Neutrosophic Sets and Systems, Vol. 72, 2024
University of New Mexico

Hybrid Neutrosophic Deep Learning Model for Enhanced Arabic Handwriting Recognition

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

[1] Department of Information Technology, Faculty of Information Technology and Computer Science, Sinai University, Qantara Branch., Ismailia, Egypt (mohamedgresha@hotmail.com)
[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)
* Correspondence: m.gresha@fci.zu.edu.eg ; mohamedgresha@hotmail.com

Abstract: Recognizing handwritten Arabic characters poses a significant challenge due to the complexities of the cursive script and the visual similarities between characters. While deep learning techniques have shown substantial promise, advancements in model architectures are essential to further enhance performance. Neutrosophic Sets (NS) have demonstrated their potential in improving classification models by effectively handling indeterminate and inconsistent data. This paper introduces a novel approach that integrates Neutrosophic Sets with a hybrid deep learning model, combining Convolutional Neural Networks (CNNs) with Bidirectional Recurrent Neural Networks (Bi-LSTM and Bi-GRU). This integration allows for the extraction of spatial features and modeling of temporal dynamics in handwritten Arabic text. Experiments conducted on the Hijjaa and AHCD datasets revealed that the NS_CNN_Bi-LSTM model achieved an accuracy of 92.38% on the Hijjaa dataset, while the NS_CNN_Bi-GRU model attained 97.38% accuracy on the AHCD dataset, outperforming previous deep learning approaches. These results highlight the significant performance improvements achieved through advanced temporal modeling and contextual representation, without the need for explicit segmentation. The findings contribute to the ongoing development of highly accurate and sophisticated deep learning systems for Arabic handwriting recognition, with broad applications in areas requiring efficient extraction of text from handwritten documents.

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

1. Introduction

Handwriting recognition, a key area within Optical Character Recognition (OCR), is crucial in various sectors like industry, education, government, and healthcare. Arabic handwriting recognition, in particular, is a challenging but promising research area due to its complex, cursive nature and the variability in individual handwriting styles [1,2]. Despite these challenges, recent advancements have improved the accuracy of Arabic handwriting recognition systems.

Deep learning, particularly through Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, has revolutionized handwriting recognition. RNNs, despite initial training difficulties, benefit from LSTM networks that handle contextual information better. To address challenges in offline handwriting recognition, images can be converted into 1D sequences and processed bidirectionally with LSTM networks. Integrating Convolutional Neural Networks

(CNNs) with RNNs has further enhanced the ability to recognize and process handwritten characters effectively [3].

Neutrosophic Sets (NS) and Neutrosophic Numbers (NN) provide a robust mathematical framework, finding applications in various fields, including Decision-Making, Image Processing, and Information Systems. These frameworks are particularly relevant in contexts involving indeterminate, inconsistent, or incomplete information [4–9]. By extending classical and fuzzy set theories to include the indeterminacy component, neutrosophic sets offer a flexible, generalized tool that more accurately reflects the complexities of real-world scenarios, making them particularly useful where traditional approaches may be inadequate.

This study aims to develop an advanced system for offline Arabic handwriting recognition by harnessing the capabilities of CNNs, Bidirectional Long Short-Term Memory (Bi-LSTM) networks, and Bidirectional Gated Recurrent Units (Bi-GRU), while also incorporating Neutrosophic Sets (NS) and Neutrosophic Numbers (NN) into existing models. The proposed system is specifically engineered to tackle the unique challenges inherent in Arabic handwriting recognition.

The primary contributions of this research are as follows:

- Examination of Neutrosophic Sets and Neutrosophic Numbers, including their applications across different domains.
- Development and implementation of innovative novel models designed to enhance recognition accuracy.
- Evaluation and comparison of the proposed models with state-of-the-art approaches and previous studies.

The paper is structured as follows: Section 2 presents a comprehensive survey of the current advancements in Arabic handwriting recognition. Section 3 provides an overview of Neutrosophic Sets and Neutrosophic Numbers. Section 4 outlines the details of our proposed system. In Section 5, we report the results of our experiments. Section 6 offers an in-depth discussion and a systematic evaluation of the system's performance. Section 7 summarizes our key findings and suggests potential avenues for future research. Finally, Section 8 revisits and elaborates on these proposed directions for future investigation in this field.

2. Related work

This research represents the first application of Neutrosophic Sets and Neutrosophic Numbers in the field of Recognition of Handwritten Arabic Characters. Consequently, this section is divided into two parts: The first part reviews 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. The second part examines the latest research on Neutrosophic Sets and Neutrosophic Numbers with a focus on Image Classification. The reviewed studies primarily focus on introducing various approaches to tackle this challenging task.

2.1. Recognition of Handwritten Arabic Characters

In 2017, El-Sawy et al. [10] 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 2020 study conducted by Altwaijry et al. [11], 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 [10] 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.

Nayef et al. [12] 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 [12], achieving accuracy rates of 99%, 95%, and 90% on AHCD, the researchers' dataset, and Hijja, respectively.

2.2. Neutrosophic in Image Classification

In recent years, Neutrosophic has attracted considerable attention within the scientific community. In [13], the authors proposed a membership-based neutrosophic approach for supervised fingerprint image classification. The model applies neutrosophic logic to handle uncertainty in fingerprint features, enhancing classification accuracy. The study evaluated the approach using standard datasets and reported improved performance compared to traditional techniques. The method effectively handled noise and ambiguous data, proving its robustness for fingerprint image analysis.

Elatawy et al.[14] developed a system to recognize alphabet Arabic sign language using neutrosophic techniques and fuzzy c-means. The system enhances input images with a Gaussian filter, then converts them to a neutrosophic domain for feature extraction. This leads to the classification of corresponding letters, with a reported accuracy of 91%. The combination of neutrosophic image processing and fuzzy clustering effectively improves sign language recognition accuracy.

In 2023, the authors proposed a deep learning model combining Neutrosophic and CNN for image and text classification. They used a neutrosophic approach to preprocess input data, enhancing feature extraction and noise reduction. The model was evaluated on multiple datasets and achieved notable improvements in accuracy compared to traditional methods. Data augmentation techniques, such as rotation and flipping, were applied to improve generalization. The authors also experimented with various optimizers, with the Adam optimizer showing the best results. Their model demonstrated its robustness in handling complex classification tasks[4].

3. Neutrosophic Sets and Neutrosophic Numbers

Neutrosophic Sets (NS), introduced by Florentin Smarandache in 1998[15], represent a significant extension of classical set theory and fuzzy set theory. The core idea behind neutrosophic sets is to handle the uncertainty, imprecision, incompleteness, and inconsistency that often arise in real-world situations. While classical set theory strictly dichotomizes elements into members and non-members, and fuzzy set theory introduces the notion of partial membership, neutrosophic sets allow for a more nuanced representation by incorporating three independent components: truth-membership (T), indeterminacy-membership (I), and falsity-membership (F)[15–17].

3.1. Formal Definition of Neutrosophic Sets

A Neutrosophic Set (NS) can be characterized as a three-part structure:

- T: The degree of truth.
- I: The degree of indeterminacy.
- F: The degree of falsity.

Each element in the neutrosophic set is represented by a triplet $x(T, I, F)$. And Formally, a Neutrosophic Set A in a universe of discourse U is defined as:

$$A = \{\langle x, T_A(x), I_A(x), F_A(x)\rangle : x \in U\}, \quad (1)$$

where:

- $T(x) \in [0,1]$ is the truth-membership degree of the element x.
- $I(x) \in [0,1]$ is the indeterminacy-membership degree of x.

- $F(x) \in [0,1]$ is the falsity-membership degree of x.

Unlike in Fuzzy Sets, where $T_A(x) + F_A(x) = 1$, Neutrosophic Sets allow the sum $T_A(x) + I_A(x) + F_A(x)$ to take any value in the real number space, thus offering flexibility in modeling situations where the information is incomplete or conflicting.

For example, if we consider an element x in a Neutrosophic Set A, the membership degrees might be:

$$T_A(x) = 0.6, I_A(x) = 0.3, F_A(x) = 0.2$$

In this case, $T_A(x) + I_A(x) + F_A(x) = 1.1$, indicating that the sum of the three components is not necessarily constrained to 1, unlike in classical or Fuzzy Sets. This ability to model indeterminacy $I_A(x)$ independently of the truth $T_A(x)$ and falsity $F_A(x)$ components make Neutrosophic Sets particularly powerful in contexts where uncertainty is a dominant factor.

Building on the concept of Neutrosophic Sets, Neutrosophic Numbers (NN) have been developed to represent uncertain quantities in mathematical and engineering contexts. A Neutrosophic Number N is defined as a triplet:

$$N = \langle T_N, I_N, F_N \rangle, \quad (2)$$

where T_N , I_N , and F_N are the truth, indeterminacy, and falsity components, respectively. These components are often intervals or real numbers, reflecting the uncertainty and vagueness associated with the quantity.

Neutrosophic Numbers can be used in Arithmetic Operations, Decision-Making processes, and various Engineering Applications where conventional numbers fail to capture the complexity of the situation.

3.2. Types of Neutrosophic Sets

There are several variations of neutrosophic sets, designed to capture different aspects of uncertainty:

3.2.1. Single-Valued Neutrosophic Sets (SVNS)

A **Single-Valued Neutrosophic Set (SVNS)** is an extension of the traditional Fuzzy Set, designed to handle uncertainty, vagueness, and indeterminacy[18]. Each element x in a Neutrosophic Set is characterized by three independent membership functions:

- **Truth Membership** $T(x)$, which measures the degree of truth.
- **Indeterminacy Membership** $I(x)$, which quantifies the level of indeterminacy or uncertainty about the element's membership.
- **Falsity Membership** $F(x)$, which measures the degree of falsity.

Mathematically, for a given element x, we have:

$$T(x), I(x), F(x) \in [0,1], \quad (3)$$

These values represent the truth, indeterminacy, and falsity respectively, with the constraint:

$$0 \leq T(x) + I(x) + F(x) \leq 3, \quad (4)$$

An SVNS provides a flexible way to express uncertainty in data, where each of the three values reflects different aspects of the element's membership in the set. SVNS are especially useful in problems like Image Processing, Decision-Making, and Classification, where managing imprecise or noisy data is crucial.

3.2.2. Interval-Valued Neutrosophic Sets (IVNS)

Interval-Valued Neutrosophic Sets (IVNS) further extend the SVNS by allowing the membership functions—truth, indeterminacy, and falsity—to take interval values instead of single

values. This reflects greater uncertainty, as each membership is now an interval rather than a precise number[19].

For an element x, the truth membership $T(x)$, indeterminacy membership $I(x)$, and falsity membership $F(x)$ are intervals defined as:

$$T(x) \in [T_{\min}, T_{\max}],\ I(x) \in [I_{\min}, I_{\max}],\ F(x) \in [F_{\min}, F_{\max}], \tag{5}$$

with the condition that:

$$T_{\min}, I_{\min}, F_{\min} \geq 0,\ T_{\max}, I_{\max}, F_{\max} \leq 1, \tag{6}$$

and:

$$0 \leq T_{\max} + I_{\max} + F_{\max} \leq 3, \tag{7}$$

This representation allows the model to capture a range of possible values for each membership, reflecting uncertainty or variability in the data more effectively than a single value. IVNS are useful in cases where the data itself is inherently imprecise, such as Expert Decision-Making with vague or uncertain information.

3.2.3. Multi-Valued Neutrosophic Sets

In **Multi-Valued Neutrosophic Sets**, the membership values for truth, indeterminacy, and falsity are drawn from a discrete set rather than a continuous interval[20]. For an element $\boldsymbol{x}$, we have:

$$T(x) \in \{t_1, t_2, \dots, t_n\},\ I(x) \in \{i_1, i_2, \dots, i_m\},\ F(x) \in \{f_1, f_2, \dots, f_p\}, \tag{8}$$

where $\boldsymbol{t_i}$, $\boldsymbol{i_j}$, and $\boldsymbol{f_k}$ are discrete values typically chosen from a finite set of real numbers. This framework is appropriate for problems where the degree of truth, falsity, and indeterminacy must take on specific predefined values rather than being continuous. For instance, multi-valued neutrosophic sets are useful in Logical Systems or Decision Models where categorical values are required.

3.2.4. Generalized Neutrosophic Sets

Generalized Neutrosophic Sets extend the neutrosophic model by allowing the truth, indeterminacy, and falsity membership functions to take values outside the traditional interval [0, 1] [21]. For an element x, the memberships are defined as:

$$T(x), I(x), F(x) \in \mathbb{R}, \quad \text{with no restrictions,} \tag{9}$$

This type of set allows membership values to exceed 1 or be negative, which can represent extreme certainty or extreme falsity, respectively. Generalized Neutrosophic Sets are more abstract and are used in cases where a wider range of membership values is necessary, often in philosophical or highly theoretical applications.

3.2.5. Refined Neutrosophic Sets

Refined Neutrosophic Sets provide a more granular view of the truth, indeterminacy, and falsity memberships by dividing each of these components into sub-components[22]. For example, truth membership could be refined into positive truth and negative truth, capturing more subtle nuances of the element's relationship with the set. For an element $\boldsymbol{x}$, the truth membership could be defined as:

$$\boldsymbol{T(x) = T_1(x) + T_2(x)},\ \text{where}\ \ \boldsymbol{T_1(x)}\ \text{is positive truth, and}\ \boldsymbol{T_2(x)}\ \text{is negative truth.}, \tag{10}$$

This allows for a more detailed representation of uncertainty and is useful in complex Decision-Making problems where the fine-grained nature of the data is important.

In summary, **SVNS** are simpler and computationally efficient but less flexible in representing complex uncertainty compared to **IVNS**, **Generalized**, or **Refined Neutrosophic Sets**, which provide more nuanced and detailed representations of uncertainty at the cost of computational complexity.

4. Research Methodology

In this research, we employ the most utilized strategies and techniques, 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.

4.1. 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], created by Saudi Arabian schoolchildren [11], comprises 47,434 images of 108 Arabic letters in various word positions. Despite initial issues with letter placement and alignment, the dataset was resized to 32x32 pixels, balanced for letter representation, and organized into 29 folders without vocalization diacritics. In contrast, the Arabic Handwritten Characters Dataset[2] (AHCD) includes 16,800 characters written by 60 participants, scanned at 300 dpi, and divided into training and test sets [10]. The AHCD dataset's pre-processing involved converting images to grayscale and applying filters to handle variations in writing styles and character shapes.

4.2. Proposed Models

The final models represent a sophisticated Deep Learning model architecture designed to incorporate Neutrosophic Logic principles into a CNNs structure, followed by a series of (Bi-LSTM/Bi-GRU) layers. The model leverages the unique capabilities of both CNNs and (LSTM/GRU) to capture spatial features and temporal dependencies, making it suitable for complex tasks such as Image Recognition, where uncertainty and imprecision are inherent.

4.2.1. Neutrosophic Transformation and Layer

In neutrosophic image processing, we aim to represent an image in terms of **truth** (T), **indeterminacy** (I), and **falsity** (F), which provide a more nuanced interpretation of image characteristics compared to classical binary or fuzzy representations. Below is a detailed explanation of each step in the conversion of an image into a neutrosophic representation, including the relevant equations.

4.2.1.1. Preprocessing the Input Image

The input is a grayscale image, where pixel values range from 0 to 255. The first step in converting the image into a neutrosophic image involves normalizing these pixel intensities to a scale between 0 and 1. This normalization makes the subsequent calculations easier to handle numerically.

$$N(x,y) = \frac{I(x,y)}{255}, \tag{11}$$

Where:

- $N(x,y)$is the normalized pixel intensity at position (x,y).

[1] https://github.com/israksu/Hijja2

[2] https://www.kaggle.com/datasets/mloey1/ahcd1

- $I(x, y)$ is the original intensity of the pixel at position (x, y), which lies between 0 and 255.

4.2.1.2. Calculation of Local Mean Intensity

To assess the **local intensity** around each pixel, we compute the local mean of pixel values in a neighborhood window (usually a 3x3 or 5x5 window) around the pixel (x, y).

The local mean $\mu(x, y)$ is given by:

$$\mu(x, y) = \frac{1}{|W|} \sum_{(x', y') \in W} N(x', y'), \quad (12)$$

Where:

- $\mu(x, y)$ is the local mean intensity around the pixel (x, y).
- W is the neighborhood window centered around (x, y).
- $|W|$ is the number of pixels in the neighborhood window.

4.2.1.3. Calculation of Local Standard Deviation

Next, we compute the **local standard deviation** $\sigma(x, y)$ in the same neighborhood. The local standard deviation measures the variation in pixel intensities within the neighborhood window. It provides insight into how homogeneous or inhomogeneous the local region is.

$$\sigma(x, y) = \sqrt{\frac{1}{|W|} \sum_{(x', y') \in W} (N(x', y') - \mu(x, y))^2}, \quad (13)$$

Where:

- $\sigma(x, y)$ is the local standard deviation at pixel (x, y).
- $\mu(x, y)$ is the local mean intensity computed in Step 2.
- W is the neighborhood window.

4.2.1.4. Calculation of Truth Membership (T)

The **truth membership** (T) evaluates how similar the intensity of a pixel is to its surrounding neighborhood. A pixel whose intensity is close to the local mean will have a high truth value, indicating that it likely belongs to the background or a homogenous region.

The truth membership is defined as:

$$T(x, y) = \frac{N(x,y) - min(N)}{max(N) - min(N)}, \quad (14)$$

Where:

- $T(x, y)$ is the truth membership of the pixel at position (x, y).
- $N(x, y)$ is the normalized intensity of the pixel.
- $min(N)$ and $max(N)$ are the minimum and maximum intensity values in the neighborhood window W.

This normalization ensures that truth values are scaled between 0 and 1.

4.2.1.5. **Calculation of Indeterminacy Membership** (I)

The **indeterminacy membership** (I) quantifies the uncertainty associated with a pixel's intensity. In regions where there is high variance in intensity (e.g., edges or noisy areas), the indeterminacy will be high.

The indeterminacy membership is calculated using the local standard deviation:

$$I(x,y) = 1 - \exp\left(-\frac{\sigma(x,y)^2}{2}\right), \quad (15)$$

Where:

- $I(x,y)$ is the indeterminacy membership of the pixel at position (x,y).
- $\sigma(x,y)$ is the local standard deviation calculated in Step 3.

This formula ensures that indeterminacy increases as the standard deviation (and hence the local variation in pixel intensities) increases.

4.2.1.6. **Calculation of Falsity Membership** (F)

The **falsity membership** (F) is the complement of the truth membership. It indicates how different a pixel's intensity is from its neighbors, potentially identifying regions that do not belong to the background or homogeneous areas.

$$F(x,y) = 1 - T(x,y), \quad (16)$$

Where:

- $F(x,y)$ is the falsity membership of the pixel at position (x,y).
- $T(x,y)$ is the truth membership calculated in Step 4.

4.2.1.7. **Construction of the Neutrosophic Image**

Finally, the image is represented in a **Neutrosophic Set** by combining the truth (T), indeterminacy (I), and falsity (F) memberships for each pixel. Thus, each pixel in the image is represented as a triplet $(\boldsymbol{T}, \boldsymbol{I}, \boldsymbol{F})$, where:

- $T(x,y)$ captures how similar the pixel is to the local region.
- $I(x,y)$ captures the uncertainty or indeterminacy about the pixel.
- $F(x,y)$ captures how different the pixel is from the local region.

Final Output

The output is a **Neutrosophic Image**, where each pixel is represented as:

$$\text{Neutrosophic Image} = \{(T(x,y), I(x,y), F(x,y)) \mid x,y \in \text{Image Dimensions}\}, \quad (17)$$

This representation provides more information than classical pixel intensities by accounting for local homogeneity (truth), uncertainty (indeterminacy), and differences (falsity) within the image.

This step-by-step process allows for a comprehensive analysis of the image in terms of local features and uncertainties, making it a powerful tool for image processing tasks such as denoising, edge detection, and segmentation.

4.2.2. Convolutional Layers

Next, the model employs multiple convolutional layers to extract spatial features from the transformed input. Convolutional layers apply filters to the input tensor to detect local patterns such as edges or textures, which are crucial for understanding image content. Each convolutional operation can be represented as:

$$X_{\text{conv}}^{(l)} = \sigma(W^{(l)} * X_{\text{conv}}^{(l-1)} + b^{(l)}), \quad (18)$$

where $X_{\text{conv}}^{(l)}$ is the output of the l-th convolutional layer, $W^{(l)}$ are the filters, $*$ denotes the convolution operation, $b^{(l)}$ is the bias term, and σ is an activation function, typically ReLU.

MaxPooling layers follow some convolutional layers, downsampling the feature maps to reduce computational complexity and to focus on the most prominent features. Batch normalization is applied to stabilize and accelerate training by normalizing the input to each layer, ensuring that it has a mean of zero and a standard deviation of one.

4.2.3. Reshaping and Bidirectional recurrent neural network Layers

After the convolutional layers, the 3D tensor X_{conv} is reshaped to a 2D tensor in preparation for the GRU layers. Specifically, the tensor is reshaped from dimensions (H', W', C') to $(H' \times W', C')$, where $H' \times W'$ is treated as the sequence length, and C' represents the features at each timestep. This operation is mathematically described as:

$$X_{\text{reshape}} = \text{Reshape}(X_{\text{conv}}) = (H' \times W', C'), \tag{19}$$

4.2.3.1. Bi-LSTM

The Bi-LSTM layers then process this reshaped tensor. LSTMs (Long Short-Term Memory networks) are a type of Recurrent Neural Network (RNN) that can capture temporal dependencies by maintaining a hidden state that evolves over the sequence while also incorporating mechanisms to learn when to forget or remember certain information. The bidirectional nature of the LSTM means that it processes the input sequence in both forward and backward directions, capturing context from both past and future states. The output from each LSTM layer is:

$$H_t^{\text{Bi-LSTM}} = \text{LSTM}_{\text{fwd}}(X_{\text{reshape},t}) \oplus \text{LSTM}_{\text{bwd}}(X_{\text{reshape},t}), \tag{20}$$

where $\oplus$ denotes concatenation, LSTM_{fwd} and LSTM_{bwd} are the forward and backward LSTMs, and $\boldsymbol{H_t^{\text{Bi-LSTM}}}$ is the hidden state at time step $\boldsymbol{t}$.

4.2.3.2. Bi-GRU

The Bi-GRU layers then process this reshaped tensor. GRUs are a type of Recurrent Neural Network (RNN) that can capture temporal dependencies by maintaining a hidden state that evolves over the sequence. The bidirectional nature of the GRU means that it processes the input sequence in both forward and backward directions, capturing context from both past and future states. The output from each GRU layer is:

$$H_t^{\text{Bi-GRU}} = \text{GRU}_{\text{fwd}}(X_{\text{reshape},t}) \oplus \text{GRU}_{\text{bwd}}(X_{\text{reshape},t}), \tag{21}$$

where $\oplus$ denotes concatenation, GRU_{fwd} and GRU_{bwd} are the forward and backward GRUs, and $\boldsymbol{H_t^{\text{Bi-GRU}}}$ is the hidden state at time step $\boldsymbol{t}$.

4.2.3. Fully Connected Layers and Output

Finally, the output of the last RNN layer is flattened into a vector and passed through several dense (fully connected) layers. These layers perform non-linear transformations on the high-level features extracted by the convolutional and RNN layers, enabling the model to make predictions. The final layer applies a SoftMax activation function to produce a probability distribution over the target classes:

$$\hat{y} = \text{SoftMax}(W_{\text{out}} X_{\text{dense}} + b_{\text{out}}), \tag{22}$$

where $\hat{y}$ is the predicted output, W_{out} and b_{out} are the weights and bias of the output layer, and X_{dense} is the output of the last dense layer.

This architecture allows the model to capture both the spatial features and temporal dependencies in the data while explicitly accounting for uncertainty through neutrosophic logic. This

makes it particularly well-suited for complex tasks such as image recognition in environments where data uncertainty is a significant factor.

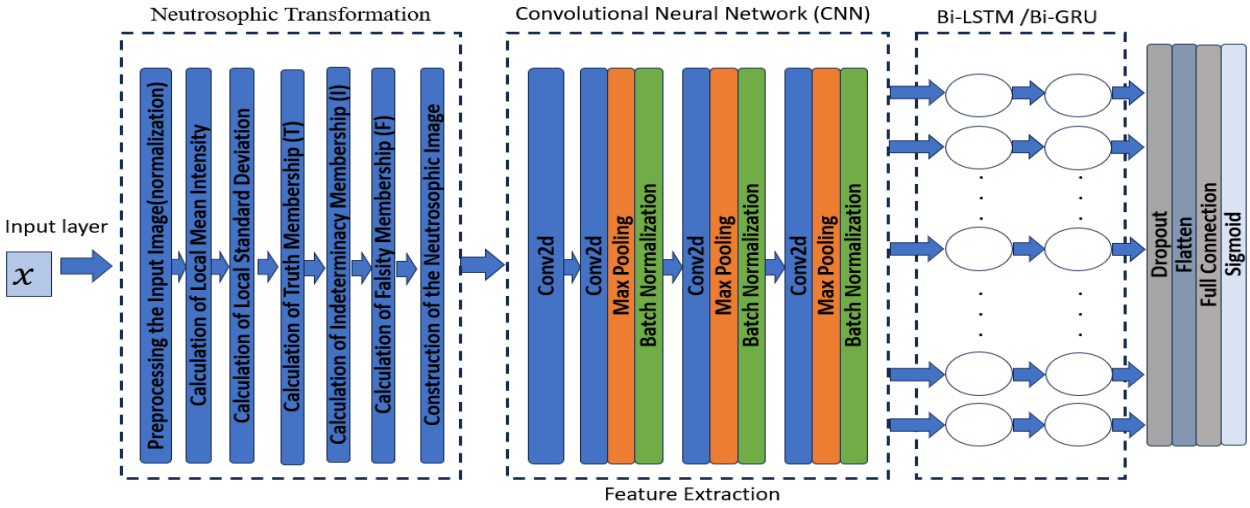

Figure 1. NS_CNN_Bi-LSTM/Bi-GRU architecture.

5. Experiments

5.1. Evaluation Phase

These evaluation metrics are employed to assess and compare the performance of classification models. Precision, recall, and F1-score provide insights into specific characteristics, such as the balance between True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), while accuracy offers a general performance overview. When interpreting and applying these metrics, it is essential to consider the problem's specific requirements and context.

1- Accuracy:

$$\textbf{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{23}$$

2- Recall:

$$\textbf{Recall} = \frac{TP}{TP + FN} \tag{24}$$

3- Precision:

$$\textbf{Precision} = \frac{TP}{TP + FP} \tag{25}$$

4- F1-score:

$$\textbf{F1} - \textbf{score} = \frac{2(\text{Recall} * \text{ Precision})}{\text{Recall} + \text{Precision}} \tag{26}$$

5.2. Experimental Setup

We conducted experiments in the Kaggle environment using a Dell G5 15 laptop, featuring an 8th generation Intel Core i7 processor with a base frequency of 2.20 GHz and a maximum frequency of 2.21 GHz. The system is equipped with an NVIDIA GeForce GTX graphics card (6 GB VRAM) and 16 GB of RAM, with settings adjusted to utilize NVIDIA TESLA P100 GPUs for all trials. While some

processes, such as those involving Pandas and Scikit-learn libraries, do not benefit from GPU acceleration, these GPUs are highly advantageous for training deep learning models.

5.3. Experimental Design

In this study, The models were compiled with a categorical cross-entropy loss function and the Adam optimizer, with accuracy as the evaluation metric. The training process involved a dataset of images and their corresponding labels. For training, a batch size of 128 was used, and the model underwent 200 epochs of training to learn from the data. A separate validation dataset, comprising images and their corresponding labels, was used to evaluate the model's capacity to generalize to previously unseen data. During training, the model's weights were adjusted according to the loss function and optimizer to minimize the loss and enhance prediction accuracy. To ensure robust performance and avoid overfitting, the validation data was continuously monitored, allowing for adjustments to the training process as needed.

6. Results and Discussion

In this study, we proposed four novel deep learning models, **CNN_Bi-LSTM, CNN_Bi-GRU, NS_CNN_Bi-LSTM, and NS_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.

In Experiment 1, the performance of four models—CNN_Bi-LSTM, CNN_Bi-GRU, NS_CNN_Bi-LSTM, and NS_CNN_Bi-GRU—was tested on the AHCD dataset. The NS_CNN_Bi-GRU model outperformed all others, achieving 97.38% accuracy on the testing set and 99.90% on the training set, with the lowest loss values (0.0034 for training and 0.2364 for testing). High training accuracies (ranging from 99.37% to 99.99%) indicate strong learning capabilities across all models, with the NS-enhanced versions performing slightly better. Test accuracies and loss metrics further confirmed the NS_CNN_Bi-GRU's superior generalization to unseen data, highlighting the impact of the NS component in improving model performance.

The models maintained consistently high macro and weighted averages for precision, recall, and F1-scores (0.96 to 0.97), demonstrating balanced performance across all character classes. Characters like ج ,ب ,ا, and س had near-perfect F1-scores, while a few, including ر ,ذ ,ت, and ظ, showed slightly lower F1-scores (0.92 to 0.96), likely due to structural similarities. Overall, the NS-enhanced models, particularly NS_CNN_Bi-GRU, exhibited better accuracy and lower losses, suggesting that GRU units may be more effective than LSTM units in character recognition tasks.

In Experiment 2, the proposed models were tested on the Hijjaa dataset, with the NS_CNN_Bi-LSTM model outperforming all others. It achieved a 92.38% accuracy on the testing set and 99.38% on the training set. The NS_CNN_Bi-LSTM also recorded the lowest loss value of 0.0206 during training, with a test loss of 0.5159, indicating its strong performance. Comparatively, CNN_Bi-LSTM, CNN_Bi-GRU, and NS_CNN_Bi-GRU had slightly lower test accuracies (ranging from 90.23% to 91.78%) and higher loss values, reflecting stable but slightly inferior generalization compared to NS_CNN_Bi-LSTM.

Class-wise, characters such as ش ,ا, and ي consistently achieved high F1-scores, indicating the models' strong ability to recognize these characters. However, characters like ذ ,د, and ء proved more challenging, particularly for CNN_Bi-LSTM and CNN_Bi-GRU models. The NS-enhanced models, especially NS_CNN_Bi-LSTM, demonstrated better performance in recognizing these difficult characters. Across all models, macro and weighted average precision, recall, and F1-scores hovered around 0.90, with NS_CNN_Bi-LSTM showing a marginally better balance and higher overall averages, suggesting its superiority in handling diverse character classes.

In summary, the results show that integrating the NS component with CNN and Bidirectional Recurrent Layers (LSTM/GRU) significantly enhances the accuracy and reduces the loss of models

for Arabic handwritten character recognition across different datasets. The NS_CNN_Bi-GRU model performs best on the AHCD dataset, achieving a strong balance between accuracy and loss, while the NS_CNN_Bi-LSTM model stands out on the Hijjaa dataset with superior accuracy and lower loss during both training and testing. Overall, the NS-enhanced models demonstrate robust performance, especially in handling difficult character classes, proving to be valuable in advancing the field of Arabic handwritten character recognition.

Table 1. The results of proposed Models on AHCD Dataset.

Model		**CNN_Bi-LSTM**			**CNN_Bi-GRU**			**NS_CNN_Bi-LSTM**			**NS_CNN_Bi-GRU**		
Char.	**Class #**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
ا	1	0.98	0.99	0.98	0.98	0.99	0.99	0.95	1	0.98	1	0.99	1
ب	2	0.98	0.98	0.98	1	0.99	1	1	0.97	0.99	0.98	0.98	0.98
ت	3	0.91	0.93	0.92	0.91	0.97	0.94	0.94	0.98	0.96	0.97	0.94	0.95
ث	4	0.93	0.95	0.94	0.96	0.97	0.97	0.97	0.98	0.98	0.98	0.99	0.98
ج	5	0.98	0.95	0.97	0.99	0.97	0.98	1	0.98	0.99	1	0.99	1
ح	6	0.94	0.98	0.96	0.94	0.99	0.97	0.94	0.97	0.96	0.94	1	0.97
خ	7	0.95	0.97	0.96	1	0.97	0.98	0.94	0.94	0.94	0.99	0.95	0.97
د	8	0.92	0.97	0.95	0.94	0.96	0.95	0.97	0.97	0.97	0.93	0.95	0.94
ذ	9	0.94	0.94	0.94	0.97	0.9	0.94	0.97	0.97	0.97	0.97	0.95	0.96
ر	10	0.97	0.94	0.95	0.91	0.99	0.95	0.94	0.99	0.97	0.91	0.98	0.95
ز	11	0.97	0.91	0.94	0.97	0.94	0.96	0.99	0.93	0.96	0.98	0.93	0.96
س	12	0.99	0.98	0.99	0.98	0.98	0.98	1	0.98	0.99	0.95	1	0.98
ش	13	0.99	0.97	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.99	0.99	0.99
ص	14	0.97	0.98	0.98	0.94	0.99	0.97	0.96	0.98	0.97	0.98	0.98	0.98
ض	15	0.96	0.97	0.96	0.99	0.93	0.96	0.99	0.93	0.96	1	0.97	0.98
ط	16	0.94	0.98	0.96	0.94	0.99	0.96	0.94	0.99	0.96	0.94	1	0.97
ظ	17	0.97	0.95	0.96	0.99	0.94	0.97	0.98	0.93	0.95	1	0.94	0.97
ع	18	0.98	0.93	0.96	0.99	0.99	0.99	0.96	0.97	0.97	0.99	0.97	0.98
غ	19	0.98	0.95	0.97	1	1	1	0.97	0.97	0.97	0.99	0.98	0.99
ف	20	0.93	0.97	0.95	0.94	0.97	0.96	0.94	0.98	0.96	0.97	0.97	0.97
ق	21	0.97	0.94	0.95	0.97	0.94	0.96	0.98	0.96	0.97	0.97	0.96	0.97
ك	22	0.94	0.97	0.95	1	0.95	0.97	0.99	0.98	0.99	0.98	0.97	0.97
ل	23	0.99	0.99	0.99	0.99	0.99	0.99	0.99	1	1	0.99	0.99	0.99
م	24	0.97	0.98	0.98	0.97	1	0.98	0.99	0.97	0.98	0.98	0.99	0.99
ن	25	0.96	0.9	0.93	0.97	0.93	0.95	0.99	0.94	0.97	0.94	0.96	0.95
ه	26	0.99	0.97	0.98	0.98	0.97	0.97	0.97	0.98	0.98	0.97	0.97	0.97
و	27	0.96	0.97	0.96	0.96	0.96	0.96	0.97	0.97	0.97	0.97	0.97	0.97
ي	28	0.97	0.97	0.97	0.99	1	1	0.99	0.98	0.99	1	0.99	1
Accuracy (train)				0.9937			0.9978			0.9999			**0.999**
Accuracy (test)				0.9616			0.9705			0.9714			**0.9738**

Loss (train)			0.023			0.0058			**0.0034**			0.0046
Loss (test)			0.2369			0.2495			**0.2364**			0.2406
Macro avg	0.96	0.96	0.96	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97
Weighted avg	0.96	0.96	0.96	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97

Table 2. The results of proposed Models on Hijjaa Dataset.

Model		**CNN_Bi-LSTM**			**CNN_Bi-GRU**			**NS_CNN_Bi-LSTM**			**NS_CNN_Bi-GRU**		
Char.	**Class #**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
ا	1	0.99	0.98	0.98	0.99	0.98	0.98	0.99	0.99	0.99	0.99	0.96	0.98
ب	2	0.93	0.97	0.95	0.92	0.95	0.94	0.93	0.94	0.93	0.94	0.94	0.94
ت	3	0.84	0.93	0.88	0.9	0.89	0.9	0.91	0.89	0.9	0.86	0.88	0.87
ث	4	0.94	0.86	0.9	0.93	0.91	0.92	0.93	0.92	0.92	0.9	0.88	0.89
ج	5	0.96	0.91	0.94	0.95	0.93	0.94	0.95	0.94	0.95	0.91	0.94	0.92
ح	6	0.85	0.89	0.87	0.84	0.9	0.87	0.88	0.89	0.89	0.85	0.84	0.84
خ	7	0.91	0.87	0.89	0.87	0.91	0.89	0.89	0.91	0.9	0.89	0.88	0.88
د	8	0.77	0.77	0.77	0.8	0.74	0.77	0.78	0.72	0.75	0.83	0.71	0.77
ذ	9	0.71	0.75	0.73	0.8	0.73	0.76	0.83	0.72	0.77	0.84	0.73	0.78
ر	10	0.87	0.89	0.88	0.86	0.91	0.88	0.95	0.88	0.91	0.88	0.89	0.89
ز	11	0.93	0.88	0.9	0.89	0.89	0.89	0.89	0.91	0.9	0.9	0.94	0.92
س	12	0.93	0.98	0.96	0.95	0.95	0.95	0.97	0.92	0.94	0.94	0.92	0.93
ش	13	0.96	0.94	0.95	0.98	0.94	0.96	0.98	0.96	0.97	0.96	0.96	0.96
ص	14	0.93	0.86	0.89	0.91	0.9	0.9	0.85	0.94	0.9	0.83	0.9	0.86
ض	15	0.95	0.87	0.91	0.92	0.9	0.91	0.91	0.91	0.91	0.92	0.86	0.89
ط	16	0.93	0.93	0.93	0.9	0.94	0.92	0.92	0.93	0.92	0.88	0.92	0.9
ظ	17	0.95	0.94	0.94	0.97	0.9	0.93	0.96	0.93	0.94	0.93	0.91	0.92
ع	18	0.81	0.88	0.84	0.88	0.85	0.87	0.86	0.9	0.88	0.81	0.84	0.82
غ	19	0.88	0.87	0.87	0.91	0.84	0.87	0.91	0.89	0.9	0.92	0.82	0.87
ف	20	0.83	0.87	0.85	0.83	0.87	0.85	0.85	0.87	0.86	0.78	0.87	0.82
ق	21	0.91	0.9	0.9	0.94	0.9	0.92	0.95	0.89	0.92	0.85	0.92	0.88
ك	22	0.87	0.91	0.89	0.91	0.89	0.9	0.92	0.92	0.92	0.9	0.89	0.9
ل	23	0.9	0.93	0.91	0.89	0.91	0.9	0.88	0.94	0.91	0.91	0.91	0.91
م	24	0.92	0.95	0.93	0.88	0.96	0.92	0.93	0.94	0.94	0.89	0.89	0.89
ن	25	0.8	0.86	0.83	0.78	0.88	0.82	0.83	0.89	0.86	0.85	0.81	0.83
ه	26	0.94	0.88	0.91	0.91	0.87	0.89	0.89	0.94	0.91	0.87	0.89	0.88
و	27	0.96	0.9	0.93	0.94	0.91	0.92	0.97	0.9	0.93	0.91	0.93	0.92
ي	28	0.96	0.94	0.95	0.93	0.93	0.93	0.95	0.95	0.95	0.94	0.93	0.93
ء	29	0.9	0.85	0.87	0.87	0.85	0.86	0.9	0.86	0.88	0.83	0.87	0.85
Accuracy (train)				0.9883			0.9911			**0.9938**			0.9666
Accuracy (test)				0.9178			0.9023			**0.9238**			0.9064

Loss (train)			0.0369			0.0273			**0.0206**			0.0935
Loss (test)			**0.4983**			0.5281			0.5801			0.5159
Macro avg	0.9	0.89	0.9	0.9	0.89	0.9	0.92	0.91	0.91	0.89	0.88	0.88
Weighted avg	0.9	0.9	0.9	0.9	0.9	0.9	0.92	0.92	0.92	0.89	0.89	0.89

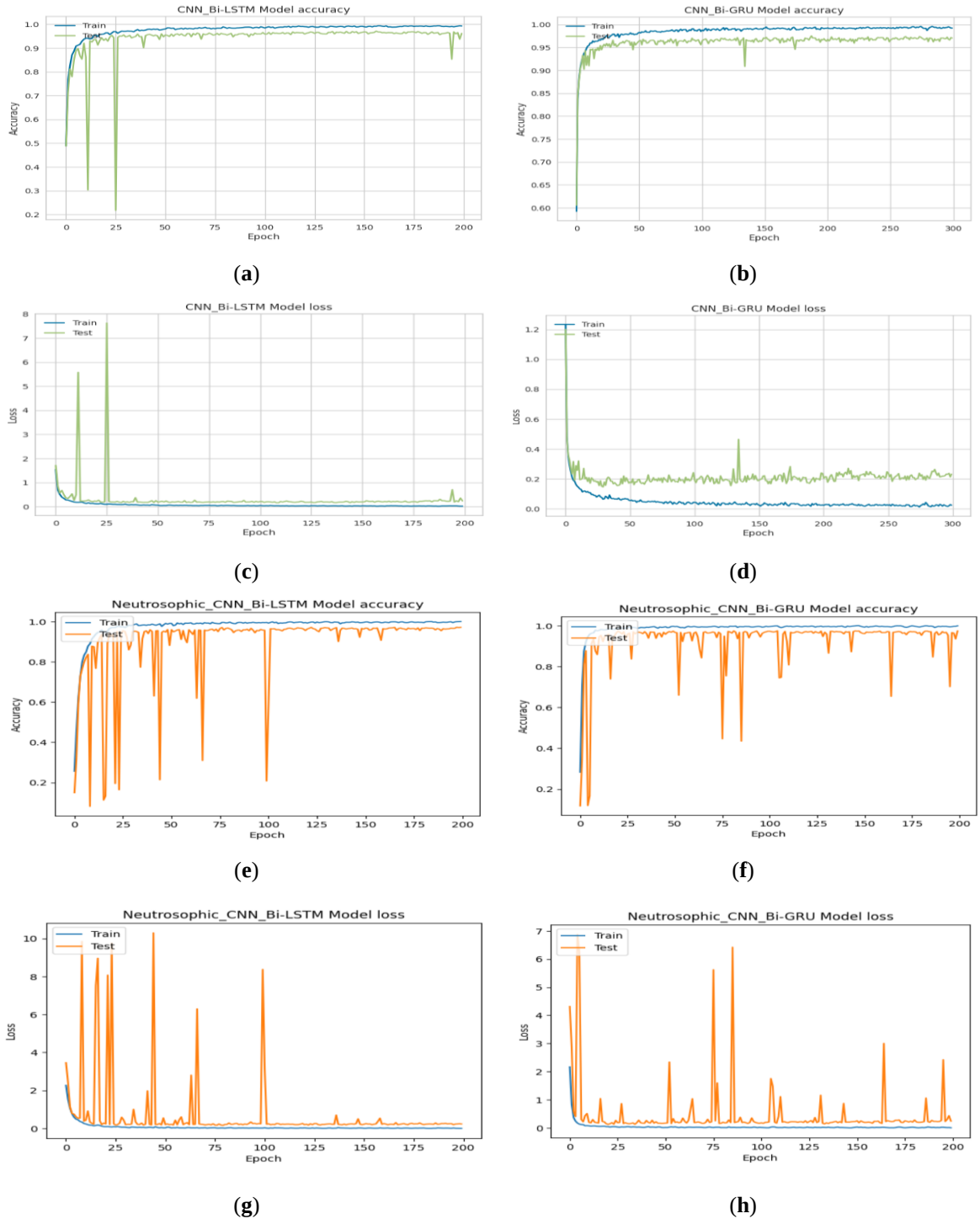

Figure 2. Model performance on the AHCL dataset: (a) Accuracy curves for CNN_Bi-LSTM during training and testing; (b) Accuracy curves for CNN_Bi-GRU during training and testing; (c) Loss curves for CNN_Bi-LSTM during

training and testing; (d) Loss curves for CNN_Bi-GRU during training and testing; (e) Accuracy curves for NS_CNN_Bi-LSTM during training and testing; (f) Accuracy curves for NS_CNN_Bi-GRU during training and testing; (g) Loss curves for NS_CNN_Bi-LSTM during training and testing; (h) Loss curves for NS_CNN_Bi-GRU during training and testing.

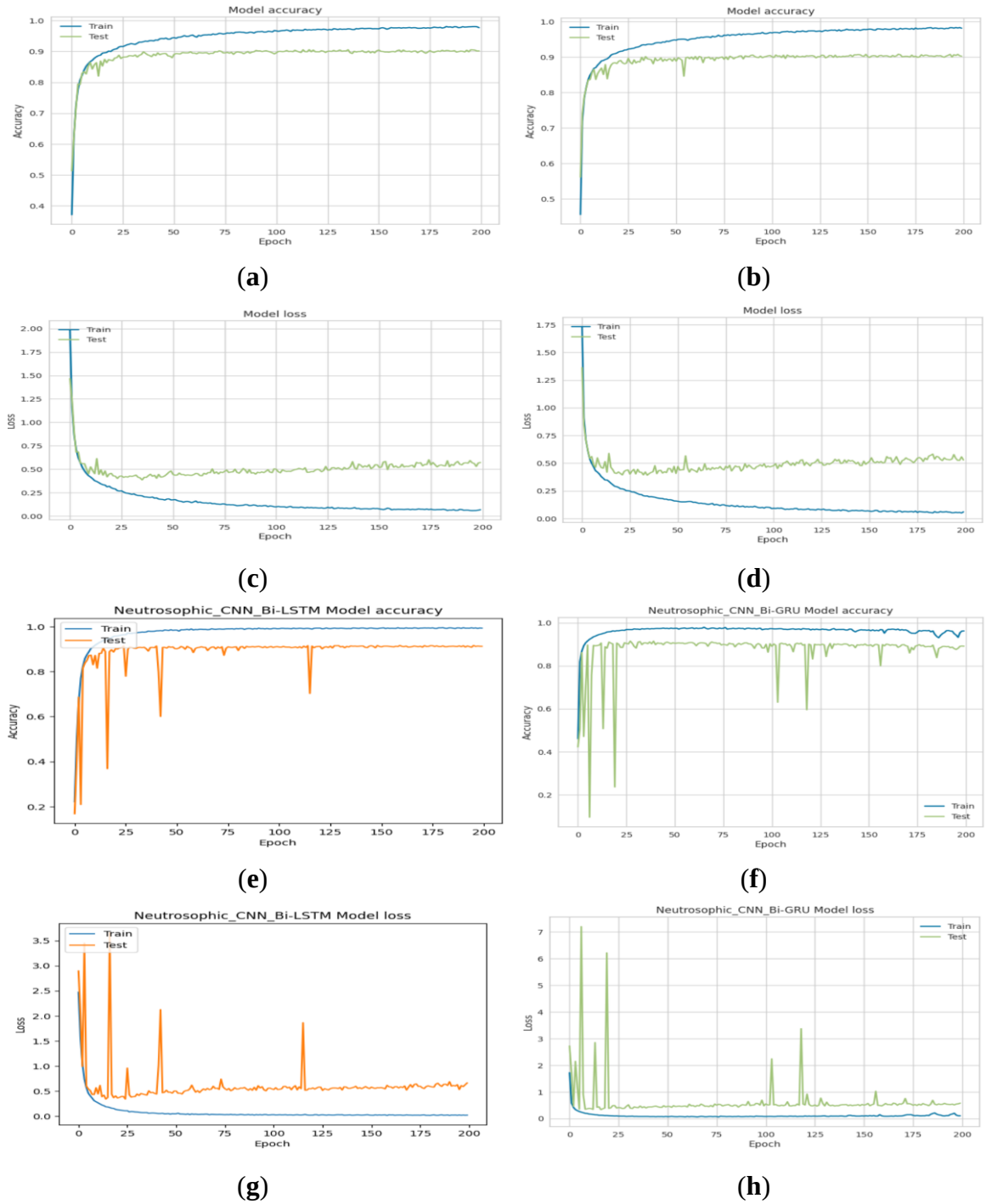

Figure 3. Model performance on the Hijjaa dataset: (a) Accuracy curves for CNN_Bi-LSTM during training and testing; (b) Accuracy curves for CNN_Bi-GRU during training and testing; (c) Loss curves for CNN_Bi-LSTM during

training and testing; (d) Loss curves for CNN_Bi-GRU during training and testing; (e) Accuracy curves for NS_CNN_Bi-LSTM during training and testing; (f) Accuracy curves for NS_CNN_Bi-GRU during training and testing; (g) Loss curves for NS_CNN_Bi-LSTM during training and testing; (h) Loss curves for NS_CNN_Bi-GRU during training and testing.

Table 3 compares the performance of the proposed CNN_Bi-LSTM, CNN_Bi-GRU, NS_CNN_Bi-LSTM, and NS_CNN_Bi-GRU models with other models in the literature on the Hijjaa and AHCD datasets. The NS_CNN_Bi-LSTM model achieves the highest accuracy on the Hijjaa dataset, with 99.38% during training and 92.38% on testing, outperforming most existing models. Similarly, the NS_CNN_Bi-GRU model excels on the AHCD dataset, achieving 99.90% training accuracy and 97.38% testing accuracy, demonstrating its superiority in both datasets. These models consistently deliver strong precision, recall, and F1-scores, making them highly effective for Arabic handwritten character recognition tasks.

Table 3. Performance comparison with models from the literature

References	Datasets		Accuracy	Precision	Recall	F1-Score
Altwaijry et al. [11]	Hijjaa		88%	88%	88%	88%
Nayef et al., [12]	Hijjaa		90%	-	-	-
Bin Durayhim et al. [23]	Hijjaa	Train	99.5%	99%	99%	99%
		Test	86.3%	85%	86%	85%
M.G. Mahdi et al. (LSTM[24]	Hijjaa	Train	96.2%	96%	96%	96%
		Test	84.6%	85%	84%	84%
M.G. Mahdi et al. (GRU)[24]	Hijjaa	Train	92.7%	92%	92%	92%
		Test	81.5%	82%	81%	81%
M.G. Mahdi et al. (Bi-LSTM) [24]	Hijjaa	Train	98.3%	98%	98%	98%
		Test	85%	85%	84%	84%
M.G. Mahdi et al. (Bi-GRU) [24]	Hijjaa	Train	97.3%	97%	97%	97%
		Test	84.6%	85%	85%	85%
CNN_Bi-LSTM	Hijjaa	Train	98.83%	98%	98%	98%
		Test	91.78%	91%	91%	91%
CNN_Bi-GRU	Hijjaa	Train	99.11%	99%	99%	99%
		Test	90.23%	90%	90%	90%
Proposed NS_CNN_Bi-LSTM	Hijjaa	Train	**99.38**	**99%**	**99%**	**99%**
		Test	**92.38%**	**92%**	**92%**	**92%**
Proposed NS_CNN_Bi-GRU	Hijjaa	Train	96.66%	96%	96%	96%
		Test	90.64%	90%	90%	90%
Altwaijry et al. [11]	AHCD		97%	97%	97%	97%
El-Sawy et al. [10]	AHCD		94.9%	-	-	-
Nayef et al., [12]	AHCD		99%	-	-	-
Bin Durayhim et al. [23]	AHCD	Train	98%	99%	99%	99%
		Test	95.5%	95%	95%	95%
M.G. Mahdi et al. (LSTM[24]	AHCD	Train	99.7%	99%	99%	99%

M.G. Mahdi et al. (GRU)[24]	AHCD	Test	94.9%	95%	95%	95%
		Train	99.8%	99%	99%	99%
		Test	93.1%	93%	93%	93%
M.G. Mahdi et al. (Bi-LSTM) [24]	AHCD	Train	99.8%	99%	99%	99%
		Test	94.9%	95%	95%	95%
M.G. Mahdi et al. (Bi-GRU) [24]	AHCD	Train	99.9%	99%	99%	99%
		Test	95.7%	96%	96%	96%
CNN_Bi-LSTM	AHCD	Train	99.37%	99%	99%	99%
		Test	96.16%	96%	96%	96%
CNN_Bi-GRU	AHCD	Train	99.78%	99%	99%	99%
		Test	97.05%	97%	97%	97%
Proposed NS_CNN_Bi-LSTM	AHCD	Train	99.99%	99%	99%	99%
		Test	97.14%	97%	97%	97%
Proposed NS_CNN_Bi-GRU	AHCD	Train	**99.9%**	**99%**	**99%**	**99%**
		Test	**97.38%**	**97%**	**97%**	**97%**

7. Conclusions

This research article introduces and evaluates two new deep learning models for Arabic handwriting recognition: CNN_Bi-LSTM and CNN_Bi-GRU. Additionally, the models were enhanced by integrating Neutrosophic sets (NS), resulting in two more models: NS_CNN_Bi-LSTM and NS_CNN_Bi-GRU. The models were designed to classify Arabic letters into 28 and 29 classes, with evaluations conducted using two datasets: AHCD and Hijjaa. The performance of the four models was compared among themselves and against existing methods in the literature.

The experimental results demonstrate the effectiveness of the proposed hybrid deep learning models in Arabic handwriting recognition. The NS_CNN_Bi-GRU model achieved a state-of-the-art accuracy of 97.38% on the AHCD dataset, while the NS_CNN_Bi-LSTM model reached 92.38% accuracy on the Hijjaa dataset, surpassing previous techniques. These results underscore the advantage of combining convolutional and bidirectional recurrent networks for capturing both spatial and temporal features in Arabic handwriting.

8. Limitation and Future work

The limitations of this research include the need for further improvements in the recognition accuracy of the hybrid models, particularly when handling noisy or complex Arabic handwriting. The models were only tested on two datasets (AHCD and Hijjaa), limiting the understanding of their generalization across more diverse datasets or real-world applications. Additionally, the focus was on overall accuracy, with little analysis of errors, failure cases, or the interpretability of the models' decisions.

To address these limitations, future research should explore alternative architectures, such as attention mechanisms or transformers, to enhance accuracy and robustness. More detailed component analysis, advanced data augmentation techniques, and evaluations on broader datasets could improve generalization. Real-world application testing in document processing and transfer learning to other languages or handwriting styles should also be considered. Efforts to optimize computational efficiency, model size, and inference speed would enable deployment in resource-constrained environments. Enhancing interpretability and expanding the scope to other Arabic language processing tasks could further extend the models' practical utility.

Conflicts of Interest: The author declares that they have no conflict of interest.

Funding: This research received no external funding.

Code Availability Statement: 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 .

References

1. Alghyaline, S. Arabic Optical Character Recognition: A Review. *CMES - Computer Modeling in Engineering and Sciences* 2023, *135*, 1825–1861.
2. Mahdi, M.G.; Sleem, A.; Elhenawy, I. Deep Learning Algorithms for Arabic Optical Character Recognition: A Survey. *Multicriteria Algorithms with Applications* **2024**, *2*, 65–79, doi:10.61356/J.MAWA.2024.26861.
3. Ghanim, T.M.; Khalil, M.I.; Abbas, H.M. Comparative Study on Deep Convolution Neural Networks DCNN-Based Offline Arabic Handwriting Recognition. *IEEE Access* **2020**, *8*, 95465–95482, doi:10.1109/ACCESS.2020.2994290.
4. Wajid, M.A.; Zafar, A.; Wajid, M.S. A Deep Learning Approach for Image and Text Classification Using Neutrosophy. *International Journal of Information Technology* **2024**, *16*, 853–859, doi:10.1007/s41870-023-01529-8.
5. Alanazi, A.A.; Abaker, A.O.I.; Abdel-Khalek, S.; Alhomayani, F.M.; Aripov, M. Neutrosophic Logic Empowered Machine Learning Algorithm with Salp Swarm Optimization for Biomedical Image Analysis. *International Journal of Neutrosophic Science* **2024**, *23*, 104–116, doi:10.54216/IJNS.230408.
6. Hassen, O.A.; Mashhadani, S.; Alhakam, I.; Darwish, S.M. A New Paradigm for Decision Making under Uncertainty in Signature Forensics Applications Based on Neutrosophic Rule Engine. *International Journal of Neutrosophic Science* **2024**, *24*, 267–282, doi:10.54216/IJNS.240224.
7. Ebtesam, E.; Al-Mansor, E. Al Leveraging Bat Algorithm with Rough Neutrosophic Soft Set for Enhanced Oral Cancer Detection and Classification. *International Journal of Neutrosophic Science* **2024**, *24*, 71–81, doi:10.54216/IJNS.240405.
8. A., A.; Sugapriya, P.; .. D.; .. S.S.; .. K.; Irudayam, F.N. Enhanced Brain Tumor Diagnosis Through Differential and Canonical Quadri –Partitioned Neutrosophic Set Classification Methods:A Comparative Study. *International Journal of Neutrosophic Science* **2024**, *24*, 277–292, doi:10.54216/IJNS.240420.
9. Straub, J.; Mohammed, H.; Kabir, D.; Alam, S.B.; Mondal, S.K.; Al-Saidi, M.; Ballagi, Á.; Hassen, O.A.; Saad, S.M. Cognitive Classifier of Hand Gesture Images for Automated Sign Language Recognition: Soft Robot Assistance Based on Neutrosophic Markov Chain Paradigm. *Computers 2024, Vol. 13, Page 106* **2024**, *13*, 106, doi:10.3390/COMPUTERS13040106.
10. El-Sawy, A.; Loey, M.; El-Bakry, H. Arabic Handwritten Characters Recognition Using Convolutional Neural Network. *WSEAS Transactions on Computer Research* **2017**, *5*, 11–19.

11. Altwaijry, N.; Al-Turaiki, I. Arabic Handwriting Recognition System Using Convolutional Neural Network. *Neural Comput Appl* **2021**, *33*, 2249–2261, doi:10.1007/s00521-020-05070-8.
12. Nayef, B.H.; Abdullah, S.N.H.S.; Sulaiman, R.; Alyasseri, Z.A.A. Optimized Leaky ReLU for Handwritten Arabic Character Recognition Using Convolution Neural Networks. *Multimed Tools Appl* **2022**, *81*, 2065–2094, doi:10.1007/s11042-021-11593-6.
13. Vinoth, D.; Devarasan, E. A Membership Based Neutrosophic Approach for Supervised Fingerprint Image Classification. *Neutrosophic Sets and Systems* **2023**, *60*, 420–445, doi:10.5281/ZENODO.10224226.
14. Elatawy, S.M.; Hawa, D.M.; Ewees, A.A.; Saad, A.M. Recognition System for Alphabet Arabic Sign Language Using Neutrosophic and Fuzzy C-Means. *Educ Inf Technol (Dordr)* **2020**, *25*, 5601–5616, doi:10.1007/S10639-020-10184-6/TABLES/3.
15. Smarandache, F. *Neutrosophy: Neutrosophic Probability, Set, and Logic: Analytic Synthesis & Synthetic Analysis*; American Research Press, 1998;
16. Ali, M.; Smarandache, F.; Vladareanu, L. Neutrosophic Sets and Logic. In *Emerging Research on Applied Fuzzy Sets and Intuitionistic Fuzzy Matrices*; Adak, A.K., Manna, D., Bhowmik, M., Eds.; IGI Global: Hershey, PA, USA, 2017; pp. 18–63 ISBN 9781522509141.
17. Zhang, X.; Li, M.; Lei, T. On Neutrosophic Crisp Sets and Neutrosophic Crisp Mathematical Morphology. *Neutrosophic Sets and Systems* **2021**, *43*, 1–11.
18. Borah, G.; Dutta, P. Fuzzy Risk Analysis in Crop Selection Using Information Measures on Quadripartitioned Single-Valued Neutrosophic Sets. *Expert Syst Appl* **2024**, *255*, 124750, doi:https://doi.org/10.1016/j.eswa.2024.124750.
19. Khatun, M.; Basu, T.M.; Manna, S.; Mondal, S.K. Extended Dombi–Hamy Mean Operator and Gaussian Likelihood-Based Ranking Approach for Solving MCDM Problems in Interval-Valued Neutrosophic Environment. *Expert Syst Appl* **2024**, *253*, 124272, doi:https://doi.org/10.1016/j.eswa.2024.124272.
20. Mahmood, A.; Abbas, M.; Murtaza, G. Multi-Valued Multi-Polar Neutrosophic Sets with an Application in Multi-Criteria Decision-Making. *Neutrosophic Sets and Systems* **2023**, *53*, 530–561, doi:10.5281/ZENODO.7536084.
21. Al-Hijjawi, S.; Ahmad, A.G.; Alkhazaleh, S.; Al-Hijjawi, S.; Ahmad, A.G.; Alkhazaleh, S. A Generalized Effective Neurosophic Soft Set and Its Applications. *AIMS Mathematics 2023 12:29628* **2023**, *18*, 29628–29666, doi:10.3934/MATH.20231517.
22. Sharma, M.; Kandasamy, I.; Vasantha, W.B. Emotion Quantification and Classification Using the Neutrosophic Approach to Deep Learning. *Appl Soft Comput* **2023**, *148*, 110896, doi:10.1016/J.ASOC.2023.110896.

23. Bin Durayhim, A.; Al-Ajlan, A.; Al-Turaiki, I.; Altwaijry, N. Towards Accurate Children's Arabic Handwriting Recognition via Deep Learning. *Applied Sciences 2023, Vol. 13, Page 1692* **2023**, *13*, 1692, doi:10.3390/APP13031692.
24. Mahdi, M.G.; Sleem, A.; Elhenawy, I.; Safwat, S. Advancing Arabic Handwriting Recognition with Convolutional and Recurrent Neural Network Ensembles. *Information Sciences with Applications* **2024**, *3*, 43–63, doi:10.61356/J.ISWA.2024.3294.

Received: May 10, 2024. Accepted: Sep 13, 2024.