Handwritten Arabic Character Recognition for Children Writing Using Convolutional Neural Network and Stroke Identification

Mais Alheraki1 · Rawan Al-Matham1 · Hend Al-Khalifa1

Received: 20 April 2023 / Accepted: 29 April 2023 / Published online: 13 May 2023
© The Author(s) 2023

Abstract
Automatic Arabic handwritten recognition is one of the recently studied problems in the field of Machine Learning. Unlike Latin languages, Arabic is a Semitic language that forms a harder challenge, especially with the variability of patterns caused by factors such as the writer’s age. Most studies have focused on adults, with only one recent study on children. Moreover, many recent machine-learning methods have focused on using Convolutional Neural Networks (CNNs), a powerful class of neural networks that can extract complex features from images. In this paper, we propose a convolutional neural network (CNN) model that recognizes children’s handwriting with an accuracy of 91% on the Hijja dataset, a recent dataset built by collecting images of Arabic characters written by children, and 97% on the Arabic Handwritten Character Dataset. The results showed a good improvement over the proposed model from the Hijja dataset authors, yet it revealed a bigger challenge to solve for children’s Arabic handwritten character recognition. Moreover, we proposed a new approach using multiple models instead of a single model based on the number of strokes in a character and merged Hijja with AHCD, which achieved an average prediction accuracy of 96%.

Keywords Convolutional neural network · CNN · Deep learning · Optical character recognition · OCR · Arabic characters

Abbreviations
ML Machine Learning
DL Deep Learning
CNN Convolutional Neural Network
Hijja Hijja dataset
AHCD Arabic Handwritten Character Dataset

1 Introduction
Handwriting recognition is a computer vision problem involving the automation of identifying handwritten script by a computer, which transforms the text from sources such as documents or touchscreens into a machine-understandable format. The input image can be offline, originating from a piece of paper or a photograph, or online if the source is digital, such as touchscreens [1].

Handwritten text for each language has many different patterns and styles from writer to writer, affected by factors such as age, background, native language, and mental state [2]. Automatic handwritten recognition has been extensively investigated using various machine learning methods, including K-nearest Neighbors (KNNs) [3], Support Vector Machines (SVMs), transfer learning [4, 5], and deep learning techniques such as Neural Networks (NNs) [6]. Recently, most studies have employed Convolutional Neural Networks (CNNs) [5, 7–9].

Latin languages have been intensively studied in the literature. Achieving state-of-the-art results [3, 10, 11]. However, the Arabic language, a Semitic language and the fourth most spoken language in the world [12], requires further investigation. Arabic has unique features, including spelling, grammar, and pronunciation, which distinguish it from other languages. Arabic writing is semi-cursive and written from right to left, with 28 characters in the alphabet. Each character has multiple shapes depending on its position in the word, making automatic handwritten recognition of Arabic script more challenging than other languages. These aspects make the automatic handwritten recognition of the Arabic
script harder than other languages. Many recent studies were conducted targeting Arabic handwritten recognition [13–15]. However, all of them focused on recognizing Arabic adult script except for [7]. They created Hijja dataset which was collected from 591 children from Arabic schools. Additionally, they proposed a CNN model to evaluate their dataset. Their achieved prediction accuracy was 87%.

In this research, we aim to improve the prediction accuracy over Hijja dataset using CNN to have a more robust model that can recognize children's Arabic script. Our experiments will answer the following research questions:

1. Using our newly proposed CNN architecture, can we enhance the accuracy for children's Arabic handwritten character recognition?
2. Using character strokes as a filter, can we enhance the accuracy for children's Arabic handwritten character recognition?

Therefore, our main contributions and novel findings are as follows:

1. We propose a new CNN architecture specifically designed for children's Arabic handwritten character recognition. This architecture demonstrates significant improvements in prediction accuracy over the Hijja dataset compared to the existing models.
2. We introduce a novel approach that uses character strokes as a filter to further enhance the accuracy of children's Arabic handwritten character recognition. This approach demonstrates the effectiveness of utilizing stroke information in conjunction with CNNs for improving recognition performance.

The rest of the paper is organized as follows: Section 2 provides background information on the Arabic language and script, Optical Character Recognition, and Convolutional Neural Networks. Section 3 offers an overview of the related work, our methodology, including the proposed solution, the utilized dataset, and the experimental setup. Results from our experimentation are presented in Sect. 4. Section 5 discusses and analyzes the results, and Sect. 6 concludes the paper, highlighting potential future work.

2 Background

In this section, we present the necessary background information needed to explain the underlying concepts of this research, including the Arabic language and Arabic script, optical character recognition, and Convolutional Neural Network.

2.1 Arabic Language and Arabic Script

Arabic is a Semitic language and the language of the Holy Qur’an. Almost 500 million people around the globe speak Arabic, and it is the language officially used in many Arab countries with different dialects. The formal written Arabic is Modern Standard Arabic (MSA). MSA is one form of classical Arabic, which is the language that was used in Qur’an, and it currently has a larger and modernized vocabulary. Because it is understood by almost everyone in the Arab world, it is being used as a formal language in the media and education. Arabic has its own features that differentiate it from other languages, including spelling, grammar, and pronunciation [2].

The calligraphic nature of Arabic script differs from that of other languages in many ways. Arabic writing is semi-cursive, and it is written from right to left. The Arabic alphabet consists of 28 characters. Their shapes change depending on the position in the word. There are 16 characters that contain dots (one, two, or three) among the 28 Arabic alphabets. These dots appear either above or below the character. Some characters may have the same body but with different dots number and/or position, as shown in Fig. 1.

Arabic characters have different shapes depending on their position in a word: initial, medial, final, or stand-alone. The initial and medial shapes are typically similar, as are the final and stand-alone shapes (see Table 1). These Aspects make recognizing Arabic script tasks more challenging than recognizing Latin script tasks. Because of this, fewer resources are created for this task, and thus the state-of-the-art is less advanced. Nevertheless, there have been some efforts to recognize Arabic handwriting in the last few years, which are covered in Sect. 3.

Fig. 1 The Arabic characters, Hamza is colored by red as it is not part of the 28 alphabets
2.2 Optical Character Recognition

Optical Character Recognition (OCR) is a pattern recognition problem that takes printed or handwritten text as an input and creates an editable machine-understandable format of text extracted from the scanned image. OCR can be used in many applications such as advanced document scanning, business applications, electronic data searching, data entry, systems for visually challenged persons, document verification, document automation, data mining, biometrics, and text storage optimization [16].

OCR are classified into offline and online recognitions. This depends on the type of data used as input for recognition. In offline recognition, the input is only an image of handwritten text with less information. In Online recognition, a special input device, such as, an electronic pen, tracks the movement of the pen during the writing process. Offline recognition is usually more challenging than online recognition [1].

2.3 Convolutional Neural Network

Convolutional Neural Network is a special type of neural network widely used in deep learning to extract features from visual data. CNNs have shown state-of-the-art results in many image classification problems. In any computer vision problem, CNNs are the best candidates for achieving considerably high accuracy compared with other machine learning (ML) algorithms. A CNN takes an image as an input in the form of a 3D matrix, with width, height, and channels, and several filters are applied to this matrix, which are called kernels and have different types to extract different features on each convolutional layer. Here are several layers in a CNN besides convolutions, which can be different from network to network, but the deeper a CNN the numerous weights it will have; therefore, the pooling layer reduces the size of the convoluted layers by either finding the maximum pixel value in a window or the average of all values, which is important for decreasing the computational resources needed in CNNs. The other layers include the activation layer, which could be ReLU or any other activation and normalization [4].

3 Literature Review

In this section, we present some of the existing datasets that were used for Arabic handwritten recognition in the literature. Additionally, some studies used CNNs for Arabic handwriting recognition.

3.1 Arabic Handwritten Recognition Datasets

There are many datasets created for Arabic handwritten recognition. One of these was introduced in [17]. It contains 5,600 images written by 50 adult writers, including a variety of shapes for each character. The DBAHCL dataset was introduced in [18]. It includes 9900 ligatures and 5500 handwritten characters and ligatures written by 50 writers.

Another study was conducted to collect Arabic handwritten diacritics (DBAHD) [19]. Another dataset, AHCD, was introduced in [20]. It includes 16800 characters written by 60 adult writers. The character images are only in isolated form [Arabic Handwritten Characters Recognition using Convolutional Neural Network] and it has been used in many studies [7, 20, 21].

Finally, the Hijja dataset was introduced and experimented with in [7] which includes 47,434 characters written by 591 children. The characters are written in isolated and connected forms, and this is the largest existing dataset. We chose the last two datasets (Hijja and AHCD) for this study.

3.2 CNN in Arabic Handwritten Recognition

In this section, we present some of the studies that used CNNs for Arabic handwriting recognition. Elleuch et al. [22] investigated two types of neural network for Arabic handwritten recognition. These are Deep Belief Network (DBN) and Convolutional Neural Networks (CNNs). The two networks used a greedy layer-wise unsupervised learning algorithm for processing. The experiments were performed on HACDB Dataset and CNN obtained the best results with 14.71 % error classification rate on the test set.

Similarly, In [9], El-Sawy et al. designed and optimized a CNN classifier by considering the learning rate and activation function (ReLU function). Their experiments were
performed on AHCD dataset, and they achieved an accuracy of 94.9% on the testing data.

Additionally, in [20], El-Sawy et al. aimed to recognize Arabic digits using a CNN based on LeNet-5 to recognize Arabic digits. Their experiments were conducted using the MADBBase database (Arabic handwritten digits images). Their model achieved high accuracy with 1% training mis-classification error rate and 12% testing miss classification error rate.

Amrouch et al. [23] used a CNN as an automatic feature extractor in the preprocessing stage and Hidden Markov Models (HMM) as the recognizer. This made the feature extraction process easier, faster, and more accurate than manual feature extraction. They achieved 89.23% accuracy.

In [5], Soumia et al. compared two approaches to Arabic handwritten character recognition. The first uses conventional machine learning with the SVM classifier. The second one used Transfer learning with ResNet, Inception V3, and VGG16 models. They also proposed a new CNN architecture and tested it. The best accuracy results were achieved with their CNN model, and 94.7, 98.3, and 95.2% were achieved for OIHACDB-28, OIHACDB-40, and AIA9K, respectively. However, all previous studies have focused on recognizing adults’ Arabic handwriting scripts.

In a previous study [24], Ahmed et al. presented CNN context-based architecture to recognize Arabic letters, words, and digits. They experimented with MADBase, CMATERDB, HACDB and SUST-ALT datasets. They aimed to reach the highest possible testing accuracy and they achieved 99% for digits, 99% for letters and 99% for words on 99% dataset. However, their proposed model was designed for offline recognition while they aim to develop a robust and lightweighted model that can be used for online recognition for Arabic children handwritten writing in real-life scenarios.

In [25], Balaha et al. proposed two different approaches. The first used 14 different CNN architectures on the HMDB dataset, and the best-acquired testing accuracy was 91.96%. Their second approach was to use transfer learning (TF) and a genetic algorithm (GA) approach named “HMB-AHCR-DLGA” was proposed to optimize the training parameters and hyperparameters in the recognition phase. The pretrained CNN models (VGG16, VGG19, and MobileNetV2) were used in the second approach. The highest achieved testing accuracy was 92.88%.

There was one study focused on children’s Arabic handwriting script [7]. Altwajiry et al. described a newly collected dataset of Arabic letters written by children aged 7–12 years. The dataset is called Hijja and includes 47,434 characters written by 591 participants. Also, they proposed a CNN model for Arabic handwritten recognition that was trained and tested on the Hijja and Arabic Handwritten Character Dataset (AHCD). Their model achieved accuracies of 97% and 88% on the AHCD dataset and the Hijja dataset, respectively.

All these studies proved that CNNs are the most suitable approach for Arabic handwritten recognition because of their power in automatic feature extraction, which makes it possible to understand difficult patterns in handwritten text. Therefore, we decided to use it in this research with a new proposed architecture to enhance the prediction accuracy on the Hijja dataset to recognize children’s handwriting scripts. We also decided to invert the images before feeding them into the model, as it had positive improvements in [15].

### 3.3 Deep learning Models for Handwritten Recognition

In this section, we will briefly cover a sample of papers that have employed deep learning techniques in handwritten recognition, highlighting their contributions and the impact they have made in advancing the field.

Weiwei Jiang and Le Zhang in their paper [26] proposed two new deep learning models, Edge-SiamNet and Edge-TripleNet, for handwritten numeral recognition. The models combine edge extraction and Siamese/Triple network structures and achieve state-of-the-art performance on seven different datasets. The results demonstrate the simplicity and effectiveness of the proposed models, which improve the network’s performance of recognizing different numbers without introducing more parameters than EdgeNet.

Another paper by [27] presents a deep convolutional neural network (CNN) architecture for character recognition in Bangla and Manipuri scripts. The proposed model achieves better accuracy than existing state-of-the-art methods on multiple datasets. The authors also introduce a new dataset for Manipuri characters called “Mayek 27” and compare the performance of different optimizers and batch sizes on the proposed model. The model is trained using Adam optimizer with a low step size of 0.0001 and achieves a classification accuracy of 99.27% on the test set. The paper also includes a detailed analysis of the computational cost and convergence rate of the proposed model. Overall, the results demonstrate the effectiveness of the proposed CNN architecture for character recognition in Indic scripts.
4 Methodology

To build a model that can recognize handwritten characters, we decided to use a deep learning approach using Convolutional Neural Networks (CNN). CNN has proven to be strong in automatic feature extraction and has reached state-of-the-art in many image classification problems.

In this section, we present a CNN architecture and two training approaches, with the goal of further enriching the results reported in the literature. In the first approach, a single model trained on multiple datasets was used. The second approach was based on the number of strokes in each character, where we divided the characters into four groups, as shown in Table 2. Each group was used to train a different model, and the number of strokes was used as a filtration step before selecting the model that would make the prediction. Therefore, we refer to them as single-model and multi-model approaches, respectively.

Table 2 shows what characters belong to each group, it’s also seen that some characters belong to 2 groups, columns represent groups, rows represent the handwriting style which could cause the number of strokes to be different. A stroke in our work starts from the writer’s stylus, pen, or whatever method is used for typing, touching the surface to form one part of the character, that part could be the main body of the character, or a dot, until the writer leaves their hand off the surface.

4.1 Datasets

There are a couple of datasets for Arabic handwritten characters. As discussed in previous sections, we used Hijja, AHCD, and a third dataset built by merging Hijja with AHCD into a new dataset, referred to as Hijja-AHCD.

The Hijja dataset [7] consists of 28 letters in the Arabic alphabet in addition to the isolated Hamza (‘), which makes it a total of 29 classes. Data were collected from 591 children in Arabic-speaking schools in Riyadh, Saudi Arabia. It contained 47,434 characters. This dataset includes both the isolated and connected forms of Arabic characters, as in Arabic script, characters could have up to three different forms based on their location in the word. This variation made it perfect for this experiment, as it allowed the network to recognize Arabic handwritten characters in all their forms.

The Arabic Handwritten Characters Dataset (AHCD) [20] is a publicly available dataset that contains 16,800 characters written by 60 participants aged ranging between 19 and 40 years. Unlike Hijja, the age range makes the style of the handwritten characters different, it’s more clear and easier to recognize by the human eye. The total number of classes was 28, matching the number of characters in the Arabic alphabet. The AHCD contains only the isolated forms of each character.

The third dataset was constructed by augmenting Hijja with AHCD; both had gray images with 32 × 32 resolution, which made the merging easy and doable. We did this by using Pandas, as both datasets were available in CSV format, we merged them into a new dataset that also has the CSV format and called it Hijja-AHCD. Concatenation was not performed programmatically before each experiment; instead, we chose to form a new dataset so that carrying out different experiments was easier.

4.1.1 Data Preprocessing

When we initially loaded AHCD and visualized a subset of it, the images were incorrect as it was rotated 90 degrees to the left and flipped vertically, as shown in Fig. 2a Sheen belongs to Groups 4 and 2, and the dots can be written separately which makes it three strokes, or merged into a curve that makes it one stroke. Figure 2b Yaa belongs to Groups 3 and 2, the same case caused by dots either separated or merged into a dash. Figure 3a shows a subset from...
the AHCD without any modification, and Fig. 3b shows the same image after transposition. We used NumPy to transpose each image, and then saved the data back to CSV, so that we did not bother later in repeating this step for each experiment. Note that the step of constructing Hijja-AHCD was done with the transposed version of the AHCD.

The last step before training was to invert the colors for all images such that it had a black background and a white foreground. This step was inspired by [20], where they achieved better results in Arabic digit recognition after applying this technique.

4.2 Model Architecture

In this section, we propose a CNN architecture that uses many convolution layers and ends with a classification feedforward network, as shown in Fig. 4.

We start with an input image of 32 × 32 size and one gray channel. The model consists of four convolutional blocks, the first two blocks have two convolution layers with 64 filters and 128, followed by activation, max pooling with a size of two, and finally a batch normalization step. The last two blocks have three convolution layers with 256 and 384 filters, respectively, followed by activation, max pooling, and batch normalization. A flattening step is then applied to prepare for feeding into the classifier. Instead of using one fully connected layer for classification, we used a feedforward network consisting of three layers, with regularization using a dropout with a probability of 0.3, to avoid overfitting. HeUniform weight initializer was used instead of the default random initialization to initialize all the weights of the convolution and fully connected (FC) layers.

The feedforward network starts with a flattening step to turn the output from the feature extractor part into a vector with one dimension, that is, $4 \times 4 \times 384$ long, and then feeds it into the first FC layer, which outputs 256 neurons, then 128, and finally 64 neurons. The very last FC layer is a classifier with a softmax function that produces the class rates for the given labels, 28 in case of AHCD, and 29 in case of Hijja and Hijja-AHCD.

The activation function used in all layers was LeakyReLU, with a slope of 0.3. LeakyReLU is a variation of ReLU that was proposed to overcome the problem of the vanishing gradient that arises in very deep networks when ReLU is used. LeakyReLU keeps some of the negative data instead of assigning Zero to them; hence it is called leaky. Our experiments, as will be seen later, showed that LeakyReLU slightly improved the results compared with the baseline.

To optimize the network, we used Categorical Cross Entropy for the loss function; because we have multiple class rates, and the labels were provided as one-hot encoded. Finally, the Adam optimizer was used to update the weights with an initial learning rate of 0.001.

4.3 Experimental Setup

This section explains the experiments conducted to recognize handwritten Arabic characters and goes over the training setup. We used the Python language and Keras framework with multiple programming libraries, such as pandas, NumPy and scikit. All work was performed on Google Colab.
with an enabled GPU. First, we followed several preprocessing approaches. Then, we built the base CNN classification model as explained in Sect. 3.2, which was used in all experiments, with fixed hyperparameters that we found to be the best after many experiments. Therefore, the evaluation results were based solely on the difference between each preprocessing and training approach.

4.3.1 Training Setup

To start the training, we need to have a validation set to help detect when the model starts overfitting on the training set. Cross validation was used as a technique to obtain the best validation set instead of splitting it manually at a fixed rate, as in [4]. We used the StratifiedKFold function from the scikit-learn library to split the training set into training and validation sets. We set the number of splits to five and shuffled to true, meaning that the training loop will run five times each time with different training and validation sets, so that we can pick the model with the highest validation accuracy. The batch size was set to 128, and we trained for 30 epochs in all five experiments, none of the models showed any improvement after 30 epoch.

During training, we used a learning rate scheduler, that is, Adam starts with an initial learning rate of 0.001, then during training, it optimizes the learning rate by itself, but we found, by experiment, that when adding a callback scheduler that is applied to the learning rate after Adam optimization on each epoch, the overall model accuracy improved. We set up this callback as a function that takes the learning rate as an input and then multiplies it by an exponential of −0.01.

4.3.2 Experiments

Figure 5 shows all experiments conducted in this study. In (A), we conducted three experiments that fall under the single model approach, starting with the model proposed by Hijja dataset [7], then the model proposed in the most recent study on Arabic handwritten recognition by Ahmed et al. [24], and finally our proposed model.

In (B), the multi-model approach, we used our proposed model to train and test the Hijja-AHCD dataset. We started by filtering the characters based on the number of strokes and splitting the data correspondingly. After grouping the characters into four different groups based on their strokes, each group was used as a standalone dataset for training the model; therefore, we say that this model is for group X. We refer to this approach as multi-model.

Finally, we experimented with transfer learning using the EfficientNetV0 pretrained model on Hijja, AHCD and Hijja-AHCD. Transfer learning helps in many problems by transferring the knowledge learned from one related or unrelated domain to another, which in many cases provides a higher convergence rate and results. Many models have been trained on the ImageNet dataset, which is a large dataset with 1000 classes of natural images, such as MobileNet, VGG19, EfficientNet, and its variants. We selected EfficientNetV0, as it has recently become a popular choice for vision problems because of its efficiency and relatively small size compared to others. The experiment started by substituting the top layer with our input size (32, 32, 3), but noticed here that EfficientNetV0 expects a three channels image so we needed to apply a preprocessing step to convert grayscale images to RGB. We also added a final classification layer to output 29 classes that matched our data. The training started with the ImageNet weights and trained all EfficientNet layers. The rest of the setup was the same as that of our CNN model.

4.4 Baseline and Evaluation Measures

To evaluate the performance of our model performed in each experiment, we calculated the prediction accuracy, recall, precision, and F1 measures of the test set for each corresponding dataset. To evaluate (C), we calculated the average prediction accuracy over the four models for each stroke-based character group. The same is true for precision, recall, and F1 score measures. We refer to Altwaijry et al. as the baseline because their model was the first to be trained and tested on the Hijja dataset.

4.4.1 Precision, Recall and F1 Score

Precision is the proportion of correctly classified characters from all characters in class X. While recall is the proportion of correctly classified characters from all characters in class X. F1 score is a function of precision and recall.

\[
\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}
\]

\[
\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}
\]
5 Results and Discussions

In this section, we present and discuss the results of our experiments. Table 3 shows the prediction accuracy for all of them where the bold values indicate the highest prediction accuracy achieved among the various models being compared, thus emphasizing the best-performing approaches in the given analysis. Refer to Appendix A for the full classification reports.

Starting from first row, where the results for carrying out full experiments on Hijja dataset can be seen. Our model outperformed Altwaijry, et al. [7], Ahmed, et al. [24], and transfer learning with EfficientNetV0. In the second row, we used AHCD dataset. Our model obtained better results than both [7] Altwaijry, [24] Ahmed, but has a similar results with EfficientNetV0. Something that worth noting here is that there’s a big difference in the accuracy of all single model experiments on Hijja versus AHCD datasets. We assume that this large difference in prediction accuracies come from the challenges Hijja dataset introduces. First, it contains different forms of each character, both isolated and connected, introducing more features and higher level of similarities between characters. Second, the characters were written by children. On the other hand, AHCD is written by adults and the characters were only in isolated forms.

The third row represents carried out experiments on the merged dataset Hijja-AHCD. Compared with with the results on Hijja dataset alone in the first row, it can be seen that merging Hijja with AHCD improved the prediction accuracy in our model, Ahmed et al. [24], and EfficientNetV0.

These results show the challenges evolving from the age difference between participants in each dataset. In Hijja, they were all children, and children’s handwritings could be very challenging to understand even for the human eye, so to capture more features and help the model learn more possible ways of writing each character, merging Hijja with AHCD experimentally helped the model learn better. We assumed that showing the model more variations of handwritings is more effective and realistic for our problem than enlarging the size of Hijja dataset with usual augmentation techniques like zooming or rotating existing images, hence we decided to merge with another similar dataset with a different age segment instead of generating new images from the same dataset.

Lastly, we introduced a new approach which is based on a prior filtration step using character strokes. We calculated the averaged prediction accuracy on all 4 character groups as shown previously in Table 2, and is reported in the last column in Table 3 as multi-model. The averaged prediction accuracy outperformed all the single model experiments and transfer learning on Hijja-AHCD dataset. The classification reports for each of the models in each character group are shown in Tables A 4, A 5, A 6, A 7 in the appendix. In order to make use of this method in a real-life application, there should be a layer to filter the number of strokes in an image prior to prediction, and based on that number, which model is used for prediction will be decided. Such thing is doable in online writing applications where the strokes can be counted as the number of times the user interacts with the screen using their fingers or a stylus. Experimentally, we split the test set as we did with the training set for evaluation.

During the evaluation, we noticed some factors of which can be said as challenges facing the CNN models with handwritten Arabic characters. First, as can be seen in Fig. 6a, the model misclassified the character Qaaf/ق as Faa’/ف, we provide the image of both characters clearly shown in Fig. 6b for comparison. We can see that Faa’, has 1 dot, and Qaaf has 2 dots.

\[ F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \]

Table 3 Prediction accuracy for all experiments

|                  | (A) Single CNN model | Transfer learning with EfficientNetV0 | Multi-model |
|------------------|----------------------|--------------------------------------|-------------|
|                  | Altwaijry et al.     | Ahmed et al. [24]                    | Our model   |                          |
| Hijja Dataset    | 87%                  | 90%                                  | 91%         | 86%                      | –                        |
| AHCD Dataset     | 96%                  | 94%                                  | 97%         | 97%                      | –                        |
| Hijja-AHCD       | –                    | 92%                                  | 93%         | 90%                      | 96%                      |

The use of [bold] formatting denotes the best accuracy prediction results among all experiments.

![Fig. 6](image-url)
has 2 dots, while the shape of the character itself is more curved than Faa‘, but many handwriting styles could make the main part of both look very similar, leaving the only difference in the dots above the curve. The image we showed the model has the 2 dots very unclear due to scanning and data preprocessing, this can be seen as a problem from two sides. Firstly, with the traditional data collecting method by scanning written characters from paper, and due to cleaning and preprocessing, it loses some quality. Secondly, the Arabic characters are a challenging problem since some characters look very similar, and even more challenging with the diverse handwriting styles.

Furthermore, the model made similar misclassifications on characters shown in Fig. 7. All of them were part of the Hijja dataset, and they are written in the medial shape, that is, they are written as if they were in the middle of a word. The model misclassified them with similar characters written in the exact same way, also in the middle position. It’s challenging even for us to classify these characters, and with the model reaching a considerable good prediction accuracy over most cases with isolated characters, it’s still unable to overcome some challenges with other positions.

Our proposal of the multi-model stroke-based solution opens the door for a new way of solving classification problems in the Arabic handwriting with the idea of a prior filtration process to overcome the shortages mentioned earlier.

6 Conclusion and Future Work

A preprint has previously been published [28]. The Arabic language introduces more challenges in the fields of Deep Learning, more studies are being carried out yet not enough to reach the advanced results in Latin languages. In this research, we discuss some approaches to get higher accuracy in Arabic handwritten character recognition using Deep Learning over two recent datasets, one of which was written by children and introduces even higher complex patterns, the Hijja dataset. We used augmentation with another dataset, AHCD, and strokes approach on the same model. Our model reached higher accuracy on both Hijja and AHCD than the baseline, moreover, augmenting Hijja with AHCD helped raise the accuracy further than on Hijja alone, which tells that the dataset made learning the handwritten characters patterns harder for the model.

We compared our model with transfer learning using EfficientNetV0 and one of the recent proposed models that gained high results on Arabic handwritten characters in the literature, nevertheless, our model outperform both their results and transfer learning. Furthermore, we also compared the multi-model approach with both the results from single models in (A) and transfer learning (C) on Hijja-AHCD and it outperformed both.

As future work, we plan to test our model in a real application as a proof of concept which could further reveal if these experiments are applicable under real conditions outside the testing environment. Particularly, using our model in an online writing application, where we present the model with a different type of input, and we anticipate it to perform well since images extracted from online handwriting are more clear than offline images, which could help the model recognize the features easily. In addition, we see that creating a new dataset based on online writing could further help improve the accuracy of our model in such applications.

Finally, our results have potential applications across various engineering sciences. These include enhancing human–computer interaction systems tailored to children, developing robots capable of interpreting children’s handwriting, integrating our model into assistive technologies for children with learning disabilities, and digitizing children’s sketches and annotations in computer-aided design. Yet, as we focused more on the complex patterns of children’s handwriting, the applications of this model in educational apps to teach dictation for children are a good use case for this research. Moreover, our work contributes to the broader field of artificial intelligence and machine learning, inspiring researchers to tackle other complex problems, including character recognition in different scripts or languages. This demonstrates the wide-ranging impact and utility of our research across multiple engineering domains.
Appendix

See tables A 1, A 2, A 3, A 4, A 5, A 6, A 7

| Table A 1 | Full classification report of our model on Hijja dataset |
|------------|----------------------------------------------------------|
| Label      | Precision | Recall | f1-score |
| Alif ﺍ     | 0.99      | 0.99   | 0.99     |
| Baa’ ﺗ     | 0.94      | 0.96   | 0.95     |
| Taa’ ﺘ     | 0.91      | 0.93   | 0.91     |
| Thaa’ ﷺ    | 0.95      | 0.92   | 0.93     |
| Jiim ﺝ     | 0.94      | 0.96   | 0.95     |
| Haa’ ﺝ     | 0.9       | 0.81   | 0.85     |
| Khaa’ خ     | 0.9      | 0.88   | 0.89     |
| Daal ﺪ     | 0.76      | 0.83   | 0.79     |
| Dhaal ﻋ     | 0.77     | 0.71   | 0.74     |
| Raa’ ﺞ     | 0.93      | 0.91   | 0.92     |
| Zaay ز     | 0.92      | 0.88   | 0.9      |
| Seen س     | 0.95      | 0.96   | 0.96     |
| Sheen ش     | 0.98      | 0.95   | 0.97     |
| Saad ص     | 0.86      | 0.94   | 0.9      |
| Daad ض     | 0.92      | 0.88   | 0.9      |
| Taa’ ﺕ     | 0.95      | 0.91   | 0.93     |
| Zaay ﺖ     | 0.93      | 0.96   | 0.94     |
| Aayn ﻤ     | 0.82      | 0.87   | 0.84     |
| Ghayn غ     | 0.92      | 0.84   | 0.88     |
| Faa’ ﻂ     | 0.82      | 0.87   | 0.84     |
| Qaa’ ق     | 0.92      | 0.93   | 0.93     |
| Kaff ك     | 0.91      | 0.89   | 0.9      |
| Lamm ﻝ     | 0.92      | 0.91   | 0.92     |
| Miim ﻎ     | 0.91      | 0.93   | 0.92     |
| Nuun ن     | 0.79      | 0.87   | 0.83     |
| Haa’ ﻆ     | 0.92      | 0.88   | 0.9      |
| Waaw و     | 0.92      | 0.94   | 0.93     |
| Yaa’ ي     | 0.97      | 0.93   | 0.95     |
| Hamza ه     | 0.84      | 0.87   | 0.86     |
| Accuracy   |           |        |          |
| Macro avg  | 0.90      | 0.90   | 0.90     |
| Weighted avg | 0.91   | 0.91   | 0.91     |

| Table A 2 | Full classification report of our model on AHCD dataset |
|------------|----------------------------------------------------------|
| Label      | Precision | Recall | f1-score |
| Alif ﺍ     | 0.99      | 1      | 1       |
| Baa’ ﺗ     | 0.99      | 0.99   | 0.99    |
| Taa’ ﺘ     | 0.93      | 0.96   | 0.94    |
| Thaa’ ﷺ    | 0.96      | 0.95   | 0.95    |
| Jiim ﺝ     | 0.99      | 0.97   | 0.98    |
| Haa’ ﺝ     | 0.97      | 0.97   | 0.97    |
| Khaa’ خ     | 1         | 0.98   | 0.99    |
| Daal ﺪ     | 0.93      | 0.98   | 0.96    |
| Dhaal ﻋ     | 0.94     | 0.95   | 0.95    |
| Raa’ ﺞ     | 0.96      | 0.97   | 0.96    |
| Zaay ز     | 0.99      | 0.92   | 0.95    |
| Seen س     | 1         | 0.99   | 1       |
| Sheen ش     | 0.98      | 1      | 0.99    |
| Saad ص     | 0.96      | 1      | 0.98    |
| Daad ض     | 0.99      | 0.96   | 0.97    |
| Taa’ ﺕ     | 0.96      | 0.98   | 0.97    |
| Zaay ﺖ     | 0.99      | 0.96   | 0.97    |
| Aayn ﻤ     | 0.98      | 0.99   | 0.98    |
| Ghayn غ     | 1         | 0.98   | 0.99    |
| Faa’ ﻂ     | 0.97      | 0.96   | 0.96    |
| Qaa’ ق     | 0.95      | 0.95   | 0.95    |
| Kaff ك     | 0.97      | 0.97   | 0.97    |
| Lamm ﻝ     | 0.98      | 1      | 0.99    |
| Miim ﻎ     | 0.99      | 0.98   | 0.99    |
| Nuun ن     | 0.97      | 0.94   | 0.96    |
| Haa’ ﻆ     | 0.98      | 0.98   | 0.98    |
| Waaw و     | 0.94      | 0.98   | 0.96    |
| Yaa’ ي     | 1         | 0.97   | 0.99    |
| Hamza ه     |           |        | 0.97    |
| Accuracy   | 0.97      | 0.97   | 0.97    |
| Macro avg  | 0.97      | 0.97   | 0.97    |
Table A 3  Full classification report of our model on Hijja-AHCD dataset

| Label | Precision | Recall | f1-score |
|-------|-----------|--------|----------|
| Alif  | 0.99      | 0.99   | 0.99     |
| Baa' | 0.97      | 0.97   | 0.97     |
| Taat | 0.93      | 0.93   | 0.93     |
| Thaa' | 0.95     | 0.93   | 0.94     |
| Jiim | 0.96      | 0.95   | 0.96     |
| Haa' | 0.89      | 0.93   | 0.91     |
| Khaa' | 0.95     | 0.93   | 0.94     |
| Daal | 0.84      | 0.87   | 0.86     |
| Dhaal | 0.81    | 0.85   | 0.83     |
| Raa' | 0.94      | 0.92   | 0.93     |
| Zaay | 0.93      | 0.89   | 0.91     |
| Seen | 0.97      | 0.97   | 0.97     |
| Sheen | 0.98    | 0.96   | 0.97     |
| Saad | 0.9       | 0.94   | 0.92     |
| Daad | 0.92      | 0.92   | 0.92     |
| Taa' | 0.93      | 0.94   | 0.94     |
| Zaa' | 0.97      | 0.92   | 0.94     |
| Aayn | 0.89      | 0.88   | 0.88     |
| Ghayn | 0.94    | 0.89   | 0.91     |
| Faa' | 0.82      | 0.92   | 0.87     |
| Qaa' | 0.94      | 0.9    | 0.92     |
| Kaff | 0.91      | 0.93   | 0.92     |
| Lamm | 0.95      | 0.92   | 0.93     |
| Miim | 0.93      | 0.95   | 0.94     |
| Nuun | 0.86      | 0.87   | 0.86     |
| Haa' | 0.93      | 0.9    | 0.92     |
| Waaw | 0.94      | 0.97   | 0.96     |
| Yaa' | 0.97      | 0.96   | 0.97     |
| Hamza | 0.87     | 0.86   | 0.86     |
| Accuracy |        |         |          |
| Macro avg | 0.92  | 0.92   | 0.92     |
| Weighted avg | 0.93 | 0.93   | 0.93     |

Table A 4  Full classification report of multi model experiment, group 1

| Label | Precision | Recall | f1-score |
|-------|-----------|--------|----------|
| Alif  | 0.99      | 0.99   | 0.99     |
| Haa' | 0.89      | 0.93   | 0.91     |
| Daal | 0.89      | 0.82   | 0.85     |
| Raa' | 0.94      | 0.93   | 0.94     |
| Seen | 0.97      | 0.95   | 0.96     |
| Saad | 0.93      | 0.93   | 0.93     |
| Taa' | 0.98      | 0.94   | 0.96     |
| Aayn | 0.89      | 0.91   | 0.9     |
| Lamm | 0.95      | 0.94   | 0.94     |
| Miim | 0.92      | 0.94   | 0.93     |
| Haa' | 0.92      | 0.93   | 0.92     |
| Waaw | 0.93      | 0.97   | 0.95     |
| Hamza | 0.92     | 0.94   | 0.93     |
| Accuracy |        |         |          |
| Macro avg | 0.93  | 0.93   | 0.93     |
| Weighted avg | 0.94 | 0.94   | 0.94     |

Table A 5  Full classification report of multi model experiment, group 2

| Label | Precision | Recall | f1-score |
|-------|-----------|--------|----------|
| Baa' | 0.97      | 0.97   | 0.97     |
| Taat | 0.92      | 0.93   | 0.92     |
| Thaa' | 0.93     | 0.95   | 0.94     |
| Jiim | 0.98      | 0.98   | 0.98     |
| Khaa' | 0.93     | 0.94   | 0.93     |
| Dhaal | 0.83    | 0.84   | 0.84     |
| Zaa' | 0.95      | 0.94   | 0.95     |
| Sheen | 0.98      | 0.96   | 0.97     |
| Daad | 0.95      | 0.94   | 0.94     |
| Zaay | 0.95      | 0.94   | 0.95     |
| Ghayn | 0.92     | 0.89   | 0.91     |
| Faa' | 0.86      | 0.88   | 0.87     |
| Qaa' | 0.93      | 0.92   | 0.93     |
| Kaff | 0.95      | 0.95   | 0.95     |
| Nuun | 0.87      | 0.88   | 0.88     |
| Yaa' | 0.97      | 0.97   | 0.97     |
| Accuracy |        |         |          |
| Macro avg | 0.93  | 0.93   | 0.93     |
| Weighted avg | 0.93 | 0.93   | 0.93     |
Appendix A

This appendix contains the full classification reports of all experiment in Table 3.

Table A 6  Full classification report of multi model experiment, group 3

| Label | Precision | Recall | f1-score |
|-------|-----------|--------|----------|
| Taa’  | 0.99      | 0.98   | 0.98     |
| Zaa’  | 0.99      | 0.98   | 0.99     |
| Qaaf  | 0.98      | 0.98   | 0.98     |
| Yaa’  | 0.99      | 1.00   | 0.99     |

Accuracy
Macro avg 0.99 0.99 0.99
Weighted avg 0.99 0.99 0.99

Table A 7  Full classification report of multi model experiment, group 4

| Label | Precision | Recall | f1-score |
|-------|-----------|--------|----------|
| Thaa’ | 0.98      | 0.99   | 0.99     |
| Sheen | 0.99      | 0.98   | 0.99     |

Accuracy
Macro avg 0.99 0.99 0.99
Weighted avg 0.99 0.99 0.99

References

1. Shekhar B. An online and offline character recognition using image processing methods-a survey. Int J Commun Comput Technol. 2016;4:102–7.
2. Habash NY. Introduction to Arabic Natural Language Processing. Synthesis lectures on human language technologies, vol. 3. Cham: Springer International Publishing; 2010. p. 1–187.
3. Baldominos A, Saez Y, Isasi P. A survey of handwritten character recognition with MNIST and EMNIST. Appl Sci. 2019;9:3169. https://doi.org/10.3390/app9153169.
4. Ma KO, Poruran S. OCR-nets: variants of pre-trained CNN for urdu handwritten character recognition via transfer learning. Procedia Computer Sci. 2020;171:2294–301.
5. Soumia F, Djamel G, Haddad M. Handwritten Arabic character recognition: comparison of conventional machine learning and deep learning approaches. In: Proceedings of the international conference of reliable information and communication technology. Cham: Springer; 2020. p. 1127–38.
6. Lamghari, N.; Charaf, M.E.H.; Raghay, S. Comparison of Neural Network Parameters for Classification of Arabic Handwritten Isolated Characters. International Journal of Computer Applications 975, 8887.
7. Altawaijry N, Al-Turaiki I. Arabic handwriting recognition system using convolutional neural network. Neural Comput & Appl. 2021;33:2249–61. https://doi.org/10.1007/s00521-020-05070-8.
8. Younis KS. Arabic handwritten character recognition based on deep convolutional neural networks. Jordan J Comput Inf Technol. 2017;3:186–200.
9. El-Sawy A, Hazem E-B, Loey M. CNN for handwritten Arabic digits recognition based on LeNet-5. In: Proceedings of the international conference on advanced intelligent systems and informatics. Cham: Springer; 2016. p. 566–75.
10. Byerly A, Kalganova T, Dear I. No routing needed between capsules. Neurocomputing. 2021;463:545–53. https://doi.org/10.1016/j.neucom.2021.08.064.
11. Jayasundara V, Jayasekara S, Jayasekara H, Rajasegaran J, Seneviratne S, Rodrigo R. TextCaps : handwritten character recognition with very small datasets. 2019 IEEE Winter Conference on Applications of Computer Vision (WACV) 2019;254–62. https://doi.org/10.1109/WACV.2019.00033
12. Saiegh-Haddad E, Joshi RM. Handbook of Arabic literacy: insights and perspectives. Dordrecht: Springer Science & Business Media; 2014.
13. Mustapha IB, Hasan S, Nabus H, Shamsuddin SM. Conditional deep convolutional generative adversarial networks for isolated handwritten Arabic character generation. Arab J Sci Eng. 2021. https://doi.org/10.1007/s13369-021-05796-0.
14. Altawaijry N, Al-Turaiki I. Arabic handwriting recognition system using convolutional neural network. Neural Comput Appl. 2021;33:2249–61.
15. Ashiquzzaman A, Tushar AK, Rahman A, Mohsin F. An Efficient Recognition Method for Handwritten Arabic Numerals Using CNN with Data Augmentation and Dropout. In: Balas VE, Sharma N, Chakrabarti A, editors. Procedings of the Data Management, Analytics and Innovation. Singapore: Springer; 2019. p. 299–309.
16. Handwritten optical character recognition (OCR): a comprehensive systematic literature review (SLR) | IEEE Journals & Magazine | IEEE Xplore Available online: https://ieeexplore.ieee.org/document/9151144. Accessed on 31 Jan 2022.

17. AlKhateeb JH. A database for Arabic handwritten character recognition. Procedia Comput Sci. 2015;65:556–61.

18. Lamghari N, Raghay S. DBAHCL: database for Arabic handwritten characters and ligatures. Int J Multimed Info Retr. 2017;6:263–9.

19. Lamghari N, Raghay S. Recognition of Arabic handwritten diacritics using the new database DBAHD. J Phys Conf Ser. 2021;1743:012023.

20. El-Sawy A, Loey M, El-Bakry H. Arabic handwritten characters recognition using convolutional neural network. WSEAS Trans Comput Res. 2017;5:11–9.

21. Younis K. Arabic handwritten characters recognition based on deep convolutional neural networks. Jordan J Comput Inform Technol. 2018.

22. Elleuch M, Tagougui N, Kherallah M. Towards Unsupervised Learning for Arabic Handwritten Recognition Using Deep Architectures. In: Dee M, editor. Proceedings of the International Conference on Neural Information Processing. Cham: Springer; 2015. p. 363–72.

23. Amrouch M, Rabi M, Es-Saady Y. Convolutional Feature Learning and CNN Based HMM for Arabic Handwriting Recognition. In: Mansouri A, El Moataz A, Nouboud F, Mammass D, editors. Proceedings of the Image and Signal Processing. Cham: Springer International Publishing; 2018. p. 265–74.

24. Ahmed R, Gogate M, Tahir A, Dashtipour K, Al-tamimi B, Hawalah A, El-Affendi MA, Hussain A. Novel deep convolutional neural network-based contextual recognition of arabic handwritten scripts. Entropy. 2021;23:340. https://doi.org/10.3390/e23030340.

25. Balaha HM, Ali HA, Youssef EK, Elsayed AE, Samak RA, Abdelhaleem MS, Tolba MM, Shehata MR, Mahmoud MR, Abdelhameed MM, et al. Recognizing Arabic handwritten characters using deep learning and genetic algorithms. Multimed Tools Appl. 2021;80:32473–509. https://doi.org/10.1007/s11042-021-11185-4.

26. Jiang W, Zhang L. Edge-SiamNet and Edge-TripleNet: new deep learning models for handwritten numeral recognition. IEICE Trans Inf Syst. 2020;103-D(3):720–3.

27. Hazra A, Choudhary P, Inunganbi SC, Adhikari M. Bangla-Meitei Mayek scripts handwritten character recognition using Convolutional Neural Network. Appl Intell. 2020;51(8):2291–311.

28. Mais Alheraki, Rawan Al-Matham and Hend Al-KhalifaHend Al-Khalifa; Handwritten Arabic character recognition for children writing using convolutional neural network and stroke identification. 2022, https://doi.org/10.48550/arXiv.2211.02119.

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.