IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 5 January 2024, accepted 10 March 2024, date of publication 20 March 2024, date of current version 24 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3379530

RESEARCH ARTICLE

ADOCRNet: A Deep Learning OCR for Arabic Documents Recognition

LAMIA MOSBAH[1], IKRAM MOALLA[1,2], (Member, IEEE), TAREK M. HAMDANI[1,3], (Senior Member, IEEE), BILEL NEJI[4], TAHA BEYROUTHY[4], AND ADEL M. ALIMI[1,5], (Senior Member, IEEE)

[1]REsearch Groups in Intelligent Machines (ReGIM-Lab), National Engineering School of Sfax (ENIS), University of Sfax, Sfax 3038, Tunisia
[2]College of Computer Science and Information Technology, Al Baha University, Al Bahah 65511, Saudi Arabia
[3]Higher Institute of Computer Science Mahdia (ISIMa), University of Monastir, Monastir 5000, Tunisia
[4]College of Engineering and Technology, American University of the Middle East, Egaila 54200, Kuwait
[5]Department of Electrical and Electronic Engineering Science, Faculty of Engineering and the Built Environment, University of Johannesburg, Johannesburg 2092, South Africa

Corresponding author: Lamia Mosbah (lamia.mosbah@regim.usf.tn)

This work was supported by Tunisian Ministry of Higher Education and Scientific Research under Grant LR11ES48.

ABSTRACT In recent years, Optical character recognition (OCR) has experienced a resurgence of interest especially for contemporary Arabic data. In fact, OCR development for printed and handwritten Arabic script is still a challenging task. These challenges are due to the specific characteristics of the Arabic script. In this work, we attempt to address these challenges by creating a deep learning OCR for Arabic document recognition called ADOCRNet. It is a novel deep learning framework whose architecture is built of layers of Convolutional Neural Networks (CNNs) and Bidirectional Long Short-Term Memory (BLSTM) trained using Connectionist Temporal Classification (CTC) algorithm. In order to assess the performance of our OCR, the proposed system is performed on two printed text datasets which are P-KHATT (text line images) and APTI (word images). It's also evaluated on a handwritten Arabic text dataset IFN/ENIT (word images). According to the practical tests, the conceived model achieves strength recognition rates on the three datasets. ADOCRNet reaches a Character Error Rate (CER) of 0.01% on the P-KHATT dataset, 0.03% on the APTI dataset and a Word Error Rate (WER) of 1.09% on the IFN/ENIT dataset, which significantly outperforms the outcomes of the current systems.

INDEX TERMS Arabic, document recognition, CNNs, CTC, deep learning, BLSTM, OCR.

I. INTRODUCTION

Optical Character Recognition (OCR) technology converts an image of printed or handwritten text into a digital format. It is widely considered as a solved problem for Latin and non-Latin scripts, with many commercial and open source softwares. Despite the recent advancements in OCR technology, it has not attained the required level of accuracy to be useful for printed and handwritten Arabic documents. In fact, for Arabic scripts there is no deep system that can read printed or handwritten text in an unconstrained environment such as unlimited, multi-size, mixed-font vocabularies and great morphological variability. This multiformity complicates the choice of features for the extraction and recognition algorithms. Nevertheless, in handwritten Arabic texts, some different characters look similar, and it is even difficult for the human eye to find the difference [1]. There are big variances between the shapes of Arabic characters example, as shown in Figure 1. Furthermore, the same character can appear different depending on its position in the word: at the beginning, in the middle or at the end. In addition, the high similarity between certain handwritten characters presents another challenge in their classification. In order to address these challenges, the accuracy of text recognition

The associate editor coordinating the review of this manuscript and approving it for publication was Marco Anisetti.

أ - ب - ت - م - س - ي

FIGURE 1. Example of arabic characters.

heavily depends on robust features, along with a robust recognizer that can effectively adapt to variations. Various approaches can be employed to address the concept of image classification. However, machine learning algorithms are considered to be the most effective among them. Years ago, the idea behind these algorithms was proposed, but due to insufficient computational power, they could not be put into practice. Deep learning enables models to be trained more effectively and discern various levels of image representation. This field was transformed by convolutional neural networks [2], which learn the fundamental shapes in the initial layers and progress to grasp image features in the deeper layers. As a result, they achieve greater precision in image classification. Nevertheless, CNNs layers are not capable of determining the process of recognition in the case of studying text image data. It works in conjunction with Recurrent Neural Networks (RNNs) [3] trained using CTC [4] to identify various image representation levels. In this paper, we propose ADOCRNet as a supervised deep learning OCR for Arabic documents recognition that is capable of appropriate line and character recognition based on CNN, BLSTM and CTC layers. Our approach builds upon this established pipeline, altering the internal architecture of each component to enhance performance and address the specific challenges of Arabic text recognition. By elucidating these distinct elements, we intend to showcase the added value and innovations that our implementation brings to the existing landscape. The rest of the study is arranged as follows: Section II provides a detailed overview of related work within the same area. The proposed system for Arabic script recognition, ADOCRNet, is explained in Section III. Section IV presents the experiments and results. The acknowledgment is described in Section V, and the conclusion and future work are reviewed in Section VI.

II. RELATED WORK

Deep learning has received exceptional attention in recent years and delivered state-of-the-art outcomes in various fields including text recognition. However, it was not accurate enough to be useful for printed and handwritten Arabic manuscripts and the results are still to be improved. There are several methods before deep learning that we will not detail in this study. This section provides a literature review covering Arabic text recognition with deep learning techniques.

A. CONVOLUTIONAL NEURAL NETWORK METHODS

Convolutional neural networks (CNNs) are specialized neural networks that are specifically designed to capture localized (spatial) information from images. A framework for recognizing handwritten words adaptable to different languages was developed by Poznanski et al. in 2016 [5]. Rather than employing a holistic approach that recognizes an entire word as a single unit, they used attribute-based encoding in their framework. This encoding method describes the image in terms of whether or not it possesses a particular set of binary attributes. A CNN network inspired from the pretrained model VGG was used. They employed multiple standalone and parallel fully connected layers. Every layer results in a distinct set of attribute estimations. Overfitting was mitigated by incorporating techniques such as Dropout, weight decay, and data augmentation, specifically rotation and shear. The system achieved 97.07% on IFN/ENIT dataset. Arabic handwritten word recognition was performed using the Multi-Column Deep Neural Network (MCDNN) by Almodfer et al. [6]. This network architecture is structured as columns, with each column being a (CNN). The CNNs share the same architecture and are trained with the same data. However, each CNN preprocesses the data in their own way. Training an Testing the constructed MCDNN with three CNN columns achieved best outcome of 8.5% CER. In their study, El-Melegy et al. [7] examined the ability of CNNs to recognize Arabic handwritten numerical values found on bank checks. Additionally, they investigated the impact of data augmentation on enhancing the model's performance. Hence, an architecture with 17 layers for CNNs created, trained, and assessed using AHDB which is an Arabic handwritten database. The dataset was split into a set of train comprising 80% of the data and a set of test containing the remaining 20%. After applying data augmentation the model's performance improved from 96.58% to 97.8%. Shams et al. [8] introduced a system specifically designed for recognizing handwritten Arabic text. The proposed architecture employed three CNN layers and three max pooling layers to effectively recognize isolated Arabic characters. To enhance performance, a dropout operation applied to extract features efficiently and reduce processing time. Subsequently, a Support Vector Machine (SVM) employed to classify the extracted features into 28 classes, corresponding to the number of Arabic letters. Impressively, their approach achieved an accuracy of 95.07%. In their study, Altwaijry and Al-Turaiki [9] introduced a CNN architecture designed for recognizing isolated handwritten Arabic characters. The CNN consists of three Conv2D layers, each utilizing ReLU activation and max pooling. Subsequently, the output is flattened and fed into a fully connected layer with an applied dropout rate of 80%. The proposed approach was tested on two datasets AHCD and Hijja, yielding recognition rates of 97% and 88% respectively. In their research, Balaha et al. [10] suggested 14 different CNN architectures for handwritten Arabic OCR. These architectures involve three Conv2D layers and three max pooling layers. The variations among these architectures primarily lie in the number of filters and fully connected layers employed. Additionally, the researchers evaluated combinations of VGG16, VGG19, and MobileNetV2 models. Among the proposed architectures, CNN-5 emerged as a notable performer, achieving an impressive accuracy of 91.96% on the HMBD dataset. Mohd et al. [11] proposed a Quranic optical character recognition (OCR) system based on CNN followed by an RNN. They developed a Quranic OCR dataset which was based on the widely known edition of the Holy Quran (Mushaf Al-Madinah). The outcome of this work shows that the proposed system obtains an accuracy of 98% on the validation data, and a WRR of 95% and a CRR of 99% in the test dataset. In their 2021 publication, Ahmed and colleagues [12] introduced a novel CNN architecture. It consists of nine Conv2D layers, each utilizing 3×3 kernels, followed by batch normalization. Additionally, five 2×2 max-pooling layers are employed. Furthermore, dropout is applied after each max-pooling layer, with rates ranging from 0.1 to 0.4. The tensor is subsequently flattened by a fully connected layer, followed by another layer of the same type. The researchers demonstrated that their proposed method achieved an excellent accuracy of 99.94% on the HACDB (Handwritten Arabic characters database for automatic character recognition) when compared with the widely recognized VGGNet-19 model. In their 2022 study, Jbrail and Tenekeci [13] designed four distinct CNN architectures for isolated handwritten Arabic character recognition. These architectures vary in the number of layers, with options of 3, 9, and 13 layers. Additionally, they explored different activation functions utilizing both Relu and Softmax, and optimization algorithms including Gradient Descent and Adam. One of the architectures incorporated a deep neural network with nine hidden layers. These layers consist of Conv2D with 3×3 filters and max pooling with 3×3 kernels, with Relu and Softmax activations. The proposed approach attained a 99.3% accuracy on the Hijja dataset.

B. MULTIDIMENSIONAL RECURRENT NEURAL NETWORKS METHODS

An artificial neural network class known as a recurrent neural network (RNN) that can remember previous inputs in memory for a huge set of Sequential data. LSTM (Long Short-Term Memory) is a modified version of RNN. It shows greater potential for identifying patterns that are characterized by temporal intervals. It works better than traditional RNNs on tasks involving long time lags. The fundamental concept behind multidimensional recurrent neural networks (MDRNNs) [14] is to replace the single recurrent connection present in standard RNNs with as many recurrent connections as there are dimensions in the data. MDLSTM is the same as MDRNN except instead of using many RNN connections we use those of LSTM. There are many works using this type of neural network citing the most important: In their paper, Ahmad et al. [1] introduced a preprocessing technique that removes redundant white spaces and corrects text-line skewness to achieve precise height normalization before the MDLSTM (Multidimensional Short Long Term Memory) and CTC layers. The system achieved a recognition rate improvement of 29%, with an accuracy rate of 75.8% CRR on text-lines from the KHATT (KFUPM Handwritten Arabic Text Database) dataset. Maalej and Kherallah [14] proposed a system for offline Arabic handwriting recognition. MDLSTM and three novel architectures featuring Dropout, ReLU (Rectified Linear Unit), and Maxout were used. To verify the validation of the suggested models, data augmentation through morphological operations was utilized. Training the MDLSTM-CTC-Maxout model on the original IFN/ENIT dataset resulted in the highest accuracy of 92.59%. Using data augmentation, this model's accuracy was improved to reach 93.46%.

C. BIDIRECTIONAL LONG SHORT-TERM MEMORY METHODS

BLSTM (Bidirectional Long Short-Term Memory) consists of the coupling of 2 recurrent neural networks. The system accepts one-dimensional signal consisting of a sequence of frames ranging from 1 to T, and incorporates two contexts into the image (past and future). Thus, contextual information is taken into account from both handwriting directions (left-to-right and right-to-left). The reason for selecting this RNN architecture over MDRNN is because its input resembles that of HMMs, which involves frames being extracted through a sliding window. Also, it is favored for its capability to access long range context, learn sequence alignment and operate without the requirement of segmented data. Fasha et al. [15] suggested an OCR software approach for printed Arabic text. The architecture of this OCR is based on five CNNs layers followed by a BLSTM. The CNN part comprise five Conv2D layers with Relu for activation and five max pooling layers. The outcome from the CNN is passed to BLSTM (BDLSTM). The BLSTM consists of two layers of LSTM. The software was evaluated on the APTI dataset and attained 85.15% CRR and 76.3% WRR. Noubigh et al., [16] introduced a character model approach with a CTC decoder to develop a new architecture that combines CNN and BLSTM. They obtained 8% as a CER and 20.1% as WER on the handwriting Arabic database KHATT. Shtaiwi et al. [17] introduced a CRNN-BLSTM-based model for the recognition of Arabic handwritten text. The MADCAT dataset was employed for precise model training. This novel approach enables concurrent text detection and segmentation. It demonstrated impressive accuracy in parsing entire pages, segmenting them at the line level, and predicting their content. To assess its performance, a sizable, demanding dataset of Arabic handwritten documents was used, yielding a character-error rate of just 3.96.

D. OTHER METHODS

In 2018, Rahal et al. [18] suggested a comprehensive text recognition system that relied on statistical characteristics. The Bag of Features (BoF) model was implemented in their approach, where Sparse Auto-Encoder (SAE) was employed for feature representation and HMM was applied for recognition. For preprocessing, they employed Gaussian

TABLE 1. Comparisons of performance obtained on printed and handwritten Arabic databases.

	Paper	Year	Dataset	Architecture	Features extraction technique	Performance
Handwritten Arabic databases	Poznanski et al.	2016	IFN/ENIT	CNN+dropout	CNN inspired from VGG	97.07% CRR
	Almodfer et al.	2017	IFN/ENIT	MCDNN	CNN	8.50% CER
	Rahal et al.	2018	IFN/ENIT	HMM+SAE	BoF	99.00%
	El-Melegy et al.	2019	AHDB	CNN + data augmentation	CNN	96.58% CRR before data augmentation 97.80% CRR after data augmentation
	Shams et al.	2020	isolated Arabic characters	three CNN layers and three max pooling layers	a dropout operation	95.07% CRR
	Altwaijry et al.	2021	AHCD and Hijja	CNN+ fully connected layer +dropout	CNN	97% CRR on AHCD and 88% CRR on Hijja
	Balaha et al.	2021	HMBD	CNN+dropout	CNN inspired from VGG	91.96% CRR
	Ahmed et al.	2021	HACDB	9Conv2D+5 max-pooing +fully connected layer	CNN	99.94% CRR
	Noubigh et al.	2021	KHATT	CNN+BLSTM +CTC	CNN	8.00% CER 20.10% WER
	Jbrail et al.	2022	Hijja	CNN architectures	CNN	99.3% CRR
	Maalej et al.	2022	IFN/ENIT	MDLSTM-CTC-Maxout	raw pixels	92.59% CRR before data augmentation 93.46% CRR after data augmentation
Printed Arabic databases	Rahal et al.	2018	P-KHATT, APTI	HMM+SAE	BoF	99.00% CRR
	Fasha et al.	2020	APTI dataset	CNN+BLSTM	CNN	85.15% CRR and 76.3% WRR.
	Mohd et al.	2021	Quranic OCR dataset	CNN+RNN	CNN	95% WRR 99% CRR

smoothing to minimize the noise commonly present in text images and image re-scaling to ensure that all images had the same height. The average recognition accuracies that were obtained are 98.92% CRR for P-KHATT dataset, 99.95% for APTI dataset and 97.85% for IFN/ENIT. Tounsi et al. [19] suggested a hybrid BoF-SAE-HMM model to recognize words in Latin and Arabic natural scenes. Handcrafted features were utilized to achieve a mean recognition accuracy of 70.5% for Arabic script and 82.3% for Latin script. The Table 1 below illustrates the state of the art described in this section. The previously detailed methods have presented a successful application of deep learning architectures to recognize Arabic text. Numerous deep neural approaches have been used in different contexts: the diversity is highlighted in the use of various Arabic databases, the enhancement of feature representation, sequence modeling layers, and the adequate transcription algorithm. All these approaches participate in the construction of deep systems with incomparable recognition rates. Regarding the previous table, the Arabic databases do not represent the same degree of difficulty due to the background, noise and writing styles. For example, regarding the APTI database, the accuracy is very high (99%), because there are no regularities for this one. Unlike APTI, P-KHATT and IFN/ENIT, reveal a low accuracy of recognition and need more effort to obtain a performant recognition rate. Thus, they present several gaps. These gaps primarily involve the absence of suitably comprehensive and annotated datasets designed for offline Arabic text recognition. These recognition systems encounter challenges in precisely identifying and segmenting ligatures, which consequently results in transcription errors. Additionally, the diversity of fonts and sizes in printed Arabic text poses difficulties for the mentioned recognition systems to adapt and perform consistently. By addressing these research gaps in offline Arabic text recognition, the field can significantly enhance its recognition capabilities. We focus in this study on the exploitation of a deep learning architecture based on CNNs, BLSTM and CTC.

III. PROPOSED SYSTEM FOR ARABIC SCRIPT RECOGNITION: ADOCRNET

To surmount several prominent issues with Arabic text, including segmentation and the extraction of handcrafted features, many of the modern works on Arabic OCR have utilized deep learning methods. Utilizing the sequence context in text is being leveraged [20], with enhanced accuracy seen through the usage of deep learning. However, achieving adequate rates in many areas remains a challenge due to cost issues, particularly the recognition of printed and handwritten text, where current state-of-the-art systems have been unsuccessful. Our proposed ADOCRNet system architecture consists of CNNs layers for feature extractions, followed by a fixed number of BLSTM layers for the sequences modeling of words and/or lines of text. The CTC, which is the last output layer, predicts probabilities for the alphabet of Arabic script. The final output is the recognized text.

TABLE 2. Output shape, number of parameters and connected layers for ADOCRNet architecture.

Layer (Type)	Number: kernel size	Output Shape	Parameters	Total
The input(input Layer)	1	(None,1244,49,1)	0	Total parameters=1,908,105 Trainable parameters=1,908,105
conv1(Conv2D)	40:3*3	(None,1244,49,16)	20	
max1(MaxPooling2D)	-:2*2	(None,622,24,16)	0	
conv2(Conv2D)	60:3*3	(None,622,24,16)	4400	
max2(MaxPooling2D)	-:2*2	(None,311,12,16)	0	
conv3(Conv2D)	60:3*3	(None,311,12,16)	6800	
max3(MaxPooling2D)	-:2*2	(None,155,6,16)	0	
reshape(Reshape)	1:-	(None,150,96)	0	
dense(Dense)	1:-	(None,150,32)	12150	
lstm(BLSTM)	250:-	(None,150,512)	19456	
dropout	0.5:-	(None,150,512)	0	
Softmax(Activation)	1:-	(None,150,32)	0	
CTC output layer	1:-	-	-	

A. SYSTEM DESCRIPTION

A deep learning model is built for a dataset of image text to study the effect of the proposed architecture on the accuracy and the precision of the system. Our suggested deep learning model involves the use of convolutional layers and recurrent layers, along with a CTC output layer, to perform printed and handwritten Arabic OCR, as shown in Figure 2. Using the CNN–BLSTM combination based on the number of layers, a fixed learning rate of 0.001 and a batch size of 20 by testing different values, are considered as the right candidate for our system. Table 2 displays the output shape of each layer in the network, the count of parameters; and Whether the layer is joined to another layer or not through the Keras function summary. This platform, which is a Python interface for artificial neural networks, is built on top of popular deep learning frameworks (TensorFlow) [21].

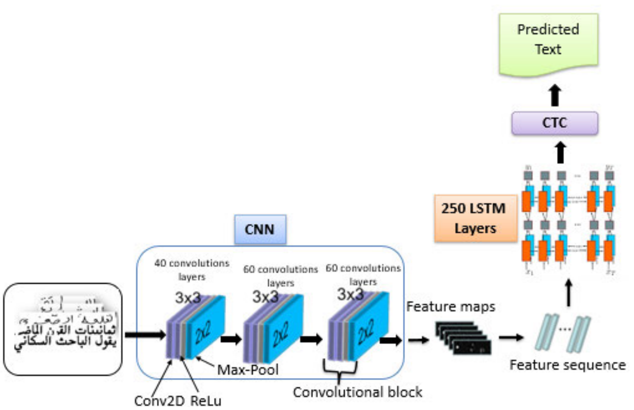


FIGURE 2. ADOCRNet architecture based on CNNs layers for features extractions, BLSTM layers for sequences modeling and CTC for specifying text images with the labeled text.

1) FEATURE EXTRACTION USING CNN

The input comprises an image of text treated as a sequence of pixels. Consequently, each image in the dataset is fed into the network as grayscale values, containing a single channel. Feature extraction is performed by applying a convolutional layer (first layer) with a 4 × 4 filter on the input. We use ReLU as the activation function [22]. This layer accelerates the neural network train phase by simplifying gradient computation to either 0 or 1. Following this, a Max Pooling layer (second layer) [23] with a 2×2 window, which reduces computational load and the number of parameters. The first and the second layer together constitute the first CNN block. A second and a third block of CNN are added to extract features that are more sophisticated. The features results of the 3 CNNs blocks are followed by the reshaping step to prepare them for the next layers. Following this stage, a dense layer takes over to classify and compress the features prior to passing them on to the BLSTM layers.

2) SEQUENCE MODELING USING BLSTM

In LSTM, 250 layers of BLSTM are used. Concatenated to the output of these layers, we apply a dropout [24] by a percentage of 0.5 to avoid overfitting. Then, a SoftMax function serves as the activation function for the output layer in neural network models that make predictions based on a multinomial probability distribution.

3) DECODING PHASE USING CTC

CTC [25] is introduced to align the frame-wise probability with labels. It employs the matrix derived from the BLSTM layers for two purposes: calculating the loss value for training and decoding it to identify the text within the input image. In this case, the CTC loss function was only provided with the output matrix and its matching ground truth text (GTT). It computed the total of whole scores by considering every feasible alignment of the GTT inside the image. Thus, the GTT score is considered elevated when the total of alignment scores yields a significant value. We use a CTC output layer in our model. It consists of a CTC decoder and a CTC loss function. The CTC loss function is applied to compare the output sequence generated by the model with the ground truth sequence of arabic characters from the handwritten input (arabic text image). Thus, during the training of our model, it is used for calcuating the disparity between the predicted sequence and the actual sequence of characters. Minimizing it using gradient descent, our model parameters are tuned to achieve a higher performance on the recognition of printed or handwritten arabic text. By permitting repetitions of characters and blank labels, the CTC decoder enables the prediction of a single label in multiple time steps. CTC generates a dataset of text line images and assigns a shape to every horizontal pixel in the image. A shape can be one character or more characters. It can recognize the pixels of the shapes without any character distinction. Therefore, CTC simplifies the training process of the system by only requiring the labeling of text-line images. The requirements regarding:

- Software:
  Keras is a high-level neural network API written in Python and designed for fast experimentation. It runs on TensorFlow and can be used in Jupyter notebooks with Python 3.6. It can also work on top of other popular deep learning libraries like Microsoft Cognitive Toolkit and Theano [26].
  Google Colaboratory is a Jupyter notebook that works on the cloud and does not require any setup, and is available for free, it was utilized with a K80 GPU for up to 12 hours per session [27].
- Hardware
  We highly recommend using a computer with a processor that is less than 5 years old, Processor: with minimum 1 GHz frequency, Ethernet connection (LAN) OR a wireless adapter (Wi-Fi), Hard Drive with Minimum 32 GB storage and a memory (RAM) of minimum 1 GB.

B. GENERAL ARCHITECTURE DESCRIPTION

In the last decade, there was a massive progress brought by the deep neural network in the computer vision community [28]. Insufficient data cannot adequately train a robust deep neural network, as the model may overfit to the training set, resulting in poor performance when tested on a different set [29]. Collecting and annotating textual data demands significant resources due to the presence of multiple characters within a text image. The data limitation affects printed and handwritten text recognition. There is a diverse range of writing styles available. Collecting large scale annotated text images is high-cost and cannot cover all diversities. The existence of these limited training samples creates a need to perform data augmentation in many deep learning approaches. Thus, data augmentation is used to increase the size of a dataset, and involves creating transformed versions of its images. Arabic writing has many styles of writing. It's difficult to find a complete database with thousands of images, nor to build it. This is the case of databases tested for this study.

1) DATA AUGMENTATION FOR TEXT IMAGE

Data augmentation is a technique of injecting knowledge by considering the invariant properties of the data with respect to certain transformations. Regarding text image transformation, the augmented data (text image) must be readable as the input images and characters must keep their original shape and not be visually unreadable. Popular data augmentation techniques could be color space transformations, kernel filters, mixing images, random erasing, feature space augmentation, adversarial training, GAN-based augmentation, neural style transfer, meta-learning schemes and geometric transformations. Existing work on augmenting text has not covered all methods of augmentation. Their effectiveness remains to be proven. In our case, we choose to apply the geometric transformations, specifically distortion, stretch and perspective transformations.

a: GEOMETRIC DISTORTION

Incorporating random scaling, cropping, flipping, and rotating into the data augmentation process can be especially challenging. Random Scaling consists of randomly changing the scale of the image between a specified range. cropping is used to handle images with varying height and width, this method was employed where a central patch was extracted from every image in the dataset. In addition, this technique can also result in an outcome that closely resembles translations. For the Flipping, implementing this augmentation technique is one of the simplest methods and has demonstrated its effectiveness on CIFAR-10 and ImageNet data. Then, the process of rotating an image to the right or left by an axis between 1° and 359° is employed to carry out rotation augmentations. The degree of rotation parameter significantly impacts the protection of utilizing rotational augmentations.

b: CONTRAST STRETCHING

Contrast stretching is a straightforward image processing technique that adjusts the contrast of an image by rescaling or "stretching" the range of intensity values to a specified range.

c: PERSPECTIVE

Perspective allows producing new images captured from any camera's viewpoints. This method can be used to simulate images captured from angles that are inaccessible to the camera.

We use data augmentation to obtain more considerable datasets. For every image of the P-KHATT dataset, we generated 12 images by the distortion transformation. Also, the same number of images was generated for the stretching method, and 12 images for the perspective transformation. We present some results of augmented data as described in the Figure 3, Figure 4 and Figure 5 below:

2) TRAINING DETAILS

In our case, we feed images of text to the neural network and get labeling text. In this section we want to follow the recognition process in the text incorporated in this image. Firstly, the text image is fitted to a CNN to abstract image features. Second, a recurrent neural network is utilized to process these features, followed by the application of a

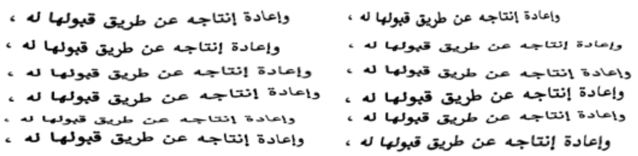


FIGURE 3. Samples of applying distortion on an image from P-KHATT database.

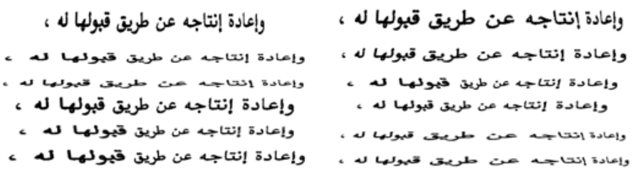


FIGURE 4. Samples of applying the stretch transformation on an image from P-KHATT database.

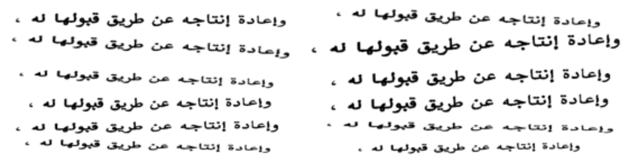


FIGURE 5. Samples of applying perspective transformation on an image from P-KHATT database.

decoding algorithm that takes the output of the BLSTM at each time step to generate the output (predicted text).

Let us consider this prototype in detail. As an example, we consider an image with the shape: height=119 and a width=449 and a number of channels equals to 1 (Figure 6). The image is passed through a feature extractor using CNN, resulting in a tensor with shape (4*4*4). Next, we do a reshape iteration. The slice tensor stretches the 4*4 matrix to a vector of 16 elements. Thus, we obtain a matrix of 4 vectors with 16 elements. Afterward, we feed the first vector to the BLSTM network and get its output. Then, a fully connected layer is applied, followed by a SoftMax layer, resulting in a vector consisting of 4 elements. The vector represents the probability distribution of observing

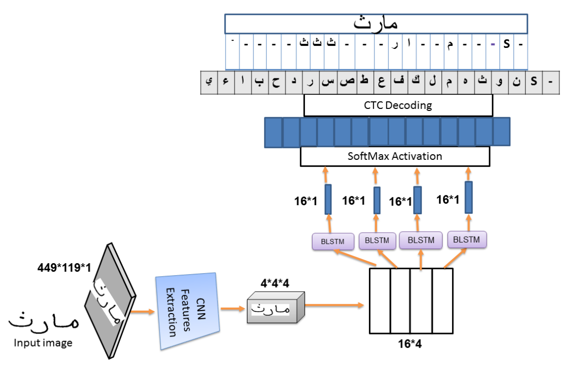


FIGURE 6. The high-level structure of the deep Neural Network used in ADOCRNet.

alphabet symbols after the first BLSTM step. We have four vectors of probabilities. They use their feedback connections to store representations of recent input characters in the form of activations, preserving the traceability of the characters. Outputs from all time steps are fitted to the decoding algorithm. At the first time step, let us select the symbol with the highest probability. Here it is م Also, we do the same at next time step and get the letter ا and repeat the same action for all other time steps. For each time step, we select the letter with the highest probability. If there are consecutive repeating characters, we combine them into a single character. We expanded our alphabet by adding blank symbols to distinguish between characters. After removing all blanks, we obtained the string of four characters, which was the predicted labeling of Arabic word(output). Therefore, we understand how to do forward paths for our network. This simple example explains the most important concepts of our architecture.

IV. EXPERIMENTS AND RESULTS

To evaluate both the performance and generality of our system, we utilized three datasets to demonstrate the outcomes across various types of data, as outlined in the subsection below A. We used APTI and IFN/ENIT to evaluate our system on word level. To evaluate ADOCRNet at text line level, we used P-KHATT with various writing styles and mixed fonts. We conducted a comparative study with other tested systems to demonstrate the effectiveness of our approach across multiple datasets. Furthermore, in subsection B, we described our experiments. Finally, we presented in subsection C, a detailed evaluation of our system

A. DATASETS

1) IFN/ENIT DATASET

The IFN/ENIT [30] is used in its v2.0p1e version, which consists of 32,492 Arabic handwritten words comprising 937 village and town names from Tunisia. More than 400 writers have contributed to the collection of handwritten words included in this database. The collection comprises over 2200 binary images of handwriting sample forms. It is divided into five sets: a-e. We present some training-testing configurations used in our experiments: abc-d and abcd-e. The evaluation of results is based on the calculation of WER.

2) APTI DATASET

APTI [31] composed of gray-scale images of words produced in 10 Arabic fonts (Arabic Transparent, Tahoma, Andalus, Advertising Bold, Simplified Arabic, Traditional Arabic, Diwani Letter, M Unicode Sara, Naskh, and DecoType Thuluth), 10 font sizes (6, 8, 10, 12, 14, 16, 18 and 24 points), and four font styles (plain, bold, italic, and combination of italic bold). APTI is selected as the benchmark dataset for evaluating OCR systems. It comprises 45’313’600 single word images summing to more than 250 million characters.

3) P-KHATT DATASET

P-KHATT (Ahmad et al., 2016) includes 9310 text line images originating from eight fonts: Akhbaar, Andalus, Naskh, Simplified Arabic, Tahoma, DecoType Thuluth, Times New Roman, and Traditional Arabic. These images were distributed on 6472 for training, 1414 for testing, and 1424 for evaluation purposes. The text is digitized at 300 dots/inch resolution. The dataset comprises 28 Arabic letters in various forms and combinations, along with spaces, 10 digits, and punctuation marks ('.', ',', ':', ';', '!', '(', ')', '?', '−', '/', '%'...).

B. EXPERIMENTS: TEST PHASE

After training ADOCRNet as described previously we follow a series of tests done on P-KHATT and APTI to test the recognition of our system on printed Arabic scripts. The IFN/ENIT is used to evaluate the recognition performance on handwritten Arabic scripts.

TABLE 3. Size of training, validation, and test partitions used in experiment datasets.

Real Datasets	Augmented datasets
P-KHATT: *6472 images for training *1424 images for evaluation *1414 images for testing	Augmented P-KHATT: *232992 images for training *1424 images for evaluation *1414 images for testing
IFN/ENIT *2603 images for training *2600 images for evaluation *6735 images for testing	Augmented IFN/ENIT *710064 images for training *2600 images for evaluation *6735 images for testing
APTI *11328 images for training *11328 images for evaluation *11328 images for testing	Augmented APTI *135936 images for training *11328 images for evaluation *11328 images for testing

C. RESULTS

1) PERFORMANCE OF ADOCRNET ON PRINTED ARABIC SCRIPT RECOGNITION

These tests are calculated on the P-KHATT. The ADOCRNet represents more flexibility than the hard clustering algorithm, the SAE+HMM developed by (Rahal et al.,2018). Table 4 displays the results obtained, which illustrate that ADOCRNet outperforms SAE and HMM. It raises an improvement of up to 0.02% compared to this system applied on PKHATT. Furthermore, the ADOCRNet yielded the highest average text recognition accuracy of 99.97% for the CRR when tested on the Tahoma mono-font. It raises a progress of up to 19% compared to (Fasha et al., 2022) system applied on KHATT.

2) PERFORMANCE OF ADOCRNET IN RECOGNITION OF DIFFERENT ARABIC FONTS

The results obtained are applied on P-KHATT. The tests are affected without training on the respective font. The results are presented in a CER expression. The obtained results prove that ADOCRNet is a very performant system in recognizing any type of font. The effectiveness of the CNN-BLSTM-CTC architecture is demonstrated in addressing the challenge of morphological variations in character features across different fonts. In Table 5, the comparison with (Rahal et al., 2018) proves an important improvement of our system for Naskh, Akbaar and Traditional Arabic. The results obtained by ABBYY and Tesseract are significantly poor and cannot be compared with our own results. For P-KHATT, CNN-BLSTM-CTC architecture enables a consistent and robust parameterization in both mono-font and mixed-font contexts. The achieved results exhibit great promise in accurately recognizing Arabic text in unconstrained environments (line or character, mono-font or mixed context).

Table 6 presents a comparison of our system with HMM-based systems reported in the literature, employing the APTI dataset. We use Arabic Transparent font for the training phase because, it is the most used in text recognition competitions for this goal. We obtain an improvement of 0.02% at the character level and improvement of 0.07% at the word level. In contrast to ABBYY and Tesseract [32], ADOCRNet demonstrates significantly improved recognition rates. The obtained rates show the stability of our selected deep learning architecture.

Table 7 displays a comparison between our system's recognition results and those of other related work on the IFN/ENIT dataset using the same parameters for training and testing. (El Moubtahij et al., 2017) yielded weak recognition rates, 21.95% for "abc-d" despite the application of two local densities and three statistical models for feature extraction. The system proposed in (Rabi and Amrouch, 2018) was more developed in (Amrouch et al., 2019) where CNN/HMM model is used to replace the hand-engineered feature/HMM model. The latter showed a WER of 11.05%, with a total reduction in the error rate of 1.02%. However, this suggested that the model was affected by the insufficient amount of data used during the training stage. In order to address this significant issue, the proposed system, based on CNN/BLSTM in (Maalej et al., 2018), resorted to the data augmentation technique, but the achieved outcome of 7.79% stay a worse result. Also, (Rahal et al., 2018) using SAE based on BoF/HMM, effects an overall decrease in the error rate with 0.8%. It remained worse than the result of our system. (Maalej et al., 2022), using MDLSTM-CTC-Maxout and the data augmentation technique. They obtained 6.64% WRR. We reach a more reduced value 1.09% as WER for abc-d configuration. For these reasons, our system's results surpass those of the previously mentioned systems. The revealed outcomes demonstrate that the recognition rates are comparable to the top proposed systems.

3) IMPACT OF DATA AUGMENTATION IN IMPROVING THE PERFORMANCE OF ADOCRNET

Confirmed to the previously announced results, we note in Figure 7 and Figure 8 that the models generated without data augmentation for the P-KHATT database present a huge number of error classes. For example, 43 in the epoch=150, 34 in the epoch=200 (an error class can be a letter confused

TABLE 4. Comparison of text recognition rates via SAE and ADOCRNet on P-KHATT and the system of Fasha et al using KHATT lines.

System	Tahoma		Mixed-font using a random number of samples from all fonts		Mixed-font using a fixed number of samples from all fonts
	CRR(%)	LRR(%)	CRR(%)	LRR(%)	LRR(%)
(Rahal et al., 2018) SAE+HMM P-KHATT	99.95	99.40	98.92	90.00	99.98
Fasha et al., 2022 CNN+BLSTM+CTC KHATT	-	-	-	-	80.15
ADOCRNet, 2023 CNN+BLSTM+CTC+ augmentation techniques P-KHATT	99.97	99.45	99.00	92.01	99.99

TABLE 5. Comparison of text recognition rates via ABBYY, Tesseract, SAE and ADOCRNet on different fonts of P-KHATT.

Font / CER	Times New Roman	Andalus	DecoType Thuluth	Tahoma	Traditional Arabic	Naskh	Akbaar	Simplified Arabic
ABBYY	–	62.47	62.29	30.09	32.34	49.78	–	32.31
Tesseract	–	74.66	67.52	61.63	52.96	59.08	–	53.25
(Rahal et al. 2018)	0.10	0.04	0.29	0.05	0.23	0.24	0.35	0.04
ADOCRNet	0.08	0.03	0.30	0.05	0.19	0.12	0.30	0.05

TABLE 6. Comparison of the performance of ADOCRNet and other OCRs using APTI dataset.

Features	Train set	Test set	CER(%)	WER(%)
Structural and statistical (Slimane et al., 2011)	Sets{1-5}	Set6	0.30	1.10
Statistical (Ahmad et al., 2016)	Sets{1;2}	Set5	0.57	2.12
SAE+BoF+HMM (Tounsi et al., 2021)	Sets{1;2}	Set5	0.05	0.17
ABBYY (commercial). 2022	Sets{1;2}	Set5	–	25.2
Tesseract (open source). 2022	Sets{1;2}	Set5	–	29
CNN+BLSTM+CTC Ours	Sets{1;2}	Set5	0.03	0.10

TABLE 7. Comparison with state-of-the-Art systems using IFN/ENIT.

Systems/ Features and models		Training-Testing configurations	
		abc-d WER(%)	abcd-e WER(%)
EL-Moubtahij et al., 2017	Local densities and statistics + HMM	21.95	-
Rabi et al, 2018	Statistical and Structural + HMM	12.07	14.28
Amrouch et al., 2018	CNN + HMM	11.05	10.77
Maalej et al., 2018	CNN + BLSTM	-	7.79
Rahal et al., 2018	SAE based BoF + HMM	2.15	5.03
Maalej et al., 2022	MDLSTM-CTC-Maxout	-	6.64
ADOCRNet	CNN+BLSTM+CTC	1.09	4.02

TABLE 8. Comparison of ADOCRNet on P-KHATT, IFN/ENIT and APTI datasets.

dataset	P-KHATT	IFN/ENIT	APTI
CER(%)	0.01	1.09	0.03

with another letter example ("yaE") predicted ("baE") or a letter confused with a sequence of letters example ("toA") predicted ("taA-laA"). The fact that we didn't use a supervised learning of characters. Figure 9 and Figure 10 present the error classes for the models generated using the augmented P-KHATT database in the epochs 60 and 90. So we conclude that ADOCRNet presents a performing model in a reduced time (epoch=90) with a limited number of error classes. Working with data augmentation, we have improved the representativeness of characters' shape. This raises the recognition and the system converge easily. Table 8 presents the CERs obtained by ADOCRNet on P-KHATT,

TABLE 9. Comparison of ADOCRNet models with real data or with augmented data using P-KhATT dataset.

Model	Epoch	Number of error classes	Figure
Model1 with real data	150	43	Figure6
Model2 with real data	200	34	Figure7
Model3 with data augmentation	60	29	Figure8
Model4 with data augmentation	90	3	Figure9

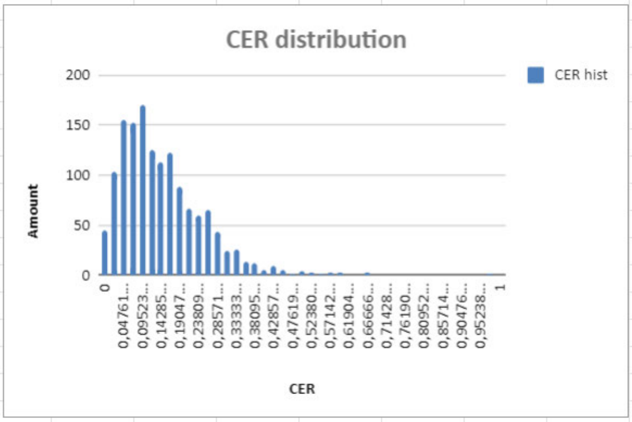


FIGURE 7. Model generated at epoch=150 for real P-KHATT.

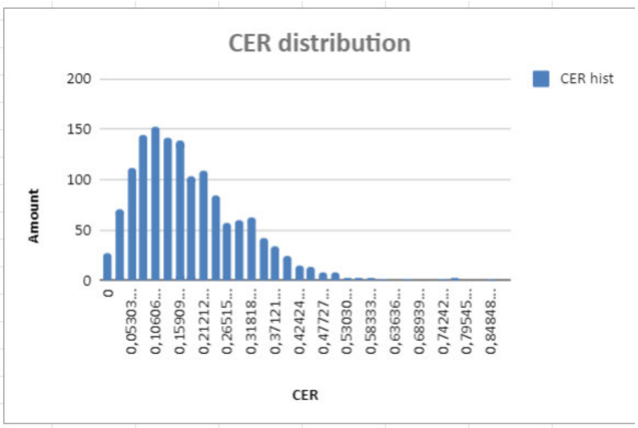


FIGURE 8. Model generated at epoch=200 for real P-KHATT.

IFN/ENIT and APTI datasets. These results demonstrate the efficiency of using morphological data augmentation techniques and the CNN-LSTM-CTC architecture.

Our research introduces an OCR engine named "ADOCR-Net". It represents a significant leap forward in text recognition accuracy and efficiency Through the fusion of morphological data augmentation techniques and adaptive CNN-LSTM-CTC architecture. ADOCRNet demonstrates unparalleled capabilities in accurately deciphering complex text formats for unlimited vocabulary, mixed-font Arabic text recognition in Arabic images.

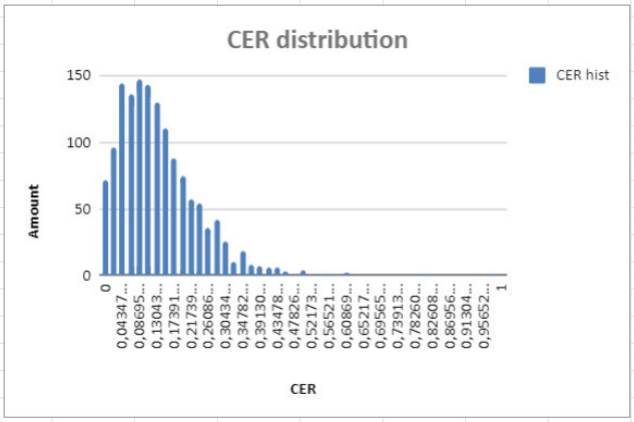


FIGURE 9. Model generated at epoch=60 for real P-KHATT.

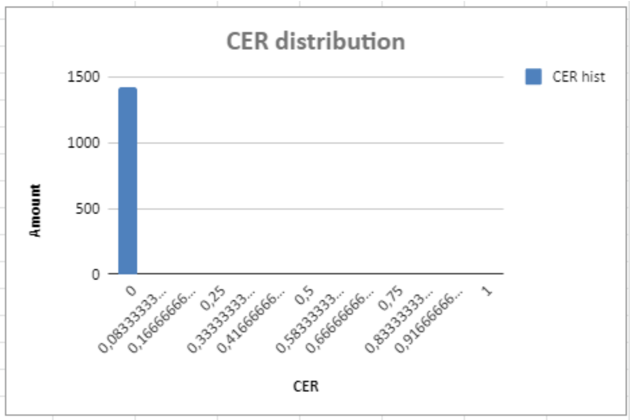


FIGURE 10. Model generated at epoch=90 for real P-KHATT.

V. CONCLUSION

This paper focuses on the presentation of ADOCRNet as a new OCR engine for unlimited vocabulary, mixed-font Arabic text recognition in Arabic images. It is based on the CNNs layers for features extractions, followed by a fixed number of BLSTM layers for the modeling of sequences and the CTC for the labeling of text. Herein, the main advantage of ADOCRNet is their ability to handle the specificity of fonts after training on mixed fonts, showing complex shapes. In other words, we investigate the exceptional role of data augmentation in the training step to obtain excellent recognition rates.The obtained results in this paper provide encouragement for more future work. The data augmentation by its different types applied on historical documents and natural scene images can be investigated. We will test the system on other scripts to show the genericity of using it in other languages.

REFERENCES

[1] R. Ahmad, S. Naz, M. Z. Afzal, S. F. Rashid, M. Liwicki, and A. Dengel, "KHATT: A deep learning benchmark on Arabic script," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 7, Nov. 2017, pp. 10–14.

[2] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022.

[3] I. J. L. Paul, S. Sasirekha, D. R. Vishnu, and K. Surya, "Recognition of handwritten text using long short term memory (LSTM) recurrent neural network (RNN)," in *Proc. AIP Conf.*, 2019, Art. no. 030011.

[4] X. Huang, L. Qiao, W. Yu, J. Li, and Y. Ma, "End-to-end sequence labeling via convolutional recurrent neural network with a connectionist temporal classification layer," *Int. J. Comput. Intell. Syst.*, vol. 13, no. 1, pp. 341–351, 2020.

[5] A. Poznanski and L. Wolf, "CNN-N-gram for HandwritingWord recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2305–2314.

[6] R. Almodfer, S. Xiong, M. Mudhsh, and P. Duan, "Multi-column deep neural network for offline Arabic handwriting recognition," in *Artificial Neural Networks and Machine Learning—ICANN 2017: 26th International Conference on Artificial Neural Networks, Alghero, Italy, September 11–14, 2017, Proceedings, Part II 26*. Springer, 2017, pp. 260–267.

[7] M. El-Melegy, A. Abdelbaset, A. Abdel-Hakim, and G. El-Sayed, "Recognition of Arabic handwritten literal amounts using deep convolutional neural networks," in *Pattern Recognition and Image Analysis: 9th Iberian Conference, IbPRIA 2019, Madrid, Spain, July 1–4, 2019, Proceedings, Part II 9*. Springer, 2019, pp. 169–176.

[8] M. Shams, A. Elsonbaty, and W. ElSawy, "Arabic handwritten character recognition based on convolution neural networks and support vector machine," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 8, pp. 144–149, 2020, doi: 10.14569/issn.2156-5570.

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

[10] H. M. Balaha, H. A. Ali, E. K. Youssef, A. E. Elsayed, R. A. Samak, M. S. Abdelhaleem, M. M. Tolba, M. R. Shehata, M. R. Mahmoud, M. M. Abdelhameed, and M. M. Mohammed, "Recognizing Arabic handwritten characters using deep learning and genetic algorithms," *Multimedia Tools Appl.*, vol. 80, nos. 21–23, pp. 32473–32509, Sep. 2021.

[11] M. Mohd, F. Qamar, I. Al-Sheikh, and R. Salah, "Quranic optical text recognition using deep learning models," *IEEE Access*, vol. 9, pp. 38318–38330, 2021.

[12] R. Ahmed, M. Gogate, A. Tahir, K. Dashtipour, B. Al-tamimi, A. Hawalah, M. A. El-Affendi, and A. Hussain, "Novel deep convolutional neural network-based contextual recognition of Arabic handwritten scripts," *Entropy*, vol. 23, no. 3, p. 340, Mar. 2021.

[13] M. W. Jbrail and M. E. Tenekeci, "Character recognition of Arabic handwritten characters using deep learning," *J. Stud. Sci. Eng.*, vol. 2, no. 1, pp. 32–40, Mar. 2022.

[14] R. Maalej and M. Kherallah "New MDLSTM-based designs with data augmentation for offline Arabic handwriting recognition," *Multimed Tools Appl.*, vol. 81, pp. 10243–10260, Mar. 2022, doi: 10.1007/s11042-022-12339-8.

[15] M. Fasha, B. Hammo, N. Obeid, and J. Widian, "A hybrid deep learning model for Arabic text recognition," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 8, 2020, doi: 10.14569/IJACSA.2020.0110816.

[16] Z. Noubigh, A. Mezghani, and M. Kherallah, "Contribution on Arabic handwriting recognition using deep neural network," in *Hybrid Intelligent Systems: 19th International Conference on Hybrid Intelligent Systems (HIS 2019) Held in Bhopal, India, December 10–12, 2019*. Springer, 2021, pp. 123–133.

[17] R. E. Shtaiwi, G. A. Abandah, and S. A. Sawalhah, "End-to-end machine learning solution for recognizing handwritten Arabic documents," in *Proc. 13th Int. Conf. Inf. Commun. Syst. (ICICS)*, Irbid, Jordan, Jun. 2022, pp. 180–185.

[18] N. Rahal, M. Tounsi, and A. M. Alimi, "Neural probabilistic system for text recognition," 2018, *arXiv:1812.03680*.

[19] M. Tounsi, I. Moalla, U. Pal, and A. M. Alimi, "Arabic and Latin scene text recognition by combining handcrafted and deep-learned features," *Arabian J. Sci. Eng.*, vol. 47, no. 8, pp. 9727–9740, Aug. 2022.

[20] M. E. Mustafa and M. Khalafallah, "A deep learning approach for handwritten Arabic names recognition," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 1, pp. 678–682, 2020, doi: 10.14569/ijacsa.2020.0110183.

[21] K. Ramasubramanian and A. Singh, "Deep learning using keras and tensorflow," in *Machine Learning Using R*. Berkeley, CA, USA: Apress, 2019, doi: 10.1007/978-1-4842-4215-5_11.

[22] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing*, vol. 503, pp. 92–108, Sep. 2022.

[23] Y. Dogan, "A new global pooling method for deep neural networks: Global average of top-K max-pooling," *Traitement du Signal*, vol. 40, no. 2, pp. 577–587, Apr. 2023.

[24] Z. Liu, Z. Xu, J. Jin, Z. Shen, and T. Darrell, "Dropout reduces underfitting," 2023, *arXiv:2303.01500*.

[25] M. Schall, "Multi-dimensional connectionist classification: Segmentation-free handwriting recognition," 2022. [Online]. Available: https://kops.uni-konstanz.de

[26] A. Gulli and S. Pal, *Deep Learning With Keras*. Packt, 2017. [Online]. Available: https://books.google.com

[27] T. Carneiro, R. V. Medeiros Da NóBrega, T. Nepomuceno, G.-B. Bian, V. H. C. De Albuquerque, and P. P. R. Filho, "Performance analysis of Google colaboratory as a tool for accelerating deep learning applications," *IEEE Access*, vol. 6, pp. 61677–61685, 2018.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[29] A. B. Shanmugavel, V. Ellappan, A. Mahendran, M. Subramanian, R. Lakshmanan, and M. Mazzara, "A novel ensemble based reduced overfitting model with convolutional neural network for traffic sign recognition system," *Electronics*, vol. 12, no. 4, p. 926, Feb. 2023.

[30] M. Pechwitz, S. S. Maddouri, V. Margner, N. Ellouze, and H. Amiri, "IFN/ENIT-database of handwritten Arabic words," in *Proc. CIFED*, vol. 2, 2002, pp. 127–136.

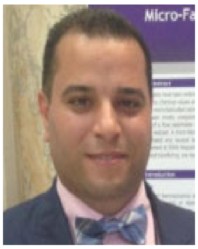
[31] F. Slimane, R. Ingold, S. Kanoun, A. M. Alimi, and J. Hennebert, "A new Arabic printed text image database and evaluation protocols," in *Proc. 10th Int. Conf. Document Anal. Recognit.*, 2009, pp. 946–950.

[32] S. Alghyaline, "Arabic optical character recognition: A review," *Comput. Model. Eng. Sci.*, vol. 135, no. 3, pp. 1825–1861, 2023.

LAMIA MOSBAH received the Engineering Diploma degree in computer science from the National Engineering School of Sfax (ENIS), University of Sfax, Sfax, Tunisia, in 2012, where she is currently pursuing the Ph.D. degree. Her research interests include the applications of intelligent methods and pattern recognition.

IKRAM MOALLA (Member, IEEE) received the Ph.D. degree in computer science from INSA, Lyon, in 2009. She has been an Assistant Professor of computer science with the National School of Engineering of Sfax (ENIS), University of Sfax, since 2008. She has been with the College of Computer Science and Information Technology in technical cooperation, Saudi Arabia, since 2015. She is currently a member of intelligent machines with the REGIM Research Laboratory. Her research interests include the applications of intelligent methods, pattern recognition, vision systems, and document understanding.

TAREK M. HAMDANI (Senior Member, IEEE) was born in Tunis, Tunisia, in 1979. He received the M.Sc., Ph.D., and H.D.R. degrees in computer science engineering from the National Engineering School of Sfax, Tunisia, in 2003, 2011, and 2019, respectively. He is currently an Associate Professor of computer science with the University of Monastir, Monastir. He focuses his research on applying intelligent methods (neural networks, fuzzy logic, and evolutionary algorithms) to machine learning, computer vision, natural language processing, intelligent pattern recognition, and analysis of large-scale complex systems. He is a Senior Member of IEEE CIS, IEEE SMC, IEEE RAS, and IEEE Computer Societies. He is a Reviewer of *Pattern Recognition Letters*, *Neurocomputing*, *Neural processing Letters*, IEEE Transactions on Neural Networks and Learning Systems, and *Soft Computing*.



BILEL NEJI was born in Tunisia, in May 1983. He received the B.Eng. degree in electrical engineering from the National Engineering School of Sfax, Tunisia, in cooperation with the University of Valenciennes, France, in 2007, the M.S. degree in new technologies in computer systems from the National Engineering School of Sfax, in cooperation with the University of Lille, France, in 2008, the Ph.D. degree in electrical engineering from the National Engineering School of Sfax, in cooperation with the University of Würzburg, Germany, in 2014, with a focus on scientific satellites' subsystems, and the Ph.D. degree in electrical engineering from The State University of New York, Buffalo, NY, USA, in 2015, with a focus on MEMS and micro/nano sensors design and fabrication. He co-founded BAK USA Technologies Corporation, NY, USA, in 2014, and was the Director of engineering, until 2018. He joined American University of the Middle East (AUM), Kuwait, in September 2018, as an Assistant Professor. He is currently an Associate Professor with AUM and also the Research Office Coordinator of the College of Engineering and Technology, AUM. He has been conducting research in different areas including embedded systems, MEMS sensor design, artificial intelligence, and renewable energy. He received the Fulbright Science Technology Award from U.S. Department of State, USA, in 2011, and several other prestigious awards worldwide.

TAHA BEYROUTHY received the M.Eng. degree in microelectronics from IMT Atlantique (Telecom Bretagne), in 2006, and the Ph.D. degree in micro and nanoelectronics from Grenoble Institute of Technology, in 2009. He is currently an Associate Professor of electrical engineering and the Dean of the College of Engineering and Technology, American University of the Middle East. Before joining AUM, he was an Assistant Professor and the Lead Member of the CIS Research Team, TIMA Laboratory, Grenoble. Later, he joined EASii IC as a Senior ASIC Designer, where he focused on improving power efficiency and security in digital systems, through novel signal sampling schemes, and designing asynchronous cryptographic circuits. He has established AUM's AI and Robotics Research Center and has published numerous publications in micro and nanoelectronics, robotics, artificial intelligence (AI), and applied physics. His current research interests include the application of AI in STEM education, brain–computer interfacing, and machine learning in cybernetics.

ADEL M. ALIMI (Senior Member, IEEE) received the degree in electrical engineering, in 1990, and the Ph.D. and H.D.R. degrees in electrical and computer engineering, in 1995 and 2000, respectively. He has been a Full Professor of electrical engineering with the National Engineering School of Sfax (ENIS), University of Sfax, since 2006. He has been the Director of Tunisia Erasmus+ Office, since 2018. He is currently the Founder and the Director of intelligent machines of the Research REGIM-Laboratory. His research interests include applications of intelligent methods (neural networks, fuzzy logic, and evolutionary algorithms) to pattern recognition, robotic systems, vision systems, and industrial processes.