

Computers, Materials & Continua DOI:10.32604/cmc.2021.015489 Article

Quranic Script Optical Text Recognition Using Deep Learning in IoT Systems

Mahmoud Badry^{1,*}, Mohammed Hassanin^{1,2}, Asghar Chandio^{2,3} and Nour Moustafa²

¹School of Engineering and Information Technology, UNSW Canberra. ACT, Newcastle NSW, 2620, Australia

²Faculty of Computers and Information, Fayoum University, Fayoum, Egypt

³Quaid-e-Awam University of Engineering, Science and Technology, Nawabshah, Pakistan

*Corresponding Author: Mahmoud Badry. Email: mma18@fayoum.edu.eg

Received: 17 November 2020; Accepted: 22 December 2020

Abstract: Since the worldwide spread of internet-connected devices and rapid advances made in Internet of Things (IoT) systems, much research has been done in using machine learning methods to recognize IoT sensors data. This is particularly the case for optical character recognition of handwritten scripts. Recognizing text in images has several useful applications, including content-based image retrieval, searching and document archiving. The Arabic language is one of the mostly used tongues in the world. However, Arabic text recognition in imagery is still very much in the nascent stage, especially handwritten text. This is mainly due to the language complexities, different writing styles, variations in the shape of characters, diacritics, and connected nature of Arabic text. In this paper, two deep learning models were proposed. The first model was based on a sequence-to-sequence recognition, while the second model was based on a fully convolution network. To measure the performance of these models, a new dataset, called QTID (Quran Text Image Dataset) was devised. This is the first Arabic dataset that includes Arabic diacritics. It consists of 309,720 different 192×64 annotated Arabic word images, which comprise 2,494,428 characters in total taken from the Holy Quran. The annotated images in the dataset were randomly divided into 90%, 5%, and 5% sets for training, validation, and testing purposes, respectively. Both models were set up to recognize the Arabic Othmani font in the QTID. Experimental results show that the proposed methods achieve state-of-the-art outcomes. Furthermore, the proposed models surpass expectations in terms of character recognition rate, F1-score, average precision, and recall values. They are superior to the best Arabic text recognition engines like Tesseract and ABBYY FineReader.

Keywords: OCR; quranic script; IoT; deep learning

1 Introduction

The Internet of Things (IoT) is based on a set of network and physical systems as well as machine intelligent methods that can analyze and infer data for certain purposes. It seeks to



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

build an intelligent environment that facilitates making the proper decision. IoT applications are particularly required in the visual recognition fields such as Intelligent Transport Systems (ITS) and video surveillance [1].

Optical character recognition (OCR) is the process of converting an image that contains text into machine-readable text. It has many useful applications including document archiving, searching, content-based image retrieval, automatic number plate recognition, and business card information extraction. OCR is also considered as a tool that can assist blind and visually impaired people. The OCR system's process includes some pre-processing of the input image file, text areas extraction, and recognition of extracted text using feature extraction and classification methods. Arabic is a widely spoken language throughout the world with 420 million speakers. Compared to Latin text recognition, not much research has been done or published on Arabic text recognition, and it is topic requiring more analysis [2,3]. The Arabic OCR system provides many applications such as archiving of historical writings and how to search them. Arabic text recognition is still under development, especially in the hand-written text [4] and this situation needs to improve. The Arabic language has some special features such as it is written from right to left, the words consist of connected Arabic characters, and each character may have up to four different forms based on its position in a word. Furthermore Arabic characters have variable sizes and fonts, which make their recognition a more difficult task to understand than Latin-derived languages. Different OCR challenges such as varied text perspectives, different backgrounds, different font shapes, and writing styles need more robust feature engineering methods to improve the system's overall performance. On the other hand, deep learning models require less highlevel engineering and extract the relevant features automatically. Although a model in the latter approach can learn deep representations from the image files automatically, it demands large-scale annotated data to train and generalize efficiently [5].

The Holy Quran is the religious scripture that Muslims throughout the world follow. Approximately one and half billion people around the world recite the Holy Quran. Most of the existing versions of the Quran have been published in the Arabic language rather than the Quranic script. The Holy Quran with Othmani font represents the main source for Arabic language rules in the form of a hand-written script. This Othmani font is chosen due to three major reasons: (1) it is one of the major grammar sources of the Arabic language, (2) it contains different words, characters, and diacritics from all over the Arabic language, and (3) it contains all the recitation styles' letters and vowels. The challenges associated with the Quranic scripts can be summarized as follows:

-Traditional image problems

Since OCR processes images, it is beset by long-lasting visual computing challenges such as poor quality images and background noises.

-Different shapes and forms

Fig. 1a shows various shapes of the diacritics for each letter. Fig. 1b illustrates different shapes of the same Arabic letter depending on its position within a word and Fig. 1c depicts the set of Arabic characters with a unique integer ID used in this research study.

-Non-pattern scripts

Arabic handwritten text does not follow the defined patterns and depends on the quality of the writer's text. For instance, using handwritten text for the bio-metric signature reveals a great dissimilarity for the same script.

			l	Diacrit	tics Sh	apes			0	٤.	1	i	2		3	1	4	10	5
Fath	1	5	1	4	1	1	۲		_	×	1	ं	-		2	1	-	3	-
Kasr	1	:	1	"	-	3	1-	e	6	Ļ	7	5	8	ت	9	ٹ	10	ē	11
Damm	۶	*	\$	붯	Ģ	3	12	9	-	_				-				1165	-
Sukkon			0			•	8		12	ż	13	د	14	ذ	15	ر	16	j	17
Other diacritics		5	У	بہ	^	قلے	ž	3	18	ش	19	ص	20	ض	21	ط	22	ظ	23
	6	2		(a)	27	à	-		24	ė	25	-	26	ف	27	ق	28	ك	29
									30	4	31	ن	32	\$	33	و	34	ى	35
Let	ter	Is	olated	Be	Letter's Lo gin	Middle	En	d	36	*	37		38		39	1 ^a	40		41
ç			s ī		5	ء ٦	. 4	ء 1	42		43	e.	44	•	45	•	46	i.	47
Ĩ			,																
1 و			, و	-	ۇ	ۇ	•	-وأ	48	5	49	•	50	•	51	,	52		53

Figure 1: (a) Various shapes of diacritics for every letter (b) different shapes of one letter show the difficulty of processing Arabic letters, (c) the set of letters that were covered in this research study

-Dynamic sizes

Broadly speaking, Arabic letters' sizes in the same script depend on the font and location in the word. For this reason, segmenting these letters is not an easy task.

-Diacritics

Arabic letters' pronunciations are controlled by the diacritics. It ranges from four to eight forms according to the type of diacritics and location of the letter as shown in Fig. 1a. More specifically, the |Holy Quran has 43 diacritics and this leads to a sophisticated problem.

-Lack of resources

In contrast to English, the number of research studies on this language is very small. This prevents new technologies from being applied since there is a definite shortage of resources.

-Datasets problem

The most recent paradigms such as deep learning algorithms require a massive amount of data to train and evaluate the networks. The recognition of Quranic letters still lacks the availability of large datasets. Intuitively, deep learning algorithms work better in the case of large datasets in comparison to small datasets [6]. To the best of our knowledge, this is the first study to introduce a large dataset for the Quran scripts.

In this paper, two deep learning-based techniques with convolutional neural network (CNN) and long short-term memory (LSTM) networks have been proposed to enhance the Arabic word image text recognition. It does this by using the Holy Quran corpus with Othmani font. The Othmani font is chosen for three key reasons: Firstly, it is one of the major grammar sources of

the Arabic language; secondly, it contains different words, characters, and diacritics from all over the Arabic language; and thirdly, the Mus'haf—Holy Quran book is written in Othmani font, which is a handwritten text that contains various shapes for each character. The Arabic word is written with a white font on a 192×64 black background image. The input image is assumed to be without any noise and skew.

The first model which is known as Quran-seq2seq-Model consists of an encoder named Quran-CNN-encoder and a multi-layer LSTM decoder. This model is like the image captioning models [7] that were developed with deep learning techniques. The second model called Quran-Full-CNN- Model implements the same encoder as in model 1, but it uses a fully connected layer followed by a Softmax activation in the networker's decoder part. The proposed methods recognize characters and diacritics from one word at a time. The key contributions of this paper are as follows:

- 1. Developing two end-to-end deep learning models that recognize Arabic text images in Quran Text Image Dataset (QTID) dataset.
- 2. Creation and evaluation of a dataset called QTID that was taken from the Holy Quran corpus.
- 3. Experimental results demonstrate that the proposed models outperform than best OCR engines like Tesseract [8] and ABBYY FineReader1.

2 Related Works

In the last few decades, research and commercial organizations have proposed several devices to create an accurate Arabic OCR for printed and handwritten text. Some of them have achieved a recognition accuracy of 99% or more for printed text but handwritten recognition is still under development.

Tesseract OCR [8] supports more than 100 languages including Arabic script with Unicode Transformation Format (UTF-8) encoding. The current Tesseract 4.0.0 version uses a deep LSTM recurrent neural network for the character recognition tasks and has much better accuracy and performance. ABBYY FineReader1 OCR can recognize 190 different languages including Arabic script. Further, a customized Arabic character OCR called Sakhr is created that supports all languages that contain Arabic characters such as Farsi and Urdu languages. The Sakhr OCR can recognize Arabic diacritics as well.

An offline font-based Arabic OCR system proposed in [9] used pre-processing, segmentation, thinning, feature extraction, and classification steps. The line and letter segmentation accuracies were 99.9% and 74%, respectively. They used a decision tree algorithm as a classifier and reported 82% accuracy. The overall system performance only reached 48.3%. In [10] an Arabic OCR combined computer vision and sequence learning methods to recognize offline handwritten text from IFN/ENIT dataset [11]. They used a multi-dimensional (MDLSTM) network to learn the handwritten text sequences from the word images and achieved an overall word recognition accuracy of 78.83%. In [12], a different word recognition OCR for the embedded Arabic words in news television was proposed. The embedded Arabic word images have many challenges including different foregrounds, backgrounds, and fonts. They used deep autoencoders and convolutional neural networks to recognize the words without applying any prior pre-processing or character segmentation on the word level. The overall accuracy obtained was only 71%.

Recognition of handwritten Quranic text is more complex than printed Arabic text. The Quranic text contains ligatures, overlapped text, and diacritics. It has more writing variations

and styles. Further, a letter with the same style may have different aspect ratios. The challenges associated with handwritten Quranic text recognition are described in [13]. In [14], a similarity check method was proposed to recognize Quranic characters and diacritics recognition in images. A projection method is used to recognize the Quranic characters, while a region-based method is applied so that diacritics can be detected. An optimization method is used to further improve recognition accuracy. Results obtained were compared with a standard Mushaf al Madinah benchmark. The overall accuracy of the system was 96.42%. An online Quranic verses authentication system proposed in [15] used a hashing algorithm to verify the integrity of the Quranic verses in the files, datasets, or disks. Further, an information retrieval system is developed to search the possible Quranic verses all over the Internet and verify with the authentication system if there is any fraud or changes in the Quranic verses. In [16], a word-based segmentation method using histogram of oriented gradient (HOG) and local binary pattern (LBP) was proposed to classify and recognize handwritten Quranic text. The text was written in one of the common scripts of the Arabic language named Kufic script. Polynomial kernel, a type of support vector machine (SVM) was applied for the classification. The word recognition accuracy achieved was 97.05%.

3 Proposed Arabic Text Recognition Methodology

Considering the Arabic word image text recognition problem as a sequence-to-sequence deep learning problem, the proposed methodology is based on the encoder and decoder model architectures. The encoder part in both models uses a deep CNN similar to VGG-16 [17] network, while the decoder part in the first model implements an LSTM network and the second model implements a fully connected neural network.

3.1 Quran-seq2seq-Model

The first model called Quran-seq2seq-Model consists of an encoder named Quran-CNNencoder and a multi-layer LSTM decoder. It is similar to the image captioning models [8] that were developed with deep learning techniques. The encoder and decoder architectures are described here. The Quran-CNN-encoder takes a gray-scale image(s) as an input, applies convolution and max-pooling operations to it, and finally outputs a vector that represents the input image features. The convolutional filter sizes used are 3×3 with the padding and stride value of 'SAME.' The initial layers use smaller values of convolutional filters, i.e., 64, while the last layers use a larger value of convolutional filters, i.e., 512. Unlike the model in [17], max-pooling layers in the proposed model use 4×4 filter size and a stride of 4×4 . The fully connected layers employed in [17] are not included in the Quran-seq2seq-Model. However, a bottleneck 1×1 convolution layer is used instead of fully connected layers, which diminishes the number of parameters and improves the efficiency of the model. As the width and height of the images are different, the filter size of the first bottleneck layer is set to 1×3 . Fig. 2 illustrates the architecture of the proposed Ouran-CNN-encoder. The decoder of the model consists of two LSTM layers followed by a fully connected layer and a Softmax activation in the end. The first LSTM has 4,448,256 parameters while the second one has 3,147,776. The number of time steps for the Multi-LSTM is 22. At each time step, these layers will try to establish the probability of the next character given an image and a set of predicted characters as illustrated in the following equation.

$$\log p(C|I) = \sum_{t=0}^{22} \log p_t(C_t | I, C_0, \dots, C_{t-1})$$
(1)

where I is the input image, C is the ground truth characters of the image, and p_t (C_t) is the probability of the character C_t in the probability distribution p_t . Fig. 3 illustrates the framework of the proposed methodology of Quran-seq2seq-Model with encoder and decoder parts of the network. At t = 0, a gray image matrix is passed through the Quran-CNN-encoder. The output matrix with $1 \times 1 \times 400$ dimensions is then passed through a fully connected layer, which makes it a vector comprising 61 elements. This vector is then passed to the first LSTM, which has 512 neurons. The first LSTM will output a vector with 512 elements, which are passed to the second LSTM. The second LSTM uses 1024 neurons and outputs a vector with 1024. Finally, this vector is passed to a fully connected layer that shrinks the vector to 61 elements so that a Softmax can be applied to it and the first character of the image can be predicted. At t = 1, the first row of the input one-hot matrix is passed directly through the two LSTM layers which have saved activations from the previous state at time t = 0, which helps the model to recognize the next character in a sequence. Similarly, the output of the second LSTM layer is passed to the fully connected layer for the character prediction using the SoftMax activation function. This process is repeated until t = 22 that includes all the 21 characters of the one-hot matrix. The final output of the model is the concatenated results of all the SoftMax activations.



Figure 2: Architecture of the Quran-CNN-encoder. 'Conv' represents the convolutional, 'BN' represents the batch normalization, and 'RELU' represents the rectified linear unit

In the training phase of Quran-seq2seq-Model, its parameters are optimized for the inputs and outputs from the training set. To optimize the model's loss function, the optimization algorithm and the learning rate must be specified. Since this is a multi-class classification problem, the loss function applied is the cross-entropy, which is defined as:

$$L(I, C) = -\sum_{t=0}^{22} \log p_t(C_t)$$
(2)

where p_t is the probability distribution of the character C_t at time t. The accuracy metric selected was the categorical accuracy, which calculated the mean accuracy rate across all the predictions and compared it with the actual outputs. The model loss function was minimized using an Adam optimization algorithm with a stochastic gradient descent (SGD) variant. This optimizer beta 1 value was set to 0.9 while beta 2 was set to 0.999. The learning rate was set to 0.0005 and the mini-batch size was 32 with no usage of the learning decay technique.



Figure 3: Framework of the proposed Quran-seq2seq-Model with encoder and decoder

3.2 Quran-Full-CNN-Model

The sequence-to-sequence models work efficiently in text recognition problems such as Arabic handwritten text. However, their character prediction and concatenation time is more than the fully convolutional models. The fully convolutional models can do the predictions at once, which makes the network training easier and helps in faster predictions. Further, these fully convolutional models take the advantage of GPU parallelization, as it does not need to wait for the previous time step. Besides, the number of parameters in these models is smaller when compared to the sequence-to-sequence models. However, the fully convolutional models are limited to the fixed number of output units.



Figure 4: The architecture of the Quran-Full-CNN-Model

The Quran-Full-CNN-Model expands the same Quran-CNN-encoder as discussed in Section 3.1. However, instead of LSTM layers, this model includes a fully connected layer followed by a Softmax activation as illustrated in Fig. 4. The output of the fully connected layer is converted into a 22×61 matrix and then, the same Softmax activation is applied to each row in this matrix. In this case, the total number of output units is 22 characters. Moreover, the number of parameters in the fully connected layers is 538,142, which when added to the encoder model increased the total number of network parameters to 1,454,062. This number is much lower than the parameters in the sequence-to-sequence model. The small number of network parameters does help to reduce the training time of the network.

In the training phase of the Quran-Full-CNN-Model, we use the same loss function and other metrics as used in the Quran-seq2seq-Model. Similarly, Adam optimization algorithm served to optimize the loss function with the same beta 1 and beta 2 values. The learning rate of the model was set to 0.001 and the mini-batch size was 32 with no usage of the learning decay technique.

4 Implementation Details

The Quran-seq2seq and the Quran-Full-CNN models along with training phases were implemented using Keras framework with a TensorFlow in the backend. The training phase for the Quran-seq2seq-Model took around 6 h for the ten epochs, which minimized the loss value from 32.0463 to 0.0074. The network evaluation metric shows that the recognition process has 99.48% accuracy on the training set. Moreover, this evaluation process took 587 s.

The Quran-Full-CNN-Model was implemented with the same development features as the Quran-seq2seq-Model. However, the training phase took around 2 h to minimize the loss value from 24.0282 to 0.0074 in ten epochs. The network evaluation metric shows that the recognition process has 99.41% accuracy on the training set. Apart from this the evaluation process took 345 s for the whole training set on the same machine specifications. The network training for both models was performed on an IntelR CoreTM i7 with 3.80 GHz with A 4 GB GTX 960 Nvidia GPU and a 16 GB DDR5 RAM.

5 Experimental Results and Discussions

To demonstrate the effectiveness of the Quran-seq2seq and Quran-Full-CNN models, different experiments were conducted on the QTID dataset.

5.1 Quran Text Image Dataset

To train, validate, and test the proposed models, a new Arabic text recognition dataset is created. The dataset can be used as a benchmark to measure the current state of recognizing Arabic text. Moreover, it is the first Arabic dataset that uses diacritics along with handwritten Arabic words. The Holy Quran corpus with Othmani font is used as the source to create the QTID dataset. This font contains different words, characters, and diacritics from all over the Arabic language. Moreover, the Mus'haf Holy Quran is written in Othmani font, which is a handwritten text, where each character is represented in various shapes.

5.2 Evaluation Metrics

To evaluate the proposed models, five different evaluation metrics have been used. The first evaluation metric is a character recognition rate (CRR), which is defined as follows:

$$CRR = \frac{\#characters - \sum \text{Levenshtein Distance(RT, GT)}}{\#characters}$$
(3)

where (RT) is the recognized text and (GT) is the ground truth text. The Levenshtein Distance function measures the distance between two strings as the minimum number of single-character edits. The other four measures are accuracy, average precision, average recall, and average F1 score, which are defined as follows:

$$Accuracy = \frac{\# \text{correctly recognized } characters}{\# characters} \tag{4}$$

$$Precision_{i} = \frac{\# \text{correctlyrecognized} character_{i}}{\# \text{correctlyrecognized} ascharacter_{i}}$$
(5)

$$Recall_i = \frac{\# correctly recognized character_i}{\# characteriappearacnces}$$
(6)

$$F1_i = \frac{2}{\frac{1}{\text{Precision}1_i} + \frac{1}{\text{Recall}_i}} \tag{7}$$

The F1 score takes the harmonic average of the precision and recall for a specific character.

5.3 Evaluation of the Quran Text Image Dataset

The performance of the proposed models on the QTID dataset has been evaluated and compared with state-of-the-art commercial OCR systems. The Quran-seq2seq and Quran-Full-CNN models, Tesseract, and ABBYY FineReader 12 were evaluated using the metrics as described in Section 5.2. Since Tesseract and ABBYY FineReader 12 cannot recognize the Arabic diacritics, an additional test set was created. This additional test set contained the same Arabic text images as in the target test set, but the diacritics were removed from the ground truth text.

All the images were converted to grayscale in the test sets and fed to the four different models to get the possible predictions. All the text predictions along with the ground truth text were saved in two lists: One for the standard test set with Arabic diacritics and the other for the additional test set without Arabic diacritics. With each model, two evaluations on the test sets with and without Arabic diacritics were performed, which led to eight different lists. An average prediction time for the models developed in this paper was 30 s for each image in the test set.

5.3.1 Character Recognition Rate

The evaluation results using the character recognition rate (CRR) metric with the proposed models and the commercial OCR systems are shown in Tab. 1. The CRR of the Quran-seq2seq-Model with Arabic diacritics is 97.60%, while without diacritics it is 97.05%. Similarly, the CRR of the Quran-Full-CNN-Model with Arabic diacritics is 98.90%, while without diacritics it is 98.55%. The CRR of the Tesseract and ABBY FineReader 12 OCR systems with and without Arabic diacritics is 11.40%, 20.70%, 6.15%, and 13.80%, respectively. These results indicate that the proposed deep learning models out-perform on the QTID than the commercial state-of-the-art OCR systems.

System/model	CRR (W-D) (%)	CRR (W-N-D) (%)				
Tesseract 4.0	11.40	20.70				
ABBY FineReader 12	6.15	13.80				
Quran-seq2seq-Model	97.60	97.05				
Quran-Full-CNN-Model	98.90	98.55				

Table 1: Character recognition rate results for the two different Arabic test sets. The first test set (W-D) includes the Arabic diacritics, while the other test set (W-N-D) does not

5.3.2 Other Evaluation Metrics

To calculate overall accuracy, average precision, average re-call, and average F1 score, some pre-processing for the predicted text and the ground truth text has been done. The predicted and ground truth text in both test sets have been aligned using a sequence algorithm so that each text instance should have the same length and each character in the predicted text and ground truth text be mapped.

Tabs. 2–4 show the evaluation results for overall accuracy, average precision, average recall, and average F1 metrics, respectively. The overall accuracy of the proposed Quran-seq2seq and Quran-Full-CNN models with and without Arabic diacritics was 95.65%, 98.50%, 95.85%, and 97.95%, respectively. Meanwhile for the commercial Tesseract and ABBAY FineReader 12 OCR systems, it was 10.67%, 2.32%, 17.36%, and 5.33%, respectively. The proposed Quran-seq2seq and Quran-Full-CNN models with and without Arabic diacritics outperform in terms of average precision and recall than the commercial OCR systems as shown in Tab. 3. This confirms that the proposed models recognize normal Arabic characters much better than Arabic diacritic characters.

Table 2: Overall accuracy results for the two different test sets. The first test set (W-D) includes the Arabic diacritics, while the other (W-N-D) does not

System/model	CRR (W-D) (%)	CRR (W-N-D) (%)				
Tesseract 4.0	10.67	17.36				
ABBY FineReader 12	2.32	5.33				
Quran-seq2seq-Model	95.65	95.85				
Quran-Full-CNN-Model	98.50	97.95				

Table 3: Average (Avg) precision and average recall results for two different test sets. The first test set (W-D) includes the Arabic diacritics, while the other (W-N-D) does not

System/model	Avg Precision	Avg Precision	Avg Recall	Avg Recall
	(W-D) (%)	(W-N-D) (%)	(W-D) (%)	(W-N-D) (%)
Tesseract 4.0	56.73	50.59	18.44	14.6
ABBY FineReader 12	34.37	37.64	4.06	6.14
Quran-seq2seq-Model	91.85	97.55	87.35	97.53
Quran-Full-CNN-Model	90.46	98.66	89.64	97.41

CMC, 2021, vol.68, no.2

The average F1 score of the proposed Quran-seq2seq and Quran-Full-CNN models with and without Arabic diacritics as shown in Tab. 4 was 89.55%, 90.05%, 95.88%, and 98.03%, respectively. In the meantime the average F1 score of the commercial OCR systems with and without Arabic diacritics was 27.83%, 7.28%, 22.66%, and 10.55%, respectively.

Table 4: Average F1 score results for two different test sets. The first test set (W-D) includes the Arabic diacritics, while the other (W-N-D) does not

System/model	Avg F1 score (W-D) (%)	Avg F1 score (W-N-D) (%)
Tesseract 4.0	27.83	22.66
ABBY FineReader 12	7.27	10.55
Quran-seq2seq-Model	89.55	95.88
Quran-Full-CNN-Model	90.05	98.03

The results documented in Tabs. 2–4 suggest the strong feature extraction capability of the CNN models, which leads to an improvement in the recognition accuracy of the Arabic text in images.

6 Conclusion and Future Work

Optical character recognition systems are supposed to deal with all kinds of languages in imagery and then convert them to their corresponding machine-readable text. Arabic text recognition for the OCR systems has not yet reached state-of-the-art standard compared to Latin text. This is mainly due to the language complexities and other challenges associated with Arabic text. This paper proposed two deep learning-based models to recognize Arabic Quranic word text in the images. The first model is a sequence-to-sequence model and the other is a fully convolutional model. A new large-scale dataset named QTID was developed from the words of the Holy Quran to improve the recognition accuracy of Arabic text from images. This is the first dataset to contain Arabic diacritics. The dataset consists of 309,720 images, which were split into training, validation, and testing sets, respectively. Both models were trained and tested on the QTID dataset. To compare the performance of the proposed model, QTID test set was evaluated on the two commercial OCR systems. The subsequent results show that the proposed models outperform commercial OCR systems. Although the proposed models outperform on the QTID, these models have some limitations such as: The text must be at the center of the input image, the foreground color of the input image should be white, while the text color is black. In the future, more Arabic images with diverse text directions will be included. The Arabic word text images with different foreground and background colors will be added. An end-to-end system will be proposed for recognizing sentence-level Arabic text in images. Further, a few more deep learning models will be evaluated on the proposed QTID dataset.

Funding Statement: This work has been funded by the Australian Research Data Common (ARDC), project code—RG192500 that will be used for paying the APC of this manuscript.

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

References

- M. Irfan, Amin Ullah, M. Khan, D. Der-Jiunn, W. Meng *et al.*, "Efficient image recognition and retrieval on IoT-assisted energy-constrained platforms from big data repositories," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9246–9255, 2019.
- [2] N. Islam, Z. Islam and N. Noor, "A survey on optical character recognition system," Journal of Information & Communication Technology, vol. 10, no. 2, pp. 1–4, 2017.
- [3] M. T. Parverz and S. A. Mahmoud, "Offline Arabic handwritten text recognition: A survey," ACM Computing Surveys, vol. 45, no. 2, pp. 1–35, 2013.
- [4] R. Ahmed, K. Dashtipour, K. Gogate, M. Raza, R. Zhang *et al.*, "Offline Arabic handwriting recognition using deep machine learning: A review of recent advances," in *Proc. BICS*, Guangzhou, China, pp. 457–468, 2019.
- [5] M. Jaderberg, K. Simonyan, A. Vedaldi and A. Zisserman, "Reading text in the wild with convolutional neural networks," *International Journal of Computer Vision*, vol. 116, no. 1, pp. 1–20, 2016.
- [6] J. Deng, W. Dong, R. Socher, L. Lj, K. Li *et al.*, "Imagenet: A large-scale hierarchical image database," in *Proc. CVPR*, Florida, US, pp. 248–255, 2009.
- [7] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proc. CVPR*, Boston, Massachusetts, US, pp. 3128–3137, 2015.
- [8] R. Smith, "An overview of the Tesseract OCR engine," in *Proc. ICDAR*, Parana, Brazil, pp. 629–633, 2007.
- [9] I. Supriana and A. Nasution, "Arabic character recognition system development," in *Proc. ICEEI*, Selangor, Malaysia, pp. 334–341, 2013.
- [10] A. Graves, "Offline Arabic handwriting recognition with multidimensional recurrent neural networks," in V. Märgner, H. El Abed (Eds.), *Guide to OCR for Arabic Scripts*, vol. 1. London, England: Springer, pp. 297–313, 2012.
- [11] M. Pechwitz, S. S. Maddouri, V. Märgner, N. Ellouze and H. Amiri, "IFN/ENIT-database of handwritten Arabic words," in *Proc. CIFED*, Hammamet, Tunis, pp. 127–136, 2002.
- [12] S. Yousfi, S. A. Berrani and C. Garcia, "Deep learning and recurrent connectionist-based approaches for Arabic text recognition in videos," in *Proc. ICDAR*, Tunis, Tunisia, pp. 1026–1030, 2015.
- [13] A. Iqbal and A. Zafar, "Offline handwritten Quranic text recognition: A research perspective," in Proc. AICAI, Dubai, UAE, pp. 125–128, 2019.
- [14] F. Alotaibi, M. T. Abdullah, R. B. H. Abdullah, R. W. B. O. Rahmat, I. A. T. Hashem et al., "Optical character recognition for Quranic image similarity matching," *IEEE Access*, vol. 6, pp. 554–562, 2017.
- [15] I. Alsmadi and M. Zarour, "Online integrity and authentication checking for Quran electronic versions," *Applied Computing and Informatics*, vol. 13, no. 1, pp. 38–46, 2017.
- [16] A. Zafar and A. Iqbal, "Application of soft computing techniques in machine reading of Quranic Kufic manuscripts," *Journal of King Saud University-Computer and Information Sciences*, vol. 4, no. 6, pp. 49, 2020.
- [17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, San Diego, CA, US, 2015.