

Article

Not peer-reviewed version

Machine Learning Approach for Arabic Handwritten Recognition

[A.M. Mutawa](#)*, [Mohammad Allaho](#), [Monirah Al-Hajeri](#)

Posted Date: 3 September 2024

doi: 10.20944/preprints202408.2233.v1

Keywords: Machine learning; handwritten recognition systems; Arabic handwriting; BiLSTM; ResNet; natural language processing



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Machine Learning Approach for Arabic Handwritten Recognition

A.M. Mutawa^{1,2,*}, Mohammad Allaho¹ and Monirah Al-Hajeri¹

¹ Department of Computer Engineering, College of Engineering and Petroleum, Kuwait University, Kuwait

² Computer Sciences Department, University of Hamburg, Germany

* Correspondence: dr.mutawa@ku.edu.kw

Abstract: Text recognition is an important area of the pattern recognition field. Natural language processing (NLP) and pattern recognition have been utilized efficiently in script recognition. Much research has been done on handwritten script recognition. However, the research on the Arabic language for handwritten text recognition received little attention compared with other languages. Therefore, it is crucial to develop a new model that can recognize Arabic handwritten text. Most of the existing models used to acknowledge Arabic text are based on traditional machine learning techniques. Therefore, we implemented a new model using deep machine learning techniques by integrating two deep neural networks. The architecture of the Residual Network (ResNet) model is used to extract features from raw images. Then, the Bidirectional Long Short-Term Memory (BiLSTM) and connectionist temporal classification (CTC) are used for sequence modeling. Our system improved the recognition rate of Arabic handwritten text compared to other models of similar type with a character error rate of 13.2% and character error rate of 27.31%. In conclusion, the domain of Arabic handwritten recognition is advancing swiftly with the use of sophisticated deep learning methods.

Keywords: machine learning; handwritten recognition systems; Arabic handwriting; BiLSTM; ResNet; natural language processing

1. Introduction

Optical Character Recognition (OCR) is a field of research in Pattern Recognition (PR). The goal of an OCR system is to automatically read text from a scanned paper and convert it to a digital format that can be readable and editable using electronic applications.

Nowadays, technologies are spread worldwide, and almost all the essential processes are being done electronically. Also, the Arabic language is the official language of 23 countries and is spoken by more than 400 million people worldwide [1]. This raises the need for an efficient Arabic text recognizer, which can be helpful in many institutions such as educational, governmental, and economic organizations. For example, it is essential to convert old and/or new documents with handwritten scripts into digital Arabic text in some institutions. The OCR system helps professionally complete office tasks while saving time and effort. Moreover, recognizing Arabic handwritten text is helpful in the procedure of automatic reading of bank checks [2].

It is challenging to recognize human writing of Arabic text because of the cursive nature of Arabic handwriting scripts. The different shapes of Arabic letters depend on their location in the word and the special marks used in some Arabic letters, such as 'Hamza' and 'Maada.' Also, a lot of Arabic letters have the same shape but are differentiated by dots, which can be one, two, or three dots placed either above or below the character [3].

The Arabic script is written from right to left, so it is essential to recognize the words in the same direction. Due to the challenges of recognizing Arabic text and the various characteristics of Arabic writing from other writings, it is difficult to apply the techniques developed for identifying other languages in Arabic scripts. Therefore, we implemented a new model that will recognize offline Arabic handwritten text.

Arabic text recognition systems can be either based on segmenting the word (analytical approach) or without segmentation. Most of the current systems are segmentation-based systems, which require segmenting the word into different characters. Then, after the segmentation process, each character is recognized. And due to the cursive nature of the Arabic handwritten text, it is challenging to segment words into characters [4]. On the other hand, the segmentation-free models (holistic approach) recognize the word as a whole word image without any segmentation processes. The holistic approach is preferred for data with small vocabulary sizes, such as recognition of bank checks. While the analytical method is ideal for recognition systems that consist of large-sized vocabularies [5,6].

Traditional approaches in developing Arabic handwritten recognition systems are based on shallow learning techniques. Using these techniques makes it difficult to deal with the challenges of recognizing Arabic handwritten words. This is because they only extract the sample features of the word image. However, deep machine learning approaches have a better performance in many systems since they can extract more complex features from the word image [7,8]. Thus, using machine learning approaches is helpful to handle the challenges of recognizing Arabic handwritten words [9,10].

The identification of Arabic characters poses persistent challenges due to several factors, and continuous research is being conducted to enhance the performance of current systems. Several methodologies are constrained to proprietary datasets or the identification of individual words or paragraphs, which complicates the evaluation of their effectiveness on authentic Arabic literature [11].

This work implements a segmentation-based model using deep machine learning techniques to have a high accuracy rate. The system is evaluated using KHATT [12] and AHTID/MW datasets [13]. These are challenging datasets that contain text line images. These datasets cover different writing styles of other writers; the system will recognize Arabic handwritten text and words from a text line image.

The contribution of the study are as follows:

- We implemented ResNet model to extract the features of every individual character in the textual image. The BiLSTM with CTC model is employed for the purpose of sequence modeling. Ultimately, a language model (LM) is employed during the post-processing phase to improve the forecasted outcome derived from the classification phase.
- We performed testing of the model on two distinct datasets: KHATT and AHTID/MW datasets. The utilization of several datasets underscores the model's capacity to extrapolate across diverse manifestations of Arabic handwriting.

2. Literature Review

Many optical character recognition systems have been designed to recognize Arabic handwritten characters and words. Convolutional Neural Network (CNNs) have been widely employed in handwritten recognition due to their capacity to autonomously acquire hierarchical features from unprocessed pixel input [14,15]. A combination of two classifiers, which are CNN and Support Vector Machine (SVM) with a dropout regularization technique, is used to recognize offline Arabic handwriting characters [16]. The authors use SVM to adjust the trainable classifier of CNN. The dropout is applied before supplying the output into an SVM classifier. The authors tested their design on a character dataset, HACDB, and on a word dataset that is IFN/ENIT. The experimental results showed that the HACDB dataset had a 5.83%-character error rate. The IFN/ENIT dataset had a 7.05%-character error rate.

An Arabic handwriting recognition system was proposed based on multiple BLSTM-CTC combinations [17]. In this study, two different extraction techniques were used. The first method is segment-based features. The second is Distribution-Concavity (DC) based features trained on different levels of the BLSTM-CTC combination. The combination levels were low-level fusion, Mid-level combination methods, and high-level fusion. The experiments were performed on the KHATT

dataset. The results showed that the high-level fusion had a better recognition rate than the other combination levels, with a 29.13% Word Error Rate (WER) and a 16.27% Character Error Rate (CER).

BenZeghiba, Louradour, and Kermorvant used a hybrid Hidden Markov Model and Artificial Neural Network (HMM and ANN) framework to recognize Arabic handwritten text [18]. The type of Artificial Neural Network (ANN) used in their system is Multi-Dimensional Long Short Term Memory Networks (MDLSTMs). The hybrid model extracts the pixel values of text line images by scanning the text line images in four directions. A connectionist temporal classification (CTC) is used during the training process. The Viterbi algorithm [19], a decoding strategy, is used to generate the best hypothesis of a character sequence. They added a hybrid language model that consists of words and Part-of-Arabic-Words (PAWs). The KHATT dataset was used to evaluate the system, and the result was a 33% Word Error Rate (WER).

A recognition system for Arabic handwritten text was proposed by Stahlberg and Vogel [20]. A sliding window is used to extract features from text line images. The window's width is 3 pixels with an overlap of 2 pixels. The parts are extracted using two different strategies. The first strategy is pixel-based features extracted from raw grayscale pixel values. The second strategy is segment-based features, consisting of centroid and height features. Kaldi toolkit, which is used in speech recognition systems and is based on deep neural networks, is used for classification [21]. The best word error rate was obtained from pixel-based features with a 30.5% WER on the KHATT corpus.

Wigington et al. introduced two data augmentation and normalization methods: a novel profile normalization strategy for both word and line images and an augmentation of existing text images using random perturbations on a regular grid [22]. These techniques were used with a CNN-LSTM architecture to enhance handwriting text recognition. Contemporary youngsters frequently utilize technology, and their distinctive characteristics in handwriting differ from those of adults. Therefore, the study by Altwaijry et al. [23] has been trained on children handwriting.

The work by Khayati et al., [24] emphasizes the development and efficiency of several CNN architectures in tackling the distinct difficulties presented by Arabic script, such as writing with cursive and the existence of diacritical markings. Lamia et al. [9] developed a CNN-graph theory method for Arabic handwritten character segmentation. They address the difficulty of segmenting linked and overlapping cursive Arabic letters, a major obstacle to effective character identification. In a study by AlShehri [25], DeepAHR improves feature extraction and recognition with a complex neural network design. The model excels at Arabic script character segmentation and contextual changes. DeepAHR outperformed prior models in accuracy and processing speed.

In another study by Alghyaline [26], the Arabic handwritten recognition was implemented using different CNN pretrained models such as VGG, ResNet, and Inception on three different datasets (HIJJA, AHCD, and AIA9K). Each dataset achieved accuracies 93.05%, 98.30%, and 96.88% on VGG model. The transformer transducer and the typical transformer design that makes use of cross-attention were the two end-to-end architectures that were explored in the work by Momeni and BabaAli [27]. They employed KHATT dataset and obtained a CER of 18.45%.

3. Materials and Methods

The Arabic handwritten text recognition system should have multiple stages to convert a handwritten text image into a digital format. This system consists of four consecutive processing stages: preprocessing, feature extraction, classification, and post-processing. The output of each process is used as an input to the following process. First, preprocessing techniques are applied to the scanned image to improve the readability of the text. After that, the ResNet model is used to extract the features of each character in the text image. These features are the input for the classification stage. The BiLSTM-CTC network converts the visual features into contextual features. It predicts the sequence of characters with the help of the predefined classes in the database. Finally, an LM is used in the post-processing stage to enhance the predicted result from the classification stage. Figure 1 shows the workflow of our system. Each stage will be discussed in detail in the following subsections.

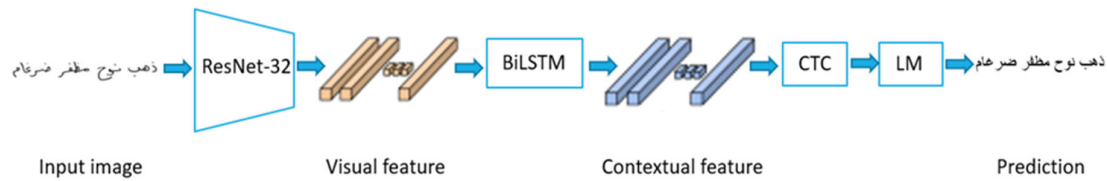


Figure 1. Our system workflow.

3.1. Datasets Description

The number of available Arabic handwritten text databases is limited. In our model, we have used two different databases, KHATT and AHTID/MW datasets, to train and test our model. These datasets contain all the Arabic characters written in different writing styles by different writers.

3.1.1. KHATT Database

The King Fahd University of Petroleum and Minerals (KFUPM) Handwritten Arabic Text database (KHATT) is one of the challenging Arabic handwritten text databases published by KFUPM [12]. The KHATT database is an offline Arabic handwritten text database consisting of text line images extracted from handwritten forms filled out by 1000 different writers. The writers are from different regions, gender, age, left/right-handedness, and educational background. The database consists of 300 Dots Per Inch (DPI) grayscale images of 2000 unique text paragraphs (randomly selected) and 2000 fixed text paragraphs (similar text). Also, the database contains 300 DPI binary text line images extracted from the paragraphs [28]. Figure 2 shows some samples from the KHATT dataset.

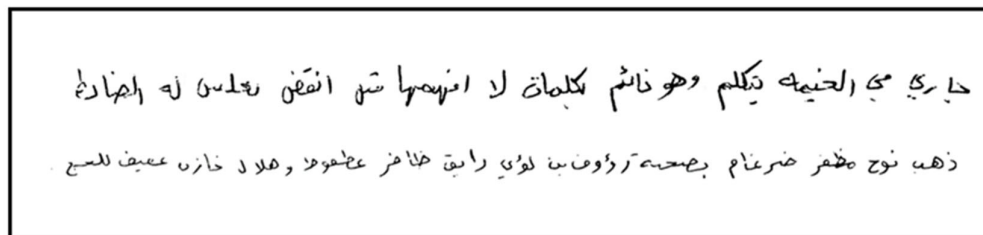


Figure 2. Samples from the KHATT dataset.

3.1.2. AHTID/MW Database

The Arabic Handwritten Text Images Database written by Multiple Writers (AHTID/MW) developed by [13] includes 3710 handwritten Arabic text lines and 22,896 words written by 53 native Arabic writers with different ages and educational levels. The dataset consists of grayscale text line images with 300 DPI resolution. The dataset contains a variety of Arabic handwriting writing styles, as shown in Figure 3.

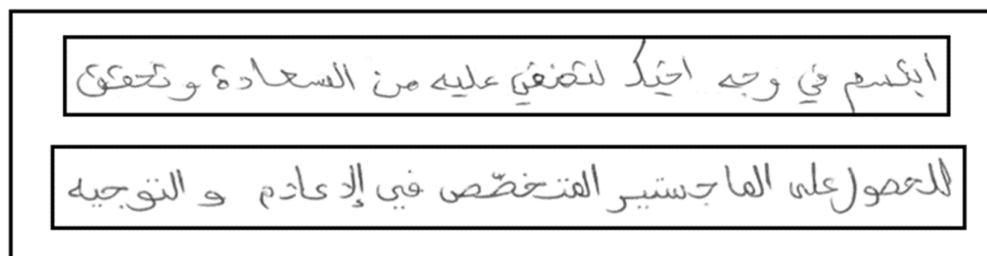


Figure 3. Samples from the AHTID/MW dataset.

3.2. Preprocessing

The preprocessing step for scanned text images is very critical. It improves the accuracies in handwritten text recognition systems. First, image binarization is applied to convert grayscale images. Each image uses values ranging from 0 to 256 for each pixel, to a black and white image represented by 1 and 0, respectively.

Arabic text line images datasets have high skew and extra white spaces. We removed the extra white regions by scanning the image from top to bottom to locate the first black pixel at the top and the position of the lowest black pixel at the bottom. After detecting the highest and lowest points of the black pixels the text line images are cropped [29]. The exact process is repeated from the left to the right side. Figure 4 shows a sample image from the KHATT dataset after removing the white spaces.

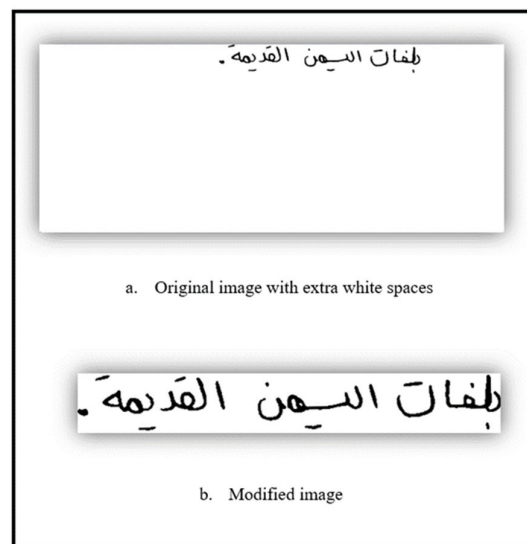


Figure 4. Sample image from KHATT dataset after removing the white spaces.

Handwritten text is freestyle handwriting; therefore, noises or unwanted text such as random lines and dots may exist in the text image. Some images in the dataset contain a straight horizontal line on top of the text. In this work, we removed the horizontal line by computing an image difference in the horizontal direction, and from the image difference, we can find the horizontal line in the image by searching the continuous difference value. Doing so, we can filter and get the horizontal line by setting a threshold value to determine whether a horizontal area can be considered a line. Suppose the length of the horizontal line is greater than the image width size divided by 10. In that case, the filtered horizontal line is removed from the image. Figure 5 shows a sample image from the KHATT dataset where the upper horizontal line is removed from the text line image.



Figure 5. Sample image from KHATT dataset after removing the upper line.

Noise filtering techniques are applied to remove noises from the text line images. Max and Min filters, also known as erosion (minimum) and dilation (maximum) filters, are used in the preprocessing stage to remove noises from the text images efficiently [30]. These filters are morphological transformation filters that define the neighborhood around each pixel. Erosion and dilation are two basic morphological operators [31].

First, erosion is applied to the text image to erode the foreground object. It makes it smaller, that is, removing small pixels (noises) near the boundaries of the foreground object (characters). Then, the dilation process is used to increase the size of foreground pixels (characters). We used erosion followed by dilation because erosion removes the noises in the text image, but it also shrinks the characters. Therefore, after the noises are removed from the text image, we dilate it. The dilation process enhances the distinctness of the characters and helps in joining broken parts of the text image together. An example of the Max and Min filter applied on a sample image from the KHATT dataset is shown in Figure 6.

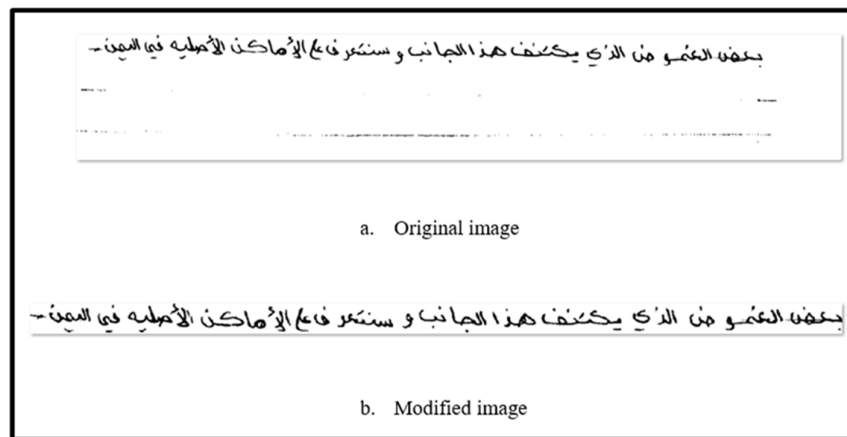


Figure 6. Sample image from KHATT dataset after applying Max and Min filter.

Image normalization is performed, which helps reduce the text image's skewness and facilitates characters' visual learning features. Arabic text contains two baselines: the upper and lower baselines, as shown in Figure 7. The two baselines identify the core zone, the upper region with the ascenders, and the lower region include the descenders. The core zone typically has a significant fraction of foreground pixels.

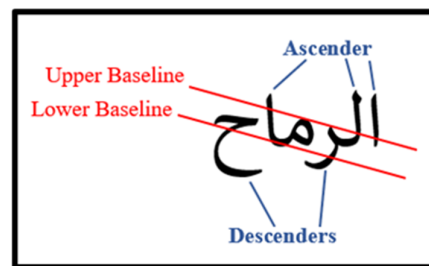


Figure 7. Upper and lower baselines in Arabic text.

We used a method proposed by Stahlberg and Vogel [32] for baseline estimation by finding stripes in the image with the dense foreground. First, we detect the baseline for the whole image and rotate the image so that the detected baseline is horizontal. After that, split the image vertically into smaller slices. Then, detect the baseline for each slice of the image separately and rotate the sliced images such that the baseline becomes horizontal. Finally, concatenate all the sliced images with a straight horizontal baseline to a single image.

3.3. Feature Extraction

Feature extraction is the second phase after the data has been preprocessed. Features are the main point around which the whole system is built. They are the target for all the previous stages and the input to the classification phase. Different feature extraction methods are applied in Arabic text recognition systems. Some approaches used handcrafted feature extraction techniques using statistical [33] or structural [34] features. Other approaches computed both statistical and structural features [35–37].

Recently, the new trend has shifted from handcrafted feature extraction methods towards machine learning techniques for feature extraction and text recognition systems. Deep networks, one of the most advanced machine learning techniques, simulate human brain activity and automatically extract features from text images. However, deep models for Arabic handwritten text recognition systems are rare compared to other languages due to their complexity and cursive writing style [38]. A convolutional neural network (CNN) is an artificial neural network used in the pattern recognition field for image processing and recognition. CNN's have proven their effectiveness in understanding image content, providing state-of-the-art image recognition and detection [39]. Therefore, since CNN's have shown their ability to learn interpretable and powerful features from an image [40], we adopted a CNN architecture in our model to extract features from the text image.

In our system, we used a ResNet-based model [41], which is a robust CNN architecture, to extract features from the text images. The CNN state-of-the-art architecture goes deeper each year since Krizhevsky et al. [42] presented AlexNet 2012. While AlexNet consisted of only five convolutional layers, the VGG (Visual Geometry Group) network had 16-19 convolutional layers [43]. Googlenet consisted of 22 deep convolutional layers [44].

However, enabling the model to learn better and more features by increasing network depth is not as simple as stacking more layers together. Deep networks are hard to train because of the vanishing/exploding gradient problem, where, as the gradient is backpropagated to earlier layers in the network, frequent multiplication might make the gradient infinitely small (i.e., vanish or explode) [45] [46].

The vanishing/exploding gradient problem makes it hard to learn and tune the parameters of the earlier layers in the network, which impedes convergence from the beginning. This results in the inability of models with deep layers to learn on a given dataset. The network performance with deep layers gets saturated or even starts degrading rapidly.

The ResNet model was introduced to overcome these issues. The main idea behind ResNet models is to use residual blocks to improve the accuracy of the models. Residual blocks are based on the concept of “identity shortcut connection” that skips/bypasses one or more layers. The input of the residual block is denoted by x , and the output is $H(x)$, which is the desired underlying mapping. The difference or residual between them is:

$$F(x) = \text{Output} - \text{Input} = H(x) - x \quad (1)$$

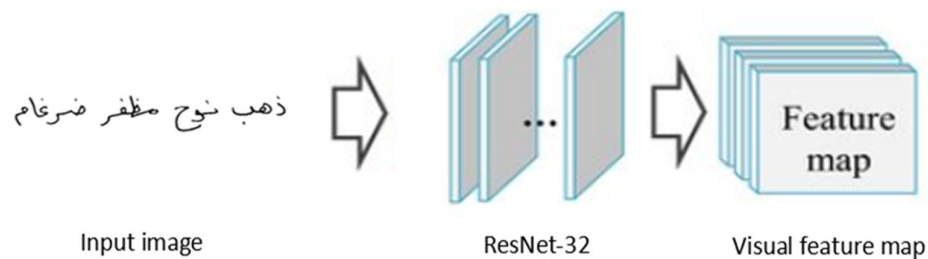
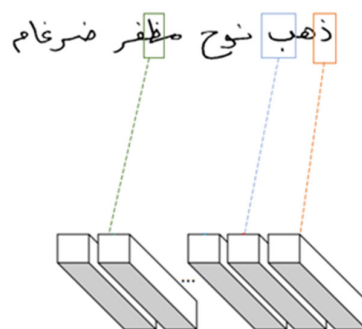
where $F(x)$ is the mapping of the stacked nonlinear layers. The original mapping is rearranged into $H(x) = F(x) + x$. The additional x operates as a residual, thus “residual block”. Therefore, ResNet solves degradation by adding the input of a layer to its output. As a result, ResNet improves the efficiency of deep neural networks with more layers and avoids poor accuracy as the model becomes deeper.

In our model, we build a 32-layer ResNet-based model to extract character features [47]. The details of the network are illustrated in Table 1. The convolutional layers in the table are shown in the following format: (kernel size, stride (width) × stride (height), pad (width) × pad (height), channels). The max-pooling layers are shown in the following format: (kernel size, stride (width) × stride (height), pad (width) × pad (height)). The residual blocks in the ResNet model are shown in Table 1 with a gray background having the following format: [kernel size, channels]. Each convolution layer in the residual blocks has stride 0 and zero padding.

Table 1. ResNet model architecture.

Layers	Configurations	Output
Conv1	$3 \times 3, 1 \times 1, 1 \times 1, 16$	64×1048
	$3 \times 3, 1 \times 1, 1 \times 1, 32$	
Pool 1: $2 \times 2, 2 \times 2, 0 \times 0$		
Conv2	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 1$	32×524
	$3 \times 3, 1 \times 1, 1 \times 1, 64$	
Pool 2: $2 \times 2, 2 \times 2, 0 \times 0$		
Conv3	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	16×262
	$3 \times 3, 1 \times 1, 1 \times 1, 128$	
Pool 3: $2 \times 2, 1 \times 2, 1 \times 0$		
Conv4	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 5$	8×263
	$3 \times 3, 1 \times 1, 1 \times 1, 256$	
Conv5	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 3$	3×263
	$2 \times 2, 1 \times 2, 1 \times 0, 256$	
	$2 \times 2, 1 \times 1, 0 \times 0, 256$	

The ResNet model is trained from scratch. The input of this stage is the normalized image. The output is a visual feature map containing each character's characteristic features in the image, as shown in Figure 8 and Figure 9.

**Figure 8.** ResNet model for text features extraction.**Figure 9.** Feature map for characters.

3.4. Classification

In this stage, after the features are extracted and the feature map is produced, which contains the qualities and characteristics of a sequence of characters, a classifier is used to generate characters with the help of predefined classes. Since we are dealing with long sentences that contain a sequence of characters and words, we used Bi-directional Long-Short-Term Memory (BiLSTM) to capture the context information of the sentence.

The BiLSTM model was proposed by Graves and Schmidhuber [48] and is robust as a classifier and in sequence recognition models in different natural-language processing (NLP) tasks such as speech recognition [49], natural language understanding [50], machine translation [51] and sentiment analysis [52].

The BiLSTM model consists of two LSTMs to process sequence information in two directions. The first is taking the sequence of inputs in the forward direction (past to future). The other is taking the sequence of inputs in the backward direction (future to past). Therefore, BiLSTM will efficiently extract the full-text context information since it has access to the previous and following context. Thus, we have used BiLSTM in our Arabic handwritten text recognition system.

Moreover, multiple BiLSTMs can be stacked together to have a deep BiLSTM model. The deep BiLSTM model allows a higher level of abstraction in data than a shallow model. Having a deep BiLSTM model improved the performance of the speech recognition system [53].

After the last BiLSTM layer, each column of contextual features is mapped to an output label. The Connectionist Temporal Classification (CTC) output layer, which was proposed by Graves et al. [54], is adopted to predict the probability of an output label sequence. CTC layer has several character outputs and one additional output known as 'blank.' The additional 'blank' output is helpful to avoid making decisions in uncertain zones, that is, in a low context area instead of being trained to constantly predict a character.

The Arabic language has 28 letters, and each letter has one to four forms. We added the 28 Arabic letters with their different forms for each letter, number, and punctuation mark and ended up with a class size of 135.

The probability given by CTC is defined as for input sequence Y , where $Y = y_1, \dots, y_T$, and T is the length of the sequence, the output is the probability of π , which is defined as:

$$p(\pi|Y) = \prod_{t=1}^T y_{\pi_t}^t \quad (2)$$

where $y_{\pi_t}^t$ is the probability of having a character π_t At time step t [55]. A sequence-to-sequence mapping function M is defined on the sequence π . The mapping function M maps π onto l , where l is the final prediction output, by removing the repeated characters first, then the blanks are removed. For example, M maps "--مم-ث-ال-لل-ل" onto "مثال," where "--" represents blank. The conditional probability is defined as the total sum of probabilities of all π that are mapped by M onto Y , as shown in Equation 3.

$$p(l|Y) = \sum_{\pi: M(\pi)=l} p(\pi|Y) \quad (3)$$

3.5. Language Model

To improve recognition accuracy, LMs are used in many NLP models, such as handwritten text recognition and speech recognition. In our system, we used an n-gram language model, which is a statistical language modeling technique. The n-gram language model is a probabilistic model that predicts the probability of a sequence of words in a text.

The n-gram language models are simple in their structure, easy to calculate the word occurrence probability and work best with high performance when trained on large amounts of data. In this work, a 3-gram language model was trained on the training corpus of the KHATT and AHTID/MW datasets. KenLM language model toolkit [56] was used to build the 3-gram language model. The KenLM toolkit is faster and uses less memory than other existing toolkits such as SRILM and IRSTLM, improving system runtime performance.

4. Results and Discussions

4.1. System Settings and Parameters

We used Python 3 and PyTorch tools and libraries to implement our model. The code was implemented using Amazon Web Services (AWS). We used Amazon Elastic Compute Cloud (EC2) with a 16GB NVIDIA V100 GPU.

The network configuration of our model is shown in Table 1. We used the architecture of the ResNet model to construct 32 trainable layers, which are a combination of convolutional layers with ReLU (Rectified Linear Unit) activation function, Max pooling layers with 2x2 filters, and batch normalization layers. It is beneficial to add the batch normalization technique for training our intense neural network. The batch normalization layers have the effect of stabilizing the learning process and accelerating the training process of the neural network. Our system applied a dropout layer after the ResNet model with a dropout ratio of 0.2. The second dropout layer is applied after the BiLSTM layers with a dropout ratio of 0.2.

Dropout, a stochastic regularization technique, is applied in our neural network. The dropout technique helps prevent overfitting and reduce interdependent learning amongst the neurons in neural networks by dropping out units (i.e., neurons) from the neural network during the training process.

The output of the ResNet model, which contains the extracted sequence of visual features from normalized text line images, is fed into the BiLSTM model with 512 hidden units to generate the contextual sequence. Different depths of BiLSTM layers are used to compare the performance of our model when adding more bidirectional LSTM layers. The first experiment was done using 2-layers of BiLSTM, and the second experiment was done on 3-layers of BiLSTM. The BiLSTM network is followed by the CTC decoder to translate the contextual feature sequence to the character sequence. The CTC decoder has 135 output units to generate characters and predict words.

Finally, we added a 3-gram language model to our system to improve the recognition accuracy. The KenLM toolkit, a fast and memory-efficient language model, is used to build a 3-gram language model. The language model will compare the weights assigned by CTC and LM. The predicted word with the highest weight will be replaced.

For optimization, we adopt the ADADELTA optimizer, which is a robust learning rate method that does not require the manual setting of a learning rate. We set the training batch size to 24, and all images are scaled to 1048x64 in both training and testing.

4.2. Performance Evaluation

The performance of handwriting recognition systems was evaluated in terms of Word Error Rate (WER) and Character Error Rate (CER). We used these two metrics to assess the performance of our system. The WER and CER are based on the concept of Levenshtein edit distance, which is the minimum number of edit operations required to transform the output text into the ground truth text. The editing operations are substitutions, insertions, and deletions necessary to convert the source string into the reference string. The Word Error Rate (WER) is calculated as follows:

$$WER = \frac{S_w + I_w + D_w}{N_w} \times 100 \quad (4)$$

where S is the total number of substituted words, I is the total number of inserted words, D is the total number of deleted words, and N is the total number of words in the evaluation set.

The Character Error Rate (CER) is calculated as follows:

$$CER = \frac{S_c + I_c + D_c}{N_c} \times 100 \quad (5)$$

where S is the total number of substituted characters, I is the total number of inserted characters, D is the total number of deleted characters, and N is the total number of characters in the evaluation set.

4.3. Experimental Results

The last stage of developing a handwriting recognition system is testing the system. This process used scaled images as input to the Arabic handwritten text recognition system. Two different datasets of Arabic handwritten text, that is, the KHATT and AHTID/MW, were used to cover all forms of the Arabic text. Therefore, characters and words of different forms and widths were used in our experiments.

The scaled images were passed through the ResNet model, followed by the BiLSTM-CTC layers and the language model post-processing stage. Table 2 and Table 3 show the results of our system using the KHATT and AHTID/MW datasets, respectively, with different BiLSTM layers.

Table 2. Our System Results for KHATT Dataset.

Model	CER%	WER%
2-BiLSTM Layers	15.8	31.6
3-BiLSTM Layers	13.2	27.31

Table 3. Our System Results for AHTID/MW Dataset.

Model	CER%	WER%
2-BiLSTM Layers	7.4	22.79
3-BiLSTM Layers	6.6	17.42

The recognition rates are improved in both datasets when using three BiLSTM layers. The WER is reduced by 4.29% for the KHATT dataset and by 5.37% for the AHTID/MW dataset. Therefore, the proposed model performs better, and the results are improved when using 3 layers of the BiLSTM network.

Additional experiments were done on the AHTID/MW dataset to test our proposed system performance. As seen below, the best performance is obtained by using 3-BiLSTM layers, which are 17.42% WER and 6.6% CER. Figure 10 and Figure 11 show the relation between the CER and WER with the epoch number, respectively. As shown in Figure 10 and Figure 11, the CER and WER decrease as the epoch number increases during the training process until it reaches epoch number 300. The results of our proposed system on the KHATT and AHTID/MW datasets confirm the robustness of our system.

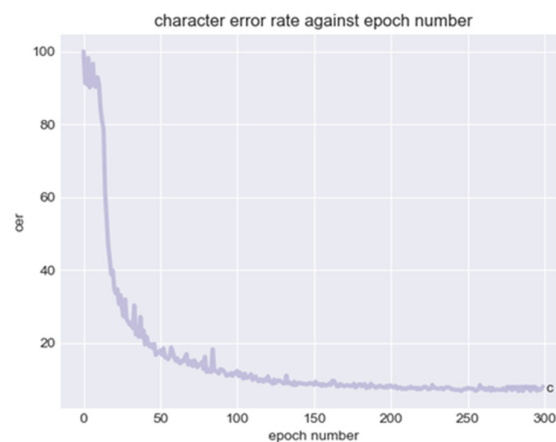


Figure 10. CER versus epoch number.

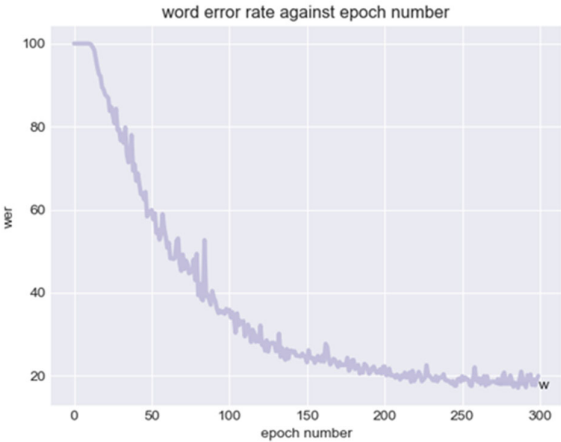


Figure 11. WER versus epoch number.

To validate our system performance, we compared our results with the most recent works of Arabic handwriting recognition systems. Table 4 shows the results of recent works obtained from the test set of the KHATT dataset. The experimental results showed that our system had an impressive recognition accuracy with a WER of 27.31% and a CER of 13.2% on the test set of the KHATT corpus. ResNet is a resilient CNN architecture designed specifically for extracting information from textual images. The fundamental concept underlying ResNet models is to employ residual blocks to enhance the precision of the models. Residual blocks rely on the idea of an "identity shortcut connection" that allows for the skipping or bypassing of one or more levels.

Table 4. Comparison Between Our Proposed System and the Existing Systems.

Reference	Year	Database	CER	WER
BenZeghiba et al. [18]	2015	KHATT Dataset	-	31.3%
Stahlberg & Vogel [20]	2015	KHATT Dataset	-	30.5%
Zeghiba [57]	2017	KHATT Dataset	-	34.3%
Jemni et al. [17]	2018	KHATT Dataset	16.27%	29.13%
Jemni et al. [58]	2019	AHTID/MW Dataset	-	18.13%
Momeni [27]	2024	KHATT Dataset	18.45%	-
Proposed Model	2024	KHATT Dataset	13.2%	27.31%
		AHTID/MW Dataset	6.6%	17.42%

5. Conclusions

We proposed a model for recognizing Arabic handwritten text. The system aims to identify Arabic handwritten text accurately by imitating the human brain to recognize text using machine learning approaches. The ResNet model was used for feature extraction, and BiLSTM-CTC sequence modeling was used for classification. Machine learning techniques are used to overcome traditional methods based on shallow learning and hand-engineered features. Moreover, machine learning approaches help overcome the challenges of recognizing Arabic handwritten text. A 3-gram language model was used in our system using the KenLM toolkit to improve the recognition accuracy of handwritten text.

Our proposed model was evaluated on the KHATT and AHTID/MW datasets. The experimental results showed that our system had an impressive recognition accuracy, 27.31%-word error rate (WER) and 13.2%-character error rate (CER) for the KHATT dataset and 17.42% WER and 6.6% CER for the AHTID/MW dataset.

5.1. Limitations and Future Works

The proposed study only employs a pretrained CNN model. Evaluating other transfer learning models is one among the future enhancement. Also, in future work different datasets can be combined to reduce the generalizability problem.

Author Contributions: Conceptualization, A.M. Mutawa and Mohammad Allaho; Data curation, Monirah Al-Hajeri; Formal analysis, A.M. Mutawa and Monirah Al-Hajeri; Methodology, Mohammad Allaho ; Project administration, A.M. Mutawa; Resources, Monirah Al-Hajeri; Software, Monirah Al-Hajeri; Supervision, A.M. Mutawa and Mohammad Allaho ; Validation, Mohammad Allaho ; Writing – original draft, Monirah Al-Hajeri; Writing – review & editing, A.M. Mutawa and Mohammad Allaho .

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are openly available [12,13].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Eberhard, D.M.; Simons, G.F.; Fennig, C.D. Gujarati. *Ethnologue: Languages of the world*, 22nd edn. Dallas: SIL International. **2019**.
2. Nashif, M.H.H.; Miah, M.B.A.; Habib, A.; Moulik, A.C.; Islam, M.S.; Zakareya, M.; Ullah, A.; Rahman, M.A.; Al Hasan, M. Handwritten numeric and alphabetic character recognition and signature verification using neural network. *Journal of Information Security* **2018**, *9*, 209.
3. El-Dabi, S.S.; Ramsis, R.; Kamel, A. Arabic character recognition system: a statistical approach for recognizing cursive typewritten text. *Pattern recognition* **1990**, *23*, 485-495.
4. Anis, M.; Maalej, R.; Elleuch, M. Recent advances of ML and DL approaches for Arabic handwriting recognition: A review. *International Journal of Hybrid Intelligent Systems* **2023**, *19*, 1-18, doi:10.3233/HIS-230005.
5. AlKhateeb, J.H.; Jiang, J.; Ren, J.; Ipson, S. Component-based segmentation of words from handwritten Arabic text. *International Journal of Computer Systems Science and Engineering* **2009**, *5*.
6. Nashwan, F.; Rashwan, M.A.; Al-Barhamtoshy, H.M.; Abdou, S.M.; Moussa, A.M. A holistic technique for an Arabic OCR system. *Journal of Imaging* **2018**, *4*, 6.
7. Boufekar, C.; Kerboua, A.; Batouche, M. Investigation on deep learning for off-line handwritten Arabic character recognition. *Cognitive Systems Research* **2018**, *50*, 180-195.
8. Alrobah, N.; Albahli, S. Arabic Handwritten Recognition Using Deep Learning: A Survey. *Arabian Journal for Science and Engineering* **2022**, *47*, 9943-9963, doi:10.1007/s13369-021-06363-3.
9. Berriche, L.; Alqahtani, A.; RekikR, S. Hybrid Arabic handwritten character segmentation using CNN and graph theory algorithm. *Journal of King Saud University - Computer and Information Sciences* **2024**, *36*, 101872, doi:https://doi.org/10.1016/j.jksuci.2023.101872.
10. Mosbah, L.; Moalla, I.; Hamdani, T.M.; Neji, B.; Beyrouthy, T.; Alimi, A.M. ADOCRNet: A Deep Learning OCR for Arabic Documents Recognition. *IEEE Access* **2024**, 1-1, doi:10.1109/ACCESS.2024.3379530.
11. Mahdi, M.G.; Sleem, A.; Elhenawy, I. Deep Learning Algorithms for Arabic Optical Character Recognition: A Survey. *Multicriteria Algorithms with Applications* **2024**, *2*, 65-79, doi:10.61356/j.mawa.2024.26861.
12. Mahmoud, S.A.; Ahmad, I.; Alshayeb, M.; Al-Khatib, W.G.; Parvez, M.T.; Fink, G.A.; Märgner, V.; El Abed, H. Khatt: Arabic offline handwritten text database. In *Proceedings of the 2012 International Conference on Frontiers in Handwriting Recognition*, 2012; pp. 449-454.
13. Mezghani, A.; Kanoun, S.; Khemakhem, M.; El Abed, H. A database for arabic handwritten text image recognition and writer identification. In *Proceedings of the 2012 international conference on frontiers in handwriting recognition*, 2012; pp. 399-402.
14. Mamouni El, M. An Effective Combination of Convolutional Neural Network and Support Vector Machine Classifier for Arabic Handwritten Recognition. *Automatic Control and Computer Sciences* **2023**, *57*, 267-275, doi:10.3103/S0146411623030069.
15. Alheraki, M.; Al-Matham, R.; Al-Khalifa, H. Handwritten Arabic Character Recognition for Children Writing Using Convolutional Neural Network and Stroke Identification. *Human-Centric Intelligent Systems* **2023**, *3*, 147-159, doi:10.1007/s44230-023-00024-4.
16. Elleuch, M.; Maalej, R.; Kherallah, M. A new design based-SVM of the CNN classifier architecture with dropout for offline Arabic handwritten recognition. *Procedia Computer Science* **2016**, *80*, 1712-1723.
17. Jemni, S.K.; Kessentini, Y.; Kanoun, S.; Ogier, J.-M. Offline Arabic handwriting recognition using BLSTMs combination. In *Proceedings of the 2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, 2018; pp. 31-36.

18. BenZeghiba, M.F.; Louradour, J.; Kermorvant, C. Hybrid word/Part-of-Arabic-Word Language Models for arabic text document recognition. In Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR), 2015; pp. 671-675.
19. Forney, G.D. The viterbi algorithm. *Proceedings of the IEEE* **1973**, *61*, 268-278.
20. Stahlberg, F.; Vogel, S. The qcri recognition system for handwritten arabic. In Proceedings of the International Conference on Image Analysis and Processing, 2015; pp. 276-286.
21. Povey, D.; Zhang, X.; Khudanpur, S. Parallel training of deep neural networks with natural gradient and parameter averaging. *arXiv preprint arXiv:1410.7455* **2014**.
22. Wigington, C.; Stewart, S.; Davis, B.L.; Barrett, W.A.; Price, B.L.; Cohen, S.D. Data Augmentation for Recognition of Handwritten Words and Lines Using a CNN-LSTM Network. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)* **2017**, *01*, 639-645.
23. Altwaijry, N.; Al-Turaiki, I. Arabic handwriting recognition system using convolutional neural network. *Neural Computing and Applications* **2021**, *33*, 2249-2261, doi:10.1007/s00521-020-05070-8.
24. El Khayati, M.; Kich, I.; Taouil, Y. CNN-based Methods for Offline Arabic Handwriting Recognition: A Review. *Neural Processing Letters* **2024**, *56*, 115, doi:10.1007/s11063-024-11544-w.
25. AlShehri, H. DeepAHR: a deep neural network approach for recognizing Arabic handwritten recognition. *Neural Computing and Applications* **2024**, *36*, 12103-12115, doi:10.1007/s00521-024-09674-2.
26. Alghyaline, S. Optimised CNN Architectures for Handwritten Arabic Character Recognition. *Computers, Materials and Continua* **2024**, *79*, 4905-4924, doi:10.32604/cmc.2024.052016.
27. Momeni, S.; BabaAli, B. A transformer-based approach for Arabic offline handwritten text recognition. *Signal, Image and Video Processing* **2024**, *18*, 3053-3062, doi:10.1007/s11760-023-02970-9.
28. Mahmoud, S.A.; Ahmad, I.; Al-Khatib, W.G.; Alshayeb, M.; Parvez, M.T.; Märgner, V.; Fink, G.A. KHATT: An open Arabic offline handwritten text database. *Pattern Recognition* **2014**, *47*, 1096-1112.
29. Ahmad, R.; Naz, S.; Afzal, M.Z.; Rashid, S.F.; Liwicki, M.; Dengel, A. Khatt: A deep learning benchmark on arabic script. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 2017; pp. 10-14.
30. Kaur, S. Noise types and various removal techniques. *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)* **2015**, *4*, 226-230.
31. Soille, P. *Morphological image analysis: principles and applications*; Springer Science & Business Media: 2013.
32. Stahlberg, F.; Vogel, S. Detecting dense foreground stripes in Arabic handwriting for accurate baseline positioning. In Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR), 2015; pp. 361-365.
33. Tavoli, R.; Keyvanpour, M.; Mozaffari, S. Statistical geometric components of straight lines (SGCSL) feature extraction method for offline Arabic/Persian handwritten words recognition. *IET Image Processing* **2018**, *12*, 1606-1616.
34. Mohamad, R.A.-H.; Likforman-Sulem, L.; Mokbel, C. Combining slanted-frame classifiers for improved HMM-based Arabic handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence* **2008**, *31*, 1165-1177.
35. Akram, H.; Khalid, S. Using features of local densities, statistics and HMM toolkit (HTK) for offline Arabic handwriting text recognition. *Journal of Electrical Systems and Information Technology* **2017**, *4*, 387-396.
36. Jayech, K.; Mahjoub, M.A.; Amara, N.E.B. Arabic handwriting recognition based on synchronous multi-stream HMM without explicit segmentation. In Proceedings of the International Conference on Hybrid Artificial Intelligence Systems, 2015; pp. 136-145.
37. Benouareth, A.; Ennaji, A.; Sellami, M. Semi-continuous HMMs with explicit state duration for unconstrained Arabic word modeling and recognition. *Pattern Recognition Letters* **2008**, *29*, 1742-1752.
38. Almodfer, R.; Xiong, S.; Mudhsh, M.; Duan, P. Multi-column deep neural network for offline Arabic handwriting recognition. In Proceedings of the International Conference on Artificial Neural Networks, 2017; pp. 260-267.
39. Zhao, Z.-Q.; Zheng, P.; Xu, S.-t.; Wu, X. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems* **2019**, *30*, 3212-3232.
40. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the European conference on computer vision, 2014; pp. 818-833.
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2016; pp. 770-778.
42. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* **2012**, *25*, 1097-1105.
43. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* **2014**, doi:10.48550/arXiv.1409.1556.
44. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2015; pp. 1-9.

45. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* **1994**, *5*, 157-166.
46. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Proceedings of the thirteenth international conference on artificial intelligence and statistics, 2010; pp. 249-256.
47. Cheng, Z.; Bai, F.; Xu, Y.; Zheng, G.; Pu, S.; Zhou, S. Focusing attention: Towards accurate text recognition in natural images. In Proceedings of the Proceedings of the IEEE international conference on computer vision, 2017; pp. 5076-5084.
48. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks* **2005**, *18*, 602-610.
49. Graves, A.; Jaitly, N.; Mohamed, A.-r. Hybrid speech recognition with deep bidirectional LSTM. In Proceedings of the 2013 IEEE workshop on automatic speech recognition and understanding, 2013; pp. 273-278.
50. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S.R. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461* **2018**.
51. McCann, B.; Bradbury, J.; Xiong, C.; Socher, R. Learned in translation: Contextualized word vectors. *arXiv preprint arXiv:1708.00107* **2017**.
52. Chen, T.; Xu, R.; He, Y.; Wang, X. Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Systems with Applications* **2017**, *72*, 221-230.
53. Graves, A.; Mohamed, A.-r.; Hinton, G. Speech recognition with deep recurrent neural networks. In Proceedings of the 2013 IEEE international conference on acoustics, speech and signal processing, 2013; pp. 6645-6649.
54. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the Proceedings of the 23rd international conference on Machine learning, 2006; pp. 369-376.
55. Shi, B.; Bai, X.; Yao, C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence* **2016**, *39*, 2298-2304.
56. Heafield, K. KenLM: Faster and smaller language model queries. In Proceedings of the Proceedings of the sixth workshop on statistical machine translation, 2011; pp. 187-197.
57. Zeghiba, M.F.B. Arabic word decomposition techniques for offline Arabic text transcription. In Proceedings of the 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), 2017; pp. 31-35.
58. Jemni, S.K.; Kessentini, Y.; Kanoun, S. Out of vocabulary word detection and recovery in Arabic handwritten text recognition. *Pattern Recognition* **2019**, *93*, 507-520, doi:10.1016/j.patcog.2019.05.003.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.