, Number plate detection, etc. [2, 3] [4]. High accuracy is achieved when applying successful Deep CNN models to handwritten character classification and recognition. Researchers who are developing Optical Character Recognition (OCR) systems are expanding text recognition to reach the main goals of obtaining high accuracy, faster [5], better achievement, and getting an error percentage close to zero, like the studies [4, 6] and [7]. Thus, researchers resort to using existing databases to compare the developed system with previous systems, along with the

possibility of expanding the quantity of the databases such as the public new dataset in [8] for various purposes.

Many challenges hinder the process of text recognition systems, such as the great similarity between Arabic letters. The difference in Arabic letter shapes based on their position in the word, makes complexity in classification. Each letter has two or four forms (isolated - beginning of the word - middle - end). In addition, there is a difference in the way the letter is written. Place and number of dots, which is an integral part of the letter's body, make it completely different in pronunciation. Another challenge that makes Arabic handwritten text recognition techniques complicated is that it leads to misclassification and unsatisfactory results [9].

Since 2012, machine learning and deep learning techniques have dominated the handwritten text recognition of various languages [10] (online, machine-printed or handwritten) and achieved better and faster performance in dealing with huge databases despite the challenges and difficulties in each language [11-13]. As a result of this, better, faster, and less error-free performance, researchers' interest has increased overtime in developing various models and architectures for handwritten text recognition task, based on the simple Convolution Neural Networks (CNN) idea. The structure of the particular method still differs in layers. The pre-trained neural network represents some of the most effective convolutional neural networks for ImageNet challenging through transfer learning, such as extracting and tuning associated features [14].

Transfer learning that is introduced in this work via deep learning approach which employs a pre-trained models, which were trained for different tasks as a starting point for another. As a result, it is a learning optimization strategy that enhances the implementation of the second task. Nevertheless, it only performs if the model features learned in the first task are general. Transfer learning may be applied for another task, either a pretrained approach in which one of the approaches accessible from different research organizations is selected as a starting point to regulate it for the other similar challenge. [15-17]. This work compares the outputs of pre-trained learning transfer approaches in classifying and identifying Arabic handwritten characters. The points listed below give a summary for the main contributions of the current study:

- 1- A comprehensive study is conducted through CNN models and transfer learning approaches to solve misclassification problems of some similar Arabic characters.
- 2- Two various databases (public and private) for Arabic handwritten characters are used to evaluate the proposed deep learning approaches.
- 3- The proposed work investigates the best Transfer Learning Approaches for classifying and recognizing the Arabic handwritten characters.

2. RELATED WORK

In recent days handwritten character recognition is carried out with deep learning based techniques which shown improved performances in classifying and recognizing tasks. Arabic handwritten characters are being addressed. The training from scratch takes a very long time; if the lexicon is too big, or the size of samples is not enough for the training the model, in such case, the machine learning algorithms to classify the characters can be a daunting task. The use of transfer learning approaches to compensate for the dataset deficiency or recognize the characters in a reasonable time. The study [6] used IFN/EINT database for samples of Arabic handwritten words. Due to the limited image samples of words, they applied the learning transfer process to recognition. They proposed an approach based on the sequential transferring of mid-level images of Arabic words. They evaluated the performance of three CNN architectures: ResNet [18], AlexNet [19], and VGG [3]. Based on their results, the ten words rated in IFN/ENIT database were accurately rated in the ImageNet database as a source dataset. The error rate of 14% was reduced to 2.5% [6]. The results showed that the ResNet model's outputs were more effective, accurate and better. The same models were applied in this study but to the Arabic characters recognition.

Applying a model experience of CNN models in learning to solve a similar task depends on resolving the character classification problems. For example, researchers at [2] used seven models of CNN techniques for learning transfer: Google Net, ResNet18, ResNet50, ResNet101, VGG16, and VGG19. AlexNet determines the most appropriate and best model to classify the handwriting images written by non-natives or natives. Their results showed that Google's model achieved the highest proportion. On the other hand, [5] evaluated many learning transform models in the classification and recognition of Devanagari characters. Their experience showed that the Inception V3 model achieved a better accuracy, which reached 99%; also, the run-time was 16.3 minutes in each epoch, while AlexNet was faster in 2.2 minutes and 98% accuracy. The moral is that more precision increases the run-time.

Many studies have applied CNN for the classification of handwritten Arabic characters using limited databases, so the dataset is insufficient for training. In this Study [7] Applied CNN for recognizing the characters of Arabic handwriting. The proposed model was trained on 16800 images of Arabic characters. The results achieved 97.2% as accuracy; the model achieved 97.7% as accuracy with data augmentation technique. The authors in [4] also utilized CNN on the similar database in [7] of Arabic handwritten characters. The database was divided into 13440 images for training and 3360 for testing. The propose model in [4] gave 5.1% misclassification in the data test and the accuracy was closed to 95%. While the proposed CNN model in [7] achieved the accuracy rate with data augmentation reached to 97.7%

The authors in [20] proposed a deep CNN named VGGnet for recognizing the alphanumeric character of Arabic handwriting, akin to the VGGNet neural network but with some improvements. Two datasets were used to train the proposed network: ADBase and HACDB. Using a model consisting of 13 convolutional layers to collect features, two max-pooling layers to minimize the image's size and dimensions, and three fully connected layers to generate a good prediction. The network employs two regularization techniques to avoid overfitting: data

augmentation and dropout. According to the study's findings, the suggested network obtained 97.32% of accuracy on HACDB database and ADBase database was 99.66% of accuracy. Based on previous studies mentioned earlier discussion. There are limited studies which dealt with solving the problem of classifying the most similar Arabic characters by using various CNN architectures to classify each character belonging to specific category. It was also noted that previous works used limited number of datasets to train deep learning models, when previously proposed models applied to a new database result were not satisfactory. The present work focused on applying four different deep CNN architectures and benefiting from the transfer learning for Arabic handwritten characters recognition and solving the classifying similar problems. The motivation of this study is to evaluate models in learning transfer to reach the optimum model in dealing with Arabic handwritten characters classification. One aim of this work is to reduce misclassifications errors of similar characters that occurred in previous studies like [6]. Based on the CNN model, DenseNet, ResNet, VGG16 transfer learning models, new deep learning approaches for classifying images of Arabic handwritten characters will be improved.

3. METHODS

3.1 Dataset:

In this part, two different datasets are described, one is public and the other is private. These two datasets are used to evaluate CNN and transfer learning models.

3.1.1 Arabic Handwritten character Database (AHCD)

This database includes 16800 images of Arabic handwritten characters. Participating to write it around 60 participants which the range of their aged 19-20 years [9]. This database was used by [9, 21] to evaluate their proposed CNN models for classifying Arabic handwritten characters, where it was split into 13440 for training and 3360 for testing.

3.1.2 Private database for this work.

The other database is a private database with images of isolated Arabic handwritten characters. These isolated characters were extracted by [22]. We collected 114200 images of characters from this private database.

The total number of character images is 131000 images to feed our proposed models for evaluation. Character images are classified into 28 classes depending on the number of Arabic alphabet 28 letters. Each class has almost 4190 images of characters, which involve the four shapes depending on their position in the word. Fig 1 shows samples of these characters. The classification model was fed in two steps: First, the number of images was reduced to 13446, split into 80% for training, 10% for testing, and 10% for validation. Second step, the number of images is 131,000 images of characters with the same split, 80% training, 10% testing and 10% validation as shown in fig 2.

Fig 1: image samples of Arabic handwritten characters isolated



3.2 Proposed method Deep CNN approaches

3.2.1 Convolutional Neural Networks

Data and functions in a CNN have extra structure. Images are used as input data for □1; □2... □n. essentially, the input image to a convolutional layer is $M \times M \times C$ image information, where M represents the image's height and width, M is the number of pixels in the image, and C represents the number of channels per pixel. Three channels of RGB image $C = 3$, while $C=1$ for a grayscale image's one channel. A CNN comprises numerous layers, including convolutional layers, pooling layers, and fully linked layers. The convolutional layer is made up of K filters (kernels) of $N \times N \times R$, where N is the height and width of the filter (kernels) and R is equal to or less than the number of image channels C and may vary for each filter (kernel). Fig 2 depicts the filter (kernel) convolved with the image to yield $M \times N+1$ k feature maps. If the input is 28×28 , the max-pooling output is 14×14 . As illustrated in Fig 1, each feature map is then pooled, often utilizing choose maximum pooling across $q \times q$, where q is the maximum value of inputs. Following the convolutional and pooling layers, any number of fully connected layers may be added, as in a normal multi-layer neural network.

Fig 2: Shows convolution steps on an input image with $M \times M$ size and multiplied with $N \times N$ size of the kernel

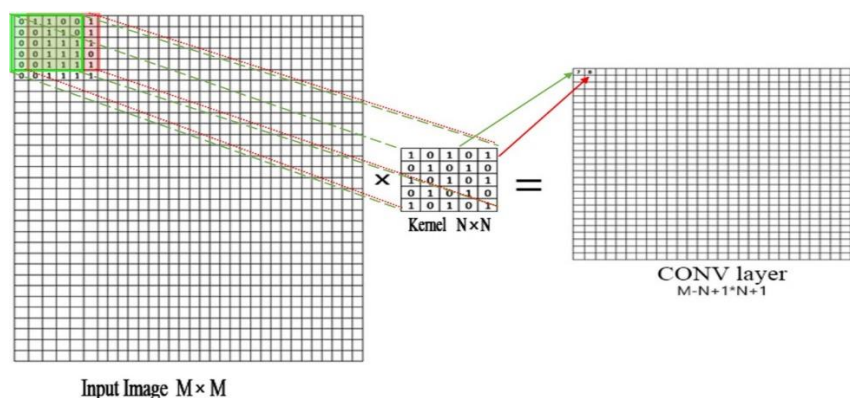
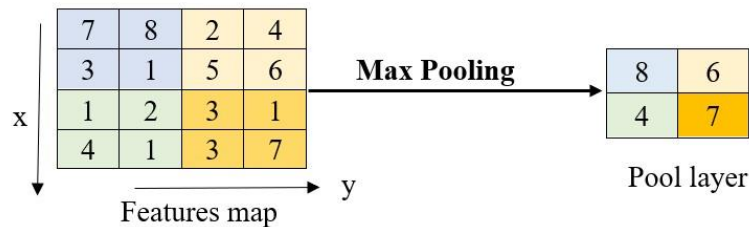


Fig 3: Shows the pooling process by selecting the largest value of the feature map to reduce the size



3.3 layers

a) Convolution Layers.

If W is a filter with an $N \times N$ kernel. The filter W is used on a convolutional layer L followed by $M \times M$ square neuron nodes the of the convolutional layer output will be defined by $(MN+1) \times (MN+1)$, resulting in the k -feature maps shown in Fig 3. The convolutional layer functions as a feature extractor that extracts prominent input characteristics like endpoints and edges, layer l_1 is used to calculate the pre-nonlinearity input to a unit. Is then calculated as follows:

$$Y_i^l = B_i^{(l)} + \sum_{a=1}^M \sum_{b=1}^M W_i X_{(i+a)(j+b)}^{l-1} \quad (1)$$

Where $B_i^{(l)}$: bias matrix, $W_i^{(l)}$: filter of size $N \times N$. Then, the convolutional layer uses its activation function as:

$$Z_i^l = \sigma(Y_i^l) \quad (2)$$

In the proposed model, the activation function Rectified Linear Unit (ReLU) is applied with non-saturating $\sigma(Y_i^l) = \max(0; Y_i^l)$. The activation function ReLU is used on the output of every convolutional layer and fully connected layer. The ReLU [27] improves the nonlinear features of the decision function and the entire structure without changing the receptive fields of the convolution layer.

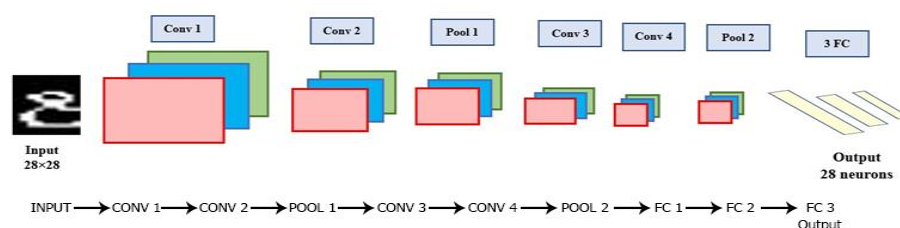
b) Pooling Layer

The pooling layer may follow convolutional layers in CNN. We get fewer parameters and hence fewer calculations by lowering spatial dimensions. Another benefit of pooling layers is that the output feature maps from the convolution layer are generalized, making the classification insensitive to direction changes and distortion effects. Pooling is accomplished by dividing the convolution layer's feature map into zones and subsampling it as a single output for each zone. There are two ways for subsampling: taking the maximum or the average of the pooled zone. Using max rather than average is a bulge and noticeable in the previous studies. The max-pooling is done entirely within a 2×2 pixel window.

c) Fully-Connected Layers

Following numerous convolution and subsampling layers, the output features are flattened into dense (fully connected layers) producing features piecing into recognizable objects. Every neuron in one fully connected layer is coupled to every neuron in the next layer. Classifiers are trained in the fully connected layer, where each neuron conducts a sum of dot products between previous layer inputs and weights; then, the activation function softmax is usually used in this part to output the prediction. Fig 3 displays the suggested architecture in the proposed model before applying the Deep CNN architectures.

Fig 4: Architectures of CNN for Arabic Handwritten Characters



It is to be pointed out here that two subsequent convolutional layers were implemented, each one with 16 kernels, followed by a pooling layer of 2×2 and stride 2. All of this is followed by two successive convolution layers, each one with 32 kernels and the last pooling layers are 2×2 of stride 2. The output of these layers is then flattened and squished in FC1 and FC2, with the final FC layer (output layer) containing 28 neurons, which correspond to the number of distinct classes of Arabic alphabet. The last fully connected layer is the input classifier.

d) Parameters of CNN

- **learning rate:** The learning rate (α) is utilized during the weight update of such architecture. This parameter is critical in influencing the neural network's convergence and generalization effectiveness. A minimal learning rate causes slow convergence and, conversely, divergence.
- **Backpropagation Optimizer:** The optimizer is the method in charge of modifying the weights of the network to decrease error. In this sense, Adam optimizer is used. Adaptive moment estimation is a more efficient form of Stochastic Gradient Descent (SGD) that requires less memory.
- **Mini-batch:** In neural network terminology, Batch refers to dividing the training set into subsets of occurrences. The optimizer computes the gradient for each subgroup (Batch) to minimize the loss. One epoch is used for the whole traverse across all of these batches. Many epochs may be required for a model to learn the instances in the dataset. Mini-batch is utilized as 16 in our model. This value is frequently chosen experimentally and requires a validation method to decide that it is optimal for the data at hand.

- **Softmax Activation Function:** In a multi-classification model, the number of neurons in the last layer (output layer) must be the same as the number of classes in the multiclassification model. It gives the probability for each character and that the target class is the character with the highest probability.

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{k=1}^N e^{x_k}} \quad (3)$$

- **Batch Normalization and Dropout:** By regulating the inputs to each layer, batch normalization provides several advantages: It hastens the learning process and the neural network may operate more excellent learning rates because batch normalization ensures that neither extraordinarily high nor low activation occurs. Batch normalization is performed to the preceding layer's output activation by removing the batch mean and dividing by the batch standard deviation. In our experiment, dropout is used to diminish interdependent learning among neurons. Also three fully connected layers are used with the proposed CNN model.

4. TRANSFER LEARNING

A dataset may be not sufficient to train CNN neural network models, obtained misclassification of the characters or inaccuracy desired of character recognition. Transfer learning approaches can be applied to character classification or recognition with high performance. Through training process on an extensive dataset is to reuse the model weights from previously trained models. The trained model was developed for standard computer vision data sets such as ImageNet recognition tasks. ImageNet is a project which aims to provide an extensive database of images for research purposes. It contains more than 14 million belonging to more than 2000 categories. The ImageNet challenge is the de-facto standard for computer vision classification algorithms when it comes to image classification [23]. Networks exclusive to this challenge have been dominated by CNN and deep learning technologies since 2012. The modern, pre-trained Of CNN models included in Keras's library represents some of the best performing CNN on ImageNet challenge through transfer learning. Such as extracting associated features and adjusting the weights. In general, transfer learning refers to a process in which a model is trained on a specific problem. This trained model is used in another task to solve a related and similar situation. In deep learning, transfer learning is the best way to introduce a CNN architecture, then retrain another task to solve a similar problem. The benefit of transfer learning is that it reduces the training time of the deep learning model [24], reduce misclassification rate. The weights of the pretrained model in reused neural networks can be used as a starting point for the training process and adapted to the new problem. The Weights in reused neural networks can be used as a starting point for the training process and adapted to the new problem. This way may be helpful when the related problem contains much more labelled data than the problem of interest. This paper uses Deep CNN's approaches for Arabic handwritten character classification. The simple CNN model proposed and compared with VGG, Resnet and Inception V3 models are used to handle the task of classification of handwritten Arabic characters.

5. TRANSFER LEARNING METHODS

There are three ways through which transfer learning can be achieved: Given a source model that has already been trained with ImageNet dataset A, a target model can be trained with dataset B by the following:

- Instead of randomly initializing the weights, use the source models architecture with weights of A dataset.
- The source model can be used as a feature extractor by replacing the layer with the number of classes in dataset B.
- Using the source model with some of the layers frozen, generally the initial layers with more available features and retraining the last layers with more specific features for dataset B. The generalization needs to be identified when the specificity becomes evident. To accomplish this, start freezing layers from top to down.

Four different CNN architectures, DenseNet121, ResNet50, Inception V3, and VGG16, were used to implement the transfer learning process. They (refers to what?) were previously trained on the ImageNet database. Then the saved weights were used to train the models on Arabic handwritten characters. The Arabic character images were classified into 28 classes according to the number of the Arabic alphabet is 28.

5.1 Deep CNN architectures:

The various CNN architectures used to classify Arabic handwritten characters are discussed in this part. Using Keras library which includes the CNN architectures.

Table 1: Shows the arguments are inputted into CNN model and transfer learning models

Arguments	inputs
Image size	28
Batch	64
epoch	5
Include_top	Fully connected
Input shape	Image_size, image_size, 3 channel
pooling	Global max pooling 'max'
Classifier_activation	Softmax
Number of classes as output	28 classes (28 Arabic alphabet)

a) DenseNet 121 [25]: Mainly, in the DenseNet, it is noticed that it commences with a convolution representation of 7×7 and 3×3 that represents pooling block maximization which precedes four dense blocks, three transition layers and a classifying layer that includes a 7×7 global average pooling and a 1000-D layer that is fully-connected. Regarding the DenseNet 121 architecture, four dense blocks with a convolutional layer count of 6, 12, 48, 32 can be noticed. Each of the dense blocks has its layer count. Transition blocks keep the transferred feature maps manageable among the four dense blocks. Utilizing a 2×2 average pooling with

a stride of 2, the number of feature maps and their sizes are reduced by half using a 1×1 convolution.

b) ResNet-50 [18]: ResNet-50, ResNet-18 and the ResNet-101. The authors in [18] proposed ResNet architecture. The ResNet is a neural network constructed from pyramidal cells in the cerebral cortex. The ResNet was applied to skip connections or shortcuts to move through many layers. The typical ResNet model was developed by skipping two or three layers, including nonlinearity (ReLU) and batch normalization. ResNet also introduced the concept of residual learning to Neural Networks (NN) to transform the competition in the NN architecture and add effective ways to train deep networks. For example, ResNet18 is a pre-trained deep learning model for image classification. The network was 18 layers deep and trained with 1 million images based on 1000 categories. ResNet50 is a pre-trained model for image classification. The network was 50 layers deep and trained with 1 million images in 1000 classes. ResNet101, on the other hand, is a pre-trained deep learning image classification model with 101 layers trained on 1 million images in 1000 categories. The ResNet-50 approach begins with a Convolution. A block consisting of a 7×7 convolution layer with 64 output channels followed by a series of layers. Batch normalization with step size 2, softmax, and 3×3 max pooling. Four modules are stacked in the center, each one has two remaining blocks. The remainder identity block applied to the output receive field is precisely the same dimension as the input receive field. The convolutional remainder block applies to the input receive field in the same size. The convolutional remainder block uses a 1×1 convolution operation to match the dimensions between the inputs and outputs of the remaining blocks. The last layer is adaptive mean pooling, followed by a fully connected layer where the number of neurons is the same number of output classes.

c) Inception V3: Inception V3 is an improved version of the fundamental model Inception V1. It was introduced in 2014 as the GoogLeNet model. Inception V3 is a pre-trained CNN with 22 ImageNet-trained layers. The network design comprises a 1 to 1 Convolution layer at the network's center. Furthermore, global average pooling was utilized at the network's conclusion [26].

d) VGG16 [3]: VGG16 (visual geometry group) is a CNN model presented by Simonyan and Zisserman [27]. This model achieves the top five test accuracy of 92.7% on ImageNet dataset. This dataset includes 1000 classes with over 14 million images. The name VGG16 comes from the fact that there are 16 layers, including a convolution layer, a max-pooling layer, an activation layer, and a fully connected layer. It has 13 convolutional layers, five max-pooling layers, and three dense layers, which sum up to 21 layers, but there are only 16 weight layers, finally, VGG19 is a pre-trained deep learning model for image classification. The network consists of 19 layers and trains with 1 million images from 1000 categories in the ImageNet database. Simonyan and Zisserman (2015) point out that VGG19 has 19 layers, specifically 16 convolutional layers, three fully connected CNNs with 3×3 filters with 1 step size and padding, and vice versa. There is a 2×2 maximum pooling layer, data augmentation was also developed and used as additional dataset generated from the existing images. As for data augmentation, two methods were used: augmentation by mirroring or creating a mirror image, and

augmentation by random crops. The experimental results and analysis will be discussed in the next section.

6. RESULTS AND DISCUSSION

This section illustrates details concerning the proposed approaches' implementation and evaluation. The proposed approaches were developed using Python 3.8.8, Google Colab, and the deep learning frameworks Keras and Tensorflow. Win 11 Pro 64 bit OS, Intel(R) Core (TM) i5-9300HF CPU 2.40 GHz, 8 GB RAM.

6.1. Evaluation Metrics

To evaluate the performance of proposed Arabic handwritten character recognition system based on confusion matrix, calculation of matrices like accuracy, precision (also called Positive Predictive Value (PPV)), Specificity (SPE), recall (called Sensitivity (SEN)), F1-score, Mathews Correlation Coefficient (MCC), and Negative Predictive Value (NPN) were used. These indices have the following mathematical definition:

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

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

$$PPV = \frac{TP}{TP+FP} \quad (6)$$

$$F1_score = \frac{TP+TN}{TP+FN+TN+FP} \quad (7)$$

$$NPN = \frac{TN}{TN+FN} \quad (8)$$

$$MCC = \frac{(TP \times FN) - (FP \times FP)}{\sqrt{(TP+FP) \times (TN+FN) \times (TP+FN) \times (TN+FP)}} \quad (9)$$

$$ACC = \frac{TP+TN}{TP+TN+FN+FP} \quad (10)$$

Where TP indicates to true positive

TN indicates to true negatives.

FP indicates to false positives.

FN indicates to false negatives. These parameters can be calculated using confusion matrices for both binary and multiclass problems [28-32].

6.2 Results

The proposed CNN model with Pertained models ResNet 50, Inception V3, DenseNet 121 and VGG16 are implemented on the collected dataset. The experiments have been explained and discussed in two stages: first, the results of CNN were proposed on our database, and secondly, the description of experiments has been applied on learning transfer approaches re-trained on our dataset.

First, regarding the CNN model results on the AHCD dataset and our dataset, in which the total images are 131000 character images. The two datasets are split into 80% training, 10% testing and 10% validation. Arabic characters were also represented with sequential numbers of (0 - 27), where 0 represents 'أ', 1 represents 'ب', 2 represents 'ت', and so on. To facilitate the carry-out process, using the equation (4), (6) and (10), the proposed CNN model obtained a test set accuracy of around 88 %, precision (Positive Predictive Value (PPV)) of 87.88%, Sensitivity (SEN) of 87.81%, and an F1-score of 87.8%, shown in table 2.

Table 2: Experiments results of CNN on for each character

NO	Character	PPV	SEN	F1_score	No	Character	PPV	SEN	F1_score
0	أ	0.99	98.	0.98	14	ض	0.84	0.86	0.85
1	ب	0.92	0.97	0.94	15	ط	0.92	0.92	0.92
2	ت	0.89	0.89	0.89	16	ظ	0.93	0.92	0.93
3	ث	0.90	0.87	0.88	17	ع	0.79	0.79	0.79
4	ج	0.89	0.92	0.91	18	غ	0.79	0.79	0.84
5	ح	0.85	0.78	0.81	79	ف	74	0.83	0.78
6	خ	0.88	0.84	0.86	20	ق	0.88	0.86	0.87
7	د	0.82	0.74	0.78	21	ك	0.87	0.90	0.89
8	ذ	0.74	0.74	0.74	22	ل	0.91	0.94	0.93
9	ر	0.83	0.93	0.88	23	م	0.89	0.90	0.89
10	ز	0.87	0.88	0.88	24	ن	0.84	0.82	0.83
11	س	0.93	0.93	0.93	25	هـ	0.86	0.87	0.86
12	ش	0.90	0.93	0.92	26	و	0.90	0.89	0.89
13	ص	0.84	0.88	0.86	27	ي	0.95	0.92	0.93
Train_acc		93 %							
Test_acc		88 %							

The transfer learning approaches are applied to the same database, the database is used as a target for pretrain transfer learning models, while the ImageNet dataset is used as the source dataset to train these models randomly.

Two experiments have been done. First, evaluating CNN and its different architectures separately to recognize Arabic handwritten characters, and secondly, evaluating all models in the Arabic characters classification. Table 3 shows the Multi-class evaluation matrices' results of accuracy, Positive Predictive Value, Sensitivity, F1-score, Mathews Correlation Coefficient and Negative Predictive Value which the trained dataset obtained is used in transfer learning techniques. Moreover, the table displays the runtime taken from each of the Deep CNN architectures.

Table 3: Balanced Multi-class evaluation metrics for CNN and each model on the combined dataset

Confusion Matrices Models	SEN	SPE	MCC	F1_score	PPV	NPV	ACC	Runtime (Hours)	
								Train	Test
DenseNet121	100	99.52	96.44	96.43	93.75	100	98.42	13:34	1:49
InceptionV3	100	99.52	96.44	96.43	93.75	100	98.94	7:14	2:48
ResNet50	100	99.71	97.69	96.73	95.83	100	97.90	0:54	0:15
VGG16	100	97.44	93.52	91.49	92.84	99	97.71	22:53	4:55
CNN		87.81			87.88		93	0:25	0:4

Fig 5: Accuracy Training and Testing for Arabic characters

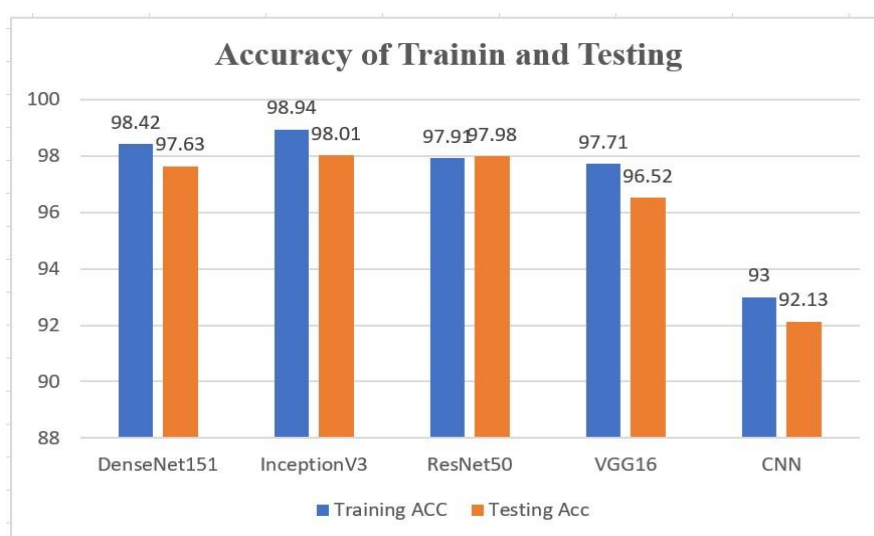


Table 3 exhibits the performance matrix computed using Equations (4 - 10). Comparisons are described in terms of accuracy and the time spent in carrying out the process. . The Inception V3 architecture outperformed the VGG, ResNet and CNN models on the combined dataset (AHCD and our dataset). Inception V3 based model achieved accuracy of 99%, precision as 93.75%, recall (SEN) 100 %, Specificity as 99.8%; the total runtime for training is almost 7 hours. While the VGG16 architecture achieved accuracy as 97.71%, 93.75%, and sensitivity 100%, Specificity (SPE) 99.8%. Comparatively higher training time of 23 hours. Therefore, in this architecture, the images of the combined dataset have been reduced from 131,000 images to 13346 images. The reason for such reduction is that Google Collab (and Spyder.4 [MSC V.1916 64 bits]) can't execute the classification or recognition process on 131,000 images using VGG16, as this VGG requests a high feature of computers. The proposed CNN is lightweight in nature when compared different models which are being used in the proposed work. The performance of CNN based model as accuracy of 93%, Positive Predictive Value 87.88%, Specificity 87.81%. CNN model's training time is nearly 25 minutes on the dataset A and

dataset B. Fig 7 displays Arabic handwritten characters recognition accuracy for the proposed CNN and transfer learning approaches.

We used two methods in regularization to prevent overfitting. First, dropout, which in some of the output nodes of a layer are ignored randomly (This technique in which random discarding occurs in the output nodes of the layer). During the training, the connections to that node are such that the network behaves as if this node does not exist. When the weights are updated, the network appears as a new network because the discarded nodes are not the same on each iteration. We made the dropout layer as 0.50 for all the transfer learning approaches. Second, Batch normalization is the process of normalizing the inputs in each layer. It is used to reduce the internal covariance shift in hidden layers because, in a deep network, there may be an internal covariance in the hidden layers due to the various inputs of the network. A value of Batch was set to 64 for all transfer learning algorithms. The data augmentation method is used when the dataset is small. In this work, this technique increases the input data. It causes crowding of input data during the training process, making the machine learning process take a very, very long time up to days. The implementation stops training for devices with low key features, and the suspension problem occurs, especially while using VGG16. For that, we do not use any data augmentation.

Fig 6: (a) Accuracy and (b) loss of DenseNet 121

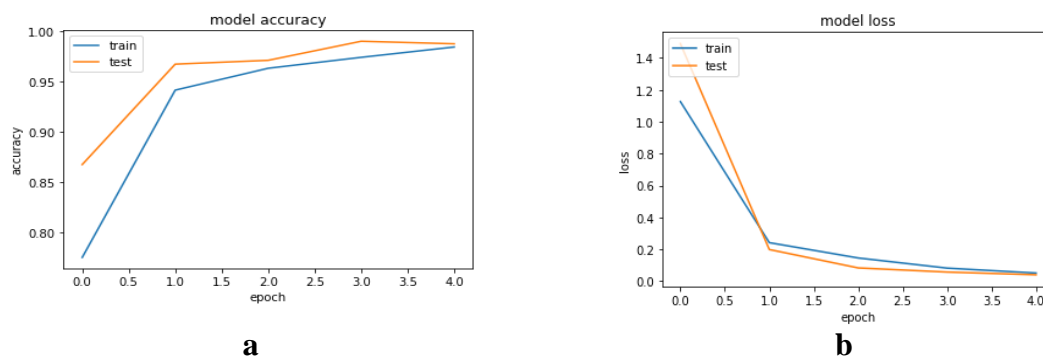


Fig 7: (a) Accuracy and (b) loss of Inception V3

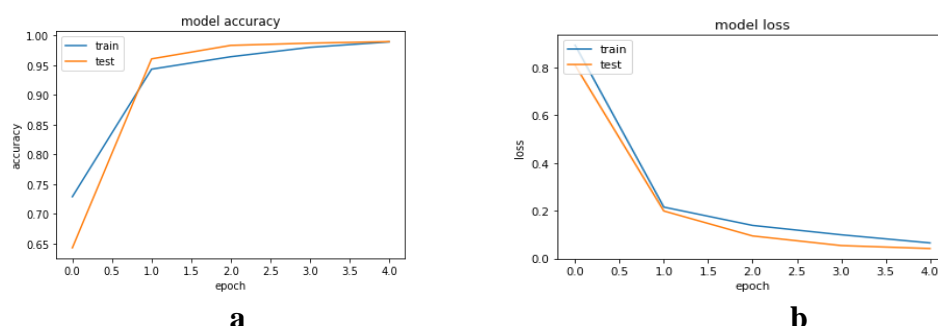


Fig 8: (a) Accuracy and (b) loss of ResNet_50

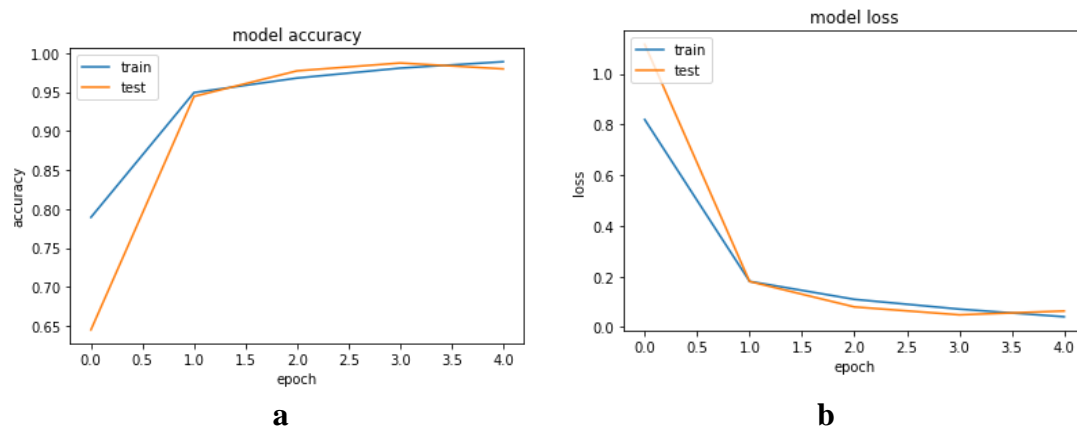
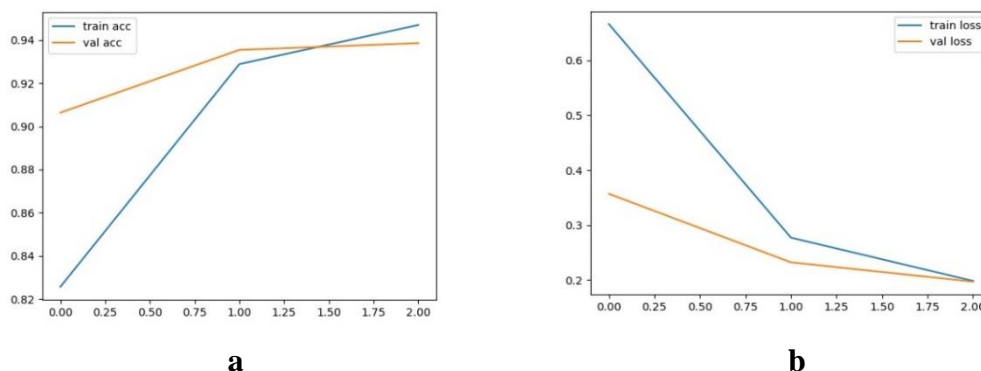


Fig 9: (a) Accuracy and (b) loss of VGG16



Because of the time spent on training and testing processes, especially in VGG16, the epochs of training have been set be 5. The accuracy of training and testing was documented and loss in shapes from 6 to 6.10. These Figs display good results. Fig 6 (a) showed that the test process was better than the training. Thus, the training result was acceptable. Likewise, (b) of Fig 6 in the loss, as the model gave a loss in the test less than training in all training repetitions. Fig 7 the training provided a better accuracy during the training and test process, the two curved lines of train and test are slightly closed together, which means that this model is perfect to recognize Arabic handwritten characters. However, the dropout was 0.5 in all used models. If the epochs increase, the prediction of this model will be better. As shown in Fig 7, ResNet-50, higher accuracy of train and test was shown as well as loss that is lower than VGG16 and CNN model. Thus, the two models, Inception V3 and ResNet-50, achieved better performance in updating the bias and weights to the recognition performance.

The following tables report the evaluation matrices reports for each model used in classifying Arabic handwritten characters, providing the comparison as well.

Table 4: Evaluation Matrices for DenseNet 121 character classification

character	SEN	SPE	ACC	MCC	F1-score	PPV	NPV
0. alif	100	98.09	93.19	0.97	95.71	96.75	100
1. baa	100	100	100	1	100	100	100
2. taa	100	100	100	1	100	100	100
3. thaa	100	100	100	1	100	100	100
4. jeem	100	100	98.55	1	100	100	100
5. hhaa	100	100	98.55	1	100	100	100
6. khaa	100	100	98.55	1	100	100	100
7. daal	100	100	98.55	1	100	100	100
8. thaal	100	100	99.88	1	98.67	100	99.64
9. raa	98.64	99.64	99.76	0.94	94.43	95.43	99.88
10. zay	98.64	99.64	99.4	0.94	98.43	98.43	99.28
11. seen	98.64	99.64	97.88	0.94	94.43	95.43	99.24
12. sheen	98.64	94.79	96.12	0.91	94.43	95.43	94.81
13. saad	99.64	99.62	98.36	0.94	94.43	95.43	99.51
14. zhaad	97.64	98.67	99.64	0.51	98.92	98.92	99.28
15. tta	98.64	99.64	99.76	0.94	98.83	98.92	99.27
16. dhaa	99.64	99.52	99.88	0.98	99.52	99.52	99.39
17. ain	98.84	99.39	99.88	0.93	99.62	99.52	99.4
18. gain	99.74	99.76	95.52	0.96	99.76	99.72	99.27
19. faa	99.03	99.03	99.88	0.51	99.03	99.52	99.28
20. qaaf	99.03	99.64	96.76	0.94	99.64	99.64	99.24
21. kaaf	93.45	94.67	98.92	0.91	94.63	93.63	94.4
22. laam	93.45	94.67	94.98	0.91	94.63	95.22	94.27
23. meem	93.45	94.28	98.29	0.91	93.43	91.66	94.27
24. noon	93.45	94.27	98.17	0.91	94.43	95.43	94.26
25. haa	93.45	94.27	96.17	0.94	94.43	95.43	94.01
26. waw	93.45	94.02	97.93	0.91	93.43	91.66	94.14
27. yaa	94.13	100	95.81	0.94	100	99.00	94.46

Table 5: Evaluation Matrices for Inception V3 character classification

character	SEN	SPE	ACC	MCC	F1-score	PPV	NPV
0. alif	100	98.09	98.19	0.857	85.71	75	100
1. baa	100	100	100	1	100	100	100
2. taa	100	100	100	1	100	100	100
3. thaa	100	100	100	1	100	100	100
4. jeem	100	100	100	1	100	100	100
5. hhaa	100	100	100	1	100	100	100
6. khaa	100	100	100	1	100	100	100
7. daal	100	100	100	1	100	100	100
8. thaal	100	99.88	99.88	0.925	92.31	85.71	100
9. raa	66.67	100	99.76	0.815	80	100	99.76
10. zay	33.33	99.88	99.4	0.688	44.44	66.67	99.52
11. seen	100	99.88	99.88	0.925	92.31	85.71	100
12. sheen	100	100	100	1	100	100	100
13. saad	66.67	100	99.76	0.815	80	100	99.76
14. zhaad	50	100	99.64	0.758	66.67	100	99.64

15. tta	83.33	99.88	99.76	0.832	83.33	83.33	99.88
16. dhaa	80	100	99.88	0.899	88.89	100	99.88
17. ain	80	100	99.88	0.899	88.89	100	99.88
18. gain	50	99.88	99.52	0.611	60	75	99.64
19. faa	83.33	100	99.88	0.913	90.91	100	99.88
20. qaaf	66.67	100	99.76	0.855	80	100	99.76
21. kaaf	100	100	100	1	100	100	100
22. laam	100	100	100	1	100	100	100
23. meem	100	100	100	1	100	100	100
24. noon	100	100	100	1	100	100	100
25. haa	100	100	100	1	100	100	100
26. waw	100	100	100	1	100	100	100
27. yaa	100	100	100	1	100	100	100

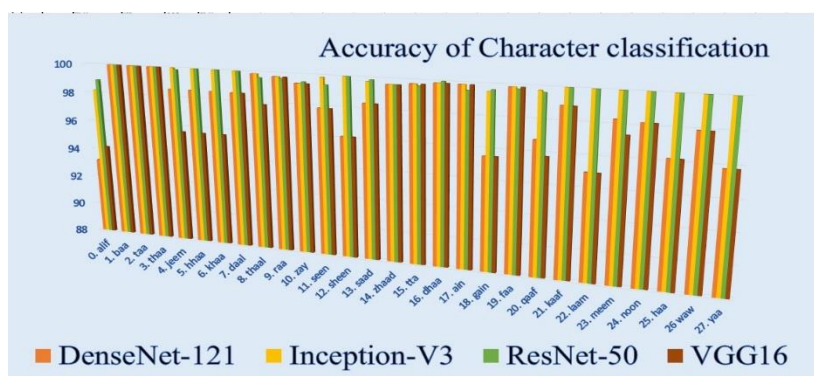
Table 6: Evaluation Matrices for ResNet-50 character classification

character	SEN	SPE	ACC	MCC	F1-score	PPV	NPV
0. alif	100	98.85	98.92	0.976	90.91	83.33	100
1. baa	100	100	100	1	100	100	100
2. taa	100	100	100	1	100.	100	100
3. thaa	100	100	100	1	100	100	100
4. jeem	98.88	100	99.88	0.997	99.44	100	99.87
5. hhaa	100	100	100	1	100	100	100
6. khaa	100	100	100	1	100	100	100
7. daal	100	100	100	1	100	100	100
8. thaal	50	100	99.64	0.758	66.67	100	99.64
9. raa	83.33	99.76	99.64	0.769	76.92	71.43	99.88
10. zay	66.67	99.76	99.52	0.664	66.67	66.67	99.76
11. seen	33.33	99.88	99.4	0.488	44.44	66.67	99.52
12. sheen	100	100	100	1	100	100.	100
13. saad	83.33	100	99.88	0.912	90.91	100	99.88
14. zhaad	66.67	99.88	99.64	0.785	72.73	80	99.76
15. tta	50	100	99.64	0.758	66.67	100	99.64
16. dhaa	100	100	100	1	100	100	100
17. ain	100	99.52	99.52	0.745	71.43	55.56	100
18. gain	50	100	99.64	0.758	66.67	100	99.64
19. faa	83.33	99.88	99.76	0.831	83.33	83.33	99.88
20. qaaf	83.33	99.76	99.64	0.797	76.92	71.43	99.88
21. kaaf	100	100	100	1	100	100	100
22. laam	100	100	100	1	100	100	100
23. meem	100	100	100	1	100	100	100
24. noon	100	100	100	1	100	100	100
25. haa	100	100	100	1	100	100	100
26. waw	100	100	100	1	100	100	100
27. yaa	100	100	100	1	100	100.	100

Table 7: Evaluation Matrices for VGG16 character classification

Character	SEN	SPE	ACC	MCC	F1-score	PPV	NPV
0. alif	100	98.85	94.19	0.976	90.91	83.33	100
1. baa	100	100	100	1	100	100	00.
2. taa	100	100	100	1	100.	100	100
3. thaa	100	100	100	1	100	100	100
4. jeem	98.88	100	95.65	0.997	99.44	100	99.87
5. hhaa	100	100	95.65	1	100	100	100
6. khaa	100	100	95.65	1	100	100	100
7. daal	100	100	98.55	1	100	100	100
8. thaal	50	100	97.88	0.758	66.67	100	99.64
9. raa	83.33	99.76	99.76	0.767	76.92	71.43	99.88
10. zay	66.67	99.76	99.4	0.664	66.67	66.67	99.76
11. seen	33.33	99.88	97.88	0.568	44.44	66.67	99.52
12. sheen	100	100	96.12	1	100	100.	100
13. saad	83.33	100	98.36	0.913	90.91	100	99.88
14. zhaad	66.67	99.88	99.64	0.785	72.73	80	99.76
15. tta	50	100	99.76	0.758	66.67	100	99.64
16. dhaa	100	100	99.88	1	100	100	100
17. ain	100	99.52	99.88	0.745	71.43	55.56	100
18. gain	50	100	95.52	0.754	66.67	100	99.64
19. faa	83.33	99.88	99.88	0.832	83.33	83.33	99.88
20. qaaf	83.33	99.76	95.76	0.767	76.92	71.43	99.88
21. kaaf	100	100	98.92	1	100	100	100
22. laam	100	100	94.98	1	100	100	100
23. meem	100	100	97.39	1	100	100	100
24. noon	100	100	98.17	1	100	100	100
25. haa	100	100	96.17	1	100	100	100
26 waw	100	100	97.93	1	100	100	100
27. yaa	100	100	95.81	1	100	100.	100

Fig 10: Arabic Character classification



By the way, Matthews Correlation Coefficient (MCC) is a statistical method used to evaluate a model, and it measures the difference between the predicted value and the actual value. Using equation (9) to obtain the MCC value. The MCC value is usually between -1 to 1. The closer

the outcome value to 1 means that the classification performance is excellent [33]. From the experiment results, we noticed that the proposed CNN model for recognizing isolated Arabic handwritten characters is an approach that can be promising and model methods to apply in the application of Arabic text recognition. This approach achieved high performance compared with traditional methods for Arabic text recognition. The most significant advantage of this approach is that it can classify and identify a large dataset. Accuracy (91-95) in identifying more than 131,000 images of Arabic letters. The misclassification occurred was due to the original image's quality. The tables from 3 - 6 display the evaluation matrices measures of each mode's performance for Arabic handwritten characters' classification.

As noticed in Fig 7, Inception V3 provides better classification. There are similarities in shapes between the characters such as 'ع', 'غ', 'ك', 'ل', 'ض', 'ص', 'ب', 'ي', 'ف', 'ق', 'س', 'ش', 'ن' in handwriting. So DenseNet 121 and VGG16 faced challenge to classify these characters. Based on the results explained above, it can be see that the CNN method and deep CNN approaches are perfect models that can be used in other OCR systems for Arabic text recognition.

6.3 Comparison

Table 8: Exhibits the comparisons between previous related studies that used the AHCD database with the proposed models

Reference	Year	Method	Size of images	Accuracy
[4]	2017	CNN	16800	94.9%
[9]	2018	Deep CNN	16800	97.6%
[7]	2021	CNN with data augmentation	16800	97.6%
Our approach (Inception V3 and ResNet 50)	2022	Transfer Learning approaches	16800 + 114200	98.94%

7. CONCLUSION

Classifying Arabic handwritten characters is a challenging task. In this paper, the classification of Arabic handwritten character images was analyzed using four deep learning approaches, which are DenseNet 121, ResNet-50, Inception V3, and VGG16 approaches of deep transfer learning. Furthermore, the proposed CNN model evaluates and verifies the ability of each model to distinguish and recognize Arabic characters. The proposed approaches were applied to 131,000 images of Arabic handwritten characters. The dataset was split into 80% training, 10% tests and 10% verification. A comparison was made between the results of the proposed models and the selection of the optimal model for recognizing Arabic characters, as shown in Figs 7 and 8. Each learning transfer model was already trained on the ImageNet database. The same weights were used in training on our dataset. The results of tables 2, 4-7 and Figs (5 10) are clarified that the Inception V3 and ResNet-50 models are the most suitable models for achieving high accuracy of almost 99% in training and 98% testing. Moreover, the perfect model for classifying Arabic characters is the DenseNet 121 model, which achieved a satisfactory accuracy rate in recognizing Arabic characters. The model VGG16 is a robust and

more reliable model for classifying Arabic characters. Still, it needs high computer specifications and a very long task process, up to days depending on the dataset's size.

Reference

- [1] Muaad, A.Y., et al., ArCAR: A Novel Deep Learning Computer-Aided Recognition for Character-Level Arabic Text Representation and Recognition. *Algorithms*, 2021. **14**(7): p. 216.
- [2] Almisreb, A.A., et al., Arabic Handwriting Classification using Deep Transfer Learning Techniques. *Pertanika Journal of Science & Technology*, 2022. **30**(1).
- [3] Sushma, L. and K. Lakshmi, An analysis of convolution neural network for image classification using different models. *Int J Eng Res Technol (IJERT)*, 2020. **9**(10).
- [4] El-Sawy, A., M. Loey, and H. El-Bakry, Arabic handwritten characters recognition using convolutional neural network. *WSEAS Transactions on Computer Research*, 2017. **5**(1): p. 11-19.
- [5] Aneja, N. and S. Aneja. Transfer learning using CNN for handwritten devanagari character recognition. in *2019 1st International Conference on Advances in Information Technology (ICAIT)*. 2019. IEEE.
- [6] Awni, M., M.I. Khalil, and H.M. Abbas, Offline Arabic handwritten word recognition: A transfer learning approach. *Journal of King Saud University-Computer and Information Sciences*, 2021.
- [7] AlJarrah, M.N., M.Z. Mo'ath, and R. Duwairi. Arabic Handwritten Characters Recognition Using Convolutional Neural Network. in *2021 12th International Conference on Information and Communication Systems (ICICS)*. 2021. IEEE.
- [8] Khan, M.A., Arabic handwritten alphabets, words and paragraphs per user (AHAWP) dataset. *Data in Brief*, 2022. **41**: p. 107947.
- [9] Younis, K.S., Arabic handwritten character recognition based on deep convolutional neural networks. *Jordanian Journal of Computers and Information Technology (JJCIT)*, 2017. **3**(3): p. 186-200.
- [10] Aneja, S., N. Aneja, and M.S. Islam. IoT device fingerprint using deep learning. in *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*. 2018. IEEE.
- [11] Lawgali, A., A survey on Arabic character recognition. 2015.
- [12] Yousif, I. and A. Shaout, Off-Line handwriting Arabic text recognition: a survey. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2014. **4**(9).
- [13] Noubigh, Z., A. Mezghani, and M. Kherallah. Transfer Learning to improve Arabic handwriting text Recognition. in *2020 21st International Arab Conference on Information Technology (ACIT)*. 2020. IEEE.
- [14] Jiang, Z.-P., et al., An Improved VGG16 Model for Pneumonia Image Classification. *Applied Sciences*, 2021. **11**(23): p. 11185.
- [15] Wang, G., The status of Chinese handwriting identification and the improvement of methodologies. *Forensic Sci*, 2019. **4**: p. 1-7.
- [16] Goularas, D. and S. Kamis. Evaluation of deep learning techniques in sentiment analysis from twitter data. in *2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)*. 2019. IEEE.
- [17] Pouyanfar, S., et al., A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 2018. **51**(5): p. 1-36.
- [18] He, K., et al. Deep residual learning for image recognition. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

- [19] Krizhevsky, A., I. Sutskever, and G.E. Hinton, Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 2012. **25**.
- [20] Mudhsh, M. and R. Almodfer, Arabic handwritten alphanumeric character recognition using very deep neural network. *Information*, 2017. **8**(3): p. 105.
- [21] Altwaijry, N. and I. Al-Turaiki, Arabic handwriting recognition system using convolutional neural network. *Neural Computing and Applications*, 2021. **33**(7): p. 2249-2261.
- [22] Boraik, O.A., M. Ravikumar, and M.A.N. Saif, Characters Segmentation from Arabic Handwritten Document Images: Hybrid Approach.
- [23] Russakovsky, O., et al., Imagenet large scale visual recognition challenge. *International journal of computer vision*, 2015. **115**(3): p. 211-252.
- [24] Aytar, Y. and A. Zisserman. Tabula rasa: Model transfer for object category detection. in 2011 international conference on computer vision. 2011. IEEE.
- [25] Huang, G., et al. Densely connected convolutional networks. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
- [26] Szegedy, C., et al. Going deeper with convolutions. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
- [27] Jaderberg, M., K. Simonyan, and A. Zisserman, Spatial transformer networks. *Advances in neural information processing systems*, 2015. **28**.
- [28] Al-Antari, M.A., S.-M. Han, and T.-S. Kim, Evaluation of deep learning detection and classification towards computer-aided diagnosis of breast lesions in digital X-ray mammograms. *Computer methods and programs in biomedicine*, 2020. **196**: p. 105584.
- [29] Al-Antari, M.A., et al., A fully integrated computer-aided diagnosis system for digital X-ray mammograms via deep learning detection, segmentation, and classification. *International journal of medical informatics*, 2018. **117**: p. 44-54.
- [30] Al-Antari, M.A., et al., Fast deep learning computer-aided diagnosis of COVID-19 based on digital chest x-ray images. *Applied Intelligence*, 2021. **51**(5): p. 2890-2907.
- [31] Al-Masni, M.A., et al., Skin lesion segmentation in dermoscopy images via deep full resolution convolutional networks. *Computer methods and programs in biomedicine*, 2018. **162**: p. 221-231.
- [32] Al-Masni, M.A., et al., Simultaneous detection and classification of breast masses in digital mammograms via a deep learning YOLO-based CAD system. *Computer methods and programs in biomedicine*, 2018. **157**: p. 85-94.
- [33] Chicco, D., N. Tötsch, and G. Jurman, The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *BioData mining*, 2021. **14**(1): p. 1-22.