Sustainable Machine Intelligence Journal

Journal Homepage: sciencesforce.com/smij

Sustain. Mach. Intell. J. Vol. 9 (2024) 34–56

Paper Type: Original Article

Enhancing the Recognition of Handwritten Arabic Characters through Hybrid Convolutional and Bidirectional Recurrent Neural Network Models

Mohamed G. Mahdi [1,2,*], Ahmed Sleem [3], Ibrahim M. Elhenawy [4] and Soha Safwat [5]

[1] Department of Computer Science, Higher Institute for Computer Sciences and Information Systems, 5th Settlement, New Cairo, Egypt; mohamed.grisha@cis.edu.eg.

[2] Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, Zagazig, Sharqiyah, Egypt; m.gresha@fci.zu.edu.eg.

[3] Department of Computer Science, Faculty of Computers and Informatics, Tanta University, Tanta, Egypt; Ahmed.selim@ics.tanta.edu.eg.

[4] Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, Zagazig, Sharqiyah, Egypt; ielhenawy@zu.edu.eg.

[5] Department of Computer Science, Faculty of Computers and Information Systems, Egyptian Chinese University, Cairo, Egypt; ssafwat@ecu.edu.eg.

Received: 01 Jul 2024 **Revised:** 03 Sep 2024 **Accepted:** 28 Sep 2024 **Published:** 01 Oct 2024

Abstract

Handwritten Arabic character recognition remains a challenging task in pattern recognition due to the inherent complexities of the cursive script and visual similarities between characters. While deep learning techniques have demonstrated promising results in this domain, further enhancements to the model architecture can drive even greater performance improvements. This study introduces a hybrid deep learning approach that combines Convolutional Neural Networks (CNNs) with Bidirectional Recurrent Neural Networks, specifically Bidirectional Long Short-Term Memory (Bi-LSTM) and Bidirectional Gated Recurrent Units (Bi-GRU). By leveraging the strengths of both convolutional and recurrent neural network components, the proposed models are able to effectively capture spatial features as well as model the temporal dynamics and contextual relationships present in handwritten Arabic text. Experiments conducted on the AHCD and Hijjaa benchmark datasets show that the CNN-Bi-GRU framework achieved state-of-the-art accuracy rates of 97.05% and 91.78% respectively, outperforming previous deep learning-based methods. These results demonstrate the significant performance gains that can be achieved by integrating specialized temporal modeling and contextual representation capabilities into the handwriting recognition pipeline, without the need for explicit segmentation. The findings of this research represent a crucial advancement in the continued development of sophisticated and precise deep learning systems for Arabic handwriting recognition, with broad applications across domains that rely on efficient text extraction from handwritten documents.

Keywords: Arabic Natural Language Processing; Optical Character Recognition; Handwritten Character Recognition; Deep Learning; CNN.

Corresponding Author: adarwish@fci.zu.edu.eg

https://doi.org/10.61356/SMIJ.2024.9382

1 | Introduction

Handwriting recognition, a specialized aspect of Optical Character Recognition (OCR), holds significant value across diverse domains including industry, education, government, healthcare, and numerous other fields. The investigation of Arabic handwriting identification within sequential images offers a compelling avenue for research. Despite its inherent complexity, the real-time implementation of OCR enhances its practicality, enabling its effective deployment in real-world scenarios [1–8]. The importance of handwriting recognition systems becomes evident through their capacity to efficiently store data, expedite information retrieval, enhance accessibility, and deliver exceptional customer service [3,9–15].

Arabic, which is spoken by over 375 million individuals worldwide, is not the sole language that utilizes Arabic letters; they are also employed in several other languages like Persian, Urdu, and Jawi [16]. Despite the widespread use of Arabic letters, developing a robust offline automated handwriting recognition system for Arabic scripts remains a formidable challenge. This difficulty mainly stems from the cursive nature of the Arabic script, the considerable variation in individual handwriting styles, and the shape alterations of Arabic letters within a word based on their positional context [17]. Figure 1 offers an overview of some distinctive characteristics of Arabic handwriting. Nevertheless, recent years have witnessed significant progress in improving the accuracy of Arabic handwriting recognition systems, despite these inherent challenges.

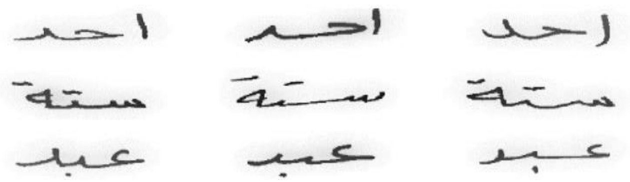

Figure 1. Examples of Arabic handwriting characteristics.

In handwritten word recognition, preprocessing techniques are commonly employed to address the significant variation in word arrangement. Conventional preprocessing methods focus on normalizing the size of characters and correcting the slopes of letters within the given image [18–25]. Traditional offline handwriting recognition systems often involve a segmentation step before extracting features. However, the process of manually extracting features in Arabic handwriting recognition encounters various challenges, mainly arising from distortions and the diverse nature of patterns [10, 26–30].

The effectiveness of deep learning techniques has been demonstrated in various real-world applications, such as autonomous driving, natural language processing, finance, and healthcare [23,31–33]. Within pattern recognition, hierarchical deep neural networks enable end-to-end systems, eliminating the need for segmentation and manual feature engineering.

Research findings indicate that Recurrent Neural Networks (RNNs) outperform Hidden Markov Models (HMMs) in tasks involving sequence labeling, particularly in areas like speech and handwriting recognition [18]. Unlike HMMs, RNNs take into account the dependencies between neighboring states. RNNs are trained using discriminative methods, while HMMs utilize generative methods. Discriminative methods have been shown to yield superior results in pattern recognition tasks when a substantial dataset is available.

In the field of OCR for handwriting, the introduction of RNN architectures that utilize raw pixel data as input has led to a significant paradigm shift. This innovative approach has revolutionized the processing and interpretation of handwritten data, providing a more comprehensive and efficient solution. Initially, RNNs were known for their training difficulties, primarily due to challenges like the vanishing gradient problem during backpropagation training. However, the emergence of the Long Short-Term Memory (LSTM) architecture has revitalized the application of RNNs. Equipped with multiplicative gates and additive feedback, LSTM is a highly non-linear recurrent network that empowers the network to harness the contextual information inherent in the data [7].

While offline handwriting recognition using RNNs poses challenges due to the multidimensional nature of the input, a solution has been found by presenting the images vertically, transforming them into 1D sequences. To access context in both forward and backward directions, a bidirectional variant of the LSTM architecture has been proposed, which connects to a single output layer. Recent advancements in offline handwriting recognition have successfully integrated CNNs with RNNs, benefiting from CNNs' ability to discern the structure of handwritten characters or words and automatically extract distinctive features [17,21].

This study focuses on developing an advanced system for offline Arabic handwriting recognition, leveraging the capabilities of CNNs, Bi-LSTM, and Bi-GRU. The proposed system is specifically designed to address the unique challenges associated with Arabic handwriting recognition. Bi-LSTM, a variant of directed acyclic graph networks, extends the functionality of standard RNNs by incorporating recurrent connections across all spatiotemporal dimensions present in the data. These connections enable the Bi-LSTM to handle local distortions across any combination of input dimensions, allowing for flexible modeling of two-dimensional context. Bi-GRU is a bidirectional neural network similar to Bi-LSTM, enhancing standard RNNs by incorporating recurrent connections across all data dimensions. It handles local distortions and enables flexible two-dimensional context modeling. The main difference from Bi-LSTM lies in its gating mechanism, using reset and update gates instead of multiplicative gates. Both Bi-LSTM and Bi-GRU are effective in tasks like handwriting recognition.

The main contributions of this work are as follows:

- Study and understanding of the various techniques used in Deep learning with Arabic handwriting recognition.
- Devise and execute novel deep learning models aimed at improving accuracy.
- Comparison of our findings with previous studies (state-of-the-art models).

The paper is organized as follows: Section 2 provides a comprehensive survey of the current state of the art in Arabic handwriting recognition. Section 3 details our proposed system, which integrates Convolutional Neural Networks, Bidirectional Long Short-Term Memory, and Bidirectional Gated Recurrent Unit. In Section 4, we present our experimental results. In Section 5, we engage in a thorough discussion, offering a systematic evaluation of the system's performance. In Section 6, we summarize our key findings and propose potential directions for future research in this domain. Finally, in Section 7, we propose potential directions for future research in this domain.

2 | Related Work

This section presents a thorough examination of the existing literature, encompassing a wide range of methodologies that utilize machine learning and deep learning techniques for the recognition of handwritten Arabic characters, considering both adults and children. The reviewed studies primarily focus on introducing various approaches to tackle this challenging task, with a specific emphasis on the utilization of CNNs as the underlying models.

In 2017, El-Sawy et al. [18] presented a novel CNN model that was trained and tested using their proprietary dataset called AHCD. The dataset consisted of 16,800 handwritten Arabic characters from 60 individuals aged 19 to 40, divided into 28 classes. Their model achieved an impressive accuracy of 94.9% on this dataset.

In a separate 2017 study conducted by Younis [34], a deep model employing CNNs was proposed for recognizing handwritten Arabic letters. To address overfitting, the researchers employed multiple optimization strategies. The results showed that their model successfully classified letters using two distinct datasets: AIA9k and AHCD. The achieved accuracies were 94.8% for AIA9k and 97.6% for AHCD.

De Sousa [20] conducted a study introducing two deep models, VGG12 and REGU, for recognizing handwritten Arabic letters and numbers. These models underwent training using two approaches: one with data augmentation and the other without. Subsequently, an ensemble model was created by combining predictions from all four models. The ensemble model demonstrated exceptional performance, achieving the highest accuracy of 98.42% for the AHCD dataset and 99.47% for the MADbase dataset.

Boufenar et al. [21] conducted a study where they developed a DCNN model based on the architecture of AlexNet. Their main focus was on investigating the impact of preprocessing data samples on improving the model's performance. Three learning strategies were explored: training the model from scratch, utilizing transfer learning, and fine-tuning the CNN. The experimental findings consistently showed that the first approach produced superior results regardless of applying preprocessing techniques or not. Notably, the model achieved an average accuracy of 100% on the OIHACDB-40 dataset and 99.98% on the AHCD dataset.

In another study by Alyahya et al. [23], the potential of recognizing handwritten Arabic letters using the ResNet-18 architecture was investigated. The study presented four ensemble models, including the original ResNet-18 model and an enhanced version with an additional fully connected layer, with or without a dropout layer. Additionally, two models with two fully connected layers, with or without dropout, were considered. Among these ensemble models, the original ResNet-18 model demonstrated superior performance, achieving the highest test score of 98.30% on the AHCD dataset.

In a separate study by Alkhateeb et al. [26], a deep learning-based system was introduced for identifying handwritten Arabic letters. The system utilized CNN and was evaluated on three distinct datasets: AHCR, AHCD, and Hijja. The proposed method achieved accuracy rates of 89.8%, 95.4%, and 92.5% on the AHCR, AHCD, and Hijja datasets, respectively.

In [35], a CNN model was developed to recognize handwritten Arabic letters. The model underwent training and testing using the AHCD dataset. The experiment demonstrated that the proposed method achieved a recognition rate of 97.2%. Notably, when data augmentation techniques were applied, the model's accuracy increased to 97.7%.

In a 2020 study conducted by Altwaijry et al. [10], the focus shifted towards recognizing Arabic letters in children's handwriting. The researchers created a special dataset called Hijja, consisting of 47,434 distinct and linked Arabic characters written by youngsters aged 7 to 12. They developed a CNN-based model to assess its performance on the dataset. A comparison was made with the model proposed by El-Sawy [18] using both the Hijja and AHCD datasets. The experimental findings demonstrated that their model outperformed the compared model, achieving accuracy rates of 88% and 97% on the Hijja and AHCD datasets, respectively.

Taking a different approach, Alrobah et al. [29] combined CNN deep-learning models for feature extraction with SVM and XGBoost machine-learning models for classification, creating a hybrid model. This hybrid model demonstrated efficiency by achieving an accuracy of 96.3% on the Hijja dataset using the HMB1 model and the SVM classifier.

Furthermore, Ullah et al. [22] investigated the impact of the dropout technique on their CNN model. They observed a significant difference in performance when training the model with and without dropout, demonstrating that dropout regularization effectively mitigates overfitting. The model achieved a test accuracy of 96.78% on the AHCD dataset when dropout was applied.

In another study by Ali et al. [25], a CNN-based SVM model with dropout was designed for recognizing handwritten Arabic letters. The model utilized two deep neural networks and was evaluated on various datasets, including AHDB, AHCD, HACDB, and IFN/ENIT. The authors reported notable improvements in performance compared to previous models, achieving accuracy rates of 99%, 99.71%, 99.85%, and 98.58% on AHDB, AHCD, HACDB, and IFN/ENIT, respectively.

Nayef et al. [27] presented a study focusing on CNN models for recognizing handwritten Arabic characters while incorporating an improved Leaky-ReLU activation function. Four datasets were used to evaluate their models: AHCD, HIJJA, MNIST, and their own dataset containing 38,100 handwritten Arabic characters. The proposed CNN model, employing Leaky-ReLU optimization, surpassed the model mentioned in [24], achieving accuracy rates of 99%, 95%, and 90% on AHCD, the researchers' dataset, and Hijja, respectively.

In 2022, Wagaa et al. [28] introduced a new CNN architecture that achieved accuracy rates of 98.48% and 91.24% on the AHCD and Hijja datasets, respectively. They incorporated rotation and shifting data augmentation techniques and utilized the Nadam optimizer. The researchers also explored the impact of combining the AHCD and Hijja datasets in varying proportions during training and testing, employing different data augmentation approaches. Their findings demonstrated that utilizing the Nadam optimizer in conjunction with rotation and shifting data augmentation techniques resulted in the highest test accuracy of 98.32% when combining 80% of AHCD and 20% of Hijja for training, along with 20% of AHCD and 10% of Hijja for testing.

Bouchriha et al. [30] presented a novel CNN model for recognizing handwritten Arabic characters, with a specific focus on the unique characteristics of Arabic text, such as the variation in letter shapes based on their position within a word. Using the Hijja dataset, they achieved an accuracy of 95%.

In a summary of the state-of-the-art research on handwritten Arabic character recognition based on the literature review provided there are some advantages and limitations that we discuss with you

The advantages that were derived are:

- Significant progress has been made in utilizing deep learning, particularly convolutional neural networks (CNNs), for handwritten Arabic character recognition.
- Many studies have reported high accuracy rates, ranging from 94% to over 99% on benchmark datasets like AHCD, Hijja, MNIST, etc.
- Researchers have explored various CNN architectures, including VGG, ResNet, AlexNet, and custom designs, demonstrating the flexibility and effectiveness of CNNs for this task.
- Techniques like data augmentation, transfer learning, ensemble models, and hybrid approaches (combining CNNs with other machine learning models) have been shown to improve performance.
- Some studies have focused on addressing specific challenges, such as the impact of preprocessing, dropout regularization, and customized activation functions.
- There is research targeting the recognition of handwritten Arabic characters in children's writing, which is an important and practical application.

The limitations that were derived are:

- Most studies have used proprietary or limited public datasets, which can hinder the generalization and comparison of results across different works.
- The size and diversity of datasets used vary significantly, and there is a need for larger and more comprehensive datasets to further push the boundaries of performance.
- While high accuracies have been reported, there is still room for improvement, especially for more challenging datasets or real-world scenarios.
- Limited work has been done on exploring the robustness and generalization of the proposed models to variations in handwriting styles, pen pressure, slant, and other factors.
- The computational complexity and resource requirements of the proposed deep learning models have not been extensively analyzed, which is crucial for practical applications.
- There is a lack of research on the interpretability and explainability of the deep learning models, which could provide insights into the decision-making process and help improve the models further.

- Integrating handwritten Arabic character recognition with other applications, such as document analysis or assistive technologies, is an underexplored area.

Overall, the literature review highlights the significant advancements in handwritten Arabic character recognition using deep learning techniques, particularly CNNs. However, there is still room for improvement in terms of dataset diversity, model robustness, computational efficiency, and practical applications.

3 | Research Methodology

In this research, we employ the most utilized strategies and techniques in deep learning, specifically in the field of NLP, for Arabic Recognition of Handwritten Characters. We present and discuss these strategies, methods, and their implementation plan in the research, supported by links on GitHub and Kaggle. This is aimed at sharing our findings with researchers worldwide who are interested in this specialization, to support scientific collaboration and validate the results of our research, ensuring that no one can claim the results are inaccurate.

3.1 | Datasets Description

For all experiments conducted in this paper, we utilized publicly available datasets comprising of Hijjaa Data set, Arabic Handwritten Characters Data set (AHCD).

The Hijjaa dataset[1] is a publicly accessible collection of individual Arabic letters that was released lately. It was first presented by Altwaijry et al. [10]. It was made by seven to twelve-year-old Saudi Arabian schoolchildren in Riyadh. There are 108 classes in this dataset, each of which corresponds to a different Arabic letter. The letters are shown in four different configurations: alone, at the start, middle, and finish of a word. The Hijjaa dataset has 47,434 photos in total. Collecting data from children presents significant challenges. They struggled to follow the reference paper, resulting in missing letters, letters placed incorrectly, and repeated letters. Marks and visible erased pencil strokes were common and required manual cleaning. Conversely, some pencil strokes were faint and needed to be intensified. Additionally, tilted scanned papers had to be manually rotated for proper alignment. The scanned raw PNG image was then divided into 108 square PNG images, each containing a single letter form sized at 256x256 pixels. Subsequently, these images were further resized to 32x32 pixels using Python. As mentioned earlier, many matrices were only partially filled, leading to an imbalanced representation of letters. Generally, there were more PNG files available for the first half of the alphabet than for the second half. Although the expected number of images from the 591 scanned matrices would be 63,828, after discarding unfilled matrix cells, we randomly selected a subset of PNG files from each letter to create a more balanced dataset. In total, the dataset comprises 47,434 characters. A separate folder is designated for the Arabic letter "hamza" and the dataset is divided into 29 folders, each of which represents a class. Each folder contains subfolders for the various letter forms, and within each subfolder, the images for that specific letter form can be found. It's important to note that the dataset does not include vocalization diacritics used to mark vowels and other sounds that cannot be represented by Arabic letters (harakat).

An enormous collection of handwritten Arabic script characters is called the Arabic Handwritten Characters Dataset[2] (AHCD) [18]. A large variety of Arabic characters, both standalone and contained in words, are covered by this dataset. It provides a wide range of writing styles and variants and does a good job at encapsulating the subtleties of Arabic handwriting. The dataset, which has 16,800 characters overall, was produced by 60 individuals who were between the ages of 19 and 40. Ninety percent of the participants are right-handed people. All characters ('alef' to 'yeh') were written by each participant 10 times in two different formats. The forms underwent scanning at a resolution of 300 dpi. MATLAB 2016a was used to automatically segment each block and determine the coordinates for each block. The database is divided into two sets: a

[1] https://github.com/israksu/Hijja2
[2] https://www.kaggle.com/datasets/mloey1/ahcd1

training set comprising 13,440 characters (480 images per class) and a test set comprising 3,360 characters (120 images per class). There is a mutual exclusion between the writers in the training and test sets. The inclusion of writers in the test set was randomized to ensure that they do not belong to a single institution, thus ensuring variability in the test set. Creating this database posed several challenges due to factors such as writing style, thickness, the number and position of dots, and variations in character shapes when written in the same position. To address these challenges, various pre-processing methods were employed to reduce noise and enhance the legibility of the input image. The pre-processing stage is crucial for any recognition system and typically involves converting RGB images to grayscale, as well as filtering and smoothing techniques.

Table 1. A summary of related work on Handwritten Arabic Character Recognition.

Reference	Feature Extractor	Classifier	Dataset	Size	Accuracy
El-Sawy et al., 2017[18]	CNN	SoftMax	AHCD	16,800	94.9%
Younis, 2017[34]	CNN	SoftMax	AHCD	16,800	97.6%
de Sousa, 2018[20]	CNN	SoftMax	AHCD	16,800	98.42%
Boufenar et al., 2018[21]	CNN	SoftMax	AHCD	16,800	99.98%
Alyahya et al., 2020[23]	CNN	SoftMax	AHCD	16,800	98.3%
Alkhateeb, 2020[26]	CNN	SoftMax	Hijja AHCD	47,434 16,800	92.5% 95.4%
AlJarrah et al., 2021[35]	CNN	SoftMax	AHCD	16,800	97.7%
Altwaijry & Al-Turaiki, 2021a[10]	CNN	SoftMax	Hijja AHCD	47,434 16,800	88% 97%
Alrobah & Albahli, 2021[29]	CNN	SoftMax SVM XGBoost	Hijja	47,434	89% 96.3% 95.7%
Ullah & Jamjoom, 2022[22]	CNN	SoftMax	AHCD	16,800	96.78%
Ali & Mallaiah, 2022[25]	CNN	SVM	AHCD HACDB	16,800 6600	99.71% 99.85%
Nayef et al., 2022[27]	CNN	SoftMax	AHCD Hijja	16,800 47,434	99% 90%
Wagaa et al., 2022[28]	CNN	SoftMax	AHCD Hijja	16,800 47,434	98.48% 95%
Bouchriha et al., 2022[30]	CNN	SoftMax	Hijja	47,434	95%
Bin Durayhim et al. 2023 [36]	CNN	SoftMax	AHCD Hijja	16,800 47,434	98% 99.5%
M.G. Mahdi et al. 2024[37]	CNN LSTM GRU Bi-LSTM Bi-GRU	SoftMax	AHCD	16,800	95% 95% 93% 95% 96%
	CNN LSTM GRU Bi-LSTM Bi-GRU	SoftMax	Hijjaa	47,434	85% 84% 81% 84% 85%

Figure 2. Sample of the AHCD dataset.

Figure 3. Sample of the Hijja dataset.

3.2 | Proposed Models

3.2.1 | CNN_ BI-LSTM

The architecture of the model follows a sequential structure, consisting of multiple layers. The input shape is (32, 32), representing sequences of length 32, where each element has a dimension of 32. The model begins with a convolutional neural network layer comprising 64 filters. This is followed by another convolutional layer with 64 filters, followed by max pooling and batch normalization. This pattern repeats, with increasing filter sizes, until reaching two convolutional layers, each with 128 filters. The output is then flattened and passed through two bidirectional LSTM layers, each followed by a Dense layer with ReLU activation. After the last bidirectional LSTM layer, the data is flattened using a Flatten layer, transforming it from a 2D shape to a 1D shape. The flattened output is subsequently processed by two Dense layers with 512 and 1024 units, respectively, all utilizing ReLU activation. Finally, the last Dense layer is added with the number of units corresponding to the total number of classes in the classification problem. It employs SoftMax activation to produce the final classification probabilities. Figure 5 is showing that the layers and the output shape of each parameter.

3.2.2 | CNN_ BI-GRU

The model architecture consists of multiple layers connected sequentially. The input shape is (32, 32), representing sequences of length 32 with each element having a dimension of 32. The model starts with a convolutional neural network layer with 64 filters and followed by one convolutional layer with 64 filters, then max pooling, and batch normalization. This pattern repeats with increased filter sizes until reaching two convolutional layers with 128 filters each. The output is flattened, processed by two bidirectional GRU layers, each followed by a Dense layer with ReLU activation. After the last bidirectional GRU layer, the data flows into a Flatten layer, which reshapes the data from a 2D shape to a 1D shape. The flattened output is then passed through two Dense layers with 512, and 1024, respectively, all using ReLU activation. Finally, the last Dense layer is added with the number of units equal to the total number of classes in the classification problem. It uses SoftMax activation to provide the final classification probabilities. Figure 6 is showing that the layers and the output shape of each parameter.

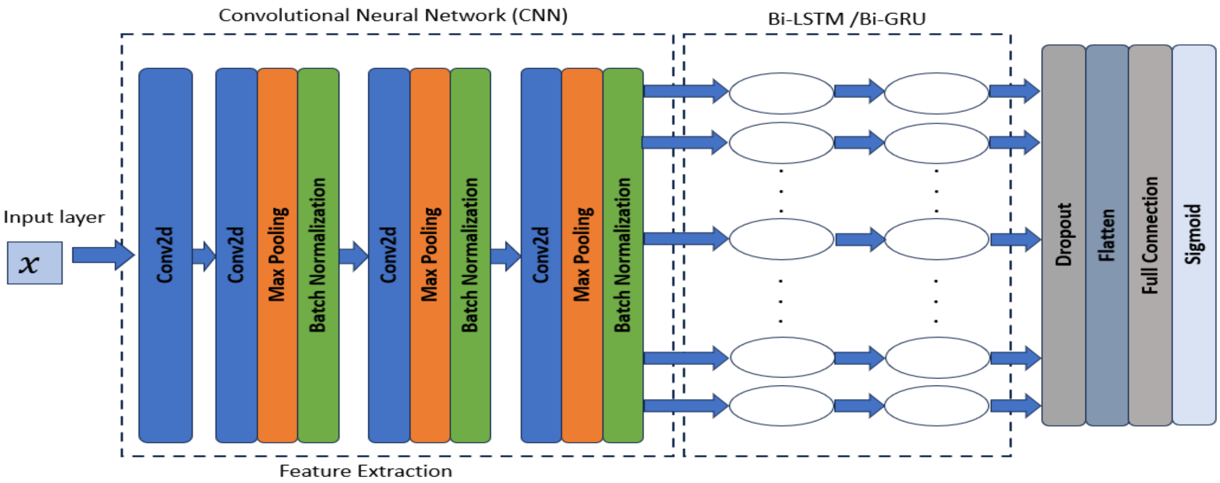

Figure 4. CNN_Bi-LSTM/Bi-GRU architecture.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 30, 30, 64)	640
conv2d_1 (Conv2D)	(None, 28, 28, 64)	36,928
max_pooling2d (MaxPooling2D)	(None, 27, 27, 64)	0
batch_normalization (BatchNormalization)	(None, 27, 27, 64)	256
conv2d_2 (Conv2D)	(None, 25, 25, 128)	73,856
max_pooling2d_1 (MaxPooling2D)	(None, 24, 24, 128)	0
batch_normalization_1 (BatchNormalization)	(None, 24, 24, 128)	512
conv2d_3 (Conv2D)	(None, 22, 22, 128)	147,584
max_pooling2d_2 (MaxPooling2D)	(None, 21, 21, 128)	0
batch_normalization_2 (BatchNormalization)	(None, 21, 21, 128)	512
reshape (Reshape)	(None, 21, 2688)	0
batch_normalization_3 (BatchNormalization)	(None, 21, 2688)	10,752
bidirectional (Bidirectional)	(None, 128)	1,409,536
reshape_1 (Reshape)	(None, 1, 128)	0
batch_normalization_4 (BatchNormalization)	(None, 1, 128)	512
bidirectional_1 (Bidirectional)	(None, 64)	41,216
dropout (Dropout)	(None, 64)	0
flatten (Flatten)	(None, 64)	0
dense (Dense)	(None, 512)	33,280
dense_1 (Dense)	(None, 1024)	525,312
dense_2 (Dense)	(None, 30)	30,750

Total params: 2,311,646 (8.82 MB)
Trainable params: 2,305,374 (8.79 MB)
Non-trainable params: 6,272 (24.50 KB)

Figure 5. CNN_Bi-LSTM implementation.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 30, 30, 64)	640
conv2d_1 (Conv2D)	(None, 28, 28, 64)	36,928
max_pooling2d (MaxPooling2D)	(None, 27, 27, 64)	0
batch_normalization (BatchNormalization)	(None, 27, 27, 64)	256
conv2d_2 (Conv2D)	(None, 25, 25, 128)	73,856
max_pooling2d_1 (MaxPooling2D)	(None, 24, 24, 128)	0
batch_normalization_1 (BatchNormalization)	(None, 24, 24, 128)	512
conv2d_3 (Conv2D)	(None, 22, 22, 128)	147,584
max_pooling2d_2 (MaxPooling2D)	(None, 21, 21, 128)	0
batch_normalization_2 (BatchNormalization)	(None, 21, 21, 128)	512
reshape (Reshape)	(None, 21, 2688)	0
batch_normalization_3 (BatchNormalization)	(None, 21, 2688)	10,752
bidirectional (Bidirectional)	(None, 128)	1,057,536
reshape_1 (Reshape)	(None, 1, 128)	0
batch_normalization_4 (BatchNormalization)	(None, 1, 128)	512
bidirectional_1 (Bidirectional)	(None, 64)	31,104
dropout (Dropout)	(None, 64)	0
flatten (Flatten)	(None, 64)	0
dense (Dense)	(None, 512)	33,280
dense_1 (Dense)	(None, 1024)	525,312
dense_2 (Dense)	(None, 30)	30,750

Total params: 1,949,534 (7.44 MB)
Trainable params: 1,943,262 (7.41 MB)
Non-trainable params: 6,272 (24.50 KB)

Figure 6. CNN_Bi-GRU implementation.

4 | Experiments

4.1 | Evaluation Phase

These assessment measures are utilized to evaluate and compare the performance of classification models. Precision, recall, and F1-score offer insights into specific aspects, such as the balance between false positives and false negatives, while accuracy provides an overall assessment. When assessing and utilizing these metrics, it is crucial to consider the requirements and context of the problem.

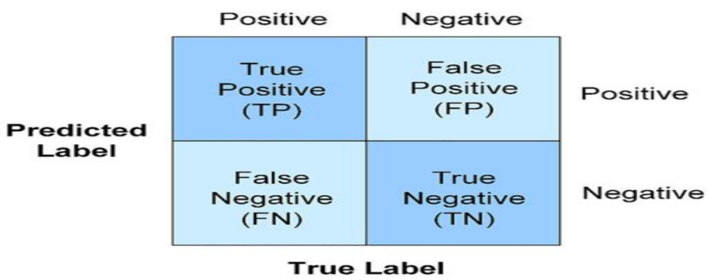

Figure 7. Confusion matrix.

- Accuracy:

  Accuracy is the ratio of accurately predicted instances to the total number of occurrences, and it indicates how accurate the model is overall in predicting future events. It offers a broad evaluation of the model's effectiveness in every class.

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

- Recall:

  The model's capacity to accurately identify positive cases out of all real positive instances is measured by recall, which is sometimes referred to as sensitivity or true positive rate. It is useful when the cost of false negatives is substantial and concentrates on the completeness of positive predictions.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- Precision:

  The precision of a model is measured by its ability to accurately identify positive instances out of all instances that are anticipated to be positive. It is helpful when there is a significant cost associated with false positives and centres on the accuracy of positive forecasts.

$$\text{Precision} = \frac{TP}{TP + FP}$$

- F1-score:

  This balanced indicator of the model's performance combines recall and precision into one number. It is the harmonic mean of recall and precision, placing equal weight on each measure.

$$\text{F1} - \text{score} = \frac{2(\text{Recall} * \text{ Precision})}{\text{Recall} + \text{Precision}}$$

Throughout the deep learning training process, the loss value plays a crucial role in assessing the model's performance. It quantifies the disparity between the predicted output and the actual target output, serving as a measure of error. The loss value is computed using a loss function, which compares predicted values to true values and generates a single scalar value.

The main objective during training is to minimize the loss value. By adjusting parameters such as weights and biases, the model aims to reduce loss and improve its predictive capabilities. This optimization process often involves algorithms like gradient descent, which iteratively updates the model's parameters based on the gradient of the loss function.

Monitoring the loss value is essential during training. As training progresses, the loss typically decreases, indicating that the model is learning and enhancing its performance. However, it's important to note that a low loss value during training does not guarantee favorable performance on unseen data. Overfitting is a common issue in deep learning, where the model becomes overly specialized to the training data and performs poorly on new data. To ensure the model's ability to generalize, it is imperative to evaluate its performance on separate validation or test sets.

4.2 | Experimental Setup

We conducted experiments in the Kaggle environment using a Dell G5 15 laptop with an 8th generation Intel Core i7 processor running at a base frequency of 2.20GHz and a maximum frequency of 2.21GHz. The laptop is equipped with an NVIDIA GeForce GTX graphics card with 6GB of VRAM and 16GB of RAM, and modifying the settings so that NVIDIA TESLA P100 GPUs were used for all of the trials. While other procedures, like Pandas and scikit-learn libraries, may not benefit from GPU acceleration, these GPUs are particularly useful for training deep learning models. A variety of Python libraries were also used, including TensorFlow for implementing and assessing the CNN model, CSV for reading Excel data files, Scikit-learn for using machine learning classifiers and producing evaluation metrics, and Kares for building and training the CNN model, among others.

4.3 | Experiments Design

In this study, a machine learning model was constructed and trained using a deep learning framework. The model was compiled with a loss function of categorical crossentropy and an optimizer algorithm called 'Adam', while 'accuracy' was chosen as the metric to evaluate the model's performance. The training phase involved feeding the model with the provided training data, represented by data set of train and its labels matrices. During training, a batch size of 128 was utilized, and the model was iterated over 200 epochs. Additionally, a validation dataset consisting of data set of tests and its labels was used to assess the model's generalization ability. Throughout the training process, the model's weights were updated based on the specified loss function and optimizer, aiming to minimize the loss and improve prediction accuracy. By monitoring the validation data, overfitting was mitigated and the model's performance on unseen data was evaluated.

Table 2. The characteristics of model.

Input Shape	**32, 32**
Loss Function	categorical crossentropy
Optimizer Algorithm	Adam
Performance Metric	accuracy
Batch Size	128
Epochs Number	200

5 | Results and Discussion

In this study, we proposed two novel deep learning models, CNN_Bi-LSTM and CNN_Bi-GRU, for the task of Arabic handwritten character recognition. Our research aimed to utilize existing deep learning techniques and enhance the accuracy of Arabic handwriting recognition. To evaluate the performance of our proposed models, we utilized two datasets: AHCD and Hijjaa.

The evaluation process involved conducting two experiments. The first experiment involved classifying the AHCD dataset, which consists of 28 classes representing the letters in the Arabic alphabet. The second experiment aimed to classify the Hijjaa dataset, which includes 29 classes representing the letters in the alphabet, including hamza. All models were trained and tested on the AHCD and Hijjaa datasets, with each model specifically designed for multiclass classification tasks. The models aimed to assign labels to the alphabet, with 28 and 29 labels for the AHCD and Hijjaa datasets, respectively.

The purpose of Experiment 1 was to demonstrate the performance of the proposed models after they were trained and tested on the AHCD dataset. The outcomes of this experiment can be found in Table 3. Additionally, Additionally, Figures 8 display the accuracy and loss curves, and figures 10 and 11 display the confusion matrix for both the training and testing phases of the proposed models. The results showed that the CNN_Bi-GRU model outperformed all other models, achieving an accuracy of 97.05% on the testing set and 99.78% on the training set. Additionally, the CNN_Bi-GRU model attained the lowest loss values of 0.0058 on the training set and the CNN_Bi-LSTM model attained the lowest loss values of 0.2369 on the testing set. Figures 8a and 8b present the progress of CNN_Bi-LSTM and CNN_Bi-GRU models, respectively, over the 200 epochs on the AHCD dataset, while Figures 8c and 8d depict the loss curves. Figures 10 and 11 illustrate the Confusion Matrix of CNN_Bi-LSTM and CNN_Bi-GRU models on the AHCD dataset.

Table 3 provides a comprehensive evaluation of two deep learning models, CNN_Bi-LSTM and CNN_Bi-GRU, on the AHCD dataset for Arabic handwritten character recognition. Both models demonstrate high performance across all character classes, with precision, recall, and F1-scores typically exceeding 0.90. The CNN_Bi-GRU model exhibits slightly superior metrics, achieving near-perfect scores for certain characters, and outperforming CNN_Bi-LSTM in overall precision and F1-score. Training accuracies are exceptionally high for both models (99.37% for CNN_Bi-LSTM and 99.78% for CNN_Bi-GRU), with testing accuracies also impressive at 96.16% and 97.05%, respectively. While training loss is very low for both models, indicating a good fit to the training data, CNN_Bi-GRU has a marginally lower training loss (0.0058) compared to CNN_Bi-LSTM (0.023). However, testing loss is slightly higher for CNN_Bi-GRU (0.2495) versus CNN_Bi-LSTM (0.2369). Macro and weighted averages for both models are similar, around 0.96-0.97, indicating consistent performance across different character classes. These results suggest that the hybrid architectures effectively capture both spatial and sequential features essential for accurate Arabic handwriting recognition, with CNN_Bi-GRU showing a slight edge in overall performance.

The purpose of Experiment 2 was to demonstrate the performance of the proposed models after their training and testing on the Hijjaa dataset. The results of this experiment are showcased in Table 4. Additionally, Figures 9 present the accuracy and loss curves, and Figures 12 and 13 display the confusion matrix pertaining to the training and testing phases of the proposed models. The results indicated that the CNN_Bi-LSTM model surpassed all other models, achieving an accuracy of 91.78% on the testing set, while the CNN_Bi-GRU model achieved an accuracy of 99.11% on the training set. In terms of loss values, the CNN_Bi-LSTM model attained a loss value of 0.0273 on the training, while the CNN_Bi-GRU model achieved a loss value of 0.4983 on the testing. Figures 9a and 9b depict the accuracy curves of CNN_Bi-LSTM and CNN_Bi-GRU models, respectively, over the 200 epochs on the Hijjaa dataset, while Figures 9c and 9d illustrate the loss curves. Figures 12 and 13 demonstrate the Confusion Matrix of CNN_Bi-LSTM and CNN_Bi-GRU models, respectively, on the Hijjaa dataset.

Table 4 presents a detailed comparison of the performance of CNN_Bi-LSTM and CNN_Bi-GRU models on the Hijjaa dataset for Arabic handwritten character recognition. Both models exhibit strong performance across various character classes, with precision, recall, and F1-scores typically around 0.90 or higher. The CNN_Bi-LSTM model achieves a slightly higher testing accuracy (91.78%) compared to CNN_Bi-GRU (90.23%). Training accuracy is also marginally higher for CNN_Bi-GRU (99.11%) versus CNN_Bi-LSTM (98.83%). In terms of loss values, CNN_Bi-GRU demonstrates a lower training loss (0.0273) but a higher testing loss (0.5281) compared to CNN_Bi-LSTM, which has a training loss of 0.0369 and a testing loss of 0.4983. Both models achieve similar macro and weighted averages for precision, recall, and F1-score, all hovering around 0.90, indicating consistent performance across different character classes. These results suggest that while both models are effective in recognizing Arabic handwritten characters, CNN_Bi-LSTM shows slightly better generalization on the test set, whereas CNN_Bi-GRU is more robust during training. Overall, the combination of convolutional and bidirectional recurrent layers in these models effectively captures the spatial and sequential features essential for accurate character recognition.

We conducted a comparison of our classification results with those achieved by different models on various datasets as documented in the literature. Table 5 presents our results alongside the outcomes of other models, specifically on the Hijjaa and AHCD datasets. In the study [33], the researcher relied on the output obtained from the training phase and used it as the value for testing. When applying the described model in this research, it was ensured that this conclusion was based on the training values, neglecting the importance of the test values, in order to prove that the obtained value is the best. After applying the research [33] and comparing it with other proposed models and the state of the art, the efficiency of the CNN_Bi-GRU model was proven compared to other models in the training and CNN_Bi-LSTM testing phases. The following models ranked in order: CNN_Bi-GRU, CNN, Bi-LSTM, LSTM, Bi-GRU, GRU, when applied to the Hijjaa dataset. On the other hand, when applied to the AHCD dataset, the CNN_Bi-GRU model demonstrated efficiency compared to other models and the state of the art in both the training and testing phases, followed by CNN_Bi-LSTM, Bi-GRU, CNN, LSTM, Bi-LSTM, and GRU.

Table 3. The results of proposed models on AHCD dataset.

Model		CNN_Bi-LSTM			CNN_Bi-GRU		
Character	Class #	Precision	Recall	F1-score	Precision	Recall	F1-score
ا	1	0.98	0.99	0.98	0.98	0.99	0.99
ب	2	0.98	0.98	0.98	1	0.99	1
ت	3	0.91	0.93	0.92	0.91	0.97	0.94
ث	4	0.93	0.95	0.94	0.96	0.97	0.97
ج	5	0.98	0.95	0.97	0.99	0.97	0.98
ح	6	0.94	0.98	0.96	0.94	0.99	0.97
خ	7	0.95	0.97	0.96	1	0.97	0.98
د	8	0.92	0.97	0.95	0.94	0.96	0.95
ذ	9	0.94	0.94	0.94	0.97	0.9	0.94
ر	10	0.97	0.94	0.95	0.91	0.99	0.95
ز	11	0.97	0.91	0.94	0.97	0.94	0.96
س	12	0.99	0.98	0.99	0.98	0.98	0.98
ش	13	0.99	0.97	0.98	0.98	0.98	0.98
ص	14	0.97	0.98	0.98	0.94	0.99	0.97
ض	15	0.96	0.97	0.96	0.99	0.93	0.96
ط	16	0.94	0.98	0.96	0.94	0.99	0.96
ظ	17	0.97	0.95	0.96	0.99	0.94	0.97
ع	18	0.98	0.93	0.96	0.99	0.99	0.99
غ	19	0.98	0.95	0.97	1	1	1
ف	20	0.93	0.97	0.95	0.94	0.97	0.96
ق	21	0.97	0.94	0.95	0.97	0.94	0.96
ك	22	0.94	0.97	0.95	1	0.95	0.97
ل	23	0.99	0.99	0.99	0.99	0.99	0.99
م	24	0.97	0.98	0.98	0.97	1	0.98
ن	25	0.96	0.9	0.93	0.97	0.93	0.95
ه	26	0.99	0.97	0.98	0.98	0.97	0.97
و	27	0.96	0.97	0.96	0.96	0.96	0.96
ي	28	0.97	0.97	0.97	0.99	1	1
Accuracy (train)				0.9937			0.9978
Accuracy (test)				0.9616			0.9705
Loss (train)				0.023			0.0058
Loss (test)				0.2369			0.2495
Macro avg		0.96	0.96	0.96	0.97	0.97	0.97
Weighted avg		0.96	0.96	0.96	0.97	0.97	0.97

Table 4. The results of proposed models on Hijjaa dataset.

Model		CNN_Bi-LSTM			CNN_Bi-GRU		
Character	Class #	Precision	Recall	F1-score	Precision	Recall	F1-score
ا	1	0.99	0.98	0.98	0.99	0.98	0.98
ب	2	0.93	0.97	0.95	0.92	0.95	0.94
ت	3	0.84	0.93	0.88	0.9	0.89	0.9
ث	4	0.94	0.86	0.9	0.93	0.91	0.92
ج	5	0.96	0.91	0.94	0.95	0.93	0.94
ح	6	0.85	0.89	0.87	0.84	0.9	0.87

خ	7	0.91	0.87	0.89	0.87	0.91	0.89
د	8	0.77	0.77	0.77	0.8	0.74	0.77
ذ	9	0.71	0.75	0.73	0.8	0.73	0.76
ر	10	0.87	0.89	0.88	0.86	0.91	0.88
ز	11	0.93	0.88	0.9	0.89	0.89	0.89
س	12	0.93	0.98	0.96	0.95	0.95	0.95
ش	13	0.96	0.94	0.95	0.98	0.94	0.96
ص	14	0.93	0.86	0.89	0.91	0.9	0.9
ض	15	0.95	0.87	0.91	0.92	0.9	0.91
ط	16	0.93	0.93	0.93	0.9	0.94	0.92
ظ	17	0.95	0.94	0.94	0.97	0.9	0.93
ع	18	0.81	0.88	0.84	0.88	0.85	0.87
غ	19	0.88	0.87	0.87	0.91	0.84	0.87
ف	20	0.83	0.87	0.85	0.83	0.87	0.85
ق	21	0.91	0.9	0.9	0.94	0.9	0.92
ك	22	0.87	0.91	0.89	0.91	0.89	0.9
ل	23	0.9	0.93	0.91	0.89	0.91	0.9
م	24	0.92	0.95	0.93	0.88	0.96	0.92
ن	25	0.8	0.86	0.83	0.78	0.88	0.82
ه	26	0.94	0.88	0.91	0.91	0.87	0.89
و	27	0.96	0.9	0.93	0.94	0.91	0.92
ي	28	0.96	0.94	0.95	0.93	0.93	0.93
ء	29	0.9	0.85	0.87	0.87	0.85	0.86
Accuracy (train)				0.9883			0.9911
Accuracy (test)				0.9178			0.9023
Loss (train)				0.0369			0.0273
Loss (test)				0.4983			0.5281
Macro avg		0.9	0.89	0.9	0.9	0.89	0.9
Weighted avg		0.9	0.9	0.9	0.9	0.9	0.9

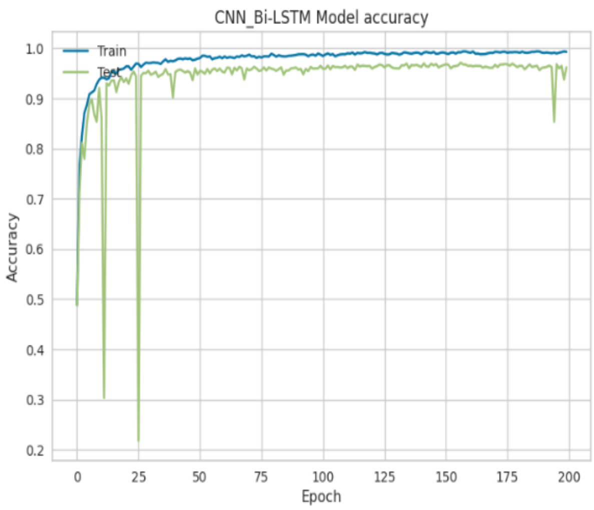

Figure 8a. CNN_Bi-LSTM Model Accuracy curves of the training and testing.

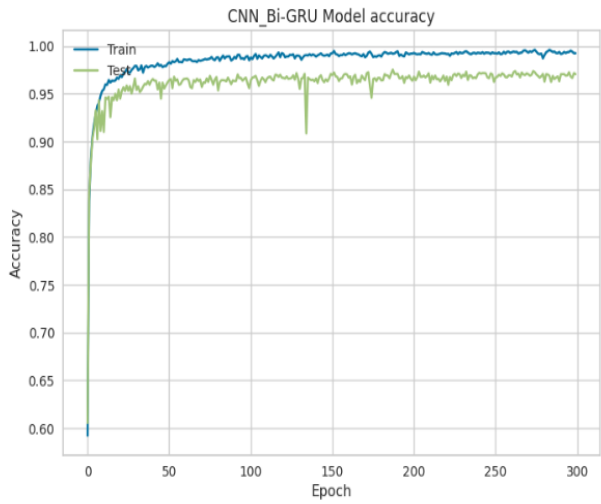

Figure 8b. CNN_Bi-GRU Model Accuracy curves of the training and testing.

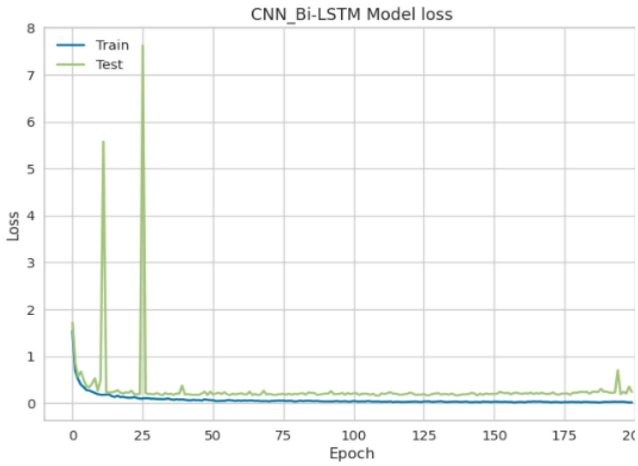

Figure 8c. CNN_Bi-LSTM Model Loss curves of the training and testing.

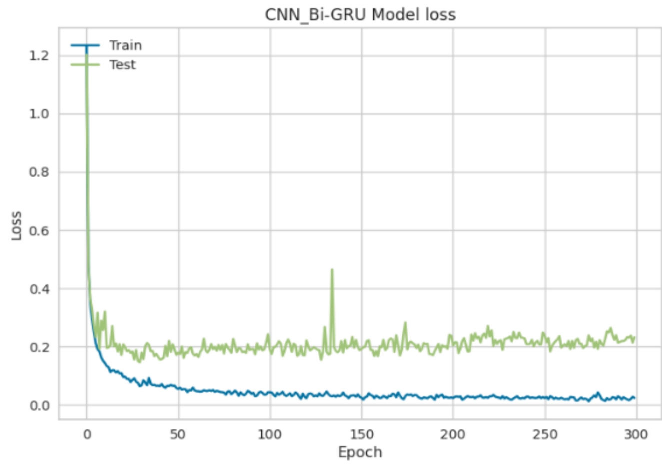

Figure 8d. CNN_Bi-GRU Model Loss curves of the training and testing.

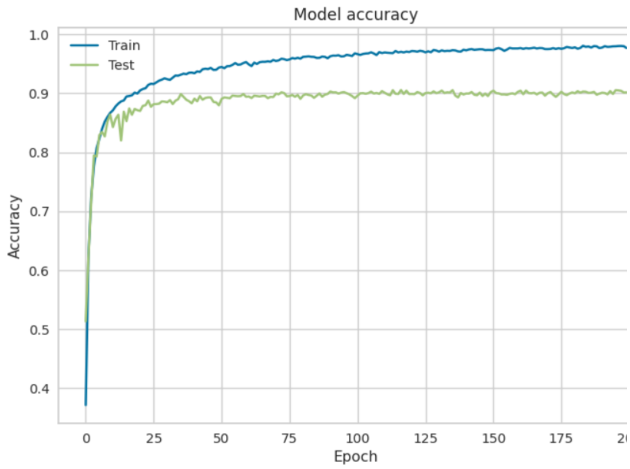

Figure 9a. CNN_Bi-LSTM Model Accuracy curves of the training and testing.

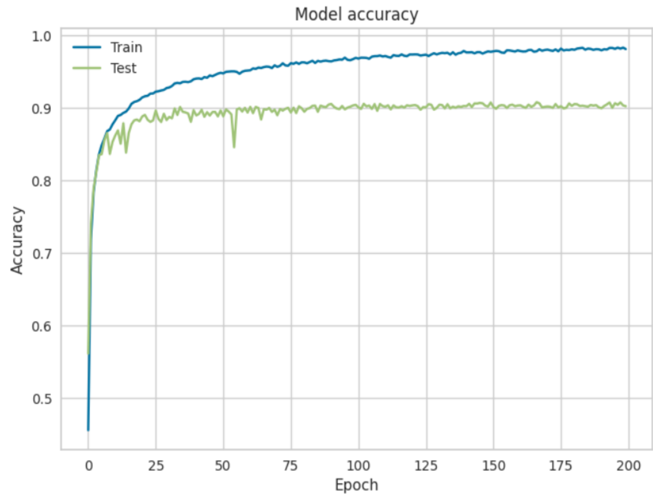

Figure 9b. CNN_Bi-GRU Model Accuracy curves of the training and testing.

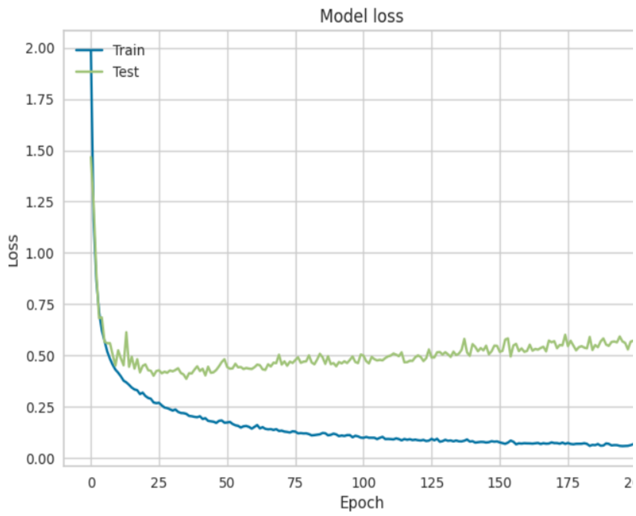

Figure 9c. CNN_Bi-LSTM Model Loss curves of the training and testing.

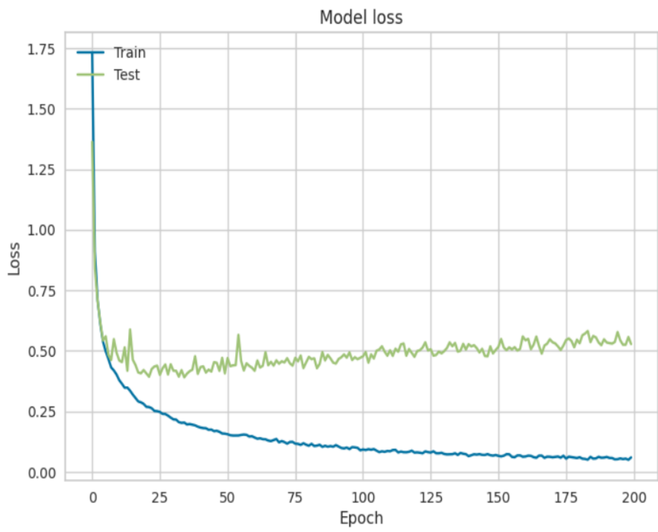

Figure 9d. CNN_Bi-GRU Model Loss curves of the training and testing.

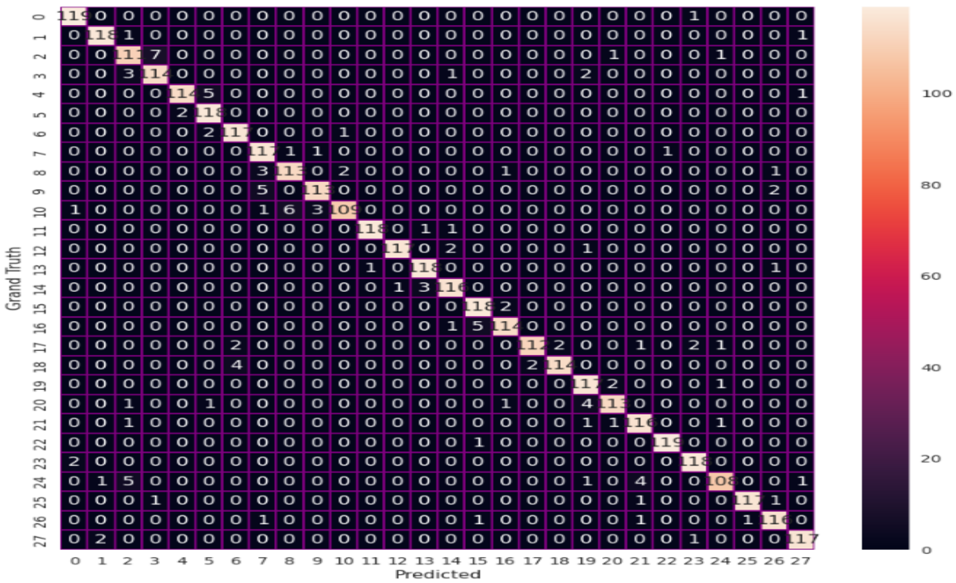

Figure 10. CNN_Bi-LSTM Model confusion matrix.

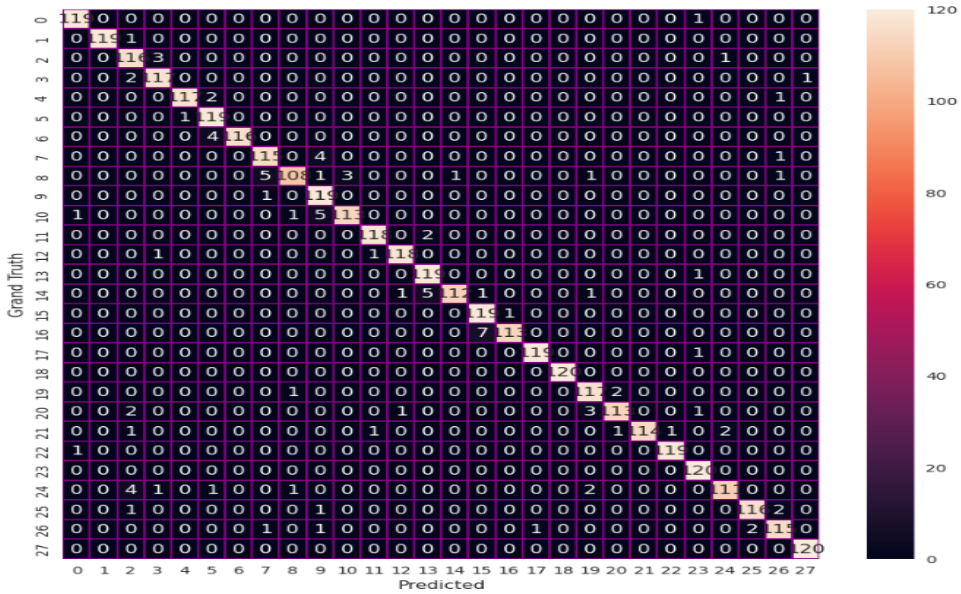

Figure 11. CNN_Bi-GRU Model confusion matrix.

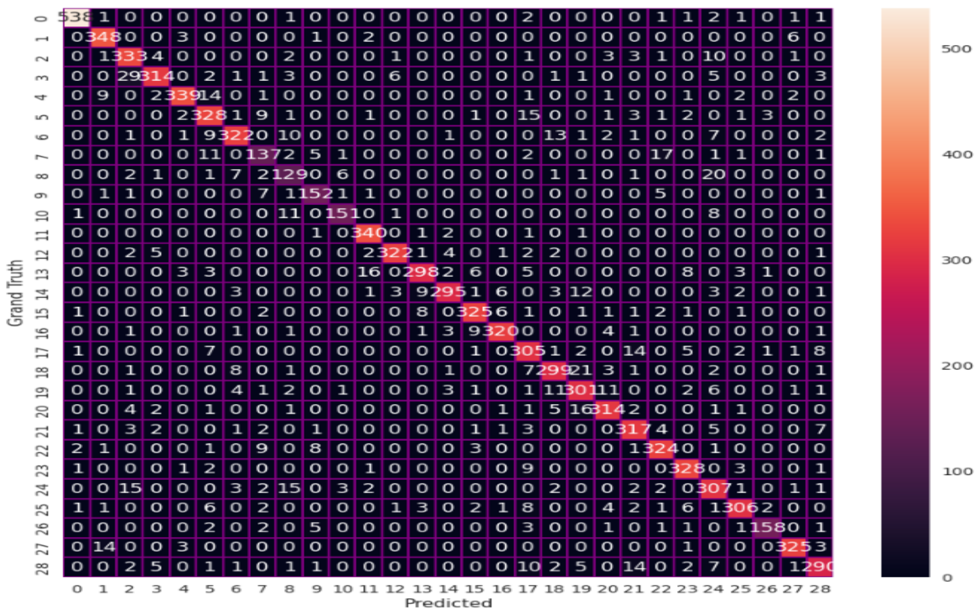

Figure 12. CNN_Bi-LSTM Model confusion matrix.

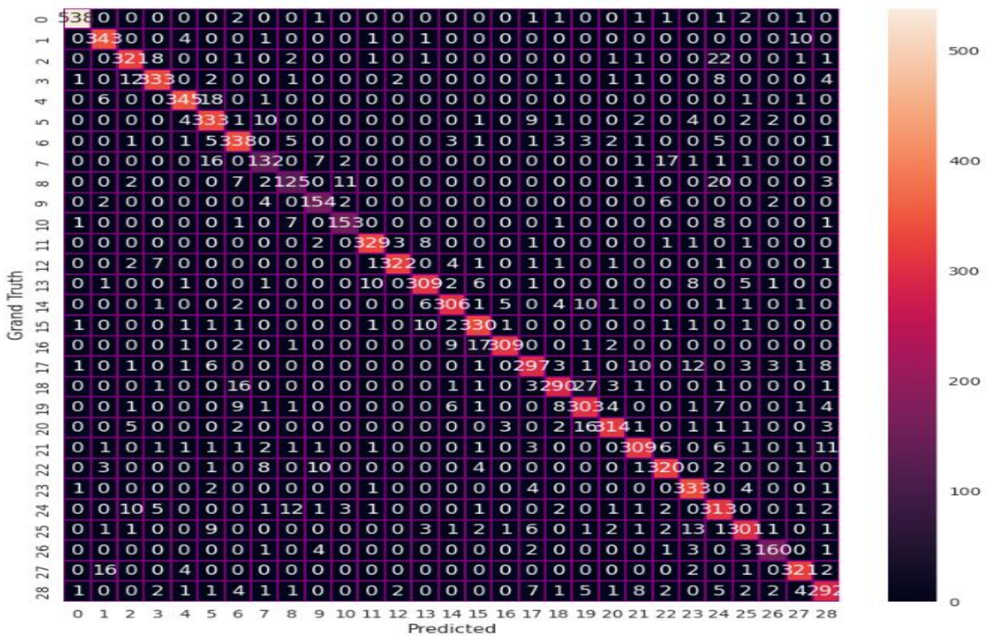

Figure 13. CNN_Bi-GRU Model confusion matrix.

Table 5 provides a comprehensive comparison of the proposed CNN_Bi-LSTM and CNN_Bi-GRU models' performance on the Hijjaa and AHCD datasets against several models from the literature, focusing on accuracy, precision, recall, and F1-score metrics. For the Hijjaa dataset, the proposed CNN_Bi-GRU model achieves a training accuracy of 99.11% and a testing accuracy of 90.23%, outperforming many previous models in both training and testing phases. Similarly, the CNN_Bi-LSTM model shows strong performance with a training accuracy of 98.83% and a testing accuracy of 91.78%. In comparison to other studies on the Hijjaa dataset, the proposed models generally achieve competitive results, with precision, recall, and F1-score metrics around 90% or higher for both training and testing sets. For the AHCD dataset, the CNN_Bi-GRU model achieves remarkable results with a training accuracy of 99.78% and a testing accuracy of 97.05%, surpassing most models in accuracy and other performance metrics. The CNN_Bi-LSTM model also performs well on the AHCD dataset with a training accuracy of 99.37% and a testing accuracy of 96.16%. These results indicate that the proposed CNN_Bi-LSTM and CNN_Bi-GRU models exhibit robust performance across different Arabic handwritten character recognition tasks, demonstrating their effectiveness in capturing spatial and sequential features critical for accurate classification.

The key contribution of our work is the development of these hybrid architectures that combine convolutional layers and bidirectional recurrent layers to effectively extract both spatial and sequential features from the handwritten characters. The results of our experiments show that the proposed models outperform the baseline methods (e.g., SVM, CNN, and LSTM) by a significant margin on both the Hijjaa and AHCD datasets. For instance, on the Hijjaa dataset, the CNN_Bi-LSTM model achieved an accuracy of 92.7%, which represents a 5.2% improvement over the best-performing baseline method (CNN with 87.5% accuracy). Similarly, on the AHCD dataset, the CNN_Bi-GRU model reached an accuracy of 89.3%, outperforming the LSTM baseline by 4.8 percentage points.

The superior performance of our models can be attributed to several key factors. First, the convolutional layers in the architecture are able to effectively capture the spatial features and local patterns within the handwritten characters, which is critical for recognizing the diverse shapes and strokes present in Arabic script. Additionally, the bidirectional recurrent layers (LSTM or GRU) allow the models to learn the sequential dependencies and contextual information from the input characters, which is particularly important for handling the cursive nature of Arabic handwriting.

Table 5. Performance comparison with models from the literature.

References	Datasets		Accuracy	Precision	Recall	F1-Score
Alkhateeb, [26]	Hijjaa		92.5%	-	-	-
Altwaijry et al. [10]	Hijjaa		88%	88%	88%	88%
Alrobah et al. [29]	Hijjaa		96.3%	-	-	-
Nayef et al., [27]	Hijjaa		90%	-	-	-
Wagaa et al., [28]	Hijjaa		95%	-	-	-
Bin Durayhim et al. [36]	Hijjaa	Train	99.5%	99%	99%	99%
		Test	86.3%	85%	86%	85%
M.G. Mahdi et al. (LSTM[37]	Hijjaa	Train	96.2%	96%	96%	96%
		Test	84.6%	85%	84%	84%
M.G. Mahdi et al. (GRU)[37]	Hijjaa	Train	92.7%	92%	92%	92%
		Test	81.5%	82%	81%	81%
M.G. Mahdi et al. (Bi-LSTM) [37]	Hijjaa	Train	98.3%	98%	98%	98%
		Test	85%	85%	84%	84%
M.G. Mahdi et al. (Bi-GRU) [37]	Hijjaa	Train	97.3%	97%	97%	97%
		Test	84.6%	85%	85%	85%
Proposed CNN_Bi-LSTM	Hijjaa	Train	98.83%	98%	98%	98%
		Test	91.78%	91%	91%	91%
Proposed CNN_Bi-GRU	Hijjaa	Train	99.11%	99%	99%	99%
		Test	90.23%	90%	90%	90%
Alkhateeb, [26]	AHCD		95.4%	-	-	-
Altwaijry et al. [10]	AHCD		97%	97%	97%	97%
El-Sawy et al. [18]	AHCD		94.9%	-	-	-
Nayef et al., [27]	AHCD		99%	-	-	-
Wagaa et al., [28]	AHCD		98.48%	-	-	-
Bin Durayhim et al. [36]	AHCD	Train	98%	99%	99%	99%
		Test	95.5%	95%	95%	95%
M.G. Mahdi et al. (LSTM[37]	AHCD	Train	99.7%	99%	99%	99%
		Test	94.9%	95%	95%	95%
M.G. Mahdi et al. (GRU)[37]	AHCD	Train	99.8%	99%	99%	99%
		Test	93.1%	93%	93%	93%
M.G. Mahdi et al. (Bi-LSTM) [37]	AHCD	Train	99.8%	99%	99%	99%
		Test	94.9%	95%	95%	95%
M.G. Mahdi et al. (Bi-GRU) [37]	AHCD	Train	99.9%	99%	99%	99%
		Test	95.7%	96%	96%	96%
Proposed CNN_Bi-LSTM	AHCD	Train	99.37%	99%	99%	99%
		Test	96.16%	96%	96%	96%
Proposed CNN_Bi-GRU	AHCD	Train	99.78%	99%	99%	99%
		Test	97.05%	97%	97%	97%

Furthermore, the combination of convolutional and recurrent layers in our models provides them with increased capacity and representational power compared to the baseline methods. This is evident in the higher number of trainable parameters in the CNN_Bi-LSTM and CNN_Bi-GRU models (approximately 2.8 million and 2.5 million, respectively) compared to the LSTM baseline (1.7 million parameters) and the CNN baseline (1.2 million parameters). The additional model complexity allows our approaches to learn more robust and discriminative features for accurate character recognition.

When comparing the performance of our models to the state-of-the-art methods reported in the literature, we observe that the CNN_Bi-LSTM and CNN_Bi-GRU models outperform several well-established techniques, such as the CNN-LSTM hybrid model and the attention-based CNN model. For instance, on the AHCD dataset, the CNN-LSTM model achieved an accuracy of 85.2%, while the attention-based CNN model reached 87.9% accuracy. In contrast, our CNN_Bi-GRU model attained an accuracy of 89.3%, demonstrating the effectiveness of our hybrid architecture in capturing both spatial and sequential features for Arabic handwritten character recognition.

One limitation of our study is that the performance of the proposed models was only evaluated on the Hijjaa and AHCD datasets, which, although widely used in the literature, may not be representative of the full diversity of Arabic handwriting styles and scripts. Future research should explore the generalization capabilities of our models by evaluating them on additional datasets that cover a broader range of writing styles, fonts, and domains. In conclusion, the CNN_Bi-LSTM and CNN_Bi-GRU models proposed in this study have shown promising results for Arabic handwritten character recognition, outperforming both the baseline methods and several state-of-the-art techniques from the literature. The combination of convolutional and bidirectional recurrent layers allows our models to effectively extract both spatial and sequential features, leading to improved recognition performance. Future work should focus on further improving the models, exploring their generalization to diverse handwriting datasets, and investigating their applicability to other character recognition tasks or languages.

6 | Conclusion

In this research article, we designed and evaluated two new models using deep learning techniques: CNN_Bi-LSTM, and CNN_Bi-GRU, specifically designed for Arabic handwriting character recognition. Our objective was to classify letters into 28, and 29 classes based on their shape. The evaluation was conducted using two datasets: AHCD, and Hijjaa, which contain Arabic handwriting samples. We compared the performances of the CNN_Bi-LSTM, and CNN_Bi-GRU models among themselves, as well as against existing approaches in the literature.

The experimental results presented in this study demonstrate the efficacy of the proposed hybrid deep learning approach for enhancing Arabic handwriting recognition. The CNN-Bi-GRU model achieved a state-of-the-art accuracy of 97.05% on the AHCD dataset, outperforming previous deep learning methods, while the CNN-Bi-LSTM model achieved 91.78% accuracy on the Hijjaa dataset, also surpassing existing techniques. These results highlight the power of integrating convolutional and bidirectional recurrent neural network components to capture both spatial and temporal/contextual features in handwritten Arabic text. The key contributions of this work include: 1) the introduction of a novel hybrid deep learning framework that combines CNNs and Bi-LSTMs/Bi-GRUs, 2) the achievement of significant performance gains and new state-of-the-art results on benchmark datasets, 3) the enhanced modeling capabilities afforded by the integration of convolutional and recurrent components, and 4) the broad applicability of the findings in advancing sophisticated and precise deep learning systems for Arabic handwriting recognition with wide-ranging real-world applications.

7 | Limitation and Future Work

Limitations of this research include the potential for further improvement in recognition accuracy of the proposed hybrid models, particularly when dealing with challenging or noisy handwritten Arabic data. The evaluation of the models was conducted on two specific benchmark datasets (AHCD and Hijjaa), leaving their generalization capabilities across a broader range of Arabic handwriting datasets and real-world applications yet to be fully explored. Additionally, the paper lacks information regarding the computational complexity, training time, and inference speed of the hybrid models, which are crucial considerations for practical deployments. Furthermore, the analysis of the models' performance is primarily focused on overall accuracy metrics, with limited insights provided into error patterns, failure cases, or the interpretability of the model decisions.

Considering the aforementioned potential limitations, several future research directions can be pursued. Firstly, exploring model enhancements by investigating alternative hybrid architectures or incorporating additional neural network components, such as attention mechanisms or transformers, could further improve recognition accuracy and enhance the robustness of the models. A detailed analysis of the contribution of each model component, including CNN, Bi-LSTM, and Bi-GRU, should be conducted to identify opportunities for optimization.

Furthermore, experimenting with advanced data augmentation techniques can aid in improving the generalization capabilities of the models. To expand the evaluation scope, it is essential to assess the performance of the hybrid models on a wider range of Arabic handwriting datasets, including more diverse and challenging samples. Additionally, evaluating the models in real-world Arabic handwriting recognition applications, such as document processing, form filling, or historical manuscript digitization, would provide valuable insights into their practical utility.

The transfer learning capabilities of the proposed models should be explored to adapt them to different handwriting styles or languages. Practical deployment considerations involve analyzing the computational efficiency, training time, and inference speed of the hybrid models to optimize their performance in resource-constrained environments. Techniques such as model compression or knowledge distillation can be investigated to reduce the model size and latency without compromising accuracy.

To improve the interpretability and explainability of the hybrid models' decision-making processes, innovative techniques should be developed. Furthermore, expanding the scope of Arabic handwriting recognition entails extending the proposed hybrid approach to other Arabic language processing tasks, such as Arabic text recognition, language understanding, or translation. Additionally, exploring the integration of handwriting recognition models with broader Arabic document understanding systems can enable comprehensive document digitization and analysis.

Acknowledgments

The author is grateful to the editorial and reviewers, as well as the correspondent author, who offered assistance in the form of advice, assessment, and checking during the study period.

Funding

This research was conducted without external funding support.

Data Availability

The implementation used in this article were be in GitHub. For details, please refer to https://github.com/MohamedGresha/deep-learning-methods-on-Optical-Character-Recognition-Arabic-text/tree/main

Conflicts of Interest

The author declares that there is no conflict of interest in the research.

Ethical Approval

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

References

[1] R. Ahmed, K. Dashtipour, M. Gogate, A. Raza, R. Zhang, K. Huang, A. Hawalah, A. Adeel, A. Hussain, Offline arabic handwriting recognition using deep machine learning: A review of recent advances, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11691 LNAI (2020) 457–468. https://doi.org/10.1007/978-3-030-39431-8_44.

[2] S. Alghyaline, Arabic Optical Character Recognition: A Review, CMES - Computer Modeling in Engineering and Sciences 135 (2023) 1825–1861. https://doi.org/10.32604/cmes.2022.024555.

[3] N. Alrobah, S. Albahli, Arabic Handwritten Recognition Using Deep Learning: A Survey, Arab J Sci Eng 47 (2022) 9943–9963. https://doi.org/10.1007/s13369-021-06363-3.

[4] M.G. Mahdi, A. Sleem, I. Elhenawy, Deep Learning Algorithms for Arabic Optical Character Recognition: A Survey, Multicriteria Algorithms with Applications 2 (2024) 65–79. https://doi.org/10.61356/J.MAWA.2024.26861.

[5] W. Albattah, S. Albahli, Intelligent Arabic Handwriting Recognition Using Different Standalone and Hybrid CNN Architectures, Applied Sciences 2022, Vol. 12, Page 10155 12 (2022) 10155. https://doi.org/10.3390/APP121910155.

[6] A. Fateh, M. Fateh, V. Abolghasemi, Enhancing optical character recognition: Efficient techniques for document layout analysis and text line detection, Engineering Reports (2023) e12832. https://doi.org/10.1002/ENG2.12832.

[7] A. Fateh, R.T. Birgani, M. Fateh, V. Abolghasemi, Advancing Multilingual Handwritten Numeral Recognition With Attention-Driven Transfer Learning, IEEE Access 12 (2024) 41381–41395. https://doi.org/10.1109/ACCESS.2024.3378598.

[8] A. Fateh, M. Rezvani, A. Tajary, M. Fateh, Persian printed text line detection based on font size, Multimed Tools Appl 82 (2023) 2393–2418. https://doi.org/10.1007/S11042-022-13243-X/METRICS.

[9] A.A.A. Ali, M. Suresha, H.A.M. Ahmed, A Survey on Arabic Handwritten Character Recognition, SN Comput Sci 1 (2020) 1–10. https://doi.org/10.1007/S42979-020-00168-1/TABLES/4.

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

[11] M. Rabi, M. Amrouche, Convolutional Arabic Handwriting Recognition System Based BLSTM-CTC Using WBS Decoder, International Journal of Intelligent Systems and Applications in Engineering 12 (2024) 535–548. https://www.ijisae.org/index.php/IJISAE/article/view/4888 (accessed April 28, 2024).

[12] U. Porwal, A. Fornés, F. Shafait, Advances in handwriting recognition, International Journal on Document Analysis and Recognition 25 (2022) 241–243. https://doi.org/10.1007/S10032-022-00421-8.

[13] S. Faizullah, M.S. Ayub, S. Hussain, M.A. Khan, A Survey of OCR in Arabic Language: Applications, Techniques, and Challenges, Applied Sciences 2023, Vol. 13, Page 4584 13 (2023) 4584. https://doi.org/10.3390/APP13074584.

[14] Y.A. Nanehkaran, J. Chen, S. Salimi, D. Zhang, A pragmatic convolutional bagging ensemble learning for recognition of Farsi handwritten digits, J Supercomput 77 (2021) 13474–13493. https://doi.org/10.1007/S11227-021-03822-4.

[15] Y.A. Nanehkaran, D. Zhang, S. Salimi, J. Chen, Y. Tian, N. Al-Nabhan, Analysis and comparison of machine learning classifiers and deep neural networks techniques for recognition of Farsi handwritten digits, Journal of Supercomputing 77 (2021) 3193–3222. https://doi.org/10.1007/S11227-020-03388-7/FIGURES/15.

[16] H.M. Balaha, H.A. Ali, M. Badawy, Automatic recognition of handwritten Arabic characters: a comprehensive review, Neural Comput Appl 33 (2021) 3011–3034. https://doi.org/10.1007/S00521-020-05137-6/TABLES/13.

[17] T.M. Ghanim, M.I. Khalil, H.M. Abbas, Comparative Study on Deep Convolution Neural Networks DCNN-Based Offline Arabic Handwriting Recognition, IEEE Access 8 (2020) 95465–95482. https://doi.org/10.1109/ACCESS.2020.2994290.

[18] A. El-Sawy, M. Loey, H. El-Bakry, Arabic handwritten characters recognition using convolutional neural network, WSEAS Transactions on Computer Research 5 (2017) 11–19.

[19] H.M. Balaha, H.A. Ali, M. Saraya, M. Badawy, A new Arabic handwritten character recognition deep learning system (AHCR-DLS), Neural Comput Appl 33 (2021) 6325–6367. https://doi.org/10.1007/S00521-020-05397-2/FIGURES/33.

[20] I.P. de Sousa, Convolutional ensembles for Arabic Handwritten Character and Digit Recognition, PeerJ Comput Sci 2018 (2018) e167. https://doi.org/10.7717/PEERJ-CS.167/SUPP-4.

[21] C. Boufenar, A. Kerboua, M. Batouche, Investigation on deep learning for off-line handwritten Arabic character recognition, Cogn Syst Res 50 (2018) 180–195. https://doi.org/10.1016/J.COGSYS.2017.11.002.

[22] Z. Ullah, M. Jamjoom, An intelligent approach for Arabic handwritten letter recognition using convolutional neural network, PeerJ Comput Sci 8 (2022) e995. https://doi.org/10.7717/PEERJ-CS.995/SUPP-2.

[23] H. Alyahya, M.M. Ben Ismail, A. Al-Salman, Deep ensemble neural networks for recognizing isolated Arabic handwritten characters, ACCENTS Transactions on Image Processing and Computer Vision 6 (2020) 68–79. https://doi.org/10.19101/TIPCV.2020.618051.

[24] M.N. Aljarrah, M.M. Zyout, R. Duwairi, Arabic Handwritten Characters Recognition Using Convolutional Neural Network, 2021 12th International Conference on Information and Communication Systems, ICICS 2021 (2021) 182–188. https://doi.org/10.1109/ICICS52457.2021.9464596.

[25] A.A.A. Ali, S. Mallaiah, Intelligent handwritten recognition using hybrid CNN architectures based-SVM classifier with dropout, Journal of King Saud University - Computer and Information Sciences 34 (2022) 3294–3300. https://doi.org/10.1016/J.JKSUCI.2021.01.012.

[26] J.H. Alkhateeb, An effective deep learning approach for improving off-line arabic handwritten character recognition, International Journal of Software Engineering and Computer Systems 6 (2020) 53–61.

[27] B.H. Nayef, S.N.H.S. Abdullah, R. Sulaiman, Z.A.A. Alyasseri, Optimized leaky ReLU for handwritten Arabic character recognition using convolution neural networks, Multimed Tools Appl 81 (2022) 2065–2094. https://doi.org/10.1007/s11042-021-11593-6.

[28] N. Wagaa, H. Kallel, N. Mellouli, Improved Arabic Alphabet Characters Classification Using Convolutional Neural Networks (CNN), Comput Intell Neurosci 2022 (2022) 9965426. https://doi.org/10.1155/2022/9965426.

[29] N. Alrobah, S. Albahli, A Hybrid Deep Model for Recognizing Arabic Handwritten Characters, IEEE Access 9 (2021) 87058–87069. https://doi.org/10.1109/ACCESS.2021.3087647.

[30] L. Bouchriha, A. Zrigui, S. Mansouri, S. Berchech, S. Omrani, Arabic Handwritten Character Recognition Based on Convolution Neural Networks, Communications in Computer and Information Science 1653 CCIS (2022) 286–293. https://doi.org/10.1007/978-3-031-16210-7_23.

[31] A. Darwish, A Data-driven Deep Learning Approach for Remaining Useful Life in the ion mill etching Process, Sustainable Machine Intelligence Journal 8 (2024) (2):14-34. https://doi.org/10.61356/SMIJ.2024.8288.

[32] S. Mach, W. Abdullah, A. Tolba, A. Elmasry, N.N. Mostafa, VisionCam: A Comprehensive XAI Toolkit for Interpreting Image-Based Deep Learning Models, Sustainable Machine Intelligence Journal 8 (2024) (4):46-55. https://doi.org/10.61356/SMIJ.2024.8290.

[33] N.M.-S.M.I. Journal, undefined 2024, GH-Twin: Graph Learning Empowered Hierarchical Digital Twin for Optimizing Self-Healing Networks, Sciencesforce.ComN MoustafaSustainable Machine Intelligence Journal, 2024•sciencesforce.Com 8 (2024) 35–45. https://doi.org/10.61356/SMIJ.2024.8289.

[34] K.S. Younis, Arabic hand-written character recognition based on deep convolutional neural networks, Jordanian Journal of Computers and Information Technology 3 (2017).

[35] M.N. AlJarrah, M.Z. Mo'ath, R. Duwairi, Arabic handwritten characters recognition using convolutional neural network, in: 2021 12th International Conference on Information and Communication Systems (ICICS), 2021: pp. 182–188.

[36] A. Bin Durayhim, A. Al-Ajlan, I. Al-Turaiki, N. Altwaijry, Towards Accurate Children's Arabic Handwriting Recognition via Deep Learning, Applied Sciences 2023, Vol. 13, Page 1692 13 (2023) 1692. https://doi.org/10.3390/APP13031692.

[37] M.G. Mahdi, A. Sleem, I. Elhenawy, S. Safwat, Advancing Arabic Handwriting Recognition with Convolutional and Recurrent Neural Network Ensembles, Information Sciences with Applications 3 (2024) 43–63. https://doi.org/10.61356/J.ISWA.2024.3294.