The Impact of Various Factors on the Convolutional Neural Networks Model on Arabic Handwritten Character Recognition

Alhag Alsayed[1], Chunlin Li[2], Ahmed Fat'hAlalim[3], Mohammed Hafiz[4],
Jihad Mohamed[5], Zainab Obied[6], Mohammed Abdalsalam[7]
School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Hubei, China[1,2,3,4,5,6,7]
University of Gadarif, Gadarif, Sudan[1]
University of Holy Quran and Islamic Science, Sudan[7]

***Abstract*—Recognizing Arabic handwritten characters (AHCR) poses a significant challenge due to the intricate and variable nature of the Arabic script. However, recent advancements in machine learning, particularly through Convolutional Neural Networks (CNNs), have demonstrated promising outcomes in accurately identifying and categorizing these characters. While numerous studies have explored languages like English and Chinese, the Arabic language still requires further research to enhance its compatibility with computer systems. This study investigates the impact of various factors on the CNN model for AHCR, including batch size, filter size, the number of blocks, and the number of convolutional layers within each block. A series of experiments were conducted to determine the optimal model configuration for the AHCD dataset. The most effective model was identified with the following parameters: Batch Size (BS) = 64, Number of Blocks (NB) = 3, Number of Convolution Layers in Block (NC) = 3, and Filter Size (FS) = 64. This model achieved an impressive training accuracy of 98.29% and testing accuracy of 97.87%.**

***Keywords*—Arabic Handwritten Character Recognition (AHCR); Optical Character Recognition (OCR); Deep Learning (DL); Convolutional Neural Network (CNN); Characters Recognition (CR)**

I. Introduction

Handwritten character recognition (HCR), also known as handwriting recognition or optical character recognition (OCR), involves converting handwritten text or characters into digital text for computer processing and comprehension. This process entails analyzing and interpreting the shapes and patterns of handwritten characters to identify specific letters, numerals, or symbols. HCR is essential for facilitating the computer understanding and interpretation of human handwriting, thereby bridging the gap between digital and analog domains [1, 2, 3]. The Arabic script, while rich in history and complexity, poses significant challenges in the domain of digital text processing, particularly in handwritten character recognition (HCR) [26]. Arabic is considered a low-resource language in computational linguistics due to the limited availability of annotated datasets and comprehensive research compared to languages like English or Chinese. This disparity stems from several inherent features of the script and the linguistic nuances of the language [5].

Arabic handwriting recognition is particularly challenging due to the script's cursive nature, where most letters within a word are connected, and the same letter can have up to four different shapes depending on its position within a word. Additionally, the presence of diacritical marks adds another layer of complexity, as these marks can significantly alter the meaning and pronunciation of words, yet they are often omitted in everyday writing [28,29].

The scarcity of extensive and varied datasets hampers the development of robust models capable of effectively handling the wide variability in handwriting styles. This gap highlights the urgent need for more focused research and resource development to enhance Arabic HCR technologies. An overview of the Arabic alphabetic characters used in such datasets can be seen in Fig. 1. This figure presents each Arabic character alongside its English transliteration, offering insights into the complexities of Arabic script recognition and the challenges involved in designing effective OCR systems for such scripts. Handwritten character recognition systems often employ machine learning algorithms and techniques to train models using extensive datasets of handwritten samples. The stages through which data progresses in handwritten character recognition systems are succinctly illustrated in Fig. 2.

This figure outlines a systematic workflow beginning with the initial step of splitting the dataset into distinct sets for training and testing. Following this, the data undergoes preprocessing, which typically involves normalization, scaling, and possibly augmentation techniques to enhance the robustness of the data before it is fed into the model. The subsequent phase involves training the Convolutional Neural Network (CNN) where the model learns to identify and classify the handwritten characters based on the features extracted from the training data. The final stage is testing, where the trained CNN is evaluated on a separate set of data to assess its performance and accuracy in recognizing new, unseen handwritten characters.

Pattern recognition (PR) involves the identification and acknowledgment of different elements in inputs such as images, sounds, or sequences of characters. The process typically includes measuring the object to identify distinctive features, extracting features related to these defining characteristics, and then comparing the outcomes with established patterns to ascertain a match or absence thereof between the two [6,7].

Convolutional Neural Networks (CNNs) are deep learning algorithms used for analyzing visual data like images and videos. They use multiple layers of interconnected neurons to

أ	ب	ت	ث	ج	ح	خ	د
Alif	Ba	Ta	Tha	Jeem	Ha	Kha	Dal
ذ	ر	ز	س	ش	ص	ض	ط
Dhal	Ra	Zay	Seen	Sheen	Sad	Dad	Ta
ظ	ع	غ	ف	ق	ك	ل	م
Za	Ain	Ghain	Fa	Qaf	Kaf	Lam	Meem
ن	ه	و	ي				
Noon	Ha	Waw	Ya				

Fig. 1. Arabic alphabet character.

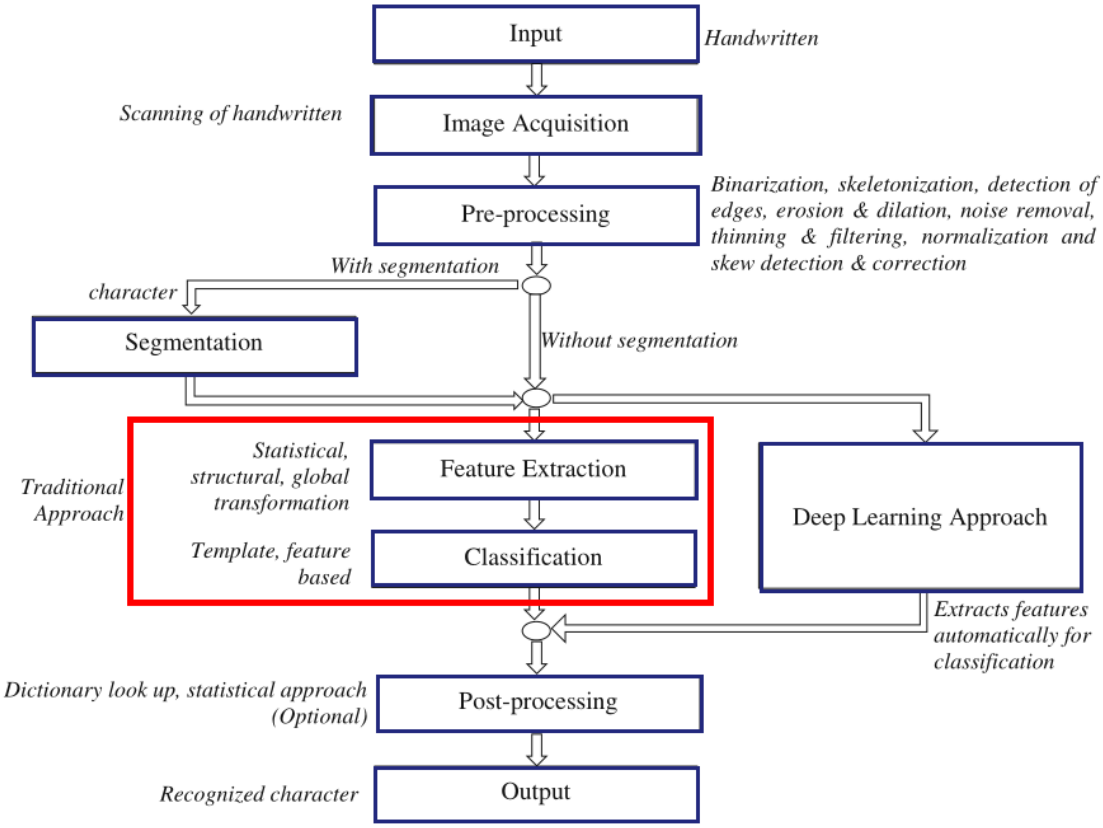

Fig. 2. Stage for handwritten character recognition.

learn and extract hierarchical representations. CNNs are effective in computer vision applications like image recognition, object detection, and segmentation. They use convolutional layers, pooling layers, and fully connected layers to learn local patterns, reduce computational complexity, and make predictions based on extracted features. as shown in Fig. 4 [8,9,10,11].

Arabic, a Semitic language from the Arabian Peninsula, is spoken by millions, particularly in the Middle East and North Africa. It is one of six UN-recognized official languages and has significant cultural, religious, and historical significance [1,12].

Arabic handwriting recognition technology is crucial in the Middle East and North Africa, enhancing communication, education, and commerce. Implementing in businesses streamlines customer orders, digitizes essential documents, and allows individuals to input Arabic text without language switches or transliteration, improving efficiency and accessibility [7,13,14,15,16]. Technological advancements in machine learning and deep neural networks are enhancing Arabic handwriting recognition, enhancing cultural heritage preservation and paving the way for future breakthroughs [13,15].

This study focuses on Arabic Handwritten Character Recognition (AHCR) using Convolutional Neural Networks (CNNs). The proposed study investigated the impact of various factors on the CNN model's performance, such as batch size, filter size, number of blocks, and the number of convolution layers within the block. The remaining sections of the paper are structured as follows: Section 2 provides an overview of related works, Section 3 details the proposed methodology, Section 4 presents the experimental results, and Section 5 concludes the study and outlines directions for future research.

II. Related Works

El-Sawy et al. [8], for this research, a substantial dataset of handwritten Arabic characters, comprising 16,800 unique character types, was acquired (referred to as AHCD) show Table I. The study suggests the implementation of a deep learning model based on Convolutional Neural Networks (CNNs). Through optimization using specific techniques, the proposed CNN model attained an accuracy level of 94.9%.Younis et al. [9], the study used deep neural networks for recognizing handwritten Arabic characters, incorporating batch normalisation and dropout techniques. The CNN model achieved accuracies of 97.2% and 94.8% for the AHCD and AIA9K datasets, demonstrating its effectiveness in this area.De Sousa et al. [17], the study demonstrates offline Arabic handwriting recognition using two extensive Arabic numeral and letter datasets. Four convolutional neural networks were used, with two training sessions and AHCD character dataset. The system achieved a validation accuracy of 98.60% and classification accuracy of 98.42%.

Najadat et al. [18], the researchers developed a Convolutional Neural Network (CNN) architecture for the AHCD dataset, enabling character classification in a unified pipeline. By adjusting CNN parameters, they achieved an accuracy of 97.2% in Arabic character recognition. Almansari et al. [19], the project aims to combine a multilayer perceptron (MLP) neural network with a convolutional neural network (CNN) to create a deep learning architecture using Python. The research evaluates the performance of the Arabic Handwritten Characters Dataset (AHCD), showing a 95.27% accuracy increase after CNN training, but a 72.08% decline after MLP training.

Alyahya et al. [20], the study evaluated the ResNet-18 architecture's effectiveness in distinguishing handwritten Arabic letters using two ensemble models. The original ResNet-18 achieved 98.30% accuracy, while ensemble models with fully connected layers and dropout layers achieved 98.00% and 98.03% accuracy.

Shams et al. [11], the study developed a hybrid model using DCNN, SVM, and k-means clustering for multi-stroke Arabic character recognition, outperforming the El-Sawy model in training and evaluation.

Altwaijry et al. [2], researchers developed a CNN-based model using the Hijja dataset and AHCD, achieving an accuracy of 88% on the Hijja dataset and 97% on the AHCD dataset, focusing on Arabic-based recognition algorithms.

AlJarrah et al. [4], a Convolutional Neural Network (CNN) was developed to recognize handwritten Arabic letters, trained on a dataset of 16,800 images. After analyzing 40 and 256 data sets, the CNN achieved an accuracy rate of 97.2%.

Kamal et al. [21], the proposed 18-layer pipeline, which includes convolution, pooling, batch normalization, dropout, global average pooling, and dense layers, achieved 96.93% accuracy on AHCD and MadBase datasets, making it suitable for real-world applications.

Wagaa et al. [22], the study proposes a convolutional neural network for Arabic letter classification using seven optimization methods. It employs data augmentation strategies and dropout regularization to address overfitting. The method outperforms existing models, achieving high recognition accuracy on the AHCD and Hijja datasets.

Elkhayati et al. [23], researchers have developed a Convolutional Neural Network (CNN) model to improve the recognition of isolated handwritten Arabic characters (IHAC) by reducing complexity and improving performance. Drawing inspiration from psychology and cognitive science, the model uses virtual max-pooling at the flattening layer, focusing on typical IHAC characteristics.

Khudeyer et al. [24], the study shows that using ResNet50 in combination with random forests yields more accurate predictions, with a 95% accuracy rate for the AIA9K, AHCD, and Hijja datasets, surpassing the modified ResNet50 architecture's 92.37%, 98.39%, and 91.64% accuracy rates.

Bin Durayhim et al. [25], it used the Arabic Handwritten Character Dataset (AHCD) to train three models: a convolutional neural network (CNN), a pre-trained CNN (VGG-16), and a fully-trained CNN. The models achieved an accuracy of 98.0% across the AHCD dataset, with even higher accuracy on the Hijja dataset.

III. Methodology

The AHCD classification problem involves identifying individual Arabic characters from isolated images. This task is a fundamental problem in the field of OCR and has extensive applications in digital text processing, automated reading systems, and educational technologies Fig. 3. In the AHCD classification task, each input is an image of a handwritten Arabic character, and the goal is to assign the correct label from a set of possible Arabic letters. The AHCD includes images of 28 basic Arabic letters, each written by numerous individuals to capture a wide range of handwriting styles. Let X represent the set of all possible images in the AHCD, where each image $x \in X$ is a grayscale bitmap of fixed size, which is 32×32 pixels. Let Y denote the set of Arabic character labels, where $Y = \{y_1, y_2, \ldots, y_{28}\}$, corresponding to the 28 letters of the Arabic alphabet. The classification task can be defined as learning a function $f : X \to Y$ that maps each image x to a label y. The function f is typically represented by a parameterized model f_θ, where θ denotes the parameters of the model. The goal of learning is to find the optimal parameters θ^* such that the predictive accuracy of f_{θ^*} on unseen data is maximized.

1) Data representation: Each image x is represented as a vector $\mathbf{x} \in \mathbb{R}^n$, where n is the number of pixels in the image (e.g., $32 \times 32 = 1024$ pixels). This vectorization is typically achieved by flattening the 2D pixel array into a 1D vector.

TABLE I. COMPARISON OF ARABIC HANDWRITTEN CHARACTER RECOGNITION STUDIES

Ref/Year	Datasets Used	Method	Result (%)	Work Limitations
El-Sawy et al.[8] 2017	AHCD	CNN	94.9	Limited optimization techniques for CNN
Younis et al.[9] 2018	AHCD, AIA9K	Deep CNN with batch normalization	97.6 (AHCD), 94.8 (AIA9K)	Susceptibility to overfitting despite regularization
De Sousa et al.[17] 2018	AHCD, MADbase digits	Ensemble of CNNs	99.74 (digits), 98.60 (AHCD)	Complexity in managing multiple CNN models
Najadat et al.[18] 2019	AHCD	CNN	97.2	Lack of advanced parameter tuning
Almansari et al.[19] 2019	AHCD	MLP and CNN	95.27, 72.08 (MLP)	MLP model reduces overall performance
Alyahya et al[20] 2020	AHCD	Deep ensemble networks with ResNet-18	98.30, 98.00, 98.03	Complexity and computational demand of ensemble models
Shams et al.[11] 2020	AHCD	DCNN, SVM, k-means clustering	95.07 CRR, 4.93% ECR	Integration challenges between DCNN, SVM, and clustering
Altwaijry et al.[2] 2021	Hijja, AHCD	CNN	88 (Hijja), 97 (AHCD)	Lower performance on children's handwriting dataset
AlJarrah et al.[4] 2021	16,800 images of Arabic letters	CNN	97.7 with data augmentation	Data augmentation may not generalize well
Kamal at el.[21] 2022	AHCD, MadBase	18-layer deep learning pipeline	96.93	Potential overfitting due to deep architecture
Wagaa at el.[22] 2022	AHCD, Hijja	CNN with optimization methods	High accuracy	Overfitting despite using multiple optimization methods
Elkhayati at el.[23] 2022	IHAC	CNN with virtual max-pooling	Improved performance	Limited exploration of virtual max-pooling impacts
Khudeyer at el.[24] 2023	AIA9K, AHCD, Hijja	ResNet50 with SVM and RF	Increased accuracy	Integration complexity of CNN with traditional ML
Bin Durayhim et al.[25] 2023	Hijja, AHCD	CNN, VGG-16, Mutqin prototype for children	Outperforms VGG-16 and literature models	Challenges in training deep networks for children's handwriting

2) Model: In this study applied classification model by using Convolutional Neural Network (CNN). The CNN takes the image vector $\mathbf{x}$ and outputs a vector $\mathbf{z}$ of raw class scores, one for each class.

3) Objective function: The learning process involves minimizing a loss function that quantifies the error between the predicted label and the true label for each image in a training set. A common choice for classification tasks is the cross-entropy loss, defined as:

$$L(\theta) = -\sum_{(x,y)\in D} \log p(y \mid x;\theta) \tag{1}$$

where, D is the training dataset and $p(y \mid x;\theta)$ is the softmax probability of the correct label y given the input x and model parameters θ.

4) Evaluated the parameter: The parameters θ are typically learned which iteratively updates the parameters to minimize the loss function.

5) Evaluation: The performance of the model f_{θ^*} is evaluated using accuracy metrics, which is the proportion of correctly classified images in a test set.

The CNN model was selected for its numerous capabilities in image recognition, such as the automatic extraction of features and classification. In this research, we investigate the application of this model in the recognition of handwritten Arabic characters using the AHCD dataset as shown in Fig. 4.

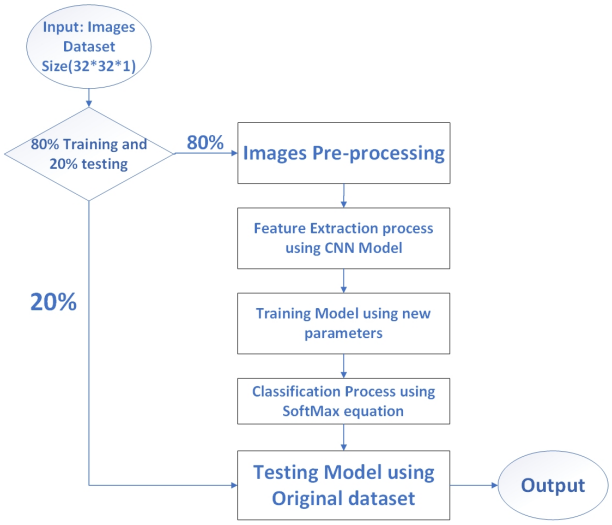

Fig. 3. The Base model.

The Eq. (2) represents the calculation of the output $X(i)$, where $X(i)$ is obtained by summing the element-wise multiplication of the input variables X_i and their corresponding weights W_i, from $i = 1$ to $k-1$, and then adding the bias term B_i.

$$X(i) = \sum_{i=1}^{k-1} X_i * W_i + B_i \tag{2}$$

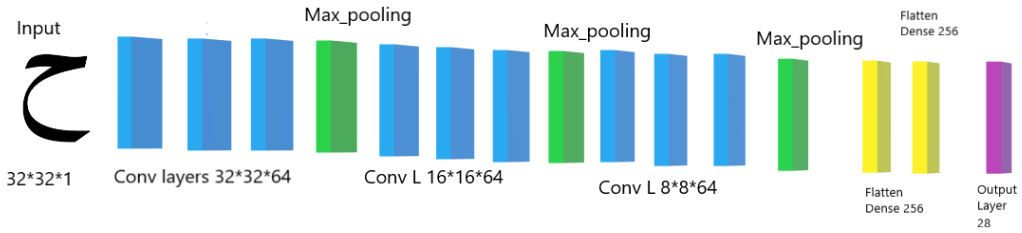

Fig. 4. CNN Model.

Eq. (3) defines the Rectified Linear Unit (ReLU) activation function, denoted as $Relu(x)$. This activation function is a fundamental component in neural networks, commonly used to introduce non-linearity into the model. The ReLU function returns the maximum of 0 and the input value x. In other words, if the input x is positive, the function outputs x, otherwise, it outputs 0. This simple yet effective activation function helps in overcoming the vanishing gradient problem and speeding up the convergence of neural network training.

$$Relu(x) = max(0, x) \tag{3}$$

The Eq. (4) defines the softmax activation function, denoted as $Softmax(x_i)$. This function is commonly used in the final layer of a neural network for multi-class classification tasks. It calculates the probability distribution over N classes for a given input vector x. The softmax function transforms the raw scores x_i into probabilities by exponentiating each score and normalizing them by the sum of all exponentiated scores across all classes. Essentially, it ensures that the output values lie between 0 and 1 and sum up to 1, representing the likelihood of the input belonging to each class. This makes it suitable for determining the class probabilities in classification tasks, allowing the model to make confident predictions about the input data[11,27].

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_{y=1}^{N} e^{x_y}} \tag{4}$$

The Arabic Handwritten Character Recognition (AHCR) application of convolutional neural networks (CNNs) is the main emphasis of this work. The suggested study looked into how different parameters, including batch size, filter size, number of blocks, and number of convolution layers per block, affected the performance of the CNN model. Using the Arabic Handwritten Characters dataset (AHCD), we ran several experiments before determining that the following parameters would yield the best model configuration:

1) We are examining the impact of batch size on the model's accuracy. The images, represented as numerical pixels, are stored on the computer, and data is processed in batches based on the chosen batch size. Initially, we implemented a batch size of 32 and conducted multiple experiments, progressively increasing the model size and the number of convolution layers. Subsequently, we repeated the experiments with a batch size of 64, following the same procedures as in the initial set of experiments as shown Fig. 10 and 13.
2) We are exploring the impact of model size on the performance. Each model comprises a series of blocks, with each block incorporating convolution layers, max-pooling, and dropout Fig. 5. We initiated the study with the most straightforward model, consisting of two blocks Fig. 7, and then progressed to models with three Fig. 6 and four blocks Fig. 5. Throughout this process, we assessed the model's accuracy while varying the number of convolution layers within each block.
3) We investigate how the number of convolution layers within each block influences the outcomes. Blocks containing a series of convolution layers were tested, starting with one layer and progressing to two and three layers. The results were carefully observed after implementing each model. Fig. 5, 8, and 9.
4) We explore the impact of filter size on the model's accuracy. In the convolution layer, a filter is utilized to identify features in the image. This study aims to assess whether the filter size influences the model's accuracy. The investigation commenced with a filter size of 16, followed by transitions to filter sizes of 32 and, ultimately, 64, with corresponding tables displaying the results.

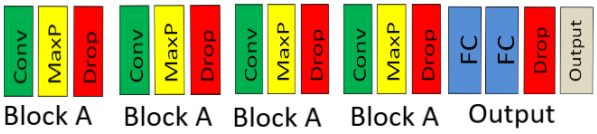

Fig. 5. CNN model with one convolutional layer and four blocks.

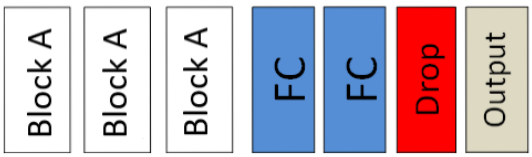

Fig. 6. CNN model with one convolutional layer and three blocks.

IV. Arabic Handwritten Characters Dataset

Arabic Handwritten Characters Datasets play a crucial role in the field of computer vision, particularly in the development

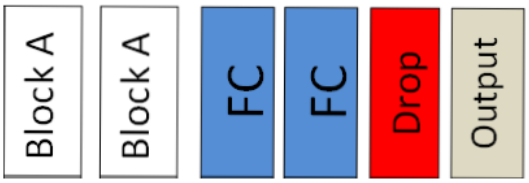

Fig. 7. CNN model with one convolutional layer and two blocks.

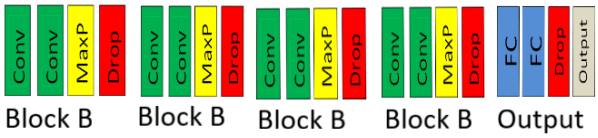

Fig. 8. CNN model with two convolutional layers and four blocks.

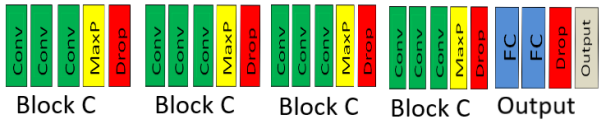

Fig. 9. CNN model with three convolutional layers and four blocks.

and enhancement of Optical Character Recognition (OCR) systems for the Arabic script. These datasets are meticulously designed to mimic real-world handwriting variability, providing a robust resource for researchers and developers to train, test, and validate various machine learning models tailored to recognize Arabic characters. A variety of datasets have been developed to address the specific needs of the Arabic OCR community, ranging from datasets comprising isolated letters to those including complete words or numerals. These datasets vary by the number of samples, the demographics of the participants (e.g., age range, native language skills), and the conditions under which the handwriting samples were collected (e.g., controlled environments versus natural writing settings). Table II describes the common datasets used in this domain, presenting the properties of each dataset.

1) AHCD: This widely used dataset is crucial for benchmarking Arabic handwritten character recognition algorithms. It comprises isolated forms of Arabic letters from multiple writers, making it a robust dataset for training models.

2) AIA9K dataset: Contains a diverse set of Arabic handwritten characters and numerals, designed to be a comprehensive resource for developing and testing recognition algorithms that handle both letters and digits.

3) MADBase dataset: Styled similarly to the popular MNIST dataset but focused on Arabic numerals, it is extensively used in digit recognition tasks and is one of the largest datasets for Arabic digits.

4) HACDB dataset: Similar to AHCD but compiled independently, this dataset is primarily used in academic research for developing new recognition technologies and for testing existing ones.

5) Hijja dataset: Specifically focuses on children's handwriting, providing a unique challenge due to the variability in young writers' script. Access is usually restricted to promote controlled studies and development.

A. The Arabic Handwritten Characters Dataset (AHCD)

The Arabic Handwritten Characters Dataset (AHCD) is an essential resource in the field of pattern recognition and machine learning, specifically geared towards the development and testing of algorithms for Arabic handwritten character recognition. This dataset is particularly valuable due to the unique characteristics of the Arabic script, which includes cursive writing and varying shapes of characters depending on their position within a word. AHCD contains images of isolated characters, which are manually segmented from handwritten words or text lines. The dataset typically includes tens of thousands of images, representing a substantial variety of handwriting styles from multiple writers. The dataset usually covers the entire standard set of Arabic characters, which includes 28 basic letters. However, it include additional forms like ligatures and character variations, depending on the dataset's specific version or extension. The characters in the dataset are often stored as grayscale images, which may vary in resolution but are typically normalized to fit a standard size (e.g., 32x32 or 64x64 pixels) to facilitate uniform processing across different machine learning models and techniques. Each image in AHCD is labeled with the corresponding Arabic character, which allows for supervised learning algorithms to be trained effectively. The dataset, accessible to the public, contains 16,800 characters written by 60 individuals aged 19-40, with 90% using right hand. Participants wrote every character ten times on two forms [8].

B. Data Pre-processing

Data preprocessing is a critical step in the pipeline of applying CNN to AHCD dataset. Effective preprocessing improves the model's learning efficiency and predictive performance by ensuring the input data is optimally conditioned. Here's an overview of the preprocessing steps commonly employed when preparing AHCD for training with a CNN.

1) Grayscale normalization: Since the AHCD consists of grayscale images, normalizing these images can be crucial for CNN performance. Normalization involves scaling the pixel values to a range [0,1] from the original range of [0,255].

2) Image resizing and verification: AHCD images are typically 32x32 pixels, which suits most CNN architectures designed for character recognition. in this step we verify and ensure that all images conform to this dimension.

3) Data augmentation: To enhance the model's ability to generalize from the training data to unseen data, applying data augmentation is a beneficial strategy. we applied various techniques such as rotation, width and height shifts, shearing, and zooming introduce a variety of transformed images derived from the original training set.

4) Label encoding: The labels for the AHCD are categorical Arabic letters. These labels need to be converted into a format suitable for classification, typically one-hot encoding.

TABLE II. COMMON ARABIC HANDWRITTEN CHARACTERS DATASETS

Dataset Name	Number of Characters	Image Format	Includes Digits?	Availability Online
AHCD	16,800 images	Grayscale, 32x32 pixels	No	Yes, freely available
AIA9K Dataset	9,000 images	Grayscale, 32x32 pixels	Yes	Restricted, limited access
MADBase Dataset	70,000 digits	Grayscale, 28x28 pixels	Yes (only digits)	Yes, freely available
HACDB Dataset	16,800 characters	Grayscale, 32x32 pixels	No	Yes, freely available
Hijja Dataset	47,434 characters	Grayscale, variable sizes	No	Yes, freely available

V. EXPERIMENTAL SETUP

A. Splitting the Dataset

It is standard practice to divide the dataset into training and testing subsets in order to evaluating the performance. In this study, we systematically partitioned AHCD into distinct subsets for training and testing purposes. Recognizing the importance of robust model training and the necessity for thorough performance evaluation, we allocated 80% of the dataset to training and 20% to testing. This configuration ensures that the model is exposed to a comprehensive variety of data during training, enhancing its ability to generalize, while also reserving a substantial portion of data for unbiased evaluation of its performance on unseen data.

B. Evaluation Metrics

In the assessment of models trained on AHCD, selecting the right evaluation metric is crucial to accurately measure the model's performance. For this study, we have chosen accuracy as the primary metric. This choice is predicated on accuracy's straightforward interpretability and its relevance in classification tasks where the distribution of classes is relatively balanced, as is the case with AHCD. Accuracy measures the proportion of total correct predictions made by the model compared to the total predictions made. In the context of handwriting recognition, where each character needs to be identified correctly from a set of 28 possibilities, accuracy provides a direct evaluation of how often the model correctly recognizes the characters. Furthermore, since the AHCD dataset is evenly distributed across different classes (i.e., each Arabic character), the use of accuracy helps to ensure that the performance measure is not biased by uneven class representation which can sometimes affect other metrics like precision and recall. Mathematically, accuracy is defined as the ratio of correctly predicted observations to the total observations. It can be expressed with the following equation:

$$\text{Accuracy } = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

Where TP denote the true positives, TN the true negatives, FP the false positives, and FN the false negative. In the specific case of a multi-class classification like the AHCD, where each instance is classified into one of many classes (and where true negatives aren't a direct consideration for each class), the formula simplifies to focusing on the true positives of each class against all classifications made. In addition to accuracy, it is essential to evaluate model performance using metrics that provide different perspectives on the behavior and efficacy of the model. Specifically, loss and error classification rate (ECR) are critical metrics for this purpose. These metrics, when used alongside accuracy, offer a more comprehensive view of a model's performance, especially in scenarios where accuracy alone might not fully capture the model's predictive capabilities. The loss function quantifies how far the predictions of the model are from the actual labels, providing a measure of the model's prediction errors. The choice of loss function can vary depending on the specific model and task, but for classification tasks involving neural networks, Cross-Entropy Loss is commonly used. This is particularly true for multi-class classification problems like those involving the AHCD dataset.

The Cross-Entropy Loss for a multi-class classification model is defined as:

$$L = -\sum_{i=1}^{N}\sum_{c=1}^{M} y_{ic} \log\left(p_{ic}\right) \tag{6}$$

Where, N is the number of samples in the dataset, M is the number of classes, y_{ic} is a binary indicator (0 or 1) if class label c is the correct classification for observation i, p_{ic} is the predicted probability of observation i being of class c. Error Classification Rate (ECR) is a straightforward metric that complements accuracy by focusing on the proportion of incorrect predictions. It is particularly useful for identifying how often the model fails, which can be critical for applications where failures have high costs. The ECR can be mathematically represented as:

$$ECR = 1 - \text{ Accuracy } = \frac{\text{Number of incorrect predictions}}{\text{Total number of predictions}} \tag{7}$$

VI. RESULTS AND DISCUSSION

In this section, we systematically analyze the performance variations of a convolutional neural network (CNN) on the Arabic Handwritten Characters Dataset (AHCD) across multiple configurations. The detailed results, spanning from Table III to Table VIII, allow us to discern the nuanced effects of altering batch sizes, filter sizes, the number of blocks, and the number of convolutional layers within each block on key performance metrics such as test accuracy and error classification rate (ECR).

Starting with Table III, we observe that the network configuration with two blocks and three convolutional layers per block achieves the highest test accuracy of 96.10% and the lowest ECR of 3.90%. This indicates that a moderate increase in network depth within a controlled number of blocks can significantly enhance the model's ability to generalize, particularly beneficial for the complex script characteristics of the Arabic language.

TABLE III. RESULTS WITH BATCH SIZE 32 AND FILTER SIZE 16

Batch Size = 32, Number of Blocks = 2, Filter Size = 16				
Number of Conv Layers	**Loss**	**Train Acc (%)**	**Test Acc (%)**	**ECR (%)**
1	0.2492	91.77	93.93	6.07
2	0.1622	94.61	95.46	4.54
3	0.1418	95.29	**96.10**	**3.90**
Batch Size = 32, Number of Blocks = 3, Filter Size = 16				
1	0.3227	89.79	94.04	5.96
2	0.1940	93.40	95.70	4.30
3	0.2299	92.79	94.67	5.33
Batch Size = 32, Number of Blocks = 4, Filter Size = 16				
1	0.6028	79.34	89.66	10.34
2	0.3242	90.23	94.08	5.92
3	0.3221	90.74	92.34	7.66

Furthermore, Table IV shows a similar trend where the configuration with three blocks and two layers achieves the highest test accuracy in the set at 97.07%. This configuration strikes an optimal balance, effectively capturing the intricate features of Arabic characters without overfitting, a common issue when excessively deep networks are employed.

In contrast, the results from Table V highlight a potential overfitting scenario as deeper networks (four blocks) yield poorer performance. The lowest test accuracy recorded is significantly less optimal when the network depth is increased without sufficient data to support the complexity, underscoring the delicate balance required between network architecture and available training data. Accordingly, the results from Table VI, where a larger batch size of 64 is utilized, suggest that while larger batches can stabilize training, they do not automatically translate into better generalization, especially when the network becomes too deep or too shallow. The optimal performance is again noted in configurations that balance depth with breadth adequately. Moreover, Table VII presents an intriguing outcome where the best test accuracy does not always correlate with the lowest ECR. This suggests that while certain configurations may optimize for one metric, they may compromise on another, highlighting the trade-offs that may need to be considered in practical applications of handwriting recognition technologies.

Lastly, the insights gathered from Table VIII reinforce the complexity of configuring CNNs for handwritten character recognition. The highest accuracy achieved (97.87%) with three blocks and two layers per block in a large batch and filter setting emphasizes that while increased resources (like larger filters) can enhance performance, the architectural setup must still avoid excessive complexity to prevent training difficulties and ensure robustness against overfitting.

Across all tables, from Table III to Table VIII, one consistent observation is that the configurations with two to three blocks generally provide better performance metrics—both in terms of higher test accuracy and lower ECR. This pattern suggests an optimal depth that effectively captures the complexities of Arabic script without causing the models to overfit or become computationally infeasible. In the context of the experiments that varied filter sizes and batch sizes, it's evident that increasing the filter size tends to improve performance, but only up to a certain point. For example, Table IV and Table VII show that a filter size of 32 offers a good compromise between capturing sufficient detail and maintaining model efficiency.

However, as seen in Table V and Table VIII, further increases in filter size sometimes result in only marginal gains or even slight reductions in performance. This outcome could be attributed to the model capturing excessive noise along with relevant features, especially when not complemented by a corresponding increase in other parameters like the number of layers or blocks. Moreover, the data points us towards an interesting trend where larger batch sizes do not always correlate with better performance, particularly when it comes to generalizing the model's capabilities to unseen data.

This is apparent from the results in Table VI compared to those in Table III. While larger batch sizes can stabilize the training process and lead to rapid convergence, they might also hinder the model's ability to navigate through narrower, potentially more optimal paths in the loss landscape during training. The highest accuracy achieved, as shown in Table VII, where configurations with moderate filter sizes and balanced block and layer setups reached an accuracy of 97.29%, underscores the necessity of tuning these parameters thoughtfully. It also highlights that while certain configurations perform exceptionally well under specific circumstances (such as with certain batch sizes or filter sizes), these configurations may not universally translate to other setups, emphasizing the bespoke nature of CNN architecture design for specific datasets like the AHCD.

A. Result Analysis

This research delves into the intricacies of recognizing Arabic handwritten characters (AHCR) using convolutional neural networks (CNNs), focusing particularly on assessing how various parameters—batch size, filter size, number of blocks, and the number of convolution layers within each block—affect the model's efficacy. The experiments conducted on the AHCD (Arabic Handwritten Characters Dataset) have illuminated some critical findings on optimizing CNN architectures for this task. Through a series of structured experiments, depicted in Tables III, IV, V, VI, VII, and VIII, alongside Fig. 10, 11, 12, 13, 14 and 15, we have comprehensively analyzed the performance impact of the aforementioned parameters. Key insights from this analysis have significantly advanced our understanding of AHCR using CNNs.

TABLE IV. BATCH SIZE 32 AND FILTER 32

Batch Size = 32, Number of Blocks = 2, Filter Size = 32				
Number Conv layers in Block	**Loss**	**Train Acc (%)**	**Test Acc (%)**	**ECR (%)**
1	0.1548	94.61	94.63	5.37
2	0.0853	97.00	96.39	3.61
3	0.1021	96.88	96.45	3.55
Batch Size = 32, Number of Blocks = 3, Filter Size = 32				
1	0.2159	93.24	95.31	4.69
2	0.1072	96.55	**97.07**	**2.93**
3	0.1371	96.11	96.74	3.26
Batch Size = 32, Number of Blocks = 4, Filter Size = 32				
1	0.2816	90.97	96.16	3.84
2	0.1490	95.45	96.11	3.89
3	3.3326	03.73	03.57	96.43

TABLE V. BATCH SIZE 32 AND FILTER 64

Batch Size = 32, Number of Blocks = 2, Filter Size = 64				
Number Conv layers in Block	**Loss**	**Train Acc (%)**	**Test Acc (%)**	**ECR (%)**
1	0.1177	96.43	95.58	4.42
2	0.0718	97.88	96.57	3.43
3	0.0806	97.62	97.00	3.00
Batch Size = 32, Number of Blocks = 3, Filter Size = 64				
1	0.1483	96.29	96.31	3.69
2	0.0867	97.81	**97.33**	**2.67**
3	0.1224	96.71	96.39	3.61
Batch Size = 32, Number of Blocks = 4, Filter Size = 64				
1	0.1789	94.59	96.17	3.83
2	3.3321	04.15	03.57	96.43
3	3.3327	03.85	03.57	96.43

TABLE VI. BATCH SIZE 64 AND FILTER 16

Batch Size = 64, Number of Blocks = 2, Filter Size = 16				
Number Conv layers in Block	**Loss**	**Train Acc (%)**	**Test Acc (%)**	**ECR (%)**
1	0.1833	93.61	94.36	5.64
2	0.1097	96.58	96.30	3.70
3	0.0930	97.07	96.53	3.47
Batch Size = 64, Number of Blocks = 3, Filter Size = 16				
1	0.2909	90.38	94.17	5.83
2	0.1604	94.89	**96.62**	**3.38**
3	0.1405	95.51	96.15	3.85
Batch Size = 64, Number of Blocks = 4, Filter Size = 16				
1	0.5266	82.00	89.56	10.44
2	0.2243	93.49	95.63	4.37
3	0.3324	3.29	3.57	96.43

1) Optimizing batch size and filter size: A critical observation from our study is that larger batch sizes generally enhance the model's learning stability and performance. Specifically, a batch size of 64 consistently showed improved accuracy across various configurations (Fig. 13, 14, and 15), suggesting that it allows the network to better generalize from the training data by averaging out noise across larger data samples per gradient update.

2) Influence of the number of blocks and layers: Our findings further indicate that an optimal number of blocks, typically around two to three (as shown in Table IX and Fig. 11 and 14), effectively balances the depth of the network with computational efficiency and prevents overfitting. Notably, increasing the number of blocks beyond this range tends to degrade performance, possibly due to the complexity and increased risk of overfitting, as observed in the lower performances in configurations with four blocks (Fig. 15).

3) The impact of filter size: The filter size also plays a pivotal role in the network's ability to capture relevant features from the handwriting images. Our results indicate that a filter size of 64 strikes a balance between capturing detailed features and avoiding the capture of irrelevant noise, particularly when paired with appropriate batch sizes and block numbers (Table IX). This size seems to provide sufficient granularity for

TABLE VII. BATCH SIZE 64 AND FILTER 32

Number Conv Layers in Block	Loss	Train Acc (%)	Test Acc (%)	ECR (%)
Batch Size = 64, Number of Blocks = 2, Filter Size = 32				
1	0.1192	95.88	96.14	3.86
2	0.0665	97.69	97.09	2.91
3	0.0556	98.19	96.71	3.29
Batch Size = 64, Number of Blocks = 3, Filter Size = 32				
1	0.1511	94.86	96.34	3.66
2	0.0720	97.87	97.00	3.00
3	0.0720	97.82	97.19	2.81
Batch Size = 64, Number of Blocks = 4, Filter Size = 32				
1	0.2298	92.26	95.83	4.17
2	0.1256	96.47	**97.29**	**2.71**
3	0.1044	97.01	97.07	2.93

TABLE VIII. BATCH SIZE 64 AND FILTER 64

Number Conv Layers in Block	Loss	Train Acc (%)	Test Acc (%)	ECR (%)
Batch Size = 64, Number of Blocks = 2, Filter Size = 64				
1	0.0839	97.27	96.31	3.69
2	0.0437	98.67	97.58	2.42
3	0.0419	98.72	97.25	2.75
Batch Size = 64, Number of Blocks = 3, Filter Size = 64				
1	0.0889	97.01	97.24	2.76
2	0.0703	97.90	97.52	2.48
3	0.0630	98.29	**97.87**	**2.13**
Batch Size = 64, Number of Blocks = 4, Filter Size = 64				
1	0.1332	95.71	97.34	2.66
2	0.0808	97.68	97.70	2.30
3	3.3324	03.64	03.57	96.43

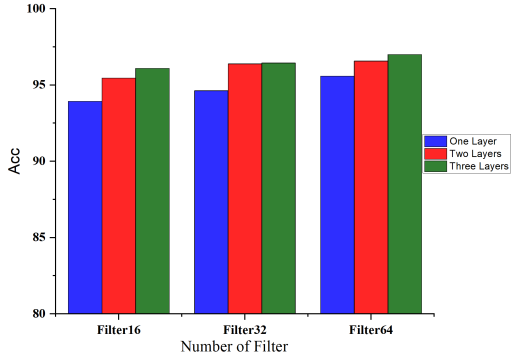

Fig. 10. Batch size 32 and two blocks.

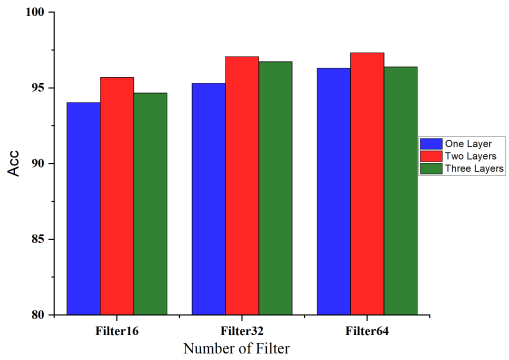

Fig. 11. Batch size 32 and three blocks.

effectively learning the intricacies of Arabic script without overwhelming the model with too much information.

The highest accuracy achieved, as illustrated in Table IX, was 97.87%, using a model configuration of three blocks and three layers per block, with a batch and filter size of 64. This configuration is recommended for achieving the best results in AHCR tasks with CNNs. It showcases the necessity of a balanced approach to CNN architecture design, especially in the context of Arabic handwriting recognition, where precision in capturing character nuances is critical. The detailed exploration of these factors provides a robust guideline for tuning CNNs for Arabic handwriting recognition. The insights gained enhance the performance of AHCR systems also contribute to the broader field of pattern recognition, offering a clear example of how careful, context-specific tuning of neural network parameters can lead to significant improvements in performance.

VII. CONCLUSION

This study extensively examined the influence of various parameters such as batch size, filter size, number of blocks, and the number of convolutional layers within each block on the performance of a convolutional neural network (CNN)

TABLE IX. THE FIVE BEST RESULT

Model Parameters				
Test Accuracy (%)	**Batch Size**	**Num of Blocks**	**Filter Size**	**Num of Layers**
97.87	64	3	64	3
97.70	64	4	64	2
97.58	64	2	64	2
97.52	64	3	64	2
97.34	64	4	64	1

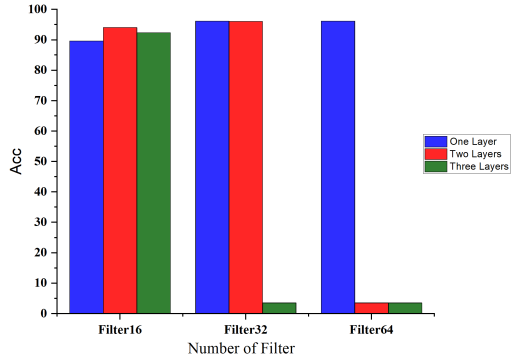

Fig. 12. Batch size 32 and four blocks.

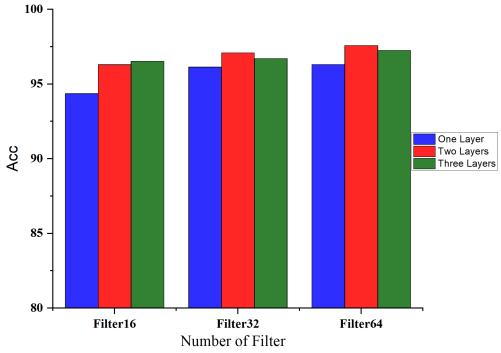

Fig. 13. Batch size 64 and two blocks.

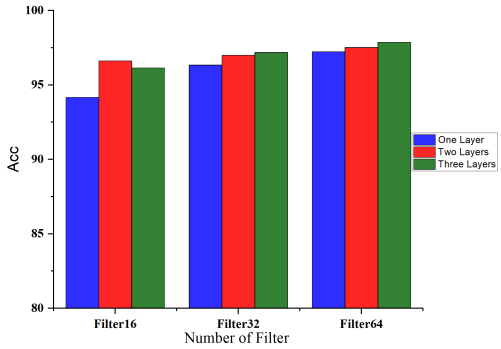

Fig. 14. Batch size 64 and three blocks.

model tailored for Arabic Handwritten Character Recognition (AHCR) using the AHCD dataset. The optimal model configuration achieved an impressive training accuracy of 98.29% and testing accuracy of 97.87%. This configuration utilized a Batch Size of 64, three blocks, three convolutional layers per block,

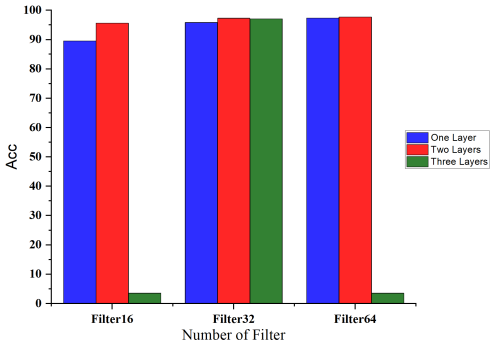

Fig. 15. Batch size 64 and four blocks.

and a filter size of 64, which proved particularly effective in capturing the complexities of Arabic script. Increasing the batch size from 32 to 64 consistently improved the model's performance across various configurations, highlighting the benefits of larger batch sizes in stabilizing training and enhancing performance in AHCR tasks. Models configured with three blocks and three convolutional layers each typically outperformed alternatives with fewer or more blocks/layers, suggesting an optimal balance between model depth and breadth, crucial for achieving good generalization without overfitting. The use of larger filter sizes (64) was advantageous over smaller sizes (16 and 32), enabling the model to better capture detailed features of the handwritten characters, critical for the accurate recognition of Arabic script.

A. Limitations and Future Work

The limitation of this study that the research was confined to the AHCD dataset. To establish the robustness and generalizability of the model, further testing is essential on a diverse array of AHCR datasets, including those featuring more variable handwriting styles or cursive text.

For future research directions, we will focus on integrating attention mechanisms, which could focus the model on relevant parts of the input data. Exploring ensemble models that combine the strengths of multiple network architectures could significantly enhance the capability to recognize challenging character classes or subtle nuances in handwriting. Additionally, extending the current model to handle continuous cursive Arabic handwriting and multi-line text recognition could substantially broaden the applicability of the research, making it more suitable for real-world scenarios, such as document analysis and automated transcription. Moreover, investigating the model's performance on mixed-language documents, particularly those combining Arabic with other scripts, could

further enhance its practicality and utility in multicultural and multilingual contexts.

Availability of Data and Materials

All data generated in this study are available in this paper.

Ethics declarations

Ethical Approval

This article does not contain any experiments with human participants or animals performed by any of the authors.

Consent to Participate

All authors approved the final manuscript.

Competing Interests

The authors declare no competing interests.

Author information

Authors and Affiliations

Wuhan university of technology, Wuhan, China

References

[1] Alrobah, N.A., Albahli, S.: Arabic handwritten recognition using deep learning: A survey. Arabian Journal for Science and Engineering 47, 9943–9963 (2022).

[2] Altwaijry, N., Al-Turaiki, I.: Arabic handwriting recognition system using convolutional neural network. Neural Comput. Appl. 33(7), 2249–2261 (2021). https://doi.org/10.1007/s00521-020-05070-8.

[3] Ahmad T. Al-Taani*, S.T.A.: Recognition of arabic handwritten characters using residual neural networks. Jordanian Journal of Computers and Information Technology (JJCIT) 07(02), 192–205(2021). https: //doi.org/10.5455/jjcit.71-1615204606.

[4] AlJarrah, M.N., Zyout, M.M., Duwairi, R.: Arabic handwritten characters recognition using convolutional neural network. In: 2021 12th International Conference on Information and Communication Systems (ICICS), pp. 182–188 (2021). https://doi.org/10.1109/ICICS52457.2021.9464596.

[5] Memon, J., Sami, M., Khan, R., Uddin, M.: Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). IEEE Access, 1–1 (2020). https://doi.org/10.1109/ACCESS.2020.3012542.

[6] Akhtar, P.: An online and offline character recognition using image processing methods-a survey mr. (2016).

[7] Ali Ahmed Ali, A., Mallaiah, S., Ahmed, H.: A survey on arabic handwritten character recognition. SN Computer Science 1 (2020). https: //doi.org/10.1007/s42979-020-00168-1.

[8] Elsawy, A., Loey, M., El-Bakry, H.: Arabic handwritten characters recognition using convolutional neural network. WSEAS TRANSACTIONS on COMPUTER RESEARCH 5, 11–19 (2017).

[9] Younis, K.: Arabic handwritten character recognition based on deep convolutional neural networks. Jordanian Journal of Computers and Information Technology (JJCIT) 3 (2018).

[10] Ahmed, R., Gogate, M., Tahir, A., Dashtipour, K., Al-Tamimi, B., Hawalah, A., El-Affendi, M.A., Hussain, A.: Novel deep convolutional neural network-based contextual recognition of arabic handwritten scripts. Entropy 23(3), 340 (2021).

[11] Shams, M., Elsonbaty, A., ElSawy, W., et al.: Arabic handwritten character recognition based on convolution neural networks and support vector machine. arXiv preprint arXiv:2009.13450 (2020).

[12] Maray, M., Al-onazi, B., Alzahrani, J., Alshahrani, S., Alotaibi, N., Alazwari, S., Othman, M., Hamza, M.: Sailfish optimizer with deep transfer learning-enabled arabic handwriting character recognition. Computers, Materials Continua 74, 5467–5482 (2023). https://doi.org/10. 32604/cmc.2023.033534.

[13] Balaha, H., Ali, H., Youssef, E., Elsayed, A., Samak, R., Abdelhaleem, M., Tolba, M., Shehata, M., Mahmoud, M., Abdelhameed, M., Mohammed, M.: Recognizing arabic handwritten characters using deep learning and genetic algorithms. Multimedia Tools and Applications 80 (2021). https://doi.org/10.1007/s11042-021-11185-4.

[14] Al-Ayyoub, M., Nuseir, A., Alsmearat, K., Jararweh, Y., Gupta, B.B.: Deep learning for arabic nlp: A survey. Journal of Computational Science 26 (2017). https://doi.org/10.1016/j.jocs.2017.11.011.

[15] Balaha, H., Ali, H., Badawy, M.: Automatic recognition of handwritten arabic characters: a comprehensive review. Neural Computing and Applications (2021). https://doi.org/10.1007/s00521-020-05137-6.

[16] Ahmed, R., Dashtipour, K., Gogate, M., Raza, A., Zhang, R., Huang, K., Hawalah, A., Adeel, A., Hussain, A.: Offline Arabic Handwriting Recognition Using Deep Machine Learning: A Review of Recent Advances, pp.457–468 (2020). https://doi.org/10.1007/978-3-030-39431-844.

[17] De Sousa, I.P.: Convolutional ensembles for arabic handwritten character and digit recognition. PeerJ Computer Science 4, 167 (2018).

[18] Hassan, N., Shboul, A., Alabed, A.: Arabic handwritten characters recnition using convolutional neural network, pp. 147–151 (2019). https://doi.org/10.1109/IACS.2019.8809122.

[19] Almansari, O.A., Hashim, N.N.W.N.: Recognition of isolated handwritten arabic characters. In: 2019 7th International Conference on Mechatronics Engineering (ICOM), pp. 1–5 (2019). https://doi.org/10.1109/ICOM47790.2019.8952035.

[20] Alyahya, H., Ismail, M.M.B., Al-Salman, A.: Deep ensemble neural networks for recognizing isolated arabic handwritten characters. ACCENTS Transactions on Image Processing and Computer Vision 6(21), 68 (2020).

[21] Kamal, M., Shaiara, F., Abdullah, C.M., Ahmed, S., Ahmed, T., Kabir, M.H.: Huruf: An application for arabic handwritten character recognition using deep learning. In: 2022 25th International Conference on Computer and Information Technology (ICCIT), pp. 1131–1136 (2022). https://doi.org/10.1109/ICCIT57492.2022.10054769.

[22] Wagaa, N., Kallel, H., Mellouli, N.: Improved arabic alphabet characters classification using convolutional neural networks (cnn). Computational Intelligence and Neuroscience 2022 (2022).

[23] Elkhayati, M., Elkettani, Y.: Uncnn: A new directed cnn model for isolated arabic handwritten characters recognition. ARABIAN JOURNAL FOR SCIENCE AND ENGINEERING (2022). https://doi.org/10.1007/s13369-022-06652-5.

[24] Khudeyer, R.S., Almoosawi, N.M.: Combination of machine learning algorithms and resnet50 for arabic handwritten classification. Informatica 46(9) (2023).

[25] Bin Durayhim, A., Al-Ajlan, A., Al-Turaiki, I., Altwaijry, N.: Towards accurate children's arabic handwriting recognition via deep learning. Applied Sciences 13(3), 1692 (2023).

[26] Alheraki, M., Al-Matham, R., Al-Khalifa, H.: Handwritten arabic character recognition for children writing using convolutional neural network and stroke identification (2022). https://doi.org/10.48550/arXiv.2211.02119.

[27] Ullah, Z., Jamjoom, M.: An intelligent approach for arabic handwritten letter recognition using convolutional neural network. PeerJ Computer Science 8, 995 (2022). https://doi.org/10.7717/peerj-cs.995.

[28] Bouchriha, L., Zrigui, A., Mansouri, S., Berchech, S., Omrani, S.: Arabic Handwritten Character Recognition Based on Convolution Neural Net- works, pp. 286–293 (2022). https://doi.org/10.1007/978-3-031-16210-7.

[29] Obied, Z., Solyman, A., Ullah, A., Fat'hAlalim, A., Alsayed, A.: Bert multilingual and capsule network for arabic sentiment analysis. In: 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), pp. 1–6 (2021). IEEE.

[30] Solyman, A., Zappatore, M., Zhenyu, W., Mahmoud, Z., Alfatemi, A., Ibrahim, A.O., Gabralla, L.A.: Optimizing the impact of data augmentation for low-resource grammatical error correction. Journal of King Saud University-Computer and Information Sciences 35(6), 101572 (2023).