A Deep Learning Approach to Convert Handwritten Arabic Text to Digital Form

Bayan N. Alshahrani, Wael Y. Alghamdi
Department of Computer Science, College of Computers and Information Technology,
Taif University, P.O.Box 11099, 21944 Taif, Saudi Arabia

***Abstract*—The recognition of Arabic words presents considerable difficulties owing to the complex characteristics of the Arabic script, which encompasses letters positioned both above and below the baseline, hamzas, and dots. In order to address these intricacies, we provide a structured approach for transforming handwritten Arabic text into a digital format. We employ a hybrid deep learning technique that combines Convolutional Neural Networks (CNNs), Bidirectional Long Short-Term Memory (BLSTM), and Connectionist Temporal Classification (CTC). We collected datasets that cover a wide range of Arabic text variations. We have also created a pre-processing pipeline. Our methodology successfully achieved an accuracy rate of 99.52%. At the level of recognizing the letters of the word, with an accuracy of 98.36% at the level of the full word. In order to evaluate the effectiveness of our suggested method for recognizing handwritten text, we utilize two essential metrics: Word Error Rate (WER) and Character Error Rate (CER) to compare its performance. The experimental research demonstrates a WER of 1.64 % and a CER of 0.48%.**

Keywords*—*Deep learning; convolutional neural networks; bidirectional long short term memory; connectional temporal classification; Arabic handwriting recognition

I. Introduction

The Arabic language, known for its intrinsic beauty and cultural significance, presents a unique challenge in the digital age. Its intricate handwritten script, defined by graceful curves and intricate loops, encapsulates centuries of wisdom, knowledge, and artistic expression [1]. However, converting handwritten Arabic text into a digital format poses a formidable task, given the language's complexity.

Arabic is celebrated for its distinctive calligraphy, which adds to the complexity of recognizing and converting handwritten text. Unlike Latin-based scripts, Arabic script is cursive, with characters that change shape based on their position within a word [2]. As shown in Fig. 1. Furthermore, Arabic words often feature descending letters, supra-line letters, and dots, making the recognition process exceptionally challenging.

Fig. 1. Examples of some arabic letters and their position.

A hybrid deep learning paradigm combining Convolutional Neural Networks (CNN), Bidirectional Long Short-Term Memory (BLSTM), and Connectional Temporal Classification (CTC) is emerging as a critical solution to this complex problem. CNN captures local spatial relationships to extract significant characteristics from Arabic handwritten text. The BLSTM component incorporates past and future contexts to model temporal dependencies and contextual information. This lets the model reflect Arabic handwriting's sequential nature, where character shapes change based on their location in a word or sentence. By aligning predicted feature sequences with ground truth labels, the CTC component allows end-to-end training and sequence alignment.

In a world where the Arabic language plays a vital role beyond linguistic communication, the importance of preserving and accessing handwritten texts is undeniable. Documents, letters, and literary works are gateways to knowledge, culture, and identity. As these texts age and grow more fragile, the urgency to preserve them intensifies [3].

In addition, the hybrid model of Deep Learning to convert handwritten Arabic text to digital form is poised to bridge the gap between the enduring legacy of the Arabic language and the boundless possibilities of the digital era. As we embark on this journey, we recognize the importance of preserving the treasures of the Arabic script, making them accessible to the world, and ensuring that the beauty of the language endures in the digital age.

The significance of this mission lies not only in its technological complexity but in the cultural responsibility it carries. The Arabic language, with its unique script and intricate calligraphy, has been a symbol of beauty and sophistication. However, as we embrace the digital age, we are faced with the challenge of transforming handwritten Arabic text into a digital format.

The problem stems from the intrinsic complexities of Arabic script, which is a cursive and context-dependent writing system. Handwritten Arabic text exhibits significant variations in writing styles, ligatures, and contextual letter forms. Arabic handwriting varies greatly between individuals, encompassing diverse writing styles, character shapes, and ligature formations, which can confound traditional OCR systems.

To tackle this problem, the development of a hybrid deep learning model that combines CNN, BLSTM, and CTC is essential. Hybrid deep learning has demonstrated success in various tasks, making it a promising approach for Arabic handwriting conversion.

Ultimately, this work not only adds to the conservation and acknowledgment of Arabic script, but also carries significant cultural and technological importance. It facilitates progress in the field of deep learning models and difficult language recognition, while also connecting the historical practice of handwritten Arabic text with the modern digital era. As we begin this endeavor, we acknowledge the significance of safeguarding the valuable elements of the Arabic script, enabling their availability to the global community, and guaranteeing the longevity of the language's elegance in the era of digital technology.

In this paper, it is organized as follows: Section II provides the background information. In addition, Section III summarizes related work. Following Section IV, where the methodology is presented, Section V showcases and reviews the experimental settings and outcomes of the methods. Finally, in Section VI, the conclusion and propositions for future work are made in Section VII.

II. Background

This section provides the essential background information required to explain the main concepts of this study, including Arabic script and Arabic language, handwriting recognition, the Convolutional Neural Network, the Bidirectional Long Short-Term Memory, and Connectional Temporal Classification.

A. Arabic Script and Arabic Language

Arabic, one of the world's most ancient and rich languages, boasts a script that holds a unique position in the tapestry of global written languages. It is not just a means of communication but a symbol of cultural heritage, religious significance, and historical depth [4].

The Arabic script is characterized by its distinctive right-to-left writing direction and an intricate system of connecting letters. Its letters change shape depending on their position within a word, as shown in Fig. 2, adding a layer of complexity that has fascinated linguists and calligraphers alike. Arabic calligraphy, with its artistic and aesthetic value, has been revered as a form of visual art for centuries [1].

The Arabic language itself is a linguistic marvel, known for its eloquence and precision. It is the language of the Quran, the holy book of Islam, and plays a central role in the spiritual lives of millions worldwide. Arabic is also the native tongue of over 400 million people, making it one of the most widely spoken languages globally. Beyond its spiritual and regional importance, the Arabic language is vital for conducting business, academic research, and fostering cultural understanding in the Arab world [6].

This study delves into the realm of Arabic script and Arabic language, exploring their intricacies and significance, particularly in the context of handwriting recognition. Understanding the challenges and nuances of Arabic script and language is a fundamental step toward developing effective recognition models and enhancing their performance. Whether the goal is to protect cultural heritage, improve human-computer interaction, or broaden the reach of artificial intelligence.

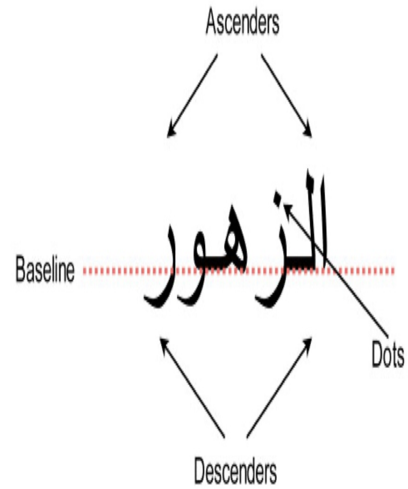

Fig. 2. Example of baseline, ascenders, and descenders in arabic script[5].

B. Handwriting Recognition

Handwriting recognition is a computer vision problem that pertains to the automation of script identification by a computer. This is achieved by converting the text from various sources, including touchscreens and documents, into a format that is comprehensible to machines. The input image may be obtained offline, from a material object like a photograph or sheet of paper, or online, from a digital source like touchscreens [7], as shown in Fig. 3.

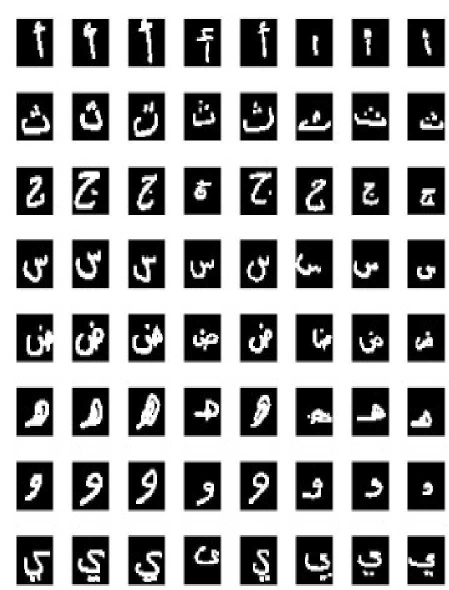

Fig. 3. Example of handwriting arabic dataset[8].

Traditional methods of handwriting recognition relied on features like edge detection, contour analysis, and statistical methods, which often required extensive feature engineering and were not as effective in handling variations in handwriting styles [9].

Deep learning, especially CNN and recurrent neural networks (RNN), revolutionized handwriting recognition by allowing the system to learn intricate patterns and representations from the raw input data, such as images of handwritten text [10].

C. Convolutional Neural Network

A CNN is a class of deep learning artificial neural networks designed specifically for tasks related to pattern recognition in images, videos, and other grid-like data structures. CNN have become the go-to architecture for image-related tasks and have demonstrated remarkable performance in various computer vision applications. Often referred to as ConvNets or Convolutional Networks, have revolutionized the field of computer vision and have extended their impact into various other domains [11].

In the digital age, our world is inundated with images, from security cameras and medical scans to social media photos. Extracting meaningful information from these images is a complex task. This is where CNN come into play. They have an innate ability to understand and recognize visual patterns, making them indispensable for applications like image classification, object detection, facial recognition, and even in emerging technologies like autonomous vehicles [12].

The CNNs involve the application of different hidden layers, each serving a specific purpose. The neural network typically consists of three primary neural layers: convolution layers, pooling layers, and fully connected layers. Each layer has a distinct function and transforms the input volume into an output neural activity volume [11], as shown in Fig. 4.

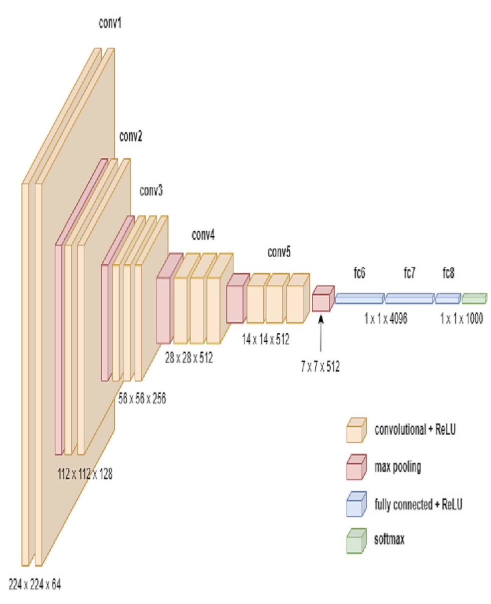

Fig. 4. CNN Components[13].

D. Bidirectional Long Short-Term Memory

An example of a neural network model used for training sequential data is the BLSTM. The system employs two separate Long Short-Term Memory LSTM models: a forward LSTM processes the input sequence in a left-to-right manner, while a backward LSTM processes the sequence in a right-to-left manner. The LSTM model was initially introduced in [14]as a solution to address the issue of gradient vanishing. The BLSTM model was proposed as a means to extract high-level features from a sequence of input features. Moreover, the BLSTM networks expand upon the LSTM by incorporating an additional layer, wherein the connections between hidden and concealed layers occur in a reverse temporal sequence. The model possesses the capability to alter both previous and future data.

E. Connectional Temporal Classification

The CTC is configured to label sequences without segmenting the input. Basically, the CTC is a softmax layer that generates probabilities at each step based on the length of the probability sequence given into it, which is T. This sequence represents all the potential label alignments in the input sequence. It complicates matters in two ways: 1) The loss value is calculated during training using both the BLSTM output matrix and the ground truth text; 2) the predicted text is provided during inference using only the output matrix. The CTC was initially presented in [15].

III. Related Work

In the literature, Deep learning models have demonstrated superior performance compared to traditional machine learning approaches on a range of handwritten recognition datasets, establishing themselves as the current leading approach.

CCNs demonstrated high efficacy in Arabic Handwritten Recognition , especially when dealing with datasets containing characters and diverse writing styles, this is shown in a study [7] and [16] and [17].

Nayef et al.[18] developed a novel convolutional neural network (CNN) architecture with an improved leaky ReLU activation function for recognizing handwritten Arabic characters. On four separate datasets, this design achieved accuracy rates over 99%, significantly higher than those of previously used methods.

Albattah et al. [8] developed and evaluated HCR deep learning and hybrid models. The hybrid models used deep learning feature extraction and machine learning classification.The best results were obtained.Where hybrid models that integrate machine learning and deep learning methods can also yield favorable outcomes on handwritten recognition problems this is shown in [19]

Transfer learning can enhance the efficacy of deep learning models on handwritten recognition problems as in a study [20] and [21]and [22].

In addition the [5] study showed that data augmentation is a powerful method that may be used to tackle class imbalance and enhance the overall performance of deep learning models by increasing their ability to generalize. . On the other hand, studies conducted in [23], [24], and [25]indicates that deep learning has great potential as a method for recognizing handwritten writing and transcribing music scores. Deep learning models have the capability to be trained in order to acquire intricate patterns in handwritten data, even when there is noise and unpredictability present. This renders them very suitable for jobs such as identifying antiquated and deteriorated handwritten papers and musical scores. Nevertheless, there remain certain obstacles that require attention and resolution.An obstacle is in the scarcity of extensive and top-notch datasets required to train these models.

IV. Methodology and Approach

This section starts with an overview of the approach to discovering handwritten text that includes complex words with ascending and descending letters and periods. Then, presents the data collection and data pre-processing steps. Afterward, includes the details of the model architecture.

A. Methodology Overview

In this paper, an approach to converting handwritten Arabic text into digital form using a hybrid deep learning model that combines CNN, BLSTM, and CTC is proposed. The approach involves several essential steps. First, the handwritten text dataset is collected, augmented, and preprocessed to prepare the data for digital conversion. Subsequently,The CNN was utilized to extract sequence features from the input photos. Moreover, the BLSTM is employed to transmit information within this sequence. It generates a matrix of character scores for each element in the sequence. The CTC procedure is established to compute the loss value for training the proposed model and to carry out the inference during this phase. Finally, the CTC algorithm reads the BLSTM output matrix to figure out what text is in the image that it is given. The presence of these two interconnected networks within the CTC enables the recognition of words at the level of individual words without the need for segmenting characters. The project's primary objective is to effectively transform handwritten Arabic text into a digital representation, ensuring accuracy and legibility. Fig. 5 illustrates the methodology framework for this project, showcasing the steps involved. Additionally, our experiments included evaluating the performance of the model and the quality of the digital output to ensure that it accurately reflects the original handwritten text.

B. Dataset

The study included the collection of data pertaining to handwritten Arabic words. The data, obtained from a sample of 30 adult participants, was manually transcribed using 60 words. The word count reached 1800. The Riyadh Dictionary, published by the King Salman Academy for the Arabic Language [26], is where the words are all in Arabic. The word set has a diverse range of intricacy, encompassing attributes such as hamzas, ascending or descending characters, and dots. In addition, Two datasets were merged for this study: the ADAB dataset introduced by Boubaker et al. [27] and the AHAWP dataset provided by Khan [28]. The finalized dataset had files in CSV format with 15009 entries.The complete dataset was divided into two subsets: one for training and one for testing.

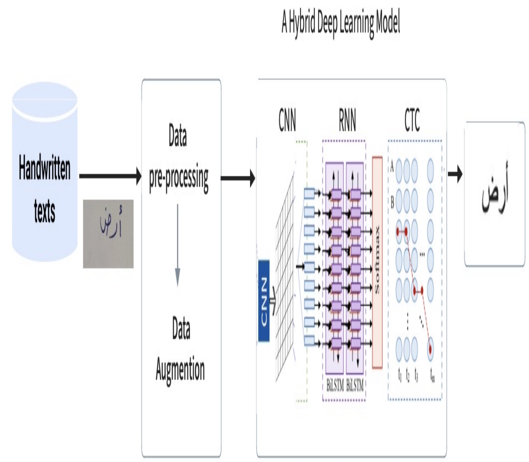

Fig. 5. Methodology framework.

C. Data Preprocessing

Prior to training the models using the dataset, several preprocessing and data augmentation techniques were applied to the data to enhance its compatibility with the models and increase its resilience to real-life scenarios. Data preprocessing is a critical component in the development of deep learning models, particularly in tasks involving complex data types such as Arabic text. This part presents a proposed data preprocessing pipeline specifically designed for Arabic text recognition tasks. The pipeline encompasses multiple stages, including filtering Arabic text, preprocessing text labels, resizing images, and encoding image labels. Python libraries such as TensorFlow and OpenCV are leveraged for efficient data manipulation. Additionally, the part provides comprehensive insight into the size of images, character encoding, and other pertinent details of the preprocessing steps. Fig. 6 shows the proposed preprocessing pipeline.

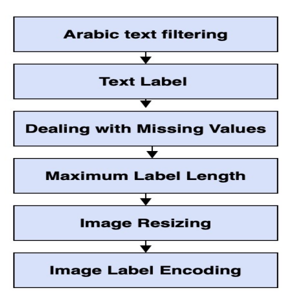

Fig. 6. Proposed preprocessing pipeline.

Arabic text filtering: The raw text data undergoes a filtering process to eliminate non-Arabic text. This process utilizes regular expressions to detect and preserve exclusively Arabic text that falls inside the Unicode range [0̆600-0̆6FF]+, which includes Arabic script and whitespace characters.

Text Label Preprocessing: Text labels are subjected to a variety of preprocessing procedures in order to guarantee the integrity and consistency of the data.Extraneous spaces surrounding text labels are removed to standardize label layout and prevent conflicts during processing.

Dealing with Missing Values: Rows containing missing or NaN labels are eliminated from the dataset to ensure data integrity and prevent errors in the following phases.

Maximum Label Length: Labels that surpass a certain maximum length, usually established at 12 characters, are eliminated in order to control computational complexity and ensure equilibrium in the dataset.

Image Resizing:Image resizing involves adjusting the dimensions of the image data to meet a defined size that is appropriate for input into a model. The process of resizing entails the subsequent steps:

Aspect Ratio Preservation: Images are scaled to a specified width and height while maintaining their original aspect ratio. This guarantees that the resized photos retain their dimensions and avoid any distortion.Target Dimensions: The target dimensions for scaled images are defined as a width of 256 pixels and a height of 64 pixels. The interpolation method used for resizing is OpenCV's INTER AREA. This method is particularly suitable for reducing the size of images and effectively maintaining image sharpness and detail.

Image Label Encoding: Image labels are encoded into integer representations using TensorFlow's StringLookup capability, similar to how text labels are encoded. The encoding procedure guarantees uniformity in data representation across text and visual modalities, enabling effortless incorporation into deep learning pipelines.

D. Data Augmentation

Data augmentation was used to enhance the quantity and variety of photos, allowing the dataset to be expanded without the need to acquire additional data [29]. The CustomDataGenerator class in this project incorporates data augmentation techniques. The code employs the ImageDataGenerator class from the Keras package, which offers a straightforward means to apply various data augmentation techniques to the photos. The employed techniques included:

Rotation: The photos underwent a rotation of a specific angle by utilizing the rotation range parameter in ImageDataGenerator. This facilitates the model's ability to discern things from various viewpoints.

Rescaling: The photos underwent resizing to various dimensions by utilizing the zoom range parameter in ImageDataGenerator. This aids the model in generalizing to objects with different scales.

Shear: The photos underwent a shearing treatment by utilizing the shear range parameter in ImageDataGenerator. This feature introduces distortions and enhances the model's ability to process objects with diverse geometries.

E. Model Architecture

The architecture of our proposed model is specifically built for the recognition of Arabic text in handwriting. Employing a hybrid deep learning architecture that integrates multiple components in order to attain precise handwriting recognition. The process commences with input layers that receive the input images and target labels utilized for training purposes. A reshape layer prepares the data, a dense layer encodes features, bidirectional LSTM layers model sequences, and a final output layer predicts class probabilities. The design is made up of convolutional layers that extract features. The training of the model is conducted via the Connectionist Temporal Classification (CTC) loss function. Fig. 7 displays the proposed architecture.

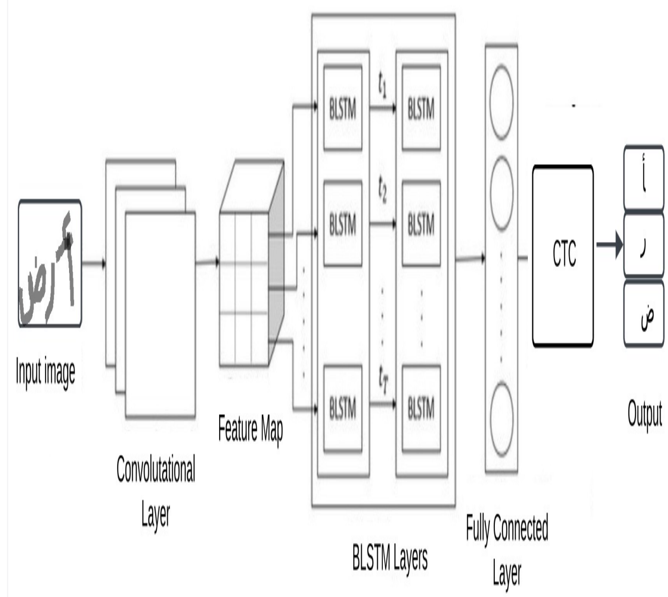

Fig. 7. Proposed model architecture.

1) Input layers: The model's "input images" layer is responsible for receiving the input images used for handwriting recognition. On the other hand, the "input labels" layer is designed to accept the target labels that correspond to the input images during the training phase.

2) Convolutional layers: The model contains three sets of convolutional layers, which are crucial for extracting information from the input images. Every set is accompanied by a maximum pooling layer. The number of filters progressively grows across the sets, beginning with 32 filters in the first set, followed by 64 filters in the second set, and ultimately reaching 128 filters in the third set. The convolutional layers employ a kernel size of 3x3, apply the ReLU activation function to introduce non-linearity, and leverage the He normal initialization method to effectively capture hierarchical features.

3) Reshape layer: The reshape layer in the model is tasked with converting the output of the convolutional layers into a format that is appropriate for subsequent processing. The function does a transformation on the output tensor, modifying its dimensions in terms of height and width. More precisely, it transforms the tensor to have a height equal to one-eighth of the input shape [0] and a width equal to the product of one-eighth of the input shape[1] and 128. The purpose of this reshaping stage is to prepare the data for the succeeding dense layer in the model.

4) Dense layer (Encoding Stage): The dense layer in our model has a vital function in transforming the retrieved characteristics into a condensed representation. The system consists of 64 units that serve as a bottleneck, capturing the most crucial characteristics. The dense layer utilizes the Rectified Linear Unit (ReLU) activation function, which introduces non-linearity and enhances the model's ability to effectively learn intricate patterns. The dense layer's weights are initialized using the He normal initialization method, which promotes a steady and efficient learning process. In order to address overfitting more effectively, a dropout layer is implemented after the dense layer. This dropout layer has a dropout rate of 0.4, which means that during training, a portion of the layer's outputs are randomly set to zero. This helps improve the model's ability to generalize and reduces its dependence on specific characteristics.

5) Bidirectional LSTM Layers (Decoding Stage): The bidirectional LSTM layers of the model are tasked with capturing the sequential relationships inherent in the encoded features. The model incorporates two stacked bidirectional LSTM layers to effectively capture both preceding and subsequent information concurrently. The LSTM layer comprises 128 units, facilitating the model's ability to proficiently capture long-range dependencies within the data. In order to mitigate overfitting and enhance generalization, a dropout rate of 0.25 is implemented on the LSTM layers. During training, the dropout rate is applied to randomly deactivate a portion of the layer's outputs. This technique helps decrease the model's dependence on specific patterns and enhances its capacity to generalize to new, unseen data.

6) Final output layer: The final output layer in the model is accountable for producing the anticipated probabilities for every class label, encompassing a distinct empty label. The layer is densely packed with num classes+1 units, where num classes indicates the total number of unique character classes. The softmax activation function is utilized to guarantee that the predicted probabilities aggregate to 1, hence rendering them interpretable as probabilities of different classes.

7) CTC Loss layer: Our model employs the CTC loss function during training. The CTC loss layer, which is implemented using the proprietary CTCLayer class, computes the CTC loss by comparing the anticipated output with the input labels. This loss function incorporates the changeable alignment between the input and target sequences, enabling the model to properly handle sequences of varied lengths.

V. Results and Discussion

The section starts by showing the experimental settings. Afterwards,The practical implementation details of the experiments were discussed.Finally,the results of the model used in the approach were analyzed.

A. Experimental Settings

In this part, the experimental parameters that were used to train the hybrid deep learning model are detailed. It goes over the optimization technique, training settings, and hyperparameters used for training.

1) Important parameters: Here are the hyperparameters that were used throughout the experiment:

Rate of Learning: A learning rate of 0.001 is used with the Adam optimizer.

Rate of Dropout: Overfitting may be reduced with the use of dropout regularization. The encoding layer has a dropout rate of 0.4, while the LSTM layers use a rate of 0.25.

Batch Size: In order to handle several samples at once, the training data is separated into batches. The training set uses a batch size of 128 whereas the testing set uses 64. The number of full iterations across the training dataset is 120 epochs, which is how long the model is trained.

Calculus of Loss: Loss function CTC (Connectionist Temporal Classification) is used.

2) Configuration for training: The configuration of the training procedure includes the following settings:

Data Generator: Both the training and testing sets use custom data generators. The generators in question are responsible for managing the loading and preparation of data in batches, hence guaranteeing optimal memory use throughout the training process.

Termination at an early stage: In order to mitigate the issue of overfitting and identify the most optimum model, the technique of early halting is used, with a duration of 10 epochs. If the validation loss does not improve for 10 consecutive epochs, the training will be terminated.

Model checkpointing: is a technique used to preserve the optimal weights throughout the training process. The weights of the model that exhibit the lowest validation loss will be preserved for further use or assessment.

Decrease in Learning Rate: The ReduceLROnPlateau callback is used to apply a technique for reducing the learning rate. If the validation loss does not improve for 5 consecutive epochs, the learning rate is decreased by a factor of 0.5. This dynamic modification aids in refining the model's performance.

Optimization:- The Adam optimizer is used for model optimization, with a learning rate of 0.001. The Adam method is well recognized for its ability to adjust the learning rate on a per-parameter basis, resulting in enhanced convergence speed and improved overall performance.

B. Implementation

The experimental implementations of all models are trained using the Google Colaboratory environment on a NVIDIA GEFORCE 64-bit computer with an Intel (R) Core (TM) i7-8565U CPU at 1.80 GHz and 1.99 GHz.

C. Evaluation Methods

In order to evaluate the performance of our model, we will compute the word-level accuracy Rate and Charcter-level Accuracy for the train and test sets of the dataset.

1) Word-level accuracy : Focuses on word-level accuracy: It evaluates the percentage of words that are successfully recognized by the system.The formula for WAR is given below Eq. 1:

$$WAC = \frac{Number of correctly recognized words}{Total number of words} \quad (1)$$

2) Character_level accuracy (CAC): Concentrates on accuracy at the character level; specifically, it evaluates the proportion of characters that the system adequately recognizes..The formula for CAR is given below Eq. 2:

$$CAC = \frac{Number of correctly recognized characters}{Total number of characters} \quad (2)$$

D. Results

The deep learning approach, which integrated CNN, BLSTM, and CTC, was implemented on a dataset consisting of 15009 handwritten Arabic words. This dataset was divided into two subsets: a training set and a test set. The model was assessed using accuracy measure.

The results indicate the model achieved a good level of precision in word identification, achieving an accuracy rate of 98.36%. This demonstrates the model's capability to precisely identify Arabic words written by hand. In the context of character recognition The model's ability to detect Arabic characters is indicated by an accuracy rate of 99.52%.

To better understand the model's performance and convergence, representations of the training and validation losses throughout the epochs were created. The CTC loss trended downward in the graphs, showing that the models learnt well from the data and became better in making predictions over time. These visuals validated that our models were properly trained as seen in Fig. 8, 9, 10.

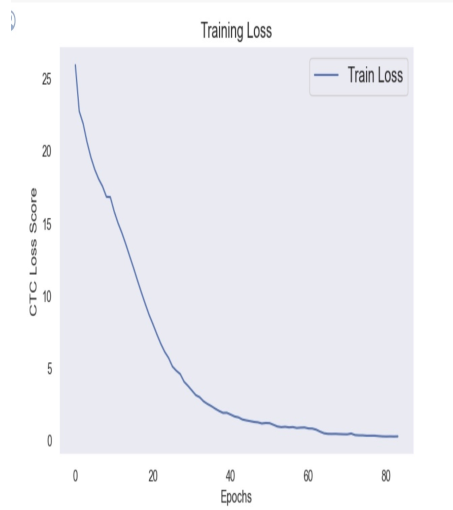

Fig. 8. Training loss.

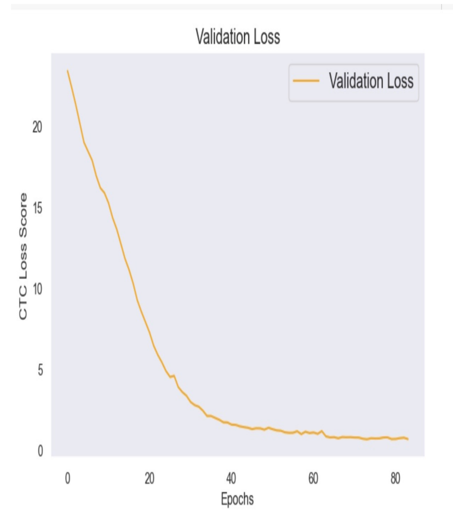

Fig. 9. Validiation loss.

As shown in Fig. 8, the graph showing the training loss of a hybrid deep learning model for transforming handwritten Arabic into digital form exhibits a constant and continuous decline over a span of 80 epochs. At the beginning, the CTC loss score is rather high, approximately 25. However, it quickly diminishes within the first 20 epochs, suggesting efficient early learning. As the training progresses, the decrease in loss becomes more slow but consistent, indicating continuous enhancements in model performance. During the last stages, the loss reaches a low value and remains constant, indicating that the model has achieved successful convergence. This evolution showcases the model's aptitude for efficient learning and mistake reduction during the training process.

As shown in Fig. 9.The graph demonstrating the validation loss for the handwriting recognition model demonstrates a noticeable and constant decline during 80 epochs, indicating successful acquisition of knowledge and ability to apply it to new examples. At the start, the CTC loss score is approximately 22, but it rapidly decreases within the first 20 epochs, indicating significant progress. As the training continues, the decrease becomes less steep but still consistent, indicating a continuous improvement of the model. During the last stages, the validation loss reaches a low value and remains constant, indicating that the model has successfully converged and the learning process has been effective. This pattern highlights the model's capacity to effectively apply its knowledge to unfamiliar data, validating its resilience and precision in transforming handwritten Arabic into digital form. In Fig. 10, the graph depicting the training and validation losses of the hybrid deep learning model demonstrates a distinct and triumphant learning path. At first, the losses reduced significantly, indicating a rapid acquisition of knowledge. As the training continues, the rate of decrease in performance slows. However, the model consistently maintains a close alignment between the losses observed during training and validation, which suggests that it is capable

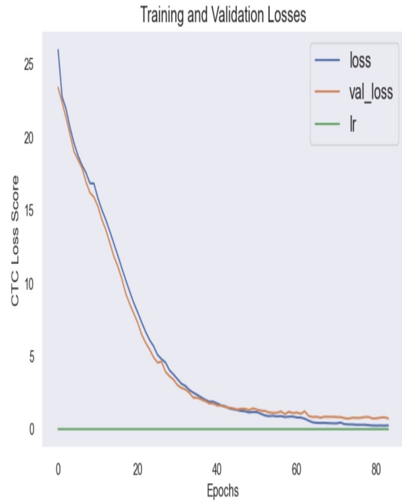

Fig. 10. Training and validiation losses.

of generalizing well. Ultimately, both losses reach a stable point at low values, indicating convergence and successful learning. The consistent learning rate facilitates a seamless and steady training procedure, highlighting the model's capacity to effectively execute handwriting recognition.

Additionally, a comparison test was conducted using two main metrics:word error rate (WER) and character error rate (CER) to assess the effectiveness of our suggested method for recognizing handwritten text and to highlight the mistakes made during recognition.In result we found that the WER was 1.64%, and the CER was 0.48%. The low error rates seen in the sample serve as evidence of the model's capacity to accurately detect words and characters.

In overall,the study's model clearly demonstrates a high level of accuracy in the conversion process, both at the word and character levels, based on the obtained results. Regrettably, the recognition process identified a few errors. Various factors, including different fonts and the complexity of the written language with dots and letters above and below the line, contribute to the errors encountered in the recognition process. There were some errors that occurred, which resulted in character substitutions that impacted the overall accuracy of the recognition process.

The current findings are quite promising, suggesting that the handwriting recognition model utilized in the study is both efficient and effective. Nevertheless, there is potential for enhancement by rectifying the specific errors and further refining the model through additional training and improvements.

VI. Conclusion

The Arabic language has additional complexities in the realm of deep learning due to its unique alphabet, which often presents issues when converting it into digital format. This study will aim to present a method for recognizing Arabic handwritten words using Hybird Deep Learning. The results have shown good performance based on the metrics used. This endeavor will contribute to the conservation of linguistic diversity while also propelling the progress of Arabic language processing.

VII. Recommendations and Future Works

As part of future tasks, the intention is to validate the model in an actual application as a proof of concept. This will help determine the feasibility of applying these experiments to real-world scenarios. Specifically, when employing the model in an online writing application, the model is provided with a distinct input type, with the expectation that it will excel in this scenario due to the superior clarity of images derived from online handwriting compared to offline images. This enhanced clarity facilitates the model's ability to readily identify distinctive attributes.Ultimately, anticipations were made that the achieved results could find application in various engineering fields. These tasks encompass improving systems for human-computer interaction, creating robots that can interpret handwritten text, integrating our model into assistive technologies for individuals with learning disabilities, and converting sketches and annotations into digital format for computer-aided design. However, as attention is directed towards the intricate patterns of handwriting, the potential applications of this model in educational apps for teaching dictation will become a compelling use case for this research.Furthermore, this work will be a valuable contribution to the wider domain of artificial intelligence and machine learning, serving as a source of inspiration for researchers to address other intricate challenges, such as sentence or language recognition. This will exemplify the extensive influence and practicality of the research across various engineering disciplines.

References

[1] M. H. Bakalla, *Arabic culture: through its language and literature*. Taylor & Francis, 2023.

[2] T. Shamma and M. Salama-Carr, *Anthology of Arabic Discourse on Translation*. Routledge, 2021.

[3] H. Hmoud, F. Shishan, Z. Qasem, and S. Bazi, "The effect of arabic language type on banking chatbots adoption," *Heliyon*, 2023.

[4] M. H. Bashir, A. M. Azmi, H. Nawaz, W. Zaghouani, M. Diab, A. Al-Fuqaha, and J. Qadir, "Arabic natural language processing for qur'anic research: A systematic review," *Artificial Intelligence Review*, vol. 56, no. 7, pp. 6801–6854, 2023.

[5] M. Eltay, A. Zidouri, and I. Ahmad, "Exploring deep learning approaches to recognize handwritten arabic texts," *IEEE Access*, vol. 8, pp. 89 882–89 898, 2020.

[6] S. Alghyaline, "Arabic optical character recognition: A review." *CMES-Computer Modeling in Engineering & Sciences*, vol. 135, no. 3, 2023.

[7] M. Alheraki, R. Al-Matham, and H. Al-Khalifa, "Handwritten arabic character recognition for children writing using convolutional neural network and stroke identification," *Human-Centric Intelligent Systems*, pp. 1–13, 2023.

[8] W. Albattah and S. Albahli, "Intelligent arabic handwriting recognition using different standalone and hybrid cnn architectures," *Applied Sciences*, vol. 12, no. 19, p. 10155, 2022.

[9] M. Ghanim, A. Mohammed, and A. Sali, "Arabic/english handwritten digits recognition using mlps, cnn, rf, and cnn-rf," *Al-Rafidain Engineering Journal (AREJ)*, vol. 28, no. 2, pp. 252–260, 2023.

[10] N. Altwaijry and I. Al-Turaiki, "Arabic handwriting recognition system using convolutional neural network," *Neural Computing and Applications*, vol. 33, no. 7, pp. 2249–2261, 2021.

[11] O. Moutik, H. Sekkat, S. Tigani, A. Chehri, R. Saadane, T. A. Tchakoucht, and A. Paul, "Convolutional neural networks or vision transformers: Who will win the race for action recognitions in visual data?" *Sensors*, vol. 23, no. 2, p. 734, 2023.

[12] L. V. Haar, T. Elvira, and O. Ochoa, "An analysis of explainability methods for convolutional neural networks," *Engineering Applications of Artificial Intelligence*, vol. 117, p. 105606, 2023.

[13] M. T. Ahad, Y. Li, B. Song, and T. Bhuiyan, "Comparison of cnn-based deep learning architectures for rice diseases classification," *Artificial Intelligence in Agriculture*, vol. 9, pp. 22–35, 2023.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[16] M. Kamal, F. Shaiara, C. M. Abdullah, S. Ahmed, T. Ahmed, and M. H. Kabir, "Huruf: An application for arabic handwritten character recognition using deep learning," in *2022 25th International Conference on Computer and Information Technology (ICCIT)*. IEEE, 2022, pp. 1131–1136.

[17] A. Alsayed, C. Li, A. Fat, AohAlalim, M. Abdalsalam, and Z. Obied, "Arabic handwritten character recognition using convolutional neural networks," 2023. [Online]. Available: https://doi.org/10.21203/rs.3.rs-3141935/v1

[18] B. H. Nayef, S. N. H. S. Abdullah, R. Sulaiman, and Z. A. A. Alyasseri, "Optimized leaky relu for handwritten arabic character recognition using convolution neural networks," *Multimedia Tools and Applications*, pp. 1–30, 2022.

[19] M. El Mamoun, "An effective combination of convolutional neural network and support vector machine classifier for arabic handwritten recognition," *Automatic Control and Computer Sciences*, vol. 57, no. 3, pp. 267–275, 2023.

[20] M. Awni, M. I. Khalil, and H. M. Abbas, "Offline arabic handwritten word recognition: A transfer learning approach," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 10, pp. 9654–9661, 2022.

[21] S. U. Masruroh, M. F. Syahid, F. Munthaha, A. T. Muharram, and R. A. Putri, "Deep convolutional neural networks transfer learning comparison on arabic handwriting recognition system," *JOIV: International Journal on Informatics Visualization*, vol. 7, no. 2, pp. 330–337, 2023.

[22] A. DAOOD, A. AL-SAEGH, and A. F. MAHMOOD, "Handwriting detection and recognition of arabic numbers and characters using deep learning methods," *Journal of Engineering Science and Technology*, vol. 18, no. 3, pp. 1581–1598, 2023.

[23] R. Malhotra and M. T. Addis, "End-to-end historical handwritten ethiopic text recognition using deep learning," *IEEE Access*, 2023.

[24] W. Jiang and L. Zhang, "Edge-siamnet and edge-triplenet: New deep learning models for handwritten numeral recognition," *IEICE Transactions on Information and Systems*, vol. 103, no. 3, pp. 720–723, 2020.

[25] A. Baró, C. Badal, P. Torras, and A. Fornés, "Handwritten historical music recognition through sequence-to-sequence with attention mechanism," in *3 rd International Workshop on Reading Music Systems*, 2022, p. 55.

[26] d. Riyadh, "Riyadh dictionary," *King Salman Academy for the Arabic Language*, 2023. [Online]. Available: https://dictionary.ksaa.gov.sa/

[27] H. Boubaker, A. Elbaati, N. Tagougui, H. El Abed, M. Kherallah, V. Märgner, and A. M. Alimi, "Adab database," 2021. [Online]. Available: https://dx.doi.org/10.21227/wpf8-dk19

[28] M. A. Khan, "Arabic handwritten alphabets, words and paragraphs per user (ahawp) dataset," *Data in Brief*, vol. 41, p. 107947, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340922001585

[29] N. Aneja and S. Aneja, "Transfer learning using cnn for handwritten devanagari character recognition," in *2019 1st International Conference on Advances in Information Technology (ICAIT)*. IEEE, 2019, pp. 293–296.