QTID: Quran Text Image Dataset

Mahmoud Badry
Faculty of Computers and Information
Fayoum University
Fayoum, Egypt

Hesham Hassan, Hanaa Bayomi
Faculty of Computers and Information
Cairo University
Cairo, Egypt

Hussien Oakasha
Faculty of Computers and Information
Fayoum University
Fayoum, Egypt

***Abstract*—Improving the accuracy of Arabic text recognition in imagery requires a big modern dataset as data is the fuel for many modern machine learning models. This paper proposes a new dataset, called QTID, for Quran Text Image Dataset, the first Arabic dataset that includes Arabic marks. It consists of 309,720 different 192x64 annotated Arabic word images that contain 2,494,428 characters in total, which were taken from the Holy Quran. These finely annotated images were randomly divided into 90%, 5%, 5% sets for training, validation, and testing, respectively. In order to analyze QTID, a different dataset statistics were shown. Experimental evaluation shows that current best Arabic text recognition engines like Tesseract and ABBYY FineReader cannot work well with word images from the proposed dataset.**

Keywords*—HDF5 dataset; Arabic script; Holy Quran text image; handwritten text recognition; Arabic OCR; text image datasets

I. Introduction

Optical character recognition (OCR) is the process of converting an image that contains text into a readable machine text. It has a lot of useful applications including document archiving and searching, automatic number plate recognition, and business card information extraction. It is also considered an assisting tool for blind and visually impaired people. Although OCR is an old problem, Arabic text recognition is still under development, especially in handwritten text [1], [2] due to many reasons including special Arabic language characteristics. Some of these characteristics are: A character may have up to four different shapes as depicted in Fig. 1, a character's width and height might change relative to its location within a word, the Arabic language is written from right to left, and some characters have the same shape except for the presence/location of dots above or below that shape. Another reason that Arabic text recognition is still under development is the lack of a standard robust comprehensive dataset [3].

This paper presents a new Arabic images dataset that can help machine learning models master the Arabic language text recognition. The dataset is generated from the Holy Quran, which contains a handwritten Arabic text including Arabic language marks. The Holy Quran is the book that Muslims believe is sent from Allah to Messenger Muhammad to guide all humans. It was written in Arabic which is the mother tongue of most of the Middle East. In fact, the Holy Quran consists of 114 Surah. Each Surah has a different number of Ayat. Moreover, an Ayah - singular of Ayat - consists of one or more Arabic words. There are 6,236 Ayat in the holy Quran, which form 77,430 Arabic words. The total number of unique words

Letter label	Isolate	Begin	Middle	End
ء	ء			
أ	أ		ـأ	
ؤ	ؤ		ـؤ	
إ	إ		ـإ	
ئ	ئ	ئـ	ـئـ	ـئ

Fig. 1. Five different Arabic characters each has different shapes for different positions in an Arabic word.

Fig. 2. Two pairs of words with same characters shape but with different marks which give them different meanings.

is 18,994.The Holy Quran was chosen in this study due to three reasons: first, it is one of the sources of the Arabic language, second, it contains different words, characters, and marks from all over the Arabic language, third, the Mus'haf -the written work of Holy Quran- is written in Othmani font which is a handwritten text that has various shapes of characters.

The presented dataset contains a large set of handwritten characters and words with different shapes and sizes. As the data is the fuel for many machine learning models, creating a big modern dataset can help data-hungry machine learning models master the Arabic text recognition. In addition, it can be used as a benchmark to measure the current state of recognizing Arabic text. What makes this dataset different is that it is the first Arabic dataset to contain Arabic marks. In Arabic, two words with the same shape but with different marks may have different meanings as shown in Fig. 2.

The proposed dataset has 309,720 different word images that were collected from the Holy Quran words in four fonts. An example of a word in the four fonts is in Fig. 3. These images have been split into training, validation, and testing

Fig. 3. The same Arabic word with font sizes 22, 24, 26, and 28 pixels, respectively.

sets then saved in an HDF5 file, which is portable and can be used across operating systems.

This paper is organized as follows: Section 2 briefly discusses some Arabic Text datasets that have been created. Full steps to the creation of the dataset are described in Section 3. Section 4 analyzes the properties of the proposed dataset and gives some statistics about it. Experimental results and evaluations are shown in Section 5. Finally, the discussion and conclusions of this paper is presented in Section 6.

II. Related Work

Through the years, Arabic text datasets have been created to help machines read Arabic text from images like any human being who knows how to read the Arabic language. These datasets played a great role in finding best methods to the mentioned goal. However, unlike English language, standard Arabic dataset is absence. Offline Arabic datasets like [4], [5], [6], [7] were made to help machine recognize text that has been scanned using a scanner or a camera and then stored digitally in an RGB or gray image. On the other hand, Online Arabic datasets like [8], [9], [10] helps machine understand text that was recorded using a digitizer as a time sequence of pen coordinates.

Datasets related to offline Arabic text recognition can be split into two groups: those that address printed text and the others that address handwritten text. This paper addresses each in turn.

Printed Arabic text recognition In this task, the purpose is recognizing Arabic characters, words, and paragraphs in which characters are typed using a printer and have a specific font. Additionally, the text is usually well structured. Many datasets have been created to solve this task. For example, a dataset with the goal of benchmarking open-vocabulary, multi-font, multi-size, and multi-style text recognition systems in Arabic, Arabic Printed Text Image (APTI) [5] was created in 2009. It contains 45,313,600 single word images that contain more than 250 million characters. This large dataset was generated using a lexicon of 113,284 words, 10 Arabic fonts, 10 font sizes, and 4-font styles. To focus on its goal each image has a clean white background with the annotation provided for each image in an XML file.

In 2015, the first public Arabic dataset ALIF [6] for recognizing Arabic text in videos was created. Creators of ALIF has extracted 6,532 text images from five famous Arabic TV channels that contain 18,041 words and 89,819 characters in total. These text images have different properties like fonts, sizes, color, backgrounds, and occlusions.

Handwritten Arabic text recognition In this task, the purpose is recognizing handwritten Arabic which was written by a human. Handwritten text recognition is harder than Printed text because the characters will have different appearances due to different writers and their styles besides one writer can produce the same character with different shapes in one sentence. Isolated Farsi/Arabic Handwritten Character Database (IFHCDB) [11] was made to help recognize Arabic isolated handwritten characters. It includes 52380 characters and 17740 numerals. The dataset is unbalanced which means that distribution of samples in each character is not uniform. Another dataset, IFN/ENIT contains 26,459 images that represent 937 names of cities and towns in Tunisia, written by 411 various writers. Each image is annotated by the ground truth text, position of word baseline, and information about each used character in a word.

A more recent dataset, KFUPM handwritten Arabic text (KHATT) [12] contains 2000 unique paragraph images, written by 1000 different writers, covering different topics like education, nature, arts, and technology. Moreover, each paragraph is segmented into lines and saved as individual images. The dataset has multiple research goals besides handwritten recognition like writer identification, noise removal techniques, binarization, and line segmentation.

Finally, A database that combines both printed and handwritten text named SmartATID [7], which stands for Smartphone Arabic Text Images Database has the purpose of recognizing Arabic text that has been captured by cell-phones cameras. The printed text version of this dataset includes 16472 document images, which are captured from 116 different paper documents with different capturing protocols like changing document layouts, cameras versions, light conditions, and position. With the similar capturing protocols, the handwritten text version of this dataset was created having 9088 images from 64 different handwritten documents.

The past pieces of work helped researchers in different Arabic OCR tasks, but none of them added the Arabic marks in their work although it is important as we have explained in the introduction section.

III. Dataset Creation

The discussion will now move to how the dataset has been created starting from a words database and fonts, ending by training, validation, and testing HDF5 files.

A. Image Generation

The first task was to generate images that represent each word in the Holy Quran with different fonts as depicted in, Fig. 4. We made a query on a complete Holy Quran database[1] that selects each word and required font to render. Each word

[1]Verified Holy Quran Database published on Github: https://github.com/quran/quran.com-images

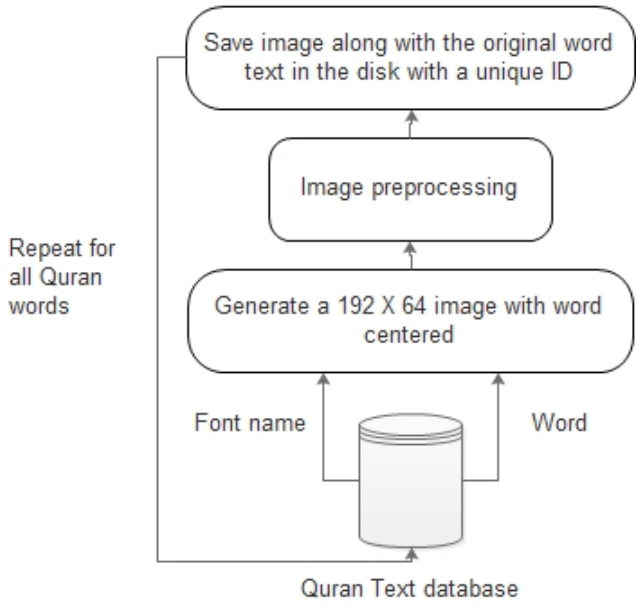

Fig. 4. Image generation steps.

is then rendered using the selected font in the middle of an image that has 192x64 dimensions. The 192x64 was selected as a dimension for the images of this study for three reasons: first, most machine learning models learn image of a fixed dimension; second, the biggest rendered word takes 180x61 dimensions; and third, the nearest numbers to 180 and 61 that can be vectorized through CPUs or GPUs are 192 and 64, respectively. The rendered images have a black background and a white text. Additionally, noise removal methods can be applied to each rendered image to make sure the images are clear. Finally, each image was given a unique id name and was saved in PNG image format on the disk along with a text file that represents the annotation of the image. The text file took the same id of the rendered image.

While reading words from the database, all the possible characters are recorded. After that, each character took a unique id to help generate one hot encoding representation of the images annotations. Ids range from 0 to 59 which represents 60 unique characters as shown in Fig. 5.

B. HDF5 files creation

The next task was to take all the generated images and then make training, validation, and testing sets which are then stored in HDF5 files as elaborated in Fig. 6. Initially, the process started by reading all the images paths into a big list. Before splitting up this list into training, validation, and testing lists, they were shuffled to make sure that random samples have been drawn for each of the lists. Afterwards, 90% of the paths list was taken for the training set, 5% for the validation set, and 5% for the testing set. The chosen percentage is based on the fact that our dataset is big enough and the trend is to take much smaller percentages for the validation and testing set as Andrew Ng[2] said on his notes[3].

[2]Andrew Ng one of the pioneers in machine learning

[3]https://www.coursera.org/learn/deep-neural-network/lecture/cxG1s/train-dev-test-sets

0 ء	1 أ	2 ؤ	3 إ	4 ئ	5 ا
6 ب	7 ة	8 ت	9 ث	10 ج	11 ح
12 خ	13 د	14 ذ	15 ر	16 ز	17 س
18 ش	19 ص	20 ض	21 ط	22 ظ	23 ع
24 غ	25 ـ	26 ف	27 ق	28 ك	29 ل
30 م	31 ن	32 ه	33 و	34 ى	35 ي
36 ً	37 ٌ	38 ٍ	39 َ	40 ُ	41 ِ
42 ّ	43 ْ	44 ٓ	45 ٔ	46 ٰ	47 ٱ
48 ٪	49 ٫	50 ٬	51 ٭	52 ٮ	53 ٯ
54 ٷ	55 ٸ	56 ۝	57 ۞	58 ۩	59 ۽

Fig. 5. Sixty Arabic characters each has a unique integer Id starting from zero.

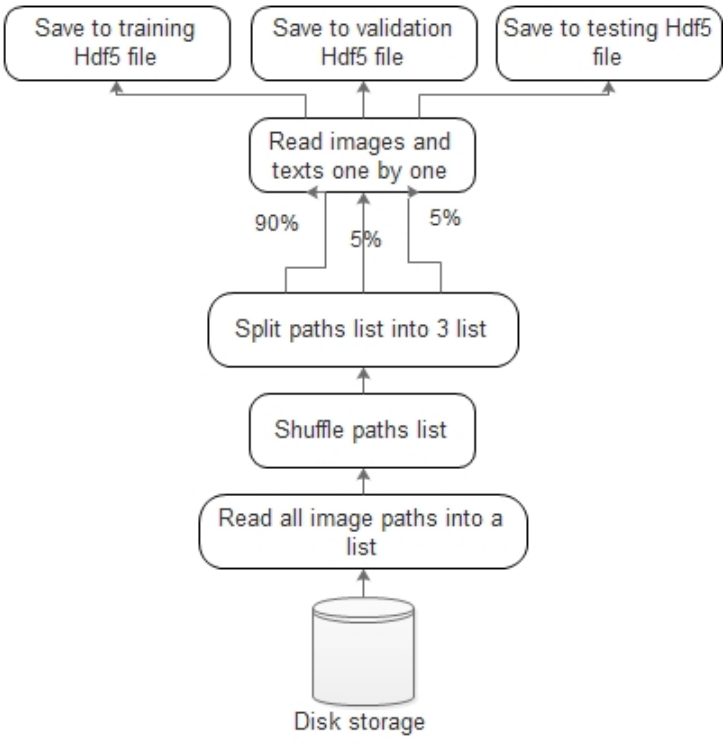

Fig. 6. HDF5 files creation steps.

Next, each image is read along with its text as they share the same unique id and then the image is converted into a 3-dimensional matrix. To make the dataset more flexible, the

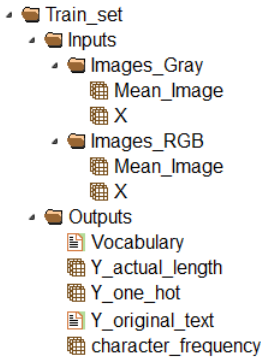

Fig. 7. Training HDF5 file architecture. Inputs consist of RGB and gray images matrices groups, while outputs contain corresponding original images annotations, text actual lengths, and one hot matrix, which corresponds to the input image matrices. Additional information includes vocabulary dictionary and character frequencies in the set.

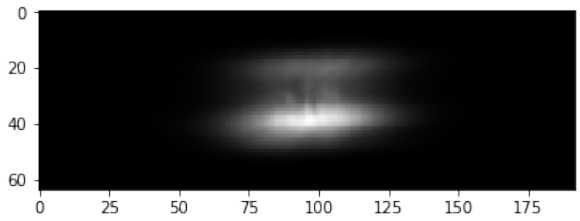

Fig. 8. Mean 192x64 gray image that can be used for normalization task.

3-dimensional matrix, which represents the RGB image, was converted into a 1-dimensional matrix that represents the gray image version of the same image. Then all the matrices are saved in an HDF5 file that has the architecture depicted in Fig. 7.

In addition to saving the text that represents each image, this text was converted into its one-hot encoding matrix using the extracted character ids, Fig. 5. To make the one-hot matrix clear, the characters ids were saved in a vocabulary list within the HDF5 files. The dataset is not balanced, that is why the characters frequencies were also saved along with the vocabulary so that each character can have its own weight in any further loss function. In addition, the lengths of these texts were saved for any further processing.

For the training set only, the mean images for the RGB and gray images have been calculated, converted to its corresponding matrices, and then saved in the training HDF5 file. Fig. 8 show the mean gray image.

IV. Dataset Statistics

Next, the properties of the Quran Text Image Dataset (QTID) were analyzed. The dataset consists of 309,720 different word image that contains 2,494,428 characters in total. Table I shows the detailed total quantity of word images and characters in the dataset. The 309,720 words were split over the training, validation, and testing sets as shown in the pie chart Fig. 9.

The number of instances per character for all the 60 characters is shown in Fig. 10. Additionally, the top 20 most repeated words are in Fig. 11 and the max length of a word is 21 characters.

TABLE I. Quantity of Words and Characters in the Dataset

	Number of Words	Number of Characters
Unique words	18,994	188,918
Holy Quran	77,430	623,607
Total (Given 4 Fonts)	309,720	2,494,428

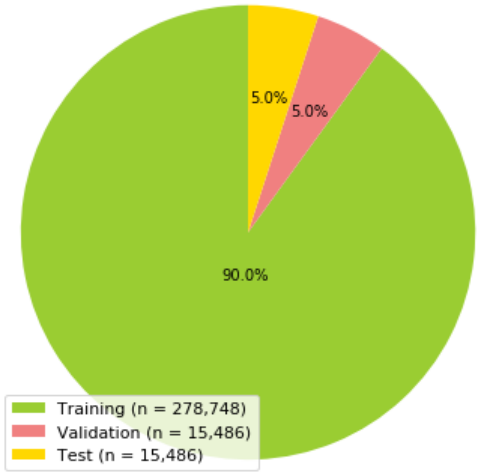

Fig. 9. Training, validation, and testing sets words distribution.

Finally, some benchmarks regarding the created HDF5 files are shown. To measure the reading time of the HDF5 file of the dataset, the testing dataset was split into 120 mini batches each consists of 128 images. Then some experiments was made to read all these mini batches 100 times. Then average of the reading time for every mini batch was calculated. The illustration is in Fig. 12.

V. Experimental Evaluation

Two of the best Arabic character recognition engines were chosen to evaluate the dataset. The first one is Tesseract [13], an optical character recognition that was developed by Google and one of the most accurate open-source engines available. It can recognize many languages, which include the Arabic language. The second one is ABBYY FineReader[4], also an optical character recognition that was developed by ABBYY for the commercial use. ABBYY team mentioned that it can recognize 190 different languages and that it has an accuracy of 99% for the Arabic language.

The dataset of this study has training, validation, and testing sets. The testing set has been evaluated only with the mentioned engines. The testing set consists of 15,486 Arabic word images, which contain 124,746 characters. In addition, it contains every possible character that is covered.

The recognition results have been evaluated using the five measures. The first one is character recognition rate (CRR) which is defined as follows:

$$CRR = \frac{\#characters - \sum LevenshteinDistance(RT, GT)}{\#characters}$$

[4]http://finereader.abbyy.com/professional/

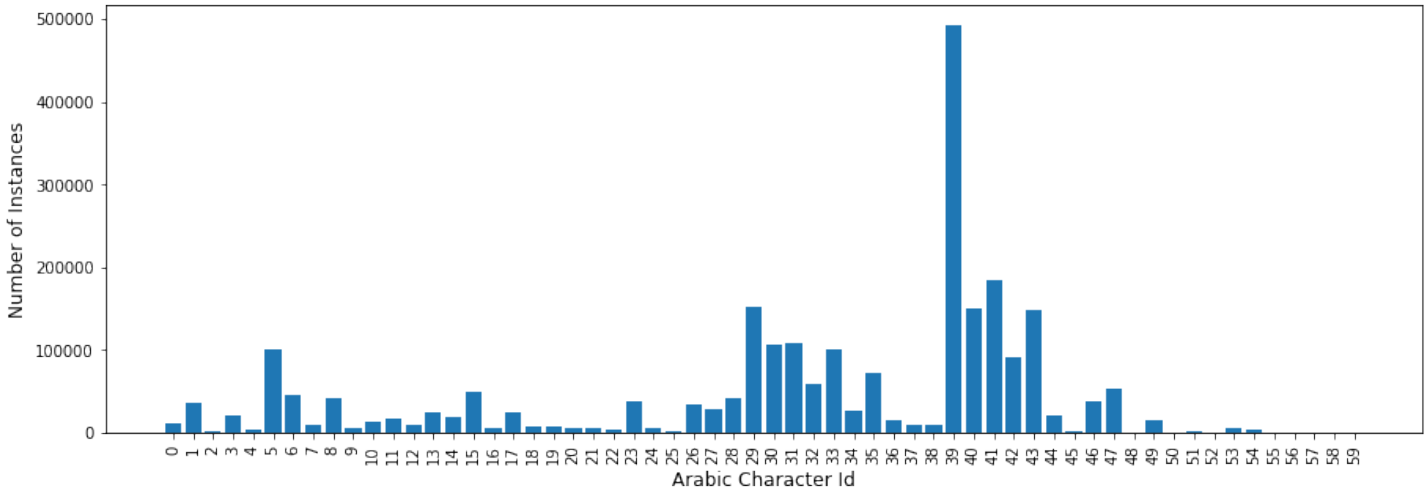

Fig. 10. Instances per character in the whole dataset.

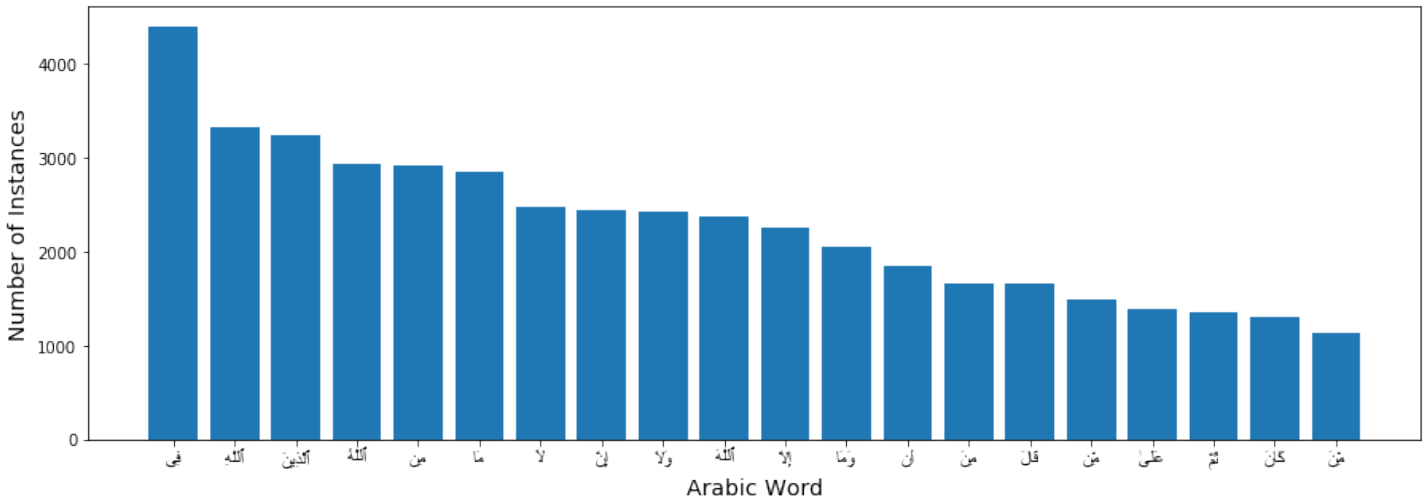

Fig. 11. Top 20 most repeated words in the whole dataset.

TABLE II. ACCURACY, PRECISION, RECALL, AND F1 RESULTS ON THE TEST SET

Engine	Accuracy (%)	Average Precision (%)	Average Recall (%)	Average F1 score (%)
Tessaract 4.0	10.67	56.73	18.44	27.83
ABBY finereader 12	2.32	34.37	4.06	7.27

Where (RT) is the recognized text and (GT) is the ground truth text. The results are in Table III. The other four measures are accuracy, average precision, average recall, and average F1 score which is defined as follows:

$$Accuracy = \frac{\#correctly\ recognized\ characters}{\#characters}$$

$$Precision_i = \frac{\#correctly\ recognized\ character\ i}{\#characters\ recognized\ as\ character\ i}$$

$$Recall_i = \frac{\#correctly\ recognized\ character\ i}{\#character\ iappearances}$$

$$F1_i = \frac{2}{1/Precision_i + 1/Recall_i}$$

Before taking these measures, all the ground truth text was first aligned with the recognized text and then the confusion matrix for all the characters was calculated to easily extract the mentioned measures. Results are in Table II. The character with 3 as Id was Tessaract top precision character while the character with 33 as Id was the top for ABBYY FineReader. To know the character Ids refer to Fig. 5.

Finally, none of the used engines has recognized a complete word image with 100% accuracy.

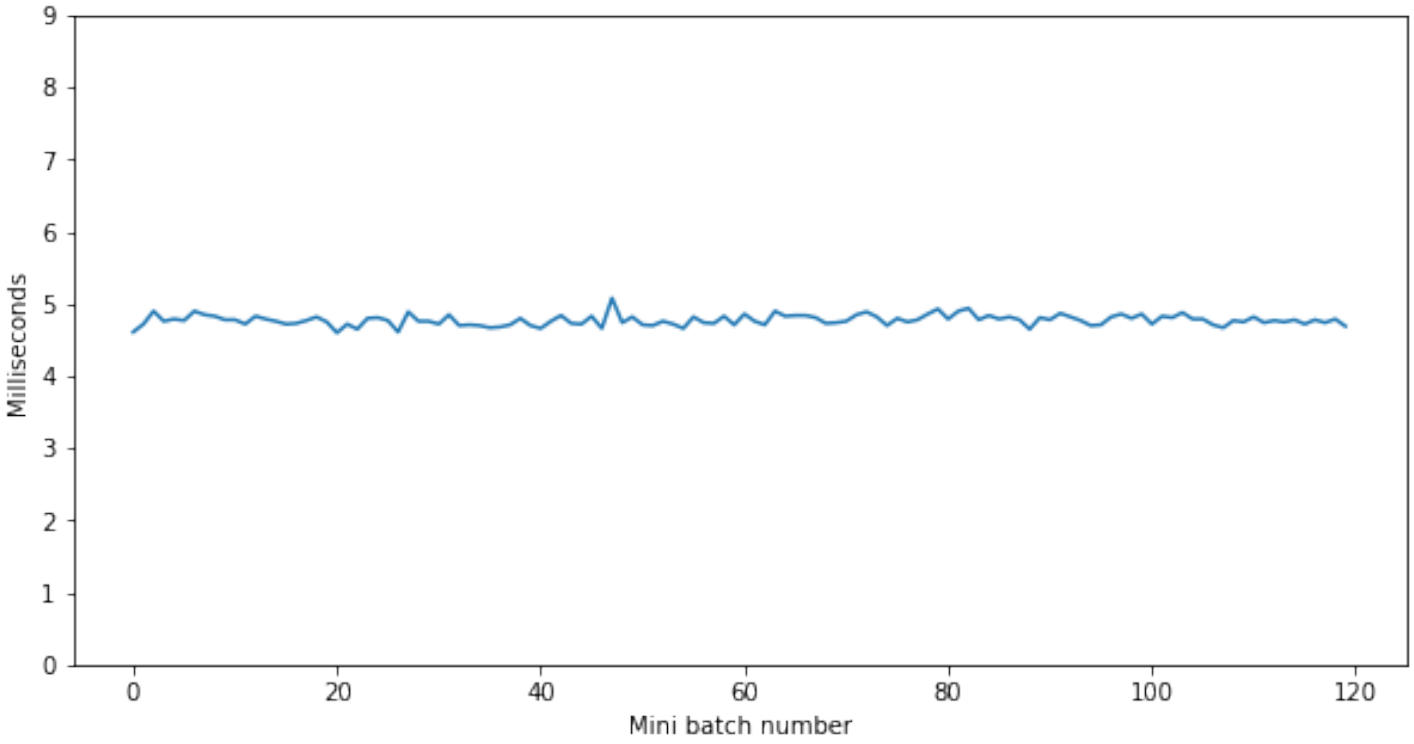

Fig. 12. HDF5 120 Mini batches reading benchmark. The experiment ran on a computer with these specifications: Intel i5 CPU, 8 DDR3 ram, and hard disk with rotation speed 7200 RPM.

TABLE III. CHARACTER RECOGNITION RATE RESULTS ON THE TEST SET

Engine	Character recognition rate (%)
Tessaract 4.0	11.4
ABBY finereader 12	6.15

VI. DISCUSSION AND CONCLUSIONS

Arabic text recognition accuracy is small compared to the accuracy of the Latin texts. In this paper, a new large dataset QTID that is made from the words of the Holy Quran was proposed in order to help machine learning models improve the accuracy of reading Arabic text from images. The dataset consists of training, validation, and testing sets that was split into 90%, 5%, and 5% respectively. It has been presented that the current best Arabic text recognition engines cannot work well with word images from the proposed dataset, which means that more work is needed to improve the Arabic text recognition task. Some of the limitations of the proposed dataset include: imbalanced characters instances, centered Arabic words images with black paddings, and dataset Arabic words do not cover the whole Arabic words dictionary. In the future, the dataset needs so improvements to the limitations by using data augmentation techniques and adding character segmentation support. In addition, the best model that can work with the proposed dataset hope to be found.

ACKNOWLEDGMENT

The authors would like to thank Alsherif Mostafa for his assistance while writing the paper, which greatly improved the manuscript.

REFERENCES

[1] T. Sobh, *Innovations and Advanced Techniques in Computer and Information Sciences and Engineering*. Springer, 2007. [Online]. Available: http://www.springer.com/gp/book/9781402062674

[2] ——, *Innovations and Advances in Computer Sciences and Engineering*. Springer, 2010. [Online]. Available: http://www.springer.com/us/book/9789048136575

[3] N. Tagougui, M. Kherallah, and A. M. Alimi, "Online arabic handwriting recognition: a survey," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 16, no. 3, pp. 209–226, Sep 2013. [Online]. Available: https://doi.org/10.1007/s10032-012-0186-8

[4] V. M. S. Mozaffari, K. Faez, "Strategies for large handwritten farsi/arabic lexicon reduction," *proceedings of the Ninth International Conference on Document Analysis and Recognition*, pp. 98–102, 2007. [Online]. Available: http://ele.aut.ac.ir/ image-proc/downloads/IFHCDB.htm

[5] F. Slimane, R. Ingold, S. Kanoun, A. M. Alimi, and J. Hennebert, "A new arabic printed text image database and evaluation protocols," in *2009 10th International Conference on Document Analysis and Recognition*, July 2009, pp. 946–950. [Online]. Available: http://diuf.unifr.ch/diva/APTI/

[6] S. Yousfi, S. A. Berrani, and C. Garcia, "Alif: A dataset for arabic embedded text recognition in tv broadcast," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, Aug 2015, pp. 1221–1225.

[7] F. Chabchoub, Y. Kessentini, S. Kanoun, V. Eglin, and F. Lebourgeois, "Smartatid: A mobile captured arabic text images dataset for multi-purpose recognition tasks," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Oct 2016, pp. 120–125.

[8] S. A. Azeem, M. El Meseery, and H. Ahmed, "Online arabic handwritten digits recognition," in *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*. IEEE, 2012, pp. 135–140.

[9] M. W. F. Sherif Abdou, "Large vocabulary arabic handwritten online character recognition competition," http://www.altec-center.org/conference/?page_id=87, 2011. [Online]. Available: http://www.altec-center.org/conference/?page_id=87

[10] H. El Abed, V. Märgner, M. Kherallah, and A. M. Alimi, "Icdar 2009 online arabic handwriting recognition competition," in *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*. IEEE, 2009, pp. 1388–1392.

[11] V. M. S. Mozaffari, K. Faez and H. El-Abed, "Strategies for large handwritten farsi/arabic lexicon reduction," *proceedings of the Ninth International Conference on Document Analysis and Recognition*, pp. 98–102, 2007. [Online]. Available: http://ele.aut.ac.ir/ imageproc/downloads/IFHCDB.htm

[12] S. A. Mahmoud, I. Ahmad, W. G. Al-Khatib, M. Alshayeb, M. T. Parvez, V. M?rgner, and G. A. Fink, "Khatt: An open arabic offline handwritten text database," *Pattern Recognition*, vol. 47, no. 3, pp. 1096 – 1112, 2014, handwriting Recognition and other PR Applications. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320313003300

[13] R. Smith, "An overview of the tesseract ocr engine," in *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, vol. 2. IEEE, 2007, pp. 629–633.