Computers, Materials & Continua
DOI: 10.32604/cmc.2023.032550
Article

An Efficient Hybrid Model for Arabic Text Recognition

Hicham Lamtougui[1,*], Hicham El Moubtahij[2], Hassan Fouadi[1] and Khalid Satori[1]

[1]LIIAN Laboratory, Faculty of Sciences Dhar-Mahraz, Fez, 30000, Morocco
[2]Modeling, Systems and Technologies of Information Team, University of Ibn Zohr, Agadir
*Corresponding Author: Hicham Lamtougui. Email: hicham.lamtougui@usmba.ac.ma
Received: 22 May 2022; Accepted: 14 July 2022

Abstract: In recent years, Deep Learning models have become indispensable in several fields such as computer vision, automatic object recognition, and automatic natural language processing. The implementation of a robust and efficient handwritten text recognition system remains a challenge for the research community in this field, especially for the Arabic language, which, compared to other languages, has a dearth of published works. In this work, we presented an efficient and new system for offline Arabic handwritten text recognition. Our new approach is based on the combination of a Convolutional Neural Network (CNN) and a Bidirectional Long-Term Memory (BLSTM) followed by a Connectionist Temporal Classification layer (CTC). Moreover, during the training phase of the model, we introduce an algorithm of data augmentation to increase the quality of data. Our proposed approach can recognize Arabic handwritten texts without the need to segment the characters, thus overcoming several problems related to this point. To train and test (evaluate) our approach, we used two Arabic handwritten text recognition databases, which are IFN/ENIT and KHATT. The Experimental results show that our new approach, compared to other methods in the literature, gives better results.

Keywords: Deep learning; arabic handwritten text recognition; convolutional neural network (CNN); bidirectional long-term memory (BLSTM); connectionist temporal classification (CTC)

1 Introduction

Recognition is a field that covers various areas such as image recognition, fingerprint recognition, facial recognition, number recognition, and character recognition. In general, the task of recognizing handwritten words is divided into two main groups: online and offline recognition. Online recognition depends on the path of the pen and the coordinates of the movement of the pen on the paper. But offline text recognition is done by analyzing the input image. Arabic writing is one of the most difficult recognition scripts in the field of text recognition due to its cursive nature which means the characters of a word are connected [1]. Despite advanced research on handwriting recognition [2], the development of a reliable system for Arabic offline handwritten words remains an open problem, it is

a complex task due to the diversity of styles, extensive vocabulary, variety of the writer's calligraphy, ligatures, overlaps, and irregular spaces.

The main objective of the handwriting recognition system is to convert handwritten text documents from the digital image format to character format documents that are readable using application processing systems.

Developing HTR systems have been the focal point of researchers for several decades. Traditional HTR approaches utilized various image processing techniques to segment the handwritten document into lines, words, or characters. After segmentation, the hand-designed features were extracted and the Hidden Markov Models (HMM) [3] were applied as sequence learning algorithms to represent the output as a sequence of characters. However, HMMs have limitations due to the Markovian hypothesis, which learns text contextual information only from the current state, making it difficult to model contextual effects. This work presents a hybrid deep learning model that uses a convolutional neural network (CNN) and a bidirectional short-term memory network (BLSTM) and it operates without requiring segmentation. Primarily, CNN and RNN algorithms have been widely used for text recognition [4]. Several studies have shown that convolutional neural networks and recurrent neural networks (RNNs) are superior to Hidden Markov Models (HMMs) for sequence labeling tasks such as handwriting and speech recognition [5] in addition, RNN long-term short-term memory (LSTM) architecture allows it to capture longer contexts, which are important for the recognition of offline text tasks. In addition, the use of the connectionist temporal classification (CTC) with the RNNs allows recognition without prior segmentation [6].

The main contribution of this study is the application of the hybrid Deep Learning CRNN (Convolutional Recurrent Neural Network) approach which, compared to other advanced systems, is more efficient for the recognition of Arabic writing, and we evaluate and compare it to other advanced systems. We present the experimental results using word and line modeling for Arabic handwriting recognition on publicly available IFN/ENIT databases [7] and Khatt [8]. To deal with class imbalances in classification problems using a new data augmentation algorithm for the recognition of handwritten Arabic texts. This document is organized as follows. Section 2 gives a discussion of related work. Section 3 presents a description of the characteristics of the Arabic text and their issues and some challenges. Section 4 describes the models, the architecture of the system, and the description of the databases. In Section 5, we report the results of the experiments. Finally, Section 6 presents the conclusion and future work.

2 Related Works

The literature shows that a fair amount of work has already been done on the performance evaluation of classifiers for the recognition of handwritten Arabic texts. Many methods have been proposed to realize a robust system for the recognition of offline handwritten characters, words, and lines by considering the experimental analysis method and the data set. This section presents related work on the IFN/ENIT and KHATT datasets.

2.1 Works Based on the IFN/ENIT Database

Graves et al. [9] combined the multidimensional LSTM with a connectionist temporal classification. The system successfully applied to English as well as Arabic, the dimensionality of the networks can be changed to match that of the data, it could in principle be used for almost any supervised sequence labeling task. According to Kessentini et al. [10], the proposed system for recognizing Arabic handwritten words is multi-stream. The proposed approach combines low-level

density-based feature streams with different widths extracted from two different sliding windows and features based on contours extracted from the bottom and top contours. The proposed system is script independent. Features are then combined according to the multi-stream paradigm. Alkhateeb et al. [11] proposes another method based on HMM has three steps namely preprocessing, feature extraction and classification, a set of intensity features is extracted from each of the segmented words, which is based on a sliding window moving over each mirrored word image. As for the Pechwitz et al. [12], he presents a recognition system for identifying Arabic words based on normalization and basic estimation, the two completely different feature sets were tested using the HMM-based recognition tool. A new system presented without segmentation based on the Hidden Markov Model (HMM). Azeem et al. [13] which incorporates a technique of dividing the image into horizontal segments followed by a technique for solving tilting problems. A robust feature set representing the foreground pixel densities in the different edge images is extracted using sliding windows. For Hamdani et al. [14], Implemented a bi-directional long-term short-term memory system (BLSTM). Gray values of pixels were used for the formation of the HMM system. They were taken out of the repositioned sliding windows. As for Abandah et al. [15], they used a system based on a process of segmentation, feature extraction and then recurrent neural network (RNN). The segmentation approach with efficient feature extraction yields better results than a holistic approach that extracts features from raw pixels. According to Jayech et al. [16] a Multi-flux Synchronous Hidden Markov Model (MSHMM) has been proposed with the advantage of efficiently modeling the temporal interaction between several features composed of a combination of structural and statistical features. Elleuch et al. [17] also proposes a new model by integrating two classifiers SVM and CNN. The results proved that the new SVM based on the architecture design of CNN with dropout performs more efficiently than the SVM model based on CNN without dropout and the standard CNN classifier. According to EL Moubtahij et al. [18], he presented a system based on Hidden Markov Models HMM Toolkit (HTK) with a word image technique of stochastic finite state automata, then he applied the sliding window technique along the image line to extract densities local and characteristic intensities. Hybrid CNN-HMM can improve the performance of a handwritten Arabic word recognition system according to Amrouch et al. [19]. For Ahmad et al. [20], he introduces a modeling that separates kernel shapes in Arabic texts from diacritics and then divides the kernel shapes into sub kernel shapes. Contextual HMM modeling uses these shapes, which demonstrates that using these shapes as models improves the contextual HMM system. For Eltay et al. [21], he uses a hybrid CRNN system and an adaptive data augmentation algorithm during the training phase which also attempts to address the problem of imbalanced classes.

2.2 Works Based on the KHATT Database

However, related work on handwriting recognition on the KHATT dataset is limited. As for the method of Mahmoud et al. [8], he presented a recognition system on the KHATT dataset using the Hidden Markov Model (HMM). They separately used the pixel density of text line images, derivatives of horizontal and vertical edges, as well as statistical characteristics and gradient characteristics. For Stahlberg et al. [22], he proposes two methods for feature extraction: a new method based on foreground segments or on the gray level intensity values of the raw pixels, and the deep neural networks discriminatively trained for modeling. According to BenZeghiba [23], he presented MDLSTM systems based on language modeling (LM) for the recognition of handwritten and printed subwords, the system used subword language modeling outperformed those using standard word LM. Jemni et al. [24], have proposed an Arabic handwriting recognition system based on the MDLSTM-CTC multiple combination. The article presented deep learning and features to create two recognition systems. The first engine uses HOG functionality. The second system relies on a cascade of CNN and MDLSTM

layers. According to Ahmad et al. [25], a network-based approach Multidimensional Long Short-Term Memory (MDLSTM) and a Connectionist Temporal Classification (CTC). MDLSTM has the advantage of scanning lines of Arabic text in all vertical and horizontal directions to cover diacritics, dashes, and periods. A new contribution for the recognition of Arabic handwriting on the KHATT dataset is presented by Noubigh et al. [26], who offers a deep CNN-BLSTM combination based on a character model approach. This low recognition rate motivates us to study the complexity of the KHATT dataset in order to improve handwriting recognition for Arabic script.

3 The Arabic Language Challenges and Characteristics

3.1 Characters and Diacritics

The recognition of Arabic writing fits into the general framework of recognition of cursive writing, with its specificities and problems. Contrary to other writing systems like the Latin or the Chinese systems, little work has been done on the recognition of Arabic writing. Additionally, Arabic characters are used in the writing of several universal languages. Arabic characters differ from other types of characters by their structure and how they are linked to form a word. Arabic writing is characterized by:

- Characters are written from right to left.
- Arabic has 28 base characters and not all caps.
- Same Arabic character can have up to four different forms (see Tab. 1).
- Some characters have the same body size, but the presence and position of a point or a group of points are the determining traits to distinguish these characters.
- An Arabic word consists of one or more related components, each of which contains one or more characters. (Fig. 1).

Table 1: Character of the Arabic alphabet

Character	Name	Ending	Middle	Beginning	Character	Name	Ending	Middle	Begining
ا	Alif	ـا	ـا	ا	ط	Ta	ـط	ـطـ	طـ
ب	Ba	ـب	ـبـ	بـ	ظ	Zae	ـظ	ـظـ	ظـ
ج	Jim	ـج	ـجـ	جـ	ف	Fae	ـف	ـفـ	فـ
د	Dal	ـد	ـدـ	دـ	ل	Lam	ـل	ـلـ	لـ
ذ	Dhal	ـذ	ـذـ	ذـ	م	Mim	ـم	ـمـ	مـ
ر	Ra	ـر	ـرـ	رـ	ن	Nun	ـن	ـنـ	نـ
ض	Dad	ـض	ـضـ	ضـ	ء	Hamza	ـئ	ـؤ	أ

As Arabic characters do not have a fixed size (height and width), their size varies from character to character to character and from shape to shape within the same character. Arabic texts are not vowelized. Readers of Arabic are used to reading these texts deducing meaning from the context. The overlap of characters and the presence of ligatures in a word, recognize a very delicate task. Each

of the separate components of a word called the Part of an Arabic Word (PAW), also is known as a pseudo-word. (see Tab. 2) shows some examples of Arabic words having a different number of PAW.

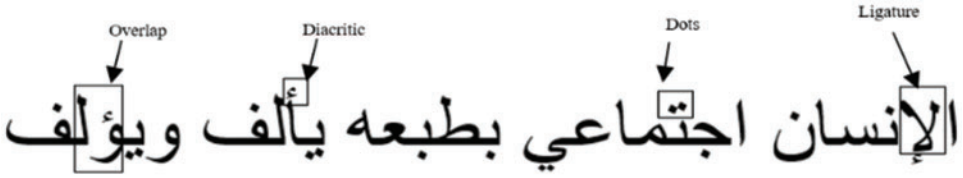


Figure 1: Some of the characteristics of Arabic script writing

Table 2: Words with multiple sub-words

Type	Examples		
Single sub-word	لغة	شبكة	مبتسم
Three sub-words	انتشار	قاموس	احتياط
Five sub-words	أزرار	أوراش	أوراقها

It is important to note that, only lam-alif is a mandatory ligature. Other ligatures are not required and therefore some authors may write them as ligatures while other authors may write the same character sequences in unligated forms, even in similar contexts. It is also possible for a writer to write a sequence of characters as a ligature in one case while writing it in an unligated form in other cases.

3.2 The Challenges of Recognizing the Arabic Manuscript

The handwritten Arabic text recognition which is a broad line of research is confronted with a certain number of challenges. Some of these problems are like those encountered by other scripts, such as handwriting variability among different authors and even for a single author, text issues, and overlap. The main challenges in recognizing handwritten Arabic text are related to the characteristics of Arabic writing.

Shapes vary depending on position: For some characters, the variations between their different shapes depending on position are not very large, while for other characters the intra-character variations are quite large as shown in Fig. 2 We can observe that the character shapes are visually very different from each other.

- Diacritics: some characters have periods above or below them. Writers write these points in diverse ways. Some authors write these points clearly, as is the case with machine-printed texts.
- Presence of ligatures: The Alif-Lam ligature should be given special attention, in most cases, it should be treated as a special character instead of treating it as two separate characters (i.e., say, alif and Lam).

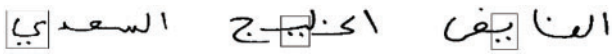


Figure 2: Different character shapes depend on the position of the character "yae" (Source IFN/ENIT)

4 The Proposed Methods

This section presents a Convolutional Recurrent Neural Network (CRNN) model for recognizing handwritten Arabic words and lines of texts and Arabic manuscript databases used for Arabic script recognition systems. The CRNN model consists of three parts: a convolutional feature extractor using a convolutional neural network (CNN) and a sliding window to extract features from a text image; recurring layers using BLSTM to predict a pre-frame from an input sequence, and a transcription layer using a CTC decoder to translate the predictions into the marker sequence [27]. Fig. 3 shows an example of the model, where CNN is first used to calculate the characteristics of the image which are transmitted to LSTM to learn the sequence and predict the output.

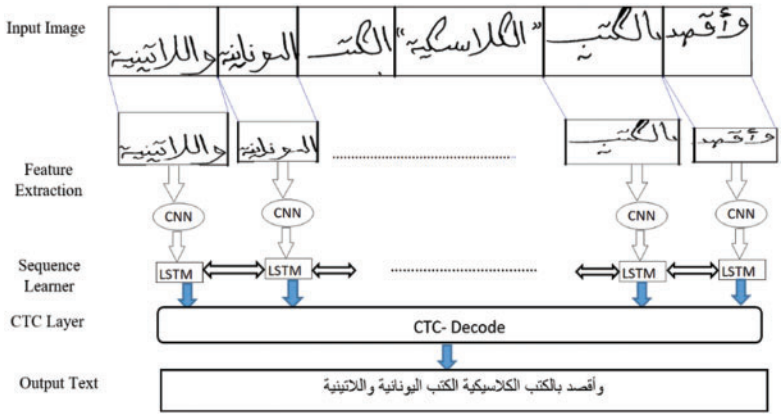


Figure 3: Architecture of CRNN with CTC layer (Source Khatt)

4.1 Convolutional Feature Extractor (CNN)

Convolutional neural networks are used successfully in a wide variety of application areas. The handwriting recognition task was one of the first applications of convolutional neural network image analysis. In addition to providing good results on object detection and image classification tasks [28], they also do well when applied to text recognition or even facial recognition [29]. Following this, it can be stated that the use of CNN remains highly effective in the detection, segmentation, and recognition of objects and regions in images (see Fig. 4).

The CNN model weights in the convolutional feature extractor are pre-trained by the Arabic handwritten databases IFN/ENIT and Khatt.

4.2 Recurrent Neural Networks (RNNs)

Recurrent neural networks are a family of neural networks that can work on sequential data [30]. Recently, many sequential labeling tasks for handwriting and offline recognition have used RNNs. The idea of these networks is to allow the sharing of parameters along the sequence to allow the extraction of information from the sequential aspect of the data [31]. Traditional RNNs (see Fig. 5) can learn complex temporal dynamics by mapping input sequences to a sequence of hidden states and hidden states to outputs via the following recurrence equations:

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$

$$y_t = f(W_{hy}h_t + b_y)$$

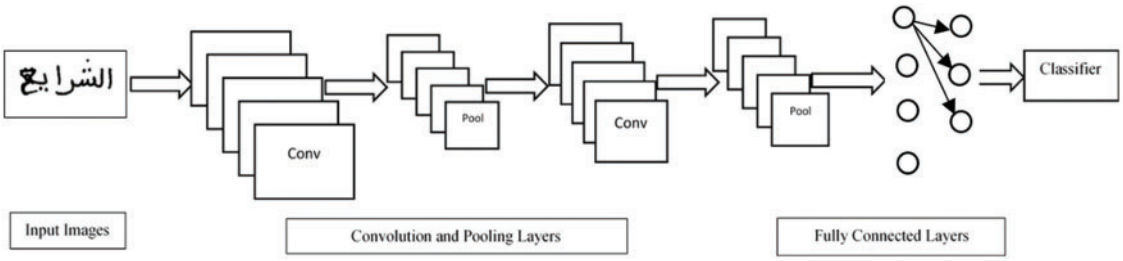


Figure 4: A general architecture of the convolutional neural network

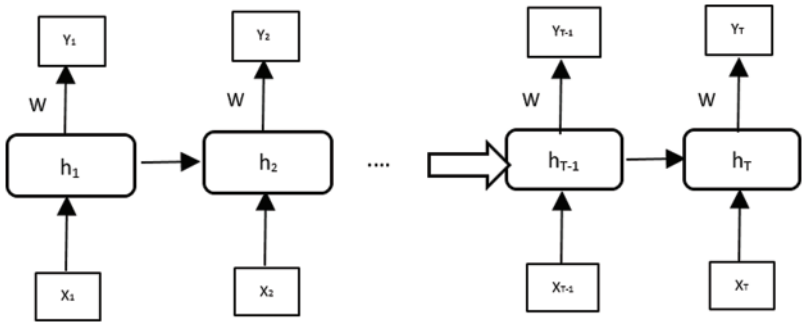


Figure 5: A diagram of a basic RNN

According to this modeling, an RNN takes as input a sequence of events $x = (x_1, x_2, \ldots, x_N)$ and defines the sequence of hidden states $h = (h_1, h_2, \ldots, h_N)$ to produce the sequence of output vectors $y = (y_1, y_2, \ldots, y_N)$ by iterating from $t = 1$ to N:

Where *N* is the total number of input vectors, the function f used in the case of RNNs is usually the hyperbolic tangent (tanh), W_{xy} is the weight matrix between the x and y, and b_y is the bias vector of layer y.

The advantage of RNNs lies in their ability to consider the past context when processing current information. However, these networks have difficulties in processing relatively long sequences [32].

Short-term and long-term memory is also a recurrent neural network, but it is different from other networks. The other networks repeat the module each time the input receives new information. However, the LSTM remembers the problem longer and has a chain structure to repeat the module.

The data transfer process is the same as standard recurrent neural networks. However, the information propagation operation is different. The main operation consists of cells and gates. An LSTM network has a memory cell, describes a layer of neurons, and three gates: an input gate, an output gate, and a forget gate. These three gates will make it possible to modulate the flow of information at the input, at output and, to be stored in an analog manner thanks to a sigmoid-type activation function. Fig. 6 is a deployed representation of the operation of an LSTM network with its three main components.

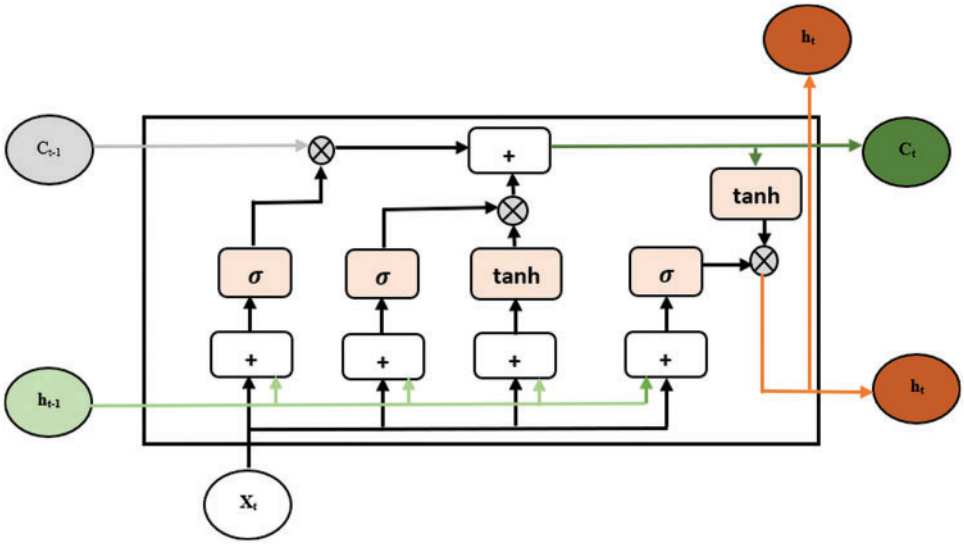


Figure 6: The structure of Long Short-Term Memory (LSTM)

Short-Term and Long-Term Memory recurrent neural networks (LSTM) are an effective neural model for a large number of applications involving temporal or sequential data. Among the many existing applications, we find handwriting recognition The idea of LSTM is to allow the network to ignore some important observations in the current prediction and to effectively model long-distance dependencies [33]. LSTMs have been shown to be effective in various fields of application. They are currently considered the state-of-the-art approach in many tasks dealing with sequential data [34]. The vector formulas for calculating LSTM units can be described as:

$$i_t = \sigma(W_{ix}x_t + w_{ih}h_{t-1} + p_i \odot c_{t-1} + b_i)$$

$$f_t = \sigma(W_{fx}x_t + w_{fh}h_{t-1} + p_f \odot c_{t-1} + b_f)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \varnothing(W_{cx}x_t + w_{ch}h_{t-1} + b_c)$$

$$g_t = \sigma(W_{gx}x_t + w_{gh}h_{t-1} + p_g \odot c_t + b_g)$$

$$h_t = g_t \odot \varnothing(c_t)$$

- The functions $\emptyset$ and σ are the hyperbolic tangent nonlinearity and logistic sigmoid, respectively. The operation $\odot$ represents element-wise multiplication of vectors.
- x_t is the input vector.
- The vectors i_t, f_t, g_t are the activation of the input, forget gates, and output, respectively.
- The $W_{.x}$ and $W_{.h}$ terms are the weight matrices for the inputs.
- x_t and the recurrent input h_{t-1}, respectively.
- The p_f, p_g, p_i are parameter vectors associated with peephole connections.

Standard LSTM can only use past context information in one direction. This can be overcome by using (BLSTM) [35] Which can learn long-range context dynamics in both input directions. BLSTM networks consist of running two LSTMs in parallel: the first network reads the input sequence from right to left and the second network in reverse from left to right. BLSTMs are built on top of the

convolutional entity extractor, as recurring layers to predict a label distribution for each entity in the entity sequence extracted from the previous component (see Fig. 7).

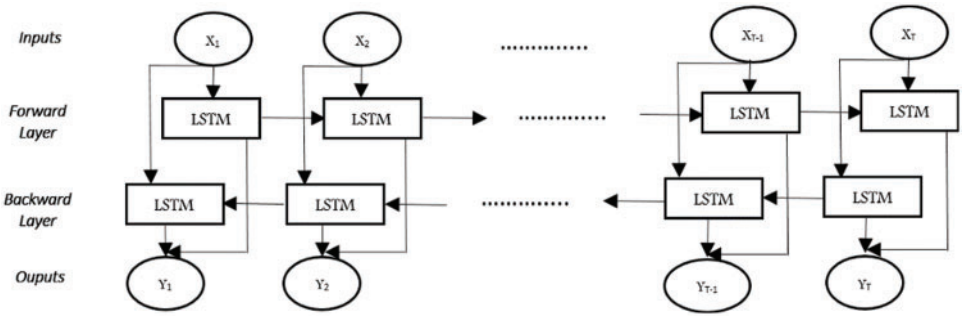


Figure 7: The architecture of a bi-directional recurrent neural network

The passes forward just for the time from t = 1 to t = N and backward from t = N to t = 1 are completed. A forward pass is also performed for the output neurons. BLSTM structures may perform better than other network structures, depending on the problem area. It has been shown to be significant success in text-based tasks where content is important.

4.3 Connectionist Temporal Classification (CTC)

Instead of trying to segment the image into letters and then extract features from each part of the image which then make it possible to classify and find the letters, Graves et al. [36] proposes to directly use the image of a word as input to a network of recurrent neurons and a sequence of letters as output for training the network. The Connectionist Temporal Classification Layer (CTC) is an output layer designed for tagging sequences with RNNs. Unlike other neural network output layers, it does not require training data to turn its outputs into transcripts. We denote the character set by $M' = M \cup \{\text{blank}\}$, where M is a fixed set of labels and "blank" represents no label. For an input sequence $x = x_1, x_2 \ldots\ldots x_N$ of length N, the conditional probability of a path π through the network of output labels on all-time steps. Finally, the probability of having a sequence of labels π is calculated for a sequence x with the probability y of observing a label at an instant t.

$$\mathbb{P}(\pi|x) = \prod_{t=1}^{T} \mathbb{P}(\pi_t|x,t) = \prod_{t=1}^{T} y_{\pi_t}^t$$

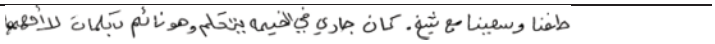
This is done without explicitly segmenting the input sequence. In cases where a dictionary is used, the tagging may be constrained to produce only complete word sequences.

$$l_{max} = B(\pi_{max}); \ \pi^t_{max} = \arg\max_k \left(y_k^t\right), \ t = 1 \ldots N$$

Problems such as the one presented in Tab. 3 will sound familiar to you, but the result is wrong; however, when one looks closely at the handwritten text, one can imagine why the system is confusing characters in the text.

To improve the results of text recognition (allow punctuation marks, avoid misspellings), there are decoding algorithms available, some also include a language model (LM).

Table 3: Example of predict and correct text (Source Khatt)

Input Text	
Output Text (Predict)	طفنا وسعينا مع شيخ. كان جاري في الخيمة يتكلم و هو نائم ت بلما لا أحهمها
Output Text (Correct)	طفنا وسعينا مع شيخ. كان جاري في الخيمة يتكلم و هو نائم بكلمات لا أفهمها

4.3.1 Best Path Decoding

Best path decoding uses only the output of the neural network and calculates an approximation by taking the most probable character at each position. It calculates the best path by taking the most probable character by time step.

4.3.2 Token Passing

By limiting the text to dictionary words, the token passing decoder solves the problem of decoding words. The most probable word in the input sequence, where the word output is limited by the dictionary. For word recognition, we expect the probability of each possible word with a character sequence based on the output of the formed network. Suppose we have L characters and the input has t time step, then the network will produce the probability that each character appears at each time step, building an $L \times t$ matrix.

4.3.3 Word Beam Search (WBS)

Word Beam Search (WBS) is introduced and helps decode features in tag sequences with the highest degree of probability [12]. WBS is a fast and powerful algorithm with an integrated language model to decode the output of the neural network in the context of text recognition. At each time step, only the best score beams from the previous time step are kept. The algorithm shows the pseudo-code for WBS.

The Beam set contains the beams of the current time step, and P contains the probabilities for the beams. P_k is the probability that the paths in a bundle end in a blank, P_{nk} that they end in a non-blank, Psom is the abbreviation of $P_k + P_{nk}$.

Algorithm 1: shows the pseudo-code for WBS decoding.

Algorithm #1: **Beam Search**
Input: matrix MT, BWS, and LM
Output: Probable result
Beam $= \{\varphi\}$;
$P_k(\varphi, 0) = 1$;
for t $= 1 \ldots$ T do
  Beams $=$ Beams (Beam, BWS);
  B $= \{\}$
  for k $\in$ Beams do
    if k! $= \varphi$ then

(Continued)

Algorithm 1: Continued

```
            P_nk (k, t) += P_1 (k, t − 1).mt(b (−1), t);
          end
          P_k(k, t) += P_nk(k, t − 1).mt(blank, t);
          Beam = Beam ∪ k;
          char_S = char_S(k)
          for i ∈ char_S do
            k′ = k + i;
            P(k′) = score (LM, k, i)
            if b(t) == i then
              P_nk(k′, t) += mt(i, t). P_k(k, t − 1)
            else
              P_nk(k′, t) += mt(i, t). P_k(k, t − 1)
            end
            Beam = Beam ∪ k′
            end
        end
    end
```

The algorithm iterates from t = 1 to t = N and creates a bundle labeling tree. An empty bundle is noted, and the last character of a bundle is indexed b_{y-1}. The best beams are obtained by sorting them against P_{som}. Separate accounting for paths ending with a blank and paths ending with non-blank counts for the CTC encoding scheme. Each beam is extended by a set of possible following characters, depending on the state of the beam. Once the algorithm has completed its time iteration, cluster tags are completed as necessary: if a cluster tagging ends with a prefix that does not represent a complete word, the prefix tree is queried to give a list of possible words that contain the prefix. There are two different ways to implement completion: either the bundle tagging is extended by the most probable word, or the bundle tagging is only completed if the list of possible words contains exactly one entry.

4.4 Dataset Preparation

This subsection presents databases of Arabic manuscripts used for recognition systems for Arabic writing. Unfortunately, most of these databases are no longer accessible, they were developed for well-defined research work. But it is the IFN/ENIT and KHATT databases, free for academic research, that have established themselves as benchmarks for comparing the performance of Arabic handwriting recognition systems.

4.4.1 Dataset IFN/ENIT

The IFN/ENIT database is the most widely used and popular database for handwritten Arabic text recognition research published by Pechwitz et al. [7]. Written by over 1000 writers, the IFN/ENIT database gathers 32,492 images of Arabic words. These words are the name of 937 Tunisian towns/villages. The IFN/ENIT is initially divided into 5 sets (see Tab. 4) and has been publicly exploited by a large number of research groups. Moreover, it has been exploited in various international competitions such as the 2009 ICDAR Offline Arabic Handwriting Recognition Competition [37].

Table 4: Some statistics of the IFN/ENIT database

Sets	IFN/ENIT	
	Words	Characters
A	6537	51984
B	6710	53862
C	6477	52155
D	6735	54166
E	6033	45169
Total	32492	257336

4.4.2 Dataset KHATT

KFUPM Handwritten Arabic TexT (KHATT) (خط) is the second database used in this work. It is proposed and developed by [12]. It is a large database of actual Arabic handwritten texts, containing 2000 paragraphs written by 1000 editors. The paragraphs are automatically segmented into a total number of 9,000 lines of text (see Fig. 8). It is available for free (khatt.ideas2serve.net). The database is divided into three disjoint sets, namely (see Tab. 5). Train (70%), validation (15%) and tests (15%). Text images are scanned at multiple resolutions (200, 300, and 600 DPI).

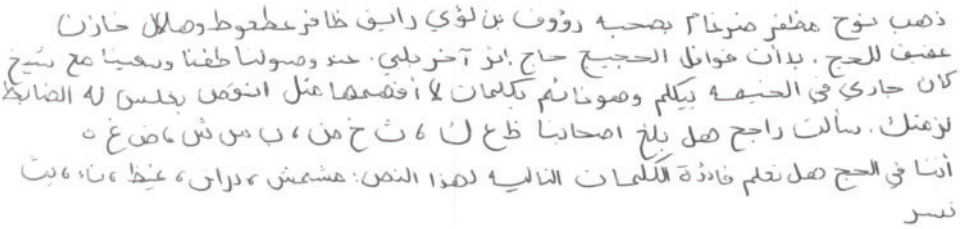


Figure 8: Sample of KHATT dataset

Table 5: Statistics in terms of number for the KHATT dataset

	Train	Test	Validation
Pages	690	141	148
Lines	9475	2007	1902
Words	129 826	26 449	26 142
Characters	605 537	122 757	121 433

4.4.3 Data Augmentation for Deep Learning

Deep learning networks require large amounts of data [38]. Preparing a sufficient amount of training data is expensive and labor-intensive. To resolve this issue, it is common practice to apply data augmentation to the training data. Data augmentation is a technique that can be used to create updated copies of images in the dataset to artificially increase the size of a training dataset. We will

be able to increase the learning field of our model, which will be able to adapt better to predict new data [39]. Different techniques can be applied on IFN/ENIT and KHATT databases, such as image rotation, image flipping, cropping, zooming, shearing, shifting, cropping, filling, and flipping to make the system more robust and able to operate in real time. Adding a variety of data goes a long way in reducing over-learning when training a machine learning model on small size and low-quality data. Keras's ImageDataGenerator class is used to achieve data augmentation when the model trains on image batches. Then by applying different random transformations it creates a new batch of randomly transformed images which is used for the training model, as shown in Fig. 9.

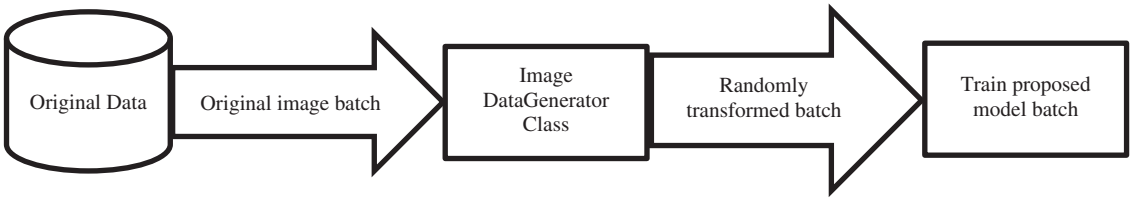
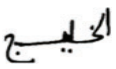
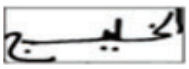


Figure 9: Data augmentation process

For the increase of data with the Keras ImageDataGenerator class, we have chosen: (see Tab. 6)

- Image rotation: This transformation consists of rotating the original image at a desired angle. In analyzing images of handwritten texts, it is common to augment a dataset with random rotations at different angles ranging, for example, from $-20°$ to $+20°$.
- Scale: When scaling or resizing, the image is resized to the given size, The image can be enlarged outward or inward.
- Cropping and zoom: These two methods visually have the same result; some parts of the image are cropped to keep only a part. During cropping, part of the image is selected, In the example given, the image is cropped in the center.

Table 6: Example of data augmentation techniques (Source IFN/ENIT)

Original image	Rotation range = 15		Scale	Cropping (resize the image)
				

However, after data augmentation, each text line in the dataset is regenerated using data augmentations techniques as explained in the next section and the size of the dataset becomes 3 times the multiple of single lines of text. The statistics for the augmented data set are shown in Tab. 7.

Table 7: The statistics of KHATT after data augmentation

Sets	Unique text lines	3 variations by data augmentation
Train set	4825	14475
Test set	966	2898
Total size	5791	17373

5 Experiments, Results, and Discussion

In the following subsections, we present experiment setups, evaluation results, and comparison with advanced systems.

5.1 Experimental Settings

The experiments are performed exclusively on the IFN/ENIT word data sets. Then we extend them to the Khatt row databases, for which we used the CRNN hybrid. We carried out our experiments by augmenting the data for the IFN/ENIT and Khatt databases. The main objective is to evaluate and analyze the performance of the different combination level strategies. Performance was measured in terms of Word Error Rate (WER). To assess the accuracy of the presented systems, we used the Levenshtein editing distance between the output text and the ground truth. The edit distance is calculated to transform a source string into a target string. To assess the performance of our system, we used the Word Accuracy Rate (WAR) word-level recognition rate. The WAR is defined by:

$$WAR = \frac{N - (S + D + I)}{N} \times 100$$

With N the number of characters in the reference images, S the number of character substitutions, D the number of character deletions, and I the number of character insertions. The WRR is calculated in the same way on the words. Both measures treat the space as a character in a line. Through our experiments, we aim to define a simple system capable of performing learning.

5.2 Implementation Details

To efficiently train the model on a larger amount of data to handle the variability of the write, the execution of our model is divided as shown in Fig. 10. Our model is the CRNN inspired by the VGG16 architecture for the HTR. We use a stack of 13 convolutional layers (3×3 filters, 1×1 stride) followed by three BLSTM layers with 512 neurons. Pooling is used after a few convolutional layers. The activation function ReLU is implemented to introduce non-linearity since convolution is a linear operation. Max Pooling Layers has been added between convolutions to reduce the width and length dimension. For text recognition, BLTSM networks are combined with the Connectionist Time Classification (CTC) function to predict output tag sequences without the need to segment the input text. Thanks to the objective function, the system is trainable from end to end. Batch normalization has been added after each convolutional layer to speed up the learning process. It basically works by normalizing each lot by both the mean and the variance. A Word Beam Search algorithm was used for the decoding.

5.3 Comparative Experiments and Results

In these experiments, the inputs are images taken from the IFN/ENIT and KHATT databases transmitted through the CNN layers followed by the BLSTM layers. The experiments are carried out on images of words and lines. In this work, we report the accuracy rate as presented in Tabs. 8 and 9. The tables show a comparative study between our recognition system and the best reported results. In this study, we introduce the approaches and the results used for each system. Our contribution to the recognition of words and lines of texts in Arabic handwriting was based on the CNN-BLSTM-CTC approach considering the IFN/ENIT and KHATT datasets as a learning and testing case. The results show that our proposed model outperforms peak results. The state of the art uses the Hidden Markov Model (HMM) ML, and DL algorithms as a classifier.

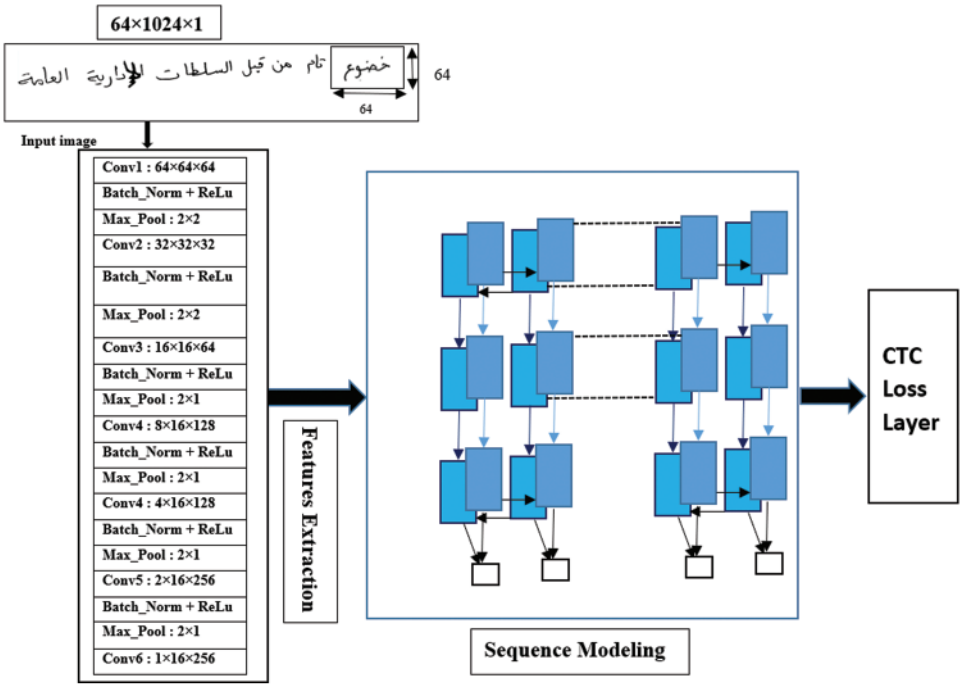


Figure 10: General architecture of the proposed model

Table 8: Comparison with state-of-the-art methods on the IFN/ENIT dataset

References	Training/Testing	Approach	Accuracy
Graves et al. [9]	abcde/f	Multi-dimensional RNN	91.43%
Kessentini et al. [10]	abcd/e	Multi-stream HMM	79.6%
Alkhateeb et al. [11]	abc/d	Handcrafted features-HMM-re-ranking	83.55%
Hamdani et al. [14]	abcd/e	Multiple HMM	81.93%
Abandah et al. [15]	abcd/e	BLSTM-RNN	75.72%
Jayech et al. [16]	abcd/e	MSHMM	72%,4%
Elleuch et al. [17]	56 classes	CNN-SVM	92.95%
El Moubtahij et al. [18]	**abc/d**	**Handcrafted features-HMM**	**78.95%**
Amrouch et al. [19]	abc/d	HMM-CNN	88.95%
Ahmad et al. [20]	abcde/f	Sub-core-shape HMM	93.32%
Eltay et al. [21]	abcde/f	BLSTM-CTC-WBS	93.57%
Our models	**abcd/e**	**CNN-BLSTM-CTC**	**92.11%**

Table 9: Comparison with state-of-the-art methods on the KHATT dataset

Reference	Model	Accuracy
Mahmoud et al. [8]	HMM	51.2%
Stahlberg et al. [22]	Pixel-based features	69.5%
BenZeghiba [23]	MDLSTM	65.7%
Jemni et al. [24]	CNN + MDLSTM	79.17%
Ahmad et al. [25]	MDLSTM + CTC	75.7%
Noubigh et al. [26]	CNN + BLSTM	79.83%
Our models	**CNN + BLSTM + CTC**	**80.15%**

6 Conclusions and Perspectives

The automatic recognition of handwritten Arabic writing remains an open field of research and a promising exploration of more methods and techniques to design optimized procedures and programs like. We note that technological advances have not however advanced research on this problem which requires more effort, and perhaps more will. In this article, we have presented two databases for Arabic handwriting. We have proposed a deep learning model to recognize handwritten words and lines. Our model had better results on the IFN/ENIT dataset and promising results against other models on the KHATT dataset, which is one of the difficult datasets that contain text documents and manuscripts in Arabic. We used a combination of convolutional and recurrent neural networks where features extracted by convolutional layers are passed to a long-term, bidirectional memory network for classification. We used the CTC layer for the alignment of the predicted labels along the most probable path with a WBS decoder. In addition, we have proposed an adaptive data augmentation algorithm that generates artificial data to improve the diversity of training data of a classifier to improve its performance. It has been shown that this method can solve the problem of data imbalance quite effectively. Our proposed method is conducted in such a way that the minority classes will be enlarged by calculating the average probability of each class. With regard to future work, we plan to use the Capsule Networks in order to ameliorate the findings.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Naz, K. Hayat, M. I. Razzak, M. W. Anwar, S. A. Madani *et al.,* "The optical character recognition of urdu-like cursive scripts," *Pattern Recognition*, vol. 47, no. 3, pp. 1229–1248, 2014.

[2] H. Almuallim and S. Yamaguchi, "A method of recognition of Arabic cursive handwriting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 715–722, 1987.

[3] A. Giménez and A. Juan, "Embedded Bernoulli mixture HMMS for handwritten word recognition," in *10th Int. Conf. on Document Analysis and Recognition*, Barcelona, Spain, pp. 896–900, 2009.

[4] Y. Wang, Y. Yang, H. Chen, H. Zheng and H. Chang, "End-to-end handwritten Chinese paragraph text recognition using residual attention networks," *Intelligent Automation & Soft Computing*, vol. 34, no. 1, pp. 371–388, 2022.
[5] M. B. Kamal, A. A. Khan, F. A. Khan, M. M. Ali Shahid, C. Wechtaisong *et al.,* "An innovative approach utilizing binary-view transformer for speech recognition task," *Computers, Materials & Continua*, vol. 72, no. 3, pp. 5547–5562, 2022.
[6] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar *et al.,* "Large-scale video classification with convolutional neural networks," in *Int. Conf. on Computer Vision and Pattern Recognition*, Columbus, OH, USA, IEEE, pp. 1725–1732, 2014.
[7] M. Pechwitz, S. S. Maddouri, V. Märgner, N. Ellouze and H. Amiri, "IFN/ENIT-database of handwritten Arabic words," in *Proc. CIFED*, Hammamet, Tunisie, pp. 127–136, 2002.
[8] S. A. Mahmoud, I. Ahmad, W. G. Al-Khatib, M. Alshayeb, M. T. Parvez *et al.,* "Khatt: An open Arabic offline handwritten text database," *Pattern Recognition*, vol. 47, no. 3, pp. 1096–1112, 2014.
[9] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Proc. NIPS*, Vancouver, British Columbia, Canada, pp. 545–552, 2008.
[10] Y. Kessentini, T. Paquet and A. Ben Hamadou, "Off-line handwritten word recognition using multi-stream hidden Markov models," *Pattern Recognition Letters*, vol. 31, no. 1, pp. 60–70, 2010.
[11] J. H. Alkhateeb, J. H. Ren, J. Jiang and H. Al-Muhtaseb, "Offline handwritten Arabic cursive text recognition using hidden Markov models and re-ranking," *Pattern Recognition Letters*, vol. 32, no. 8, pp. 1081–1088, 2011.
[12] M. Pechwitz, H. E. Abed and V. Märgner, "Handwritten Arabic word recognition using the IFN/ENIT-database," in *Guide to OCR for Arabic Scripts*, London: Springer, pp. 169–213, 2012.
[13] S. A. Azeem and H. Ahmed, "Effective technique for the recognition of offline arabic handwritten words using hidden Markov models," *International Journal on Document Analysis and Recognition*, vol. 16, no. 4, pp. 399–412, 2013.
[14] M. Hamdani, H. El Abdel, M. Kherallah and A. M. Alimi, "Combining multiple HMMs using on-line and off-line features for off-line Arabic handwriting recognition," in *10th Int. Conf. on Document Analysis and Recognition*, Barcelona, Spain, IEEE, pp. 201–205, 2009.
[15] G. A. Abandah, F. T. Jamour and E. A. Qaralleh, "Recognizing handwritten arabic words using grapheme segmentation and recurrent neural networks," *International Journal on Document Analysis and Recognition*, vol. 17, no. 3, pp. 275–291, 2014.
[16] K. Jayech, M. A. Mahjoub and N. E. B. Amara, "Arabic handwriting recognition based on synchronous multi-stream HMM without explicit segmentation," in *10th Int. Conf. on Hybrid Artificial Intelligent Systems*, Bilbao, Spain, pp. 136–145, 2015.
[17] M. Elleuch, R. Maalej and M. Kherallah, "A new design based-SVM of the CNN classifier architecture with dropout for offline Arabic handwritten recognition," *Procedia Computer Science*, vol. 80, pp. 1712–1723, 2016.
[18] H. EL Moubtahij, A. Halli and K. Satori, "Arabic handwriting text recognition offline system through using the HMM toolkit and stochastic finite-state automaton," *International Journal of Tomography and Simulation*, vol. 30, pp. 92–105, 2017.
[19] M. Amrouch and M. Rabi, "Deep neural networks features for Arabic handwriting recognition," in *17th Int. Conf. on Advanced Information Technology, Services and Systems*, Tangier, Morocco, pp. 138–149, 2017.
[20] I. Ahmad and G. A. Fink, "Handwritten Arabic text recognition using multi-stage sub-core-shape HMMs," *International Journal on Document Analysis and Recognition*, vol. 22, no. 3, pp. 329–349, 2019.
[21] M. Eltay, A. Zidouri and I. Ahmad, "Exploring deep learning approaches to recognize handwritten arabic texts," *IEEE Access*, vol. 8, pp. 89882–89898, 2020.
[22] F. Stahlberg and S. Vogel, "The QCRI recognition system for handwritten Arabic," in *Proc. ICIAP*, Genoa, Italy, pp. 276–286, 2015.
[23] M. F. BenZeghiba, "Arabic word decomposition techniques for offline Arabic text transcription," in *1st Workshop on Arabic Script Analysis and Recognition*, Nancy, France, pp. 31–35, 2017.

[24] S. K. Jemni, Y. Kessentini and S. Kanoun, "Out of vocabulary word detection and recovery in Arabic handwritten text recognition," *Pattern Recognition*, vol. 93, pp. 507–520, 2019.
[25] R. Ahmad, S. Naz, M. Afzal, S. Rashid, M. Liwicki *et al.,* "A deep learning based Arabic script recognition system: Benchmark on KHAT," *International Arab Journal of Information Technology*, vol. 17, no. 5, pp. 299–305, 2020.
[26] Z. Noubigh, A. Mezghani and M. Kherallah, "Contribution on Arabic handwriting recognition using deep neural network," in *19th Int. Conf. on Hybrid Intelligent Systems*, Bhopal, India, pp. 123–133, 2021.
[27] F. Abdurahman, E. Sisay and K. A. Fante, "AHWR-net: Offline handwritten Amharic word recognition using convolutional recurrent neural network," *SN Applied Sciences*, vol. 3, no. 8, pp. 1–11, 2021.
[28] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[29] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[30] A. Graves, S. Fernández and J. Schmidhuber, "Multi-dimensional recurrent neural networks," in *Proc. ICANN*, Heidelberg, Berlin, pp. 549–558, 2007.
[31] S. Naz, A. I. Umar, R. Ahmad, S. B. Ahmed, S. H. Shirazi *et al.,* "Urdu Nasta'liq text recognition system based on multi-dimensional recurrent neural network and statistical features," *Neural Computing and Applications*, vol. 28, no. 2, pp. 219–231, 2017.
[32] W. Simayi, M. Ibrayim and A. Hamdulla, "Character type based online handwritten Uyghur word recognition using recurrent neural network," *Wireless Networks*, pp. 1–11, 2021.
[33] Y. He, S. Nazir, B. Nie, S. Khan and J. Zhang, "Developing an efficient deep learning-based trusted model for pervasive computing using an LSTM-based classification model," *Complexity*, vol. 2020, pp. 1–6, 2020.
[34] J. Qian, M. Zhu, Y. Zhao and X. He, "Short-term wind speed prediction with a two-layer attention-based LSTM," *Computer Systems Science and Engineering*, vol. 39, no. 2, pp. 197–209, 2021.
[35] A. M. Almars, "Attention-based Bi-LSTM model for Arabic depression classification," *Computers, Materials & Continua*, vol. 71, no. 2, pp. 3091–3106, 2022.
[36] A. Graves, S. Fernández, F. Gomez and J. Schmidhuber, "Connectionist temporal classification: Labeling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, New York, NY, United States, pp. 369–376, 2006.
[37] V. Märgner and H. E. Abed, "ICDAR 2009 Arabic handwriting recognition competition," in *10th Int. Conf. on Document Analysis and Recognition*, Barcelona, Spain, pp. 1383–1387, 2009.
[38] H. Sun and R. Grishman, "Lexicalized dependency paths based supervised learning for relation extraction," *Computer Systems Science and Engineering*, vol. 43, no. 3, pp. 861–870, 2022.
[39] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.