applied sciences

MDPI

Article

Towards Accurate Children's Arabic Handwriting Recognition via Deep Learning

Anfal Bin Durayhim [1,*], Amani Al-Ajlan [1,†], Isra Al-Turaiki [2] and Najwa Altwaijry [2]

[1] Department of Information Technology, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

[2] Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

* Correspondence: anfalmohammedaldr@gmail.com

† First joint author.

Abstract: Automatic handwriting recognition has received considerable attention over the past three decades. Handwriting recognition systems are useful for a wide range of applications. Much research has been conducted to address the problem in Latin languages. However, less research has focused on the Arabic language, especially concerning recognizing children's Arabic handwriting. This task is essential as the demand for educational applications to practice writing and spelling Arabic letters is increasing. Thus, the development of Arabic handwriting recognition systems and applications for children is important. In this paper, we propose two deep learning-based models for the recognition of children's Arabic handwriting. The proposed models, a convolutional neural network (CNN) and a pre-trained CNN (VGG-16) were trained using Hijja, a recent dataset of Arabic children's handwriting collected in Saudi Arabia. We also train and test our proposed models using the Arabic Handwritten Character Dataset (AHCD). We compare the performance of the proposed models with similar models from the literature. The results indicate that our proposed CNN outperforms the pre-trained CNN (VGG-16) and the other compared models from the literature. Moreover, we developed Mutqin, a prototype to help children practice Arabic handwriting. The prototype was evaluated by target users, and the results are reported.

Keywords: automatic handwriting recognition; Arabic handwritten character recognition; machine learning; convolutional neural network; deep learning

check for updates

Citation: Bin Durayhim, A.; Al-Ajlan, A.; Al-Turaiki, I.; Altwaijry, N. Towards Accurate Children's Arabic Handwriting Recognition via Deep Learning. *Appl. Sci.* **2023**, *13*, 1692. https://doi.org/10.3390/app13031692

Academic Editor: Alessandro L. Koerich

Received: 20 November 2022
Revised: 17 January 2023
Accepted: 25 January 2023
Published: 29 January 2023

1. Introduction

Automatic handwriting recognition is important for a wide range of applications. Automatic handwriting recognition refers to the capability of a computer to recognize handwritten input [1]. Character recognition technologies are used to convert the characters into their digital counterparts, which provide an automatic technique for recognizing text on images. Handwriting data that are input via scanning are considered *offline*, and handwriting data that are input via pen tip are considered *online*. The handwriting can be input from a variety of sources, including paper documents, images, touch screens, or other devices [1]. Automatic handwriting recognition is a challenging task since different individuals' handwriting differs in most cases. Furthermore, a single writer's handwriting may vary significantly over time [1]. Arabic handwritten character recognition presents additional challenges to researchers posed by the Arabic language including variations in the character shapes according to their placement in words.

Arabic is one of the top five spoken languages in the world, and there are 315 million Arabic speakers, which means that computers must now, more than ever, be able to understand what these speakers are writing [2]. Arabic handwritten character recognition (AHCR) technology has been significantly improved by using various machine learning (ML) algorithms, such as support vector machines (SVMs) and artificial neural networks

(ANN) [2]. Additionally, AHCR models that use convolutional neural networks (CNNs) have achieved outstanding results in a variety of handwritten datasets [2]. CNNs automatically detect and extract images' distinctive and representative features, outperforming traditional ML algorithms that require the features to be manually defined (e.g., SVMs) [2]. In the fields of computer vision, machine vision, speech recognition, natural language processing, and audio recognition, deep learning has achieved excellent success [3–5]. Recently, deep learning has also demonstrated remarkable results in handwriting recognition.

Handwriting-based systems are being employed to help children in their education. Researchers have spent an abundance of time and effort developing and improving handwriting learning systems for schoolchildren in different languages and on different platforms including web pages, computer-based applications and tablet applications [6]. This is encouraged by the fact that writing using digital notebooks allows children to obtain immediate feedback, which improves their learning experience. In addition, it facilitates the personalization of the learning process for better learning outcomes. Jolly et al. [7] presented a comparative study in French preschools that focused on handwriting acquisition using digital devices versus paper. The results showed that children trained on digital devices performed significantly better than children trained on paper, particularly in terms of fluency (decreased 'in-air' time and stopping time). Despite the importance and demand for Arabic handwriting recognition systems for children's handwriting, little published research is available [1,8,9].

In this paper, we detail the development of deep learning-based models to recognize children's handwriting using the Hijja dataset [1]. This research presents the following contributions to the literature: (1) improved Arabic handwriting recognition models based on CNN. The gap that we address is the need to investigate the recognition of children's handwriting. This research direction also serves handwriting recognition in other languages that use Arabic characters, such as Persian, Tajik, Dari, and Urdu; (2) a comparative study of the performances of the developed models as compared to state-of-art models and (3) the development of a prototype, named *Mutqin*, that integrates the best-obtained model into a useful tablet application for children to practice Arabic handwriting and spelling skills, with feedback based on their answers.

The paper is divided into six sections. Section 2 gives an overview of the related work in the field of deep learning for handwriting recognition. Section 3 presents the dataset and the proposed deep learning models. Section 4 presents the performance results of the proposed deep learning models and compares the proposed models with different models from the literature. In Section 5, we describe the development of our prototype application. Finally, Section 6 presents the conclusion and future work.

2. Related Work

This section presents various machine learning and deep learning approaches to solving the handwriting recognition problem for digits and characters.

2.1. Deep Learning for Handwriting Digit Recognition

For handwriting digit recognition, In 2010, Das et al. [10] represented handwritten Arabic numerals using an algorithm consisting of 88 features. They used a Multilayer Perceptron (MLP)-based classifier to recognize handwritten Arabic digits using the CMATERDB 3.3.1 dataset. An experiment involving 3000 samples showed that the technique yielded an average recognition rate of 94.93%. In 2017, Ashiquzzaman et al. [11] presented a novel algorithm for deep learning neural networks that used an activation function based on ReLU activation and a regularization process that used dropout. Moreover, the authors modified the method outlined in [10] by implementing dropout regularization in the MLP model to reduce the overfitting between the fully connected layers. Overall, there were 10 neurons in the output layer with softmax activation. The CMATERDB 3.3.1 handwritten Arabic digits dataset was used to test all the models. The proposed model in the study had 93.8% accuracy. In 2017, Alani et al. [12] proposed an algorithm combining RBMs and

CNNs to recognize handwritten Arabic digits. In particular, they presented a two-phase approach to Arabic digit recognition. First, features were extracted from the raw data using an RBM. Then, a deep supervised learning architecture was used to train and test the extracted features. Similar to the previous study, the CMATERDB 3.3.1 handwritten Arabic digit dataset was used to train and test the proposed algorithm, which achieved an accuracy of 98.59%. In 2017, Mudhsh et al. [13] proposed an alphanumeric VGG net design with 13 convolutional layers, two max-pooling layers, and three fully connected layers for alphanumeric character recognition. To avoid overfitting, regularization dataset augmentation and dropout were used. Their VGGnet was based on the VGGNet standard model. Furthermore, their methodology employed the ADBase for digits and the HACDB for characters. For the ADBase database, the model was 99.57% accurate, while for the HACDB database, it was 97.32% accurate. In 2018, Latif et al. [14] proposed a CNN model for recognizing handwritten digits in different languages: Eastern Arabic, Persian, Devanagari, Urdu, and Western Arabic. The proposed deep learning model was tested on various datasets. For the multilanguage database, the proposed algorithm achieved an accuracy of 99.26% and a precision of 99.29% on average, while an average accuracy of 99.32% was achieved for each language individually. In 2019, Ashiquzzaman et al. [15] modified the CNN model for recognizing handwritten Arabic numerals proposed in [11] by adding data augmentation to the CMATERDB 3.3.1 dataset as well as changing the activation function from a ReLU to an exponential linear unit (ELU). Compared to previous work on this dataset, the model achieved a high accuracy of 99.4%. Table 1 summarizes these deep learning-based handwritten digit recognition studies.

Table 1. A summary of deep learning-based handwritten digit recognition models.

References	Year	Model	Dataset	Type	Size	Accuracy
Das et al. [10]	2010	MLP	CMATERDB	Digits	3000	94.93%
Ashiquzzaman et al. [11]	2017	MLP				93.8%
Alani et al. [12]	2017	RBM-CNN				98.59 %
Mudhsh et al. [13]	2017	DNN	ADBase	Digits	70,000	99.57%
			HACDB	Chars	6600	97.32%
Latif et al. [14]	2018	CNN	MADBase	Digits	70,000	99.32%
			MNIST		70,000	99.32%
			HODA		80,000	99.32%
			Urdu		8500	99.32%
			DHCD		20,000	99.32%
			Combined		20,000	99.26%
Ashiquzzaman et al. [15]	2019	CNN	CMATERDB	Digits	3000	99.4%

2.2. Deep Learning for Handwritten Character Recognition

Recently, researchers have shown an increased interest in handwritten character recognition. In 2017, El-Sawy et al. [16] collected an extensive dataset of handwritten Arabic characters to train a deep learning model. Furthermore, they implemented optimization methods on a CNN. The proposed CNN model achieved 94.9% accuracy. In 2017, Younis et al. [17] proposed a deep neural network (DNN) for handwritten Arabic character recognition that used CNN models. In addition, they used dropout and batch normalization as regularization techniques. They tested the approach with two datasets: the AHCD and the AIA9K. The experimental results showed that the CNN model achieved 97.6% accuracy for the AHCD and 94.8% accuracy for the AIA9K. As recognizing handwritten characters has become more critical, recognizing children's handwriting has emerged as a specific area of research. In 2021, Altwaijry et al. [1] explored this area by conducting a study using the Arabic language in recognition systems and creating a dataset called Hijja, which contains Arabic letters produced by children aged 7–12 years. Five hundred and ninety-one participants contributed 47,434 characters to their dataset. They also proposed a model for automatic handwriting recognition based on a CNN. The model consisted of three convolutional layers, three pooling layers, and four fully connected layers. In addition, the ReLU activation function was used after each convolutional layer along with 80% dropout in the fully connected layers to avoid overfitting. The Hijja and AHCD datasets were used to train this model. The proposed CNN model outperformed the other model presented in [16] on the Hijja and AHCD datasets, achieving 88% and 97% accuracy, respectively. In 2020, Alyahya et al. [18] studied the effectiveness of the ResNet-18 architecture in recognizing handwritten Arabic letters. They employed a fully connected layer and a dropout layer in the original architecture. They proposed two ensemble models: the first one was based on the original ResNet-18 and modified ResNet-18 with one fully connected layer with/without a dropout layer, and the second one was based on the original ResNet-18 and modified ResNet-18 with two fully connected layers added to both the original and the modified ResNet-18 with/without a dropout layer. The AHCD dataset was used to train and evaluate the proposed algorithm, with the original ResNet-18 achieving the best result of 98.30%. The ensemble model with one fully connected layer and an ensemble model with two fully connected layers paired with a dropout layer achieved an accuracy of 98.00% and 98.03%, respectively. In 2020, Elkhayati et al. [19] described a new categorization strategy based on a CNN relative neighborhood graph (RNG) with Gabriel's graph (GG). The method was evaluated using recognition of isolated handwritten Arabic characters. The method involved directing the CNN through a filtering layer, thereby narrowing the set of probable classes for a query item. For this purpose, the rules of the RNG and GG were integrated. Experiments on the IFHCDB database showed that the proposed technique performed better than a standard CNN by 97.40%. In 2020, Shams et al. [20] proposed a hybrid model based on the development of deep convolution neural networks (DCNN) followed by an SVM to recognize and classify the missing features that the DCNN did not correctly classify. They employed the k-means clustering approach to solving the multi-stroke Arabic character recognition problem, which refers to the difficulty of recognizing letters that have the same stroke but different numbers or positions of the dots. To train and assess the model, they employed AHCD [16] and 840 additional images. In terms of accuracy, the model outperformed El-Sawy et al's. [16] model, reaching an accuracy of 95.07%. In 2021, AlJarrah et al. [5] proposed a CNN for the recognition of handwritten Arabic characters, training the model on 16,800 images of handwritten Arabic characters. The proposed model was composed of six convolution layers with different functions to predict and recognize images and three fully connected layers to perform character recognition. The CNN model was applied to two batches: 40 and 256, achieving a 97.2% accuracy. After applying data augmentation, the model achieved an accuracy of 97.7%. In 2021, Alrobah et al. [2] proposed a hybrid model combining an ML model and a deep learning model. They preprocessed the Hijja dataset introduced by Altwaijry et al. [1] and developed a CNN for feature extraction of the Arabic character images. They then passed it on to a hybrid model using an SVM and eXtreme Gradient Boosting classifiers. The CNN was trained with backpropagation. The CNNs, trained first with the softmax fully connected layer, were then employed as a feature extractor to train and evaluate the ML classifier. They conducted two experiments and attained a recognition rate of 96.3%. In 2021, Balaha et al. [21] investigated the effects of architectural complexity on the recognition of handwritten Arabic characters. The researchers introduced and used a dataset of handwritten Arabic characters (HMDB) and two CNN-based architectures: HMB1 and HMB2. The HMBD, CMATER, and AIA9k datasets were used to train and evaluate the two proposed architectures. The authors performed 16 experiments to investigate the effects of the weight initializers, optimizers, data augmentation, and regularization on the model accuracy. For the HMBD, CMATER, and AIA9k, the best accuracies obtained were 98.4%, 100%, and 99.0%, respectively. In 2021, Balaha et al. [22] addressed the recognition and text segmentation phases. They presented several solutions for text segmentation. A CNN was used in the recognition phase, and they proposed 14 different native CNN architectures. The model was trained and evaluated using the HMBD database, which includes 54,115 handwritten Arabic characters. Experiments were conducted using native CNN architectures and the highest accuracy was 91.96%. In the same study, a transfer learning (TL) and genetic algorithm (GA) approach called HMB-AHCR-DLGA was also developed to improve the recognition phase's training parameters and hyperparameters. For the latter technique, pre-trained CNN models (VGG16, VGG19, and MobileNetV2) were used. The best combinations were presented after five optimization experiments. The highest testing accuracy reported was 92.88%, achieved in the VGG16 experiment using the AdaMax weights optimizer. Table 2 summarizes these deep learning-based handwritten character recognition studies.

Table 2. A summary of deep learning-based handwritten character recognition models.

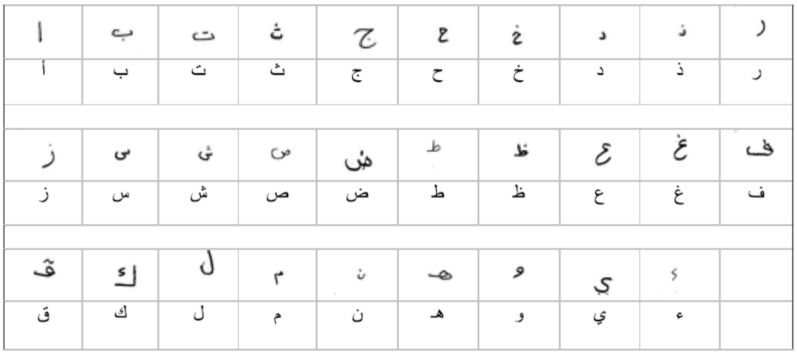
References	Year	Model	Dataset	Type	Size	Accuracy
El-Sawy et al. [16]	2017	CNN	AHCD	Chars	16,800	94.9%
Younis et al. [17]	2017	CNN	AHCD	Chars	16,800	97.6%
			AIA9K	Chars	8737	94.8%
Altwaijry et al. [1]	2021	CNN	Hijja	Chars	47,434	88%
			AHCD	Chars	16,800	97%
Alyahya et al. [18]	2020	CNN	AHCD	Chars	16,800	98.3%
Elkhayati et al. [19]	2020	CNN + CG	IFHCDB	Chars	52,380	97.4%
Shams et al. [20]	2020	DCNN + SVM	AHCD	Chars	16,800	95.07%
AlJarrah et al. [5]	2021	CNN	AHCD	Chars	16,800	97.2%
		CNN+Data augmentation	AHCD	Chars	16,800	97.7%
Alrobah et al. [2]	2021	CNN + SVM	Hijja	Chars	12,776	96.3%
Balaha et al. [21]	2021	CNN	HMBD	Chars	54,115	98.4%
			CMATER	Chars	30,000	100%
			AIA9k	Chars	89,740	99.0%
Balaha et al. [22]	2021	CNN	HMBD	Chars	54,115	91.96%
		HMB-AHCR-DLGA	HMBD	Chars	54,115	92.88%

Previous research has shown that CNNs have achieved high accuracy in recognizing numbers [14,15] and Arabic characters [1,5,18,21]. In addition, the focus has been exclusively on adult handwriting, which was used to train the Arabic handwriting recognition models. We found that the new Hijja dataset for children has not been comprehensively studied, with an accuracy of only 88% being achieved in [1]. On the other hand, in [2], the dataset was modified, and only the individual letters, i.e., 29 classes, were considered. In this case, an accuracy of 96.3% was achieved. In addition, the use of VGG16 as a pre-trained model for TL led to good accuracy for Arabic datasets, such as in [22], where an accuracy of 92.88% was achieved. The literature also shows that there are few handwriting recognition-based applications for children, and there are only two such applications for children that include Arabic. The first of these applicationss [8] depends on the stroke number of a single letter only, and the second [23] depends on tracking dots for words or letters. To the best of our knowledge, only Android-based software exists for this purpose.

3. Material and Methods

3.1. Dataset

We used the Hijja as our main dataset to train our proposed models. The Hijja is a recent and publicly available dataset of single Arabic letters introduced by Altwaijry et al. [1]. It was written by Arabic-speaking school children aged 7 to 12 years in Riyadh, Saudi Arabia. The dataset contains 108 classes that represent each Arabic letter using four different shapes for positioning at the start, center, and end of a word as well as when standing alone. The dataset contains a total of 47,434 images. The dataset is divided into 29 files, and each file contains images for one Arabic letter and one file for the Hamza. Figure 1 presents a sample of the Hijja dataset.



ا	ب	ت	ث	ج	ح	خ	د	ذ	ر
أ	ب	ت	ث	ج	ح	خ	د	ذ	ر
ز	س	ش	ص	ض	ط	ظ	ع	غ	ف
ز	س	ش	ص	ض	ط	ظ	ع	غ	ف
ق	ك	ل	م	ن	هـ	و	ي	ء	
ق	ك	ل	م	ن	هـ	و	ي	ء	

Figure 1. Sample of the Hijja dataset [2].

3.2. Proposed Deep Learning Models

The use of CNNs has been highly successful in recognizing numbers and characters, and TL models (such as the VGG16 model) have achieved good accuracy in recognizing Arabic letters; as such, we used these models to train our dataset.

3.2.1. Convolutional Neural Network Model

Our proposed CNN deep learning model was comprised of seven layers as presented in Figure 2. The first four layers were convolutional layers, while the last three were fully connected. Furthermore, there were pooling and activation layers between the convolutional and fully connected layers. The hidden layers used the nonlinearity function ReLU [24] to speed up the training phase, defined as shown in Equation (1).

$$ReLU(x) = max(0, x) \tag{1}$$

The Hijja dataset was divided into 70% for training the model, 10% for validation, and 10% for testing. Then, we conducted two experiments on handwriting recognition using the same model, the first for 29 classes and the second for 108 classes for the variations in the handwritten Arabic letters.

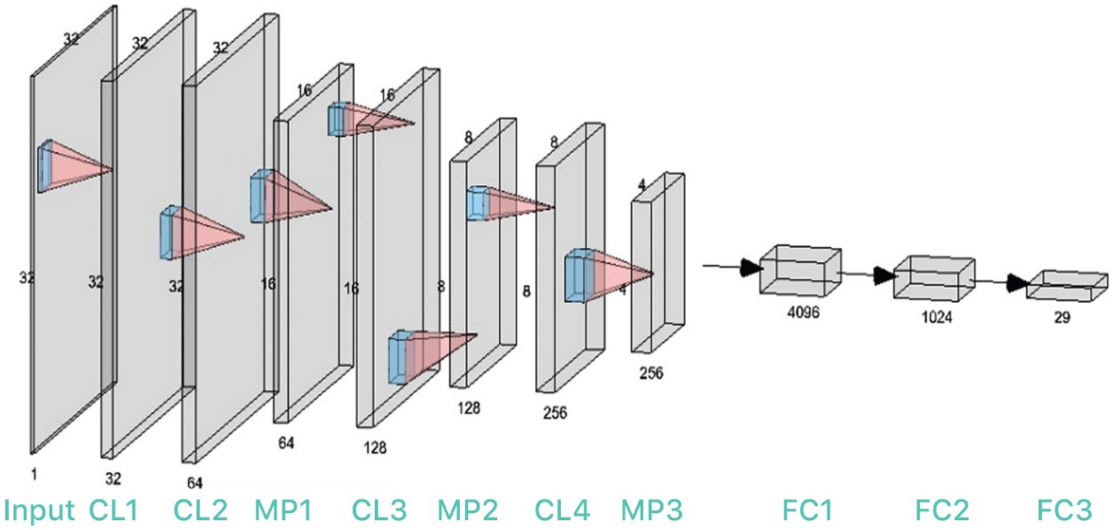


Figure 2. Our CNN architecture.

Convolutional layers 1 and 2: The input images were 32 × 32 × 1 grayscale images. At least one layer in a CNN is used for convolution. In a convolutional layer, there are several filters (kernels) to produce a feature map. Each filter functions as a feature extractor, identifying features such as vertices, edges, and endpoints. Therefore, the first convolutional layer was the first step of feature extraction. In this layer, the input image of size 32 × 32 was processed with 64 filters with a kernel size of 3 × 3. In addition, the stride size was 1, where the stride refers to the movement across the image, meaning the filter moves one pixel at a time. We used zero padding, which prevented information loss around the perimeter of the image. We also set the initial random weights of the layers using the HeNormal initializer. HeNormal draws the weights from a normal distribution $X \sim \mathcal{N}(\mu, \sigma^2)$ with the mean μ and standard deviation σ defined as in (2).

$$\begin{aligned} \mu &= 0, \\ \sigma^2 &= \sqrt{\frac{2}{fan_{in}}}, \end{aligned} \tag{2}$$

where fan_{in} is the number of input units in the weight tensor. In addition, the ReLU function (1) was used as the activation function in the hidden layers.

Pooling Layer: To minimize the size of the feature maps, a pooling layer is typically placed after the convolution layer. Pooling is applied to each feature map and can be performed in several ways, such as max-pooling, average-pooling, or min-pooling with a $q \times q$ window, where q is the filter size [1]. We used max-pooling to construct a two-dimensional pooling layer. Figure 2 shows that MP1 was the first pooling layer. This layer managed each feature map individually, resulting in an output of 64 feature maps, each with a size of 16 × 16. The layer selected the max-pooling layer over 2 × 2 area; then, a batch normalization layer was used to normalize the output.

Convolutional Layer 3: Convolutional Layer 3 was also used to extract features. This layer and its specifications were very similar to those of the previous convolutional layers. The input data, consisting of feature maps received from the first convolutional layer, were processed with 128 convolutional kernels, each with a size of 3 × 3. A stride of one pixel and zero padding were used. Thus, we extracted 128 feature maps with a size of 16 × 16. We also employed the ReLU activation function, followed by a max-pooling layer of the same size as the previous max-pooling layer, resulting in an 8 × 8 layer. An additional layer of batch normalization was used after this layer.

Convolutional Layer 4: Convolutional Layer 4 performed similar tasks to the previous convolutional layers with a filter size of 128 to extract additional features. We also employed the activation function ReLU followed by a max-pooling layer with the same size as the previous max-pooling layer, to create a 4 × 4 layer followed by a batch normalization layer. Finally, the dropout technique was used in each convolutional layer except the first one. The probability of the dropout rate in the network was 20%.

Fully Connected Layer: There can be any number of fully connected layers after the convolutional and pooling layers. Each neuron in a fully connected layer is connected to every other neuron in the layers around it. These layers are used to generate predictions and provide the final nonlinear feature combinations of the network. Therefore, in our case, the output of the previous convolutional layers was flattened to create a single long feature vector. Figure 2 shows the last three steps of the learning process, including the fully concatenated layers FC1 and FC2, which provided 512 and 1,024-dimensional feature vectors, respectively. In addition, FC3 was the output layer, and it provided 29 classes using the Softmax activation function, see Equation (3), where N was the number of output classes.

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_{k=1}^{N} e^{x_k}} \tag{3}$$

Optimization: We employed Adam, a computationally efficient optimization solver for neural network algorithms, as an optimizer [25]. Furthermore, we used the categorical cross-entropy loss function for this model to estimate the loss (error) and make comparisons and measurements of the distance between the prediction and the correct results; this was used for multiclass classification.

3.2.2. VGG-16 Model

We used a pre-trained model VGG-16 and retrained part of it on the Hijja dataset. We preprocessed the size of the image as follows: since the size of the Hijja image was 32 × 32, we resized it to 224 × 224 to fit the VGG16 model, which was trained on an image of the same size. The trained model did not work well when applied to new and unknown datasets due to various dataset constraints, such as the lack of images in the dataset. Therefore, we divided the Hijja dataset into three sections: 40% for training, 48% for validation, and 12% for testing, with training and validation accounting for most of the data.

The VGG-16 weights were pre-trained in ImageNet. In our case, we used the VGG-16 model by freezing the weights of the first four groups and using the pre-trained convolutional base. In addition, we added our layers, namely, flattening layers and a fully connected layer with a ReLU activation function, a dropout layer with a ratio of 50%, and at least one Softmax layer as an output layer to distinguish between classes. To train the proposed VGG-16, root mean square propagation was used as the optimizer. Finally, we trained the model using the Hijja dataset.

4. Results and Discussion

4.1. Evaluation Measures

The performance of the proposed deep learning model for multiclass classification was evaluated based on the precision, recall, and F1-score metrics.

1. The precision (P) is the fraction of images that are correctly classified divided by the total number of images classified:

$$Precision(P) = \frac{TP}{TP + FP}. \tag{4}$$

2. The recall (R) is the fraction of correctly classified images divided by the total number of images that belong to class x:

$$Recall(R) = \frac{TP}{TP + FN}. \tag{5}$$

3. The F1-score combines the recall and precision as follows:

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

In the above equations, the parameters are as follows: (1) TP (true positives) refers to the total number of images that were correctly labeled by the classifier as belonging to class x; (2) FP (false positives) refers to the total number of images that were incorrectly labeled as belonging to class x; (3) FN (false negatives) refers to the total number of images that were incorrectly labeled as not belonging to class x; and (4) TN (true negatives) refers to the total number of images that were correctly labeled by the classifier as not belonging to class x.

4.2. Results

The goal of our research was to improve the accuracy of children's handwriting recognition. We used the Hijja dataset to evaluate the performance of the proposed CNN models. Two experiments were conducted. The first one was to classify the dataset into 29 classes, the number of letters in the alphabet including *hamza*. The second was to classify the dataset into 108 categories, the number of letters in the alphabet in all their separate and connected forms. All the models were trained and tested on the Hijja dataset, and all the models were designed for multiclass classification of the alphabet into 29 or 108 labels.

4.2.1. Results for the 29 Classes (CNN and VGG-16)

Our aim was to demonstrate how the CNN and VGG-16 performed after being trained with the Hijja. Table 3 compares their performance measures. Regarding the classification accuracy, precision, recall, and F1-score, the CNN performed better than the VGG-16, achieving an accuracy value of 99% on the testing set. Table 3 also shows that all class numbers from 0 to 27 corresponded to the alphabet in order from *alif* to *ya*, while classification number 28 corresponded to *hamza*.

Table 3. The results and comparison of the CNN and VGG16 models for the 29 classes of the Hijja dataset.

	Model	**CNN**			**VGG-16**		
Character	**Class**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
أ	**0**	1.00	1.00	1.00	0.96	0.95	0.96
ب	**1**	0.99	1.00	0.99	0.78	0.91	0.84
ت	**2**	0.99	0.99	0.99	0.92	0.81	0.86
ث	**3**	1.00	1.00	1.00	0.92	0.89	0.90
ج	**4**	1.00	0.99	1.00	0.95	0.80	0.87
ح	**5**	0.99	1.00	0.99	0.72	0.94	0.82
خ	**6**	0.99	0.98	0.98	0.88	0.82	0.85
د	**7**	0.97	0.98	0.97	0.88	0.85	0.87
ذ	**8**	0.99	0.94	0.96	0.96	0.82	0.89
ر	**9**	0.98	0.98	0.98	0.84	0.67	0.75
ز	**10**	1.00	0.99	0.99	0.98	0.52	0.68
س	**11**	1.00	1.00	1.00	0.91	0.95	0.93
ش	**12**	1.00	1.00	1.00	0.76	0.66	0.71
ص	**13**	0.99	1.00	0.99	0.71	0.88	0.79
ض	**14**	0.99	1.00	0.99	0.85	0.88	0.86
ط	**15**	1.00	1.00	1.00	0.90	0.89	0.89
ظ	**16**	0.99	1.00	1.00	0.88	0.87	0.87
ع	**17**	0.97	0.99	0.98	0.81	0.76	0.79
غ	**18**	0.99	0.97	0.98	0.69	0.93	0.79
ف	**19**	0.98	0.99	0.98	0.88	0.87	0.87
ق	**20**	1.00	0.99	0.99	0.96	0.90	0.92
ك	**21**	1.00	0.96	0.98	0.76	0.77	0.77
ل	**22**	0.99	0.99	0.99	0.80	0.85	0.82
م	**23**	1.00	0.99	0.99	0.76	0.90	0.83
ن	**24**	0.96	0.99	0.97	0.91	0.88	0.90
هـ	**25**	1.00	0.99	1.00	0.76	0.90	0.83
و	**26**	1.00	1.00	1.00	0.84	0.79	0.82
ي	**27**	1.00	0.98	0.99	0.62	0.81	0.71
ء	**28**	0.98	0.98	0.98	0.82	0.56	0.67
	accuracy (train)			**0.96**			**0.90**
	accuracy (test)			**0.99**			**0.83**
	macro avg	0.99	0.99	0.99	0.84	0.83	0.83
	weighted avg	0.99	0.99	0.99	0.85	0.83	0.83

4.2.2. Performance Comparison with Models and Datasets from the Literature

We compared our classification results with those of different models and datasets from the literature. Table 4 shows our results alongside the results of a number of other models on the Hijja and AHCD datasets. Table 5 shows the detailed results of running our proposed models using the AHCD dataset. Here, the CNN also performed better than the VGG-16, achieving an accuracy value of 98% on the testing set. AHCD [26] is composed of 16,800 characters written by 60 participants, whose age range was between 19 and 40 years. It should be noted that Alrobah and Albahli [2] reported results for 12,776 images (out of 47,424) containing isolated letters, thereby improving their performance measures. The lower performance of the VGG-16 may be attributed to the need for resizing the input images, as well as the original size of the network, leading us to conclude that a smaller network is more desirable for a simpler classification task. The last row in the table shows the training accuracy obtained in the present study.

Table 4. Performance comparison with models from the literature.

References	Accuracy	Precision	Recall	F-Measure	Datasets
Altwaijry et al. [1]	88%	88%	88%	88%	Hijja
Alrobah et al. [2]	96.3%	-	-	-	Part of Hijja
Proposed VGG-16	83%	85%	83%	83%	Hijja
Proposed CNN	**99%**	**99%**	**99%**	**99%**	**Hijja**
El-Sawy et al. [16]	94.9%	-	-	-	AHCD
Younis et al. [17]	97.6%	-	-	-	AHCD
Alyahya et al. [18]	98.3%	98.35%	-	-	AHCD
Altwaijry et al. [1]	97%	97%	97%	97%	AHCD
Proposed VGG-16	94%	95%	94%	94%	AHCD
Proposed CNN	**98%**	**99%**	**99%**	**99%**	**AHCD**

Comparing the models shows that our model was more lightweight than that of Alrobah et al. [2], whilst it was marginally larger than that of Altwaijry et al. [1]; yet, it achieved a substantially improved performance on the Hijja dataset. Smaller models are faster to train than larger models, and yield better performance on datasets with fewer classes and smaller image sizes than the ImageNet dataset, which has over 14 million images in 22,000 classes. They also usually perform better than large models such as VGG with transfer learning on smaller datasets. The model proposed by El-Sawy et al. [16] was likely too small to handle the intricacies of a handwriting dataset, such as the Hijja, which was reflected in its performance [1]. The AHCD dataset was easier to classify than the Hijja dataset, which allowed both models in [16,17] to improve their performance on the AHCD despite the model size, and a moderate increase in model size [20] did not translate into a large improvement on the AHCD. A large model such as [18] achieved improved performance on the AHCD, and yet its initial training time and subsequent transfer learning training made our model a more attractive option, with a similar performance to a smaller model [5].

4.2.3. Results for the 108 Classes (CNN and VGG-16)

We aimed to test the performance of the proposed models in recognizing all variations of the Arabic handling of connected and disconnected letters. Therefore, we trained the models using the Hijja dataset. The VGG-16 failed to recognize many classifications due to the dataset limitations. Overall, the CNN model outperformed the VGG-16 and achieved an accuracy of 95%.

4.3. Computational Complexity

Table 6 shows the computational time for our proposed CNN model on both the Hijja and AHCD datasets. We report the average training and prediction times in seconds. Hijja is a larger and more complicated dataset compared to the AHCD. Thus, the training and prediction times were longer using Hijja. The larger number of parameters for the Hijja dataset was the result of having 29 classes instead of 28.

Table 5. The results and comparison of the CNN and VGG16 models on the AHCD dataset.

	Model		**CNN**			**VGG-16**	
Character	**Class**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
أ	**0**	1.00	1.00	1.00	1.00	0.96	0.98
ب	**1**	0.97	1.00	0.98	0.88	0.98	0.93
ت	**2**	0.99	0.99	0.99	0.95	0.85	0.90
ث	**3**	0.99	0.99	0.99	0.96	0.97	0.97
ج	**4**	1.00	0.84	0.91	0.97	0.97	0.97
ح	**5**	0.86	1.00	0.93	0.95	0.95	0.95
خ	**6**	0.99	1.00	0.99	0.96	0.96	0.96
د	**7**	0.99	1.00	0.99	0.97	0.96	0.96
ذ	**8**	0.99	0.97	0.98	0.92	0.96	0.94
ر	**9**	0.97	0.99	0.98	0.96	0.90	0.93
ز	**10**	0.98	1.00	0.99	0.98	0.97	0.97
س	**11**	1.00	0.99	1.00	0.99	1.00	1.00
ش	**12**	1.00	1.00	1.00	0.93	0.91	0.92
ص	**13**	0.98	1.00	0.99	0.87	0.93	0.90
ض	**14**	1.00	0.99	1.00	0.93	0.93	0.93
ط	**15**	0.98	0.99	0.98	0.98	0.93	0.96
ظ	**16**	1.00	0.97	0.98	0.96	0.97	0.97
ع	**17**	1.00	0.99	0.99	0.96	0.76	0.85
غ	**18**	0.99	0.99	0.99	0.94	0.95	0.95
ف	**19**	0.98	0.99	0.99	0.96	0.92	0.94
ق	**20**	0.99	0.98	0.99	0.99	0.96	0.97
ك	**21**	1.00	1.00	1.00	0.81	1.00	0.89
ل	**22**	1.00	0.98	0.99	1.00	0.88	0.94
م	**23**	1.00	1.00	1.00	0.97	0.95	0.96
ن	**24**	1.00	0.99	1.00	0.93	0.98	0.96
هـ	**25**	1.00	0.99	0.99	0.95	0.97	0.96
و	**26**	0.99	1.00	1.00	0.85	0.98	0.91
ي	**27**	1.00	1.00	1.00	0.93	0.93	0.93
	accuracy (train)			**0.97**			**0.97**
	accuracy (test)			**0.98**			**0.94**
	macro avg	0.99	0.99	0.99	0.95	0.94	0.94
	weighted avg	0.99	0.99	0.99	0.95	0.94	0.94

Table 6. Computational Complexity.

	Training	**Prediction**		**Parameters**	
	Time	**Time**	**Trainable**	**Non-Trainable**	**Total**
Hijja					
Proposed CNN	202s	1.96 s	5,123,741	896	5,124,637
AHCD					
Proposed CNN	30.24s	0.68 s	5,123,228	896	5,124,124

5. Development of the Mutqin Application

We developed a prototype that integrates the CNN model into a useful tablet application for children to practice Arabic handwriting and spelling skills. First, we created a questionnaire to investigate the need for the development of an educational application to practice writing and spelling skills in Arabic and to gather functional requirements for such an application. We collected information from eight teachers. The results indicated that there was a need for the development of an application with essential and attractive features to encourage children to practice spelling and writing skills. Moreover, the results helped us to identify the most important functional requirements to build an application appropriate for the target audience. The target user is between 7 and 12 years old. The application offers two options: practice writing Arabic letters and practice spelling skills, as presented in Figure 3. The application allows a user to practice writing by displaying the writing board, which contains pens of different colors and an eraser, as presented in Figure 4. Moreover, the user receives appropriate feedback based on the input, as presented in Figure 5. Considering children are the main audience, the application was designed to be very clear and easy to use and understand.

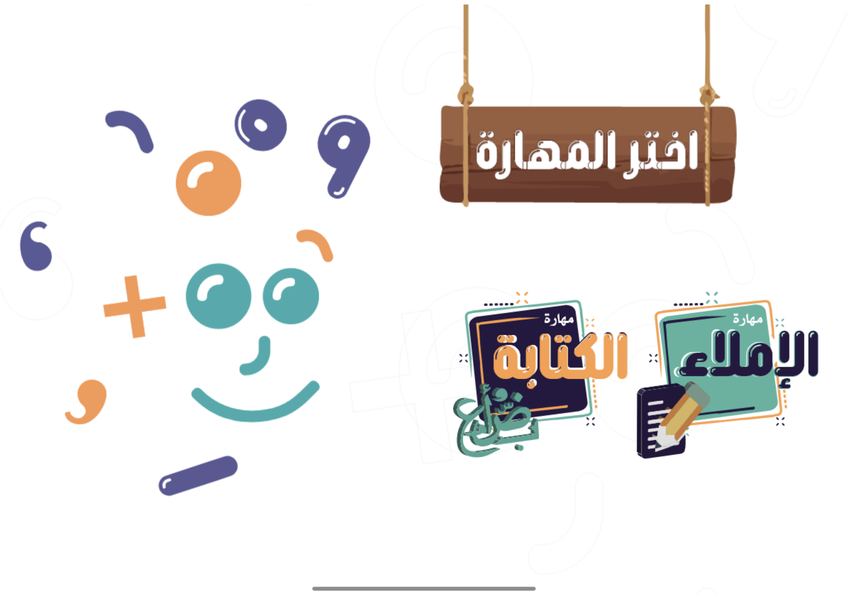


Figure 3. Homepage interface.

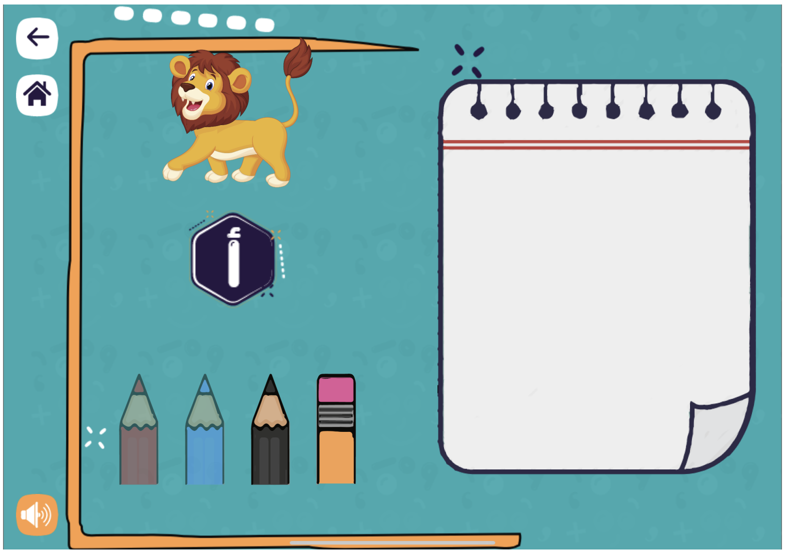


Figure 4. The writing board page interface.

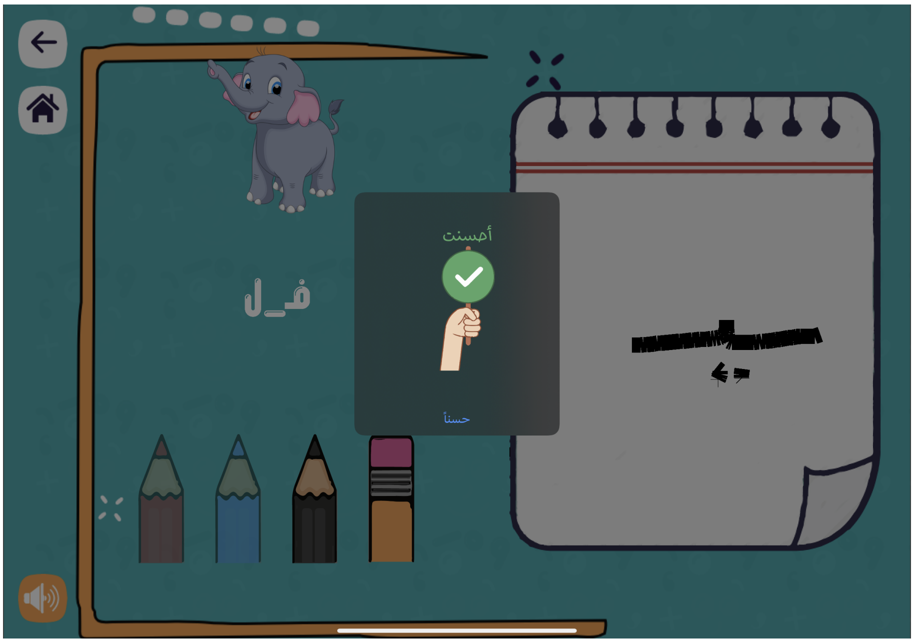


Figure 5. Feedback interface.

Then, we implemented user acceptance testing, which is a necessary step before releasing software into the real world. It is used to ensure that the application accomplishes what it aims to do and meets the requirements [27]. We conducted user acceptance testing with 11 users. The participants were between 7 and 12 years old. The evaluation was based on three usability criteria: effectiveness, efficiency, and satisfaction. We recorded the time it took users to perform a particular function to ensure efficiency. To evaluate the application's effectiveness, we calculated the average number of errors in certain tasks and the average time it took users to complete the tasks, using both the stylus and the finger, for the application's target group.

As shown in Table 7, all the functions passed the test and were completed in a reasonable amount of time; hence, it was concluded that the application was efficient and saved time. Finally, a survey was conducted to measure the user satisfaction with the application [28], and the users were satisfied.

Table 7. User acceptance test for stylus and finger tasks.

Task (Stylus)	Average Number of Errors	Average Time in Seconds	Result
Choose skill	0.00	4.03	Pass
Hear sound	0.09	5.15	Pass
Change pen color	0.00	5.58	Pass
Write on board	1.73	28.87	Pass
Erase	0.00	2.99	Pass
Task (Finger)			
Choose skill	0.00	3.45	Pass
Hear sound	0.00	4.07	Pass
Change pen color	0.00	3.16	Pass
Write on board	1.36	26.64	Pass
Erase	0.00	1.03	Pass

6. Conclusions and Further Work

In this paper, we proposed two deep-learning models. We designed a CNN and fine-tuned a VGG-16 pre-trained model for Arabic handwriting character recognition. The models were designed to classify the letters into 29 and 108 classes based on letter shape. The research was carried out using the Hijja dataset for children's handwriting. We evaluated the performances of the CNN and VGG-16 and compared them with each other. We also trained and tested our proposed models using the Arabic Handwritten Character Dataset (AHCD). We then compared the results with the results of approaches from the literature. We selected the best-performing model among them, namely, the CNN model, which was able to outperform state-of-the-art classification algorithms using the Hijja dataset and achieved 99% accuracy. Then, we developed Mutqin, a prototype for children to practice writing and spelling skills, and integrated it with our CNN model. Furthermore, we evaluated the Mutqin application using user acceptance testing to measure three criteria: effectiveness, efficiency, and satisfaction. Our results indicated that the application performed well.

This research could be extended in many ways. The developed deep learning models could be trained on a combination of datasets to improve performance. They could also be trained to recognize connected letters and words. Such models can be used in writing and dictation applications. In addition, models can be built for the recognition of Arabic calligraphy. In the future, more features can be added to the prototype, such as the ability to share results with parents and teachers, personalization with photos, and support for different platforms. The application of gesture-based handwriting could be investigated in the context of writing applications for children in Arabic and other languages.

Author Contributions: Conceptualization, I.A.-T. and N.A.; methodology, I.A.-T., N.A., A.A.-A. and A.B.D.; software, A.B.D. and N.A.; validation, A.B.D. and N.A.; discussion and results, A.A.-A., I.A.-T. and N.A.; writing—original draft preparation, A.A.-A.; writing—review and editing, A.A.-A., I.A.-T. and N.A.; supervision, I.A.-T. All authors have read and agreed to this version of the manuscript.

Funding: This research project was supported by a grant from the "Research Center of the Female Scientific and Medical Colleges", Deanship of Scientific Research, King Saud University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Altwaijry, N.; Al-Turaiki, I. Arabic handwriting recognition system using convolutional neural network. *Neural Comput. Appl.* **2021**, *33*, 2249–2261. [CrossRef]
2. Alrobah, N.; Albahli, S. A Hybrid Deep Model for Recognizing Arabic Handwritten Characters. *IEEE Access* **2021**, *9*, 87058–87069. [CrossRef]
3. Hong, D.; Gao, L.; Yao, J.; Zhang, B.; Plaza, A.; Chanussot, J. Graph Convolutional Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote. Sens.* **2021**, *59*, 5966–5978. [CrossRef]
4. Wu, X.; Hong, D.; Chanussot, J. UIU-Net: U-Net in U-Net for Infrared Small Object Detection. *IEEE Trans. Image Process.* **2023**, *32*, 364–376. [CrossRef]
5. AlJarrah, M.N.; Zyout, M.M.; Duwairi, R. Arabic Handwritten Characters Recognition Using Convolutional Neural Network. In Proceedings of the 2021 12th International Conference on Information and Communication Systems (ICICS), Valencia, Spain, 24–26 May 2021; pp. 182–188.
6. Corbille, S.; Fromont, E.; Anquetil, E.; Nerdeux, P. Integrating Writing Dynamics in CNN for Online Children Handwriting Recognition. In Proceedings of the 2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), Dortmund, Germany, 8–10 September 2020; pp. 270–275. [CrossRef]
7. Jolly, C.; Palluel-Germain, R.; Gentaz, E. Evaluation of a tactile training for handwriting acquisition in French kindergarten children: A pilot study. In *Kindergartens: Teaching Methods, Expectations and Current Challenges*; Nova Science Publishers: Hauppauge, NY, USA, 2013; pp. 161–176.

8. El-Sawy, A.; Loey, M.; EL-Bakry, H. Arab Kids Tutor (AKT) System For Handwriting Stroke Errors Detection. *Int. J. Technol. Enhanc. Emerg. Eng. Res.* **2016**, *4*, 8.
9. Alheraki, M.; Al-Matham, R.; Al-Khalifa, H. Handwritten Arabic Character Recognition for Children Writ-ing Using Convolutional Neural Network and Stroke Identification. *arXiv* **2022**, arXiv:2211.02119.
10. Das, N.; Mollah, A.F.; Saha, S.; Haque, S.S. Handwritten arabic numeral recognition using a multi layer perceptron. *arXiv* **2010**, *4*, arXiv:1003.1891.
11. Ashiquzzaman, A.; Tushar, A.K. Handwritten Arabic numeral recognition using deep learning neural networks. In Proceedings of the 2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR), Dhaka, Bangladesh, 13–14 February 2017; pp. 1–4.
12. Alani, A. Arabic Handwritten Digit Recognition Based on Restricted Boltzmann Machine and Convolutional Neural Networks. *Information* **2017**, *8*, 142. [CrossRef]
13. Mudhsh, M.; Almodfer, R. Arabic Handwritten Alphanumeric Character Recognition Using Very Deep Neural Network. *Information* **2017**, *8*, 105. [CrossRef]
14. Latif, G.; Alghazo, J.; Alzubaidi, L.; Naseer, M.M.; Alghazo, Y. Deep Convolutional Neural Network for Recognition of Unified Multi-Language Handwritten Numerals. In Proceedings of the 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR), London, UK, 12–14 March 2018; pp. 90–95.
15. Ashiquzzaman, A.; Tushar, A.K.; Rahman, A.; Mohsin, F. An efficient recognition method for handwritten arabic numerals using CNN with data augmentation and dropout. In *Data Management, Analytics and Innovation*; Balas, V.E., Sharma, N., Chakrabarti, A., Eds.; Springer: Singapore, 2019; Volume 808, pp. 299–309.
16. El-Sawy, A.; Loey, M.; El-Bakry, H. Arabic handwritten characters recognition using convolutional neural network. *WSEAS Trans. Comput. Res.* **2017**, *5*, 11–19.
17. Younis, K.; Khateeb, A. Arabic Hand-Written Character Recognition Based on Deep Convolutional Neural Networks. *Jordanian J. Comput. Inf. Technol.* **2017**, *3*, 186. [CrossRef]
18. Alyahya, H.; Al-Salman, A.; Ben Ismail, M.M. Deep ensemble neural networks for recognizing isolated Arabic handwritten characters. *ACCENTS Trans. Image Process. Comput. Vis.* **2020**, *6*, 2455–4707. [CrossRef]
19. Elkhayati, M.; Elkettani, Y. Towards directing convolutional neural networks using computational geometry algorithms: Application to handwritten Arabic character recognition. *Adv. Sci. Technol. Eng. Syst. J.* **2020**, *5*, 137–147. [CrossRef]
20. Shams, M.; Elsonbaty, A.; ElSawy, W. Arabic Handwritten Character Recognition based on Convolution Neural Networks and Support Vector Machine. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*. [CrossRef]
21. Balaha, H.M.; Ali, H.A.; Saraya, M.; Badawy, M. A new Arabic handwritten character recognition deep learning system (AHCR-DLS). *Neural Comput. Appl.* **2021**, *33*, 6325–6367. [CrossRef]
22. Balaha, H.M.; Ali, H.A.; Youssef, E.K.; Elsayed, A.E.; Samak, R.A.; Abdelhaleem, M.S.; Tolba, M.M.; Shehata, M.R.; Mahmoud, M.R.; Abdelhameed, M.M.; et al. Recognizing arabic handwritten characters using deep learning and genetic algorithms. *Multimed. Tools Appl.* **2021**, *80*, 32473–32509. [CrossRef]
23. Aizan, N.L.K.; Mansor, E.I.; Mahmod, R. Preschool children handwriting evaluation on paper-based and tablet-based settings. *Int. J. Comput. Inf. Technol.* **2014**, *3*, 7.
24. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Ft. Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
25. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2017**, arXiv:1412.6980.
26. El-Sawy, A.; Loey, M.; Hazem, E. Arabic Handwritten Characters Dataset. 2017. Available online: https://www.kaggle.com/mloey1/ahcd1 (accessed on 30 May 2019).
27. Hambling, B.; Van Goethem, P. *User Acceptance Testing: A Step-by-Step Guide*; BCS, The Chartered Institute for IT: Swindon, UK, 2013.
28. Sualim, S.A.; Yassin, N.M.; Mohamad, R. Comparative evaluation of automated user acceptance testing tool for web based application. *Int. J. Softw. Eng. Technol.* **2016**, *2*, 7.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.