JJEE
Volume 8, Number 4, December 2022
Pages 307-321
Jordan Journal of Electrical Engineering
ISSN (Print): 2409-9600, ISSN (Online): 2409-9619

Towards Building a Speech Recognition System for Quranic Recitations: A Pilot Study Involving Female Reciters

Suhad Al-Issa[1], Mahmoud Al-Ayyoub[2], Osama Al-Khaleel[3*], Nouh Elmitwally[4]

[1, 3] Department of Computer Engineering, Jordan University of Science and Technology, Irbid, Jordan
E-mail: oda@just.edu.jo
[2] Artificial Intelligence Research Center, College of Engineering and Information Technology, Ajman University, Ajman, UAE
[2] Department of Computer Science, Jordan University of Science and Technology, Irbid, Jordan
[4] School of Computing and Digital Technology, Birmingham City University, B4 7XG Birmingham, United Kingdom.
[4] Faculty of Computers and Artificial Intelligence, Cairo University, Giza 12613, Egypt

Received: February 08, 2021 *Revised: November 02, 2022* *Accepted: November 12, 2022*

***Abstract*—** This paper is the first step in an effort toward building automatic speech recognition (ASR) system for Quranic recitations that caters specifically to female reciters. To function properly, ASR systems require a huge amount of data for training. Surprisingly, the data readily available for Quranic recitations suffer from major limitations. Specifically, the currently available audio recordings of Quran recitations have massive volume, but they are mostly done by male reciters (who have dedicated most of their lives to perfecting their recitation skills) using professional and expensive equipment. Such proficiency in the training data (along with the fact that the reciters come from a specific demographic group; adult males) will most likely lead to some bias in the resulting model and limit their ability to process input from other groups, such as non-/semi-professionals, females or children. This work aims at empirically exploring this shortcoming. To do so, we create a first-of-its-kind (to the best of our knowledge) benchmark dataset called the Quran recitations by females and males (QRFAM) dataset. QRFAM is a relatively big dataset of audio recordings made by male and female reciters from different age groups and proficiency levels. After creating the dataset, we experiment on it by building ASR systems based on one of the most popular open-source ASR models, which is the celebrated DeepSpeech model from Mozilla. The speaker-independent end-to-end models, that we produce, are evaluated using word error rate (WER). Despite DeepSpeech's known flexibility and prowess (which is shown when trained and tested on recitations from the same group), the models trained on the recitations of one group could not recognize most of the recitations done by the other groups in the testing phase. This shows that there is still a long way to go in order to produce an ASR system that can be used by anyone and the first step is to build and expand the resources needed for this such as QRFAM. Hopefully, our work will be the first step in this direction and it will inspire the community to take more interest in this problem.

***Keywords*—** Holy Quran; Recitations; Speech recognition; DeepSpeech; Word error rate.

1. INTRODUCTION

The Holy Quran is an essential component in the lives of Muslims. Besides learning and structuring their lives from this holy book, Muslims recite the Holy Quran in each of their five daily prayers. Reading the Holy Quran is one of the main acts of worship in which a Muslim can partake. The Holy Quran is considered the main source of Islamic law which leads to the fairest assessment. Besides the legislative and guidance components of the Quran, many Muslims utilize the Quran as a form of art especially when it comes to its inscription and recitation. People would follow famous reciters from all over the Muslim world to learn the correct reading of the Quran.

* Corresponding author
Article's DOI: 10.5455/jjee.204-1612774767

The Holy Quran is the words of Allah (the only GOD). It is written in the classical Arabic language. It consists of 114 different chapters (or Surahs), where each Surah consists of specific number of verses (Ayahs). The Quran is also divided into 30 parts. Each part is called a Juz'. The number of unique words in the entire Quran is (77,794) words (including the Basmalah, which is a special phrase added at the start of each Surah). It is considered the main source of Arabic language vocabulary. The words in the Arabic language comprise 28 different letters (29 letters if we count the hamza, 'ء').

Automatic speech recognition (ASR) is a process by which a computer takes recorded speech in the form of signals and converts it into words. An ASR system generally consists of three basic stages: pre-processing, feature extraction, and classification [1-5]. Over the past few years, ASR systems have evolved from classical methods like CMU Sphinx [6-10] to modern methods that use deep learning (DL) and deep neural networks (DNN) [11, 12].

To function properly, ASR systems require a huge amount of data for training. Surprisingly, the data readily available for Quranic recitations suffer from major limitations. Specifically, the currently available audio recordings of Quran recitations have massive volume, but they are mostly done by male reciters (who have dedicated most of their lives to perfecting their recitation skills) using professional and expensive equipment. Such proficiency and quality in the training data - along with the fact that the reciters come from a specific demographic group; adult males - will most likely lead to some bias in the resulting model and limit their ability to process input from other groups, such as non-/semi-professionals, females or children. This work aims at empirically exploring this shortcoming.

The contributions of this work can be summarized as follows:

a) Creating a Quranic recitations dataset for reciters from both genders (male and female).
b) Applying the DeepSpeech model on the Quranic recitations' datasets and analyzing the results.

The rest of this paper is organized as follows. Section 2 provides a survey of the related work. Section 3 talks about our dataset, which is the main contribution of this work. Section 4 presents the methodology for building the ASR systems. The experimental results are provided and discussed in section 5. Finally, the paper is concluded in section 6.

2. RELATED WORK

Many researchers and practitioners have focused on the automated processing of the recitations of the Holy Quran. Some of these researchers used "traditional" ASR techniques such as the open-source CMU Sphinx framework [13]; where an ASR system for the Arabic language is introduced and expanded to consider the recitation of the Holy Quran.

The authors of [14] discussed the challenging aspects of building a speech recognition based system for verifying the verses of the Holy Quran on the Internet as well as their solutions. They examined the techniques used to deal with finite vocabulary and how modeling can avoid some complexities of the system in the phonetic domain of the language and dictionary model. A system that identifies errors in Quran recitation and shows where errors occur was presented.

For the purpose of developing a system to help its users memorize the Holy Quran without the help of others, the authors of [15] relied on speech recognition for Quran recitations. The authors differentiate between the task of simply developing a generic ASR

system and the more challenging task of developing an ASR-based language teacher. The system was implemented to be speaker-dependent through the inclusion of steps for pre-processing speech signals, feature extraction, and pattern matching. After extracting the features, they are fed to a neural network (NN) for acoustic modeling and classification.

The authors of [16] proposed a new way to extract what they considered as suitable features for Arabic speech recognition. To identify Arabic vowels, they considered the wavelet packet transform (WPT) method [17-19] with standard modular arithmetic and NN. For classification, they gave a total of 266 coefficients for probabilistic neural network (PNN). In their results, they showed that the modular wavelet packet and neural networks (MWNN) system, proposed in [20], achieved the best recognition rate.

Using a simplified set of Arabic phonetics, the authors of [21] aimed at simplifying the process of creating a language model for Quran speech recognition. CMU Sphinx 4 is used to train and evaluate the language model of the Hafs method of Quranic recitation. A 1.5% lower word error rate (WER) was achieved.

The researchers in [22] developed an intelligent learning system (ITS), which provides direct feedback to learners without the intervention of a human teacher. The system was implemented using the ITS builder (ITSB) authoring tool, which provides an intelligent educational system for teaching proper recitation and pronunciation of the Holy Quran, i.e., "Tajweed" according to the Hafs narration. Teachers and students evaluated the system by reciting the Holly Quran and the results were positive.

A model of technological application in the evaluation of the recitation of the Quran was presented in [23]. Scientific methods were applied in the analysis of the correct recitation based on appropriate rules. The work focused on detecting errors in recitation. It addressed the difficult issues of representing and classifying characters based on digital speech processing (DSP) techniques [24]. This was used to - automatically - identify, categorize and recognize the Quran recitation speech for the representation function. The main goal of the system was to help Muslims recite the Quran correctly.

Concurrently to our work, a Quranic dataset called QDAT, was recently published on the Kaggle website [25]. QDAT includes more than 1,500 audio recordings of Quranic recitations with Tajweed. The recordings cover the recitations of 165 participants (Male and Female). Forty-one males, with an age ranges from 8 to 62 years, recorded 351 audio files. On the other hand, 124 females, with an age ranges from 5 to 68 years, recorded 1,159 audio files. The audio files are available in WAV format with 11 KHz sample rate, MONO channel and 16-bit resolution. QDAT also includes a CSV file that contains further information about the audio files like the correctness of the recitation of the three recitation rules: The "Separate Extension", the "Accentuated" and the "Hiding". The final column shows the correctness of the reading based on these three rules, where the correctness of the recitation with Tajweed rules is manually annotated by an expert. The goal of this dataset is the correct application of these three Tajweed rules. The QDAT dataset covers only one verse in the Quran which is (قَالُواْ لَا عِلۡمَ لَنَآۖ إِنَّكَ أَنتَ عَلَّٰمُ ٱلۡغُيُوبِ), [Surah: Al-Ma'idah, Verse: 109].

There have been some notable efforts related to Quranic studies from a computational perspective. For example, the work in [26] proposed an automated Tajweed verification engine that is dedicated to the Quran Learn. Experiments using it were conducted on the j-QAF students in elementary schools in Malaysia. Another example is the work in [27],

which taught adolescents the correct recitation of the Holy Quran. It also familiarized them with the provisions of the reciting of the Holy Quran.

The comprehensive review in [28] presented the latest trends in technologies to recognize Arabic speech. This review focuses on machine learning and DL techniques in building ASR systems. The hybrid hidden Markov model (HMM)- DNN models, the convolutional neural networks (CNN) model, the recurrent neural vetwork (RNN) model, and the end-to-end DL models have been discussed as a revolution in improving the ASR performance. The use of the end-to-end model for Arabic speech and the Arabic language is discussed. In addition, the latest services and toolkits currently available and necessary for building comprehensive models for ASR are presented.

The authors in [29] designed and developed a phonetic dictionary generator for modern standard Arabic and Arabic disordered speech. Their system takes Arabic texts as input and outputs their corresponding phoneme sequences using pre-defined rules. The goal is to use the phonetic dictionary for ASR research and development.

3. QURAN RECITATIONS BY FEMALES AND MALES (QRFAM) DATASET

QRFAM is the benchmark dataset collected in this work as one of the main contributions made to the community. The QRFAM dataset includes a large number of widely varied audio recordings of Quranic recitations. The recordings are for males and females. The males are very famous professional adult reciters, some of whom are native speakers of Arabic. The recordings by those professional reciters cover the entire Holy Quran. However, we had to select a subset of their recordings to ensure that the dataset is balanced and homogenous across the two genders. The females, on the other hand, are semi-professional reciters who work at community centers for teaching and memorizing the Holy Quran in different Arab countries. Some of them are adults and the rest are not. The recordings by the females cover a subset of the Holy Quran (one Surah or more per reciter). The audio recordings are WAV files where each file is for only one verse from the Quran (one Ayah). For each recording, the corresponding Quranic text of each verse is added to the dataset. The Hafs narration for the audio and the Uthmani style for the text are adopted. The recordings by the males are long and of high quality compared to those by females. Table 1 presents statistical information about the QRFAM dataset.

Table 1. The QRFAM dataset statistical information.

QRFAM info.	Males	Females
Number of audio records	5,660	5,660
Number of reciters	21	21
Number of hours	24.12	13.04
Minimum recording time [s]	1:10	1:1
Maximum recording time [s]	45:90	45:67
Average recording time [s]	15:34	8:29

3.1. Data Collection

Despite having many recitations available online, most of them are recorded by professional male adult reciters. Training an ASR system using only male recordings would

not help much in recognizing a female recitation. Unfortunately, there is limited availability of annotated audio recordings, and an insufficient dataset in terms of size and quality, especially when it pertained to female reciters. Our work comes to fill this gap.

3.1.1. Males Dataset Collection

The nature of the recorded male recitations of the Holy Quran, available online, makes it difficult to directly use them for training and testing any ASR system. Most of them are continuous recitations of the entire Surah or long records of some verses. They also contain many moments of silence and many sentences are repeated during a single record. Moreover, sometimes, they are available in low quality. Fortunately, the everyayah website [30] provides Quranic records that include a recitation of each verse separately. Therefore, we use the audio records in this site for the male recitations part of our dataset. Specifically, we use the version known as "Al-Murattal" because it has shorter audio lengths.

3.1.2. Females Dataset Collection

In order to complete the dataset, we had to collect female recitations. The public availability of female recitations is rare due to religious and social reasons. Therefore, we sought the help of Quran teaching and memorization centers in our home country of Jordan. As a result, many female volunteers from these centers participated in preparing the audio recordings using the WhatsApp application. Each female was assigned several Surahs from the Quran and one verse was recorded per audio file. The collected audio recordings were not sufficient when compared with the collected male recordings. Therefore, we searched through Facebook groups looking for females willing to participate in our study. Those females are from different Arab countries like Egypt, Syria and Algeria. They volunteered to prepare the recordings the same way the Jordanian females did. With these new recordings, we were able to obtain female recordings of the entire Holy Quran. It took seven months to finalize this task.

3.2. QRFAM Preperation

To use the DL model and ensure excellent results, the data must be restructured, cleaned, transformed into a specific form and split into subsets with suitable and specific percentages.

3.2.1. Restructuring and Cleaning

In our case, the data is the audio records for the male and the female reciters. The restructuring and cleaning took place through the following phases, where we wrote Python scripts to perform these phases:

a) Creating the CSV files in Mozilla's DeepSpeech format for the unprocessed dataset. The file is made up of three columns: wav_filename, wav_filesize and transcript as shown in Fig. 1. The wav_filename column specifies the full path for each audio file; the wav_filesize column specifies the size of each audio file and the transcript column specifies the corresponding text of each audio file. The CSV file must be encoded using UTF-8 encoding. We used the Arabic Quranic text in Uthmani style.

b) Removing the duplicate audio files: many verses are repeated in the Quran throughout different Surahs, and there are many verses that are repeated in the same Surah through the Holy Quran. For example, the Ayah of (فَبِأَيِّ آلاءِ رَبِّكُمَا تُكَذِّبَانِ) from Surat Al-Rahman is repeated 31 times. We did not do any action for these two types of repetition in order to make the training more effective. We only deleted the duplicated audio files for both (الإستعاذة & البسملة) in all the records of the same reciter.
c) Deleting the empty audio files and the corrupted audio files: for each audio file, we found the audio length in seconds and then filtered out those with a time that is less than 1 s. After that, we manually deleted the empty files.
d) Transferring each audio file in the datasets into a specific format and encoding. In fact, Mozilla's DeepSpeech handles WAV audio files with a 16-bit Mono and 16 kHz sample rate.
e) Sorting the audio files ascendingly according to their names. The names are identical to the order of the verses in each Surah in the Quran. The sequence of the Surahs follows the Quran order. The reason is to facilitate the process of tracing and finding out whether the defect is in the text or in the audio file.
f) Adding a new audio_duration column that represents the time in seconds of each audio file. The processing is limited by the memory recourses that we have (12 GB). Therefore, we experimentally found out that the feasible audio length must be less than or equal to 45 s.
g) Filtering the audio files according to maximum audio_duration (45 s): all audio files with audio time greater than 45 s are filtered out from the dataset.
h) Splitting the audio files according to the reciters: we create different folders with the reciters' names. The audio files of the reciter are stored in the folder with the reciter's name. Thus, we avoid the intersection in the audio files among the reciters.
i) Dividing the audio files into (80%) for the training set, (10%) for validation and (10%) for testing. The largest training percentage is to guarantee good training for the system.
j) Preprocessing and normalization of the text: the Holy Quran contains diacritics marks, stop signs like {◌, ◌ , ◌ , ◌ , ◌ , etc.}, control and adjustment marks {◌ّ ;◌ٰ ,◌ۜ ,◌ۛ , etc.}, and other accent marks written in Uthmani style. We normalized the text in a transcript column, where the text normalization includes:
   - Removing diacritics marks, stop signs, control and adjustment marks and other accent marks.
   - Making two versions of a text: one is free of diacritics marks and the other contains some diacritics (Damma, Fatha, Kasra, Tanween al-Dam, Tanween al-Fath, Tanween al-Kasr, Sukoon and Shaddah).
   - For the letter alif (ا) , there are only the following shapes { ء , ى, ئ , آ , إ , أ , ا} .
   - Converting a special character from Uthmani style to orthographic style (الرسم الإملائي) [31], as shown in Table 2.

```
wav_filename,wav_filesize,transcript
/media/suhad/Backup/Male/full_data/test/Mohammad_al_Tablaway/001000.wav,185880,أَعُوذُ بِاللَّهِ مِنَ الشَّيْطَانِ الرَّجِيمِ
/media/suhad/Backup/Male/full_data/test/Mohammad_al_Tablaway/001001.wav,192462,بِسْمِ اللَّهِ الرَّحْمَانِ الرَّحِيمِ
/media/suhad/Backup/Male/full_data/test/Mohammad_al_Tablaway/001002.wav,192462,الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ
/media/suhad/Backup/Male/full_data/test/Mohammad_al_Tablaway/001003.wav,133948,الرَّحْمَانِ الرَّحِيمِ
```

Fig. 1. The CSV files format in Mozilla's DeepSpeech.

Table 2. Results of converting the special characters from Uthmani style to orthographic style.

Character		Symbol	Uthmani Style	Orthographic Style
The deleted Alif	(الألف المحذوفة)	'	﴿مَٰلِكِ﴾	مالك
The deleted Waw	(الواو المحذوفة)	و	﴿ٱلْغَاوُۥنَ﴾	الغاوون
The deleted Ya'a	(الياء المحذوفة)	ے	﴿لَا يَسْتَحْيِۦ﴾	لا يستحيي
The deleted Noon	(النون المحذوفة)	ن	﴿نُّنجِ﴾	ننجي
The deleted Allam	(اللام المحذوفة)	ل	﴿ٱلَّيْلِ﴾	الليل
The extra Alif	(الألف الزائدة)		﴿لِشَاْىْءٍ﴾	لشيء
The extra Waw	(الواو الزائدة)		﴿سَأُوْرِيكُمْ﴾	سأريكم
The extra Ya'a	(الياء الزائدة)		﴿بِأَيْيِدٍ﴾	بأيدٍ
Convert a letter to another letter	(قلب حرف إلى حرف)		﴿ٱلصَّلَوٰةَ﴾	الصلاة
What fell into it from the disconnection	(ما وقع فيه من القطع)		﴿مَالِ هَٰذَا﴾	مالهذا
What fell into it from the connection	(ما وقع فيه من الوصل)		﴿يَبْنَؤُمَّ﴾	يا ابن أم
Al-Hamzeh	(الهمزة)		﴿ٱلْقُرْءَانُ﴾	القرآن
Dawing Ha'a Al Ta'neeth with Al Taa Al-Mabsootah	(رسم هاء التأنيث بالتاء المبسوطة)	هـ ← ت	﴿نِعْمَتَ ٱللَّهِ﴾	نعمة الله

4. MOZILLA'S DEEPSPEECH IMPLEMENTATION

In this work, we use the celebrated neural speech recognition model known as DeepSpeech from Mozilla [32]. Implemented on Google's TensorFlow framework and released as an open-source project, Mozilla aimed at allowing the general public to have access to an industry-level ASR system. With contributions from many researchers and practitioners, the open speech-to-text (STT) engine from Mozilla is a product of fruitful community collaboration. The 0.5.1 release of the model (from 2019) was pre-trained on several publicly available American English datasets, such as Librispeech, SwitchBoard and Fisher. It was tested on the LibriSpeech clean test set and the results were excellent with only 8.22% WER.

Due to DeepSpeech's open-source nature, its proven power, flexibility, effectiveness, its very professional implementation and proper documentation, we choose it as the ASR to conduct our experiment and prove our point about the need for more diversified resources. Specifically, the 0.7.0-alpha.1 release of DeepSpeech is the one we used in our work. Unidirectional RNN with long short-term memory (LSTM) cells are used in the Mozilla

engine [26]. LSTM is an extension of RNN from two decades ago that enhanced RNN's ability to carry information across training steps by introducing cell state vectors, which are controlled by the input, output and forget gates mechanisms. This allowed the network to learn long-term dependencies and greatly helped with the vanishing gradient problem [33].

Fig. 2 (adapted from [11]) gives a general overview of the DeepSpeech model's architecture as we adopt it into our setting. Each time step of this model relies on the output of the fully connected layers (FCL) at that time, in addition to the LSTM cell state of the previous time step. Additionally, the very popular adaptive moment estimation (Adam) optimizer is used in the training process, which requires less fine-tuning of the hyper-parameters compared with the Nesterov process. The hyper-parameter values of the pre-trained models are provided in the Mozilla GitHub repository, which allows us to perform transfer learning effectively reducing the training time and computational requirements. Finally, a 5-gram Language Model (LM) is created using the KenLM tool [34].

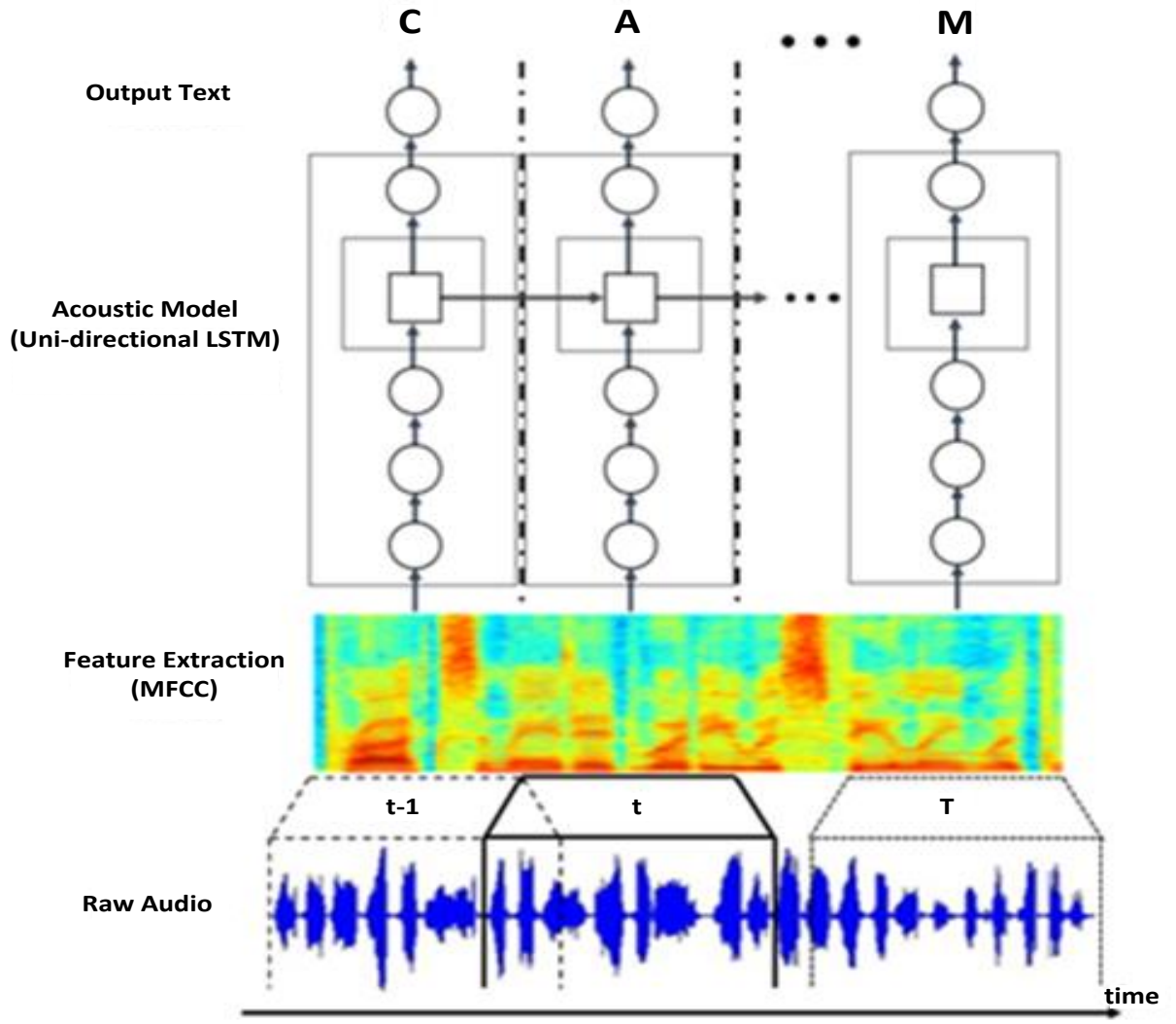

Fig. 2. Mozilla's DeepSpeech model structure over time.

As with most ASR systems, selecting the right features plays an important role in their success. For DeepSpeech, the features to be used are the Mel-frequency cepstrum coefficients

(MFCCs) instead of the spectrogram features. Built on the human peripheral auditory system, MFCC features are among the most used features for modern ASR systems [35].

The Mel-frequency cepstrum (MFC) representation is the product of a cosine transformation of the real logarithm of the short-term power spectrum expressed on a Mel-frequency scale. In terms of human perception, the Mel-frequency scale for low frequencies (up to 1 kHz) is roughly linear and for higher frequencies is logarithmic. Mel-frequency scale is used to catch the phonetically features of speech [35].

Five steps are followed to extract MFCC features from an audio clip. They are:

a) Framing blocking and windowing
b) Discrete Fourier transform (DFT) spectrum
c) Mel-Frequency warping.
d) Logarithmic operation.
e) Discrete cosine transform (DCT).

The MFCCs are the coefficients resulting from the DCT. The first few coefficients are used to represent an audio frame since that is where most of the information is maintained. E.g., the default number of MFCCs used by the hidden Markov model toolkit (HTK) from Cambridge University is 13. These include the 0th coefficient, which reflects the average energy of the spectrum [36]. With higher-order coefficients representing increased levels of spectral information, the MFCC feature vector provides a smooth version of the log energy spectrum; thus transforming the speech signal into a more compact low-dimensional representation [35].

Following common notation in the literature [11], we denote a single utterance by (x) and its label by (y) that is sampled from a training set as the following:

$$S = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots\}. \quad (1)$$

where each single utterance $x^{(i)}$ represents a time-series with length $T^{(i)}$, where each time-slice is a vector of extracted features, $x_t^{(i)}$ where $t = 1,\dots, T^{(i)}$.

Through using MFCCs for features, $x_{t,p}^{(i)}$ denotes the pth MFCC feature in the signal frame at time t. RNN's aim is to transform an input sequence x into a sequence of character probabilities for the transcription y, with $\hat{y}_t$ defined as follows:

$$\hat{y}_t = \mathbb{P}(c_t | x) = \frac{\exp(W_k^{(l)} h_t^{(l-1)} + b_k^{(l)})}{\sum_j \exp(W_j^{(l)} h_t^{(l-1)} + b_j^{(l)})} \quad (2)$$

where $c_t \in \{\text{character}, \text{space}, \text{blank}\}$.

As shown in Fig. 2, the RNN is composed of five hidden layers denoted as $h^{(l)}$. In the first layer $h_t^{(1)}$, at each time t, the output depends on the MFCC frame x_t along with frame context C on each side (C=9). The remaining layers work on independent data for each time step. So, for each time t, the first three layers are calculated by:

$$h_t^{(l)} = g(W^{(l)} h_t^{(l-1)} + b^{(l)}) \quad (3)$$

where $g(z) = \min\{\max\{0, z\}, 20\}$, a rectified linear unit (ReLU) activation function, $W^{(l)}$ is the weight matrix, and $b^{(l)}$ is the bias parameter for layer l. The fourth layer is LSTM recurrent layer [37], where it contains a set of hidden units with forward recurrence $h^{(f)}$:

$$h_t^{(f)} = g(W^{(4)} h_t^{(3)} + W^{(f)} h_{t-1}^{(f)} + b^{(4)}). \quad (4)$$

where $h^{(f)}$ is calculated sequentially from t = 1 to $t = T^{(i)}$ for the i-th utterance. The fifth layer takes the forward units as inputs as the following:

$$h^{(5)} = g(W^{(5)} h^{(f)} + b^{(5)}). \quad (5)$$

The output layer is standard logits that represents the predicted character probabilities for each time slice t and character k in the alphabet:

$$h_{t,k}^{(6)} = \hat{y}_{t,k} (W^{(6)} h_t^{(5)})_k + b_k^{(6)}. \quad (6)$$

where $b_k^{(6)}$ denotes the k-th bias and $(W^{(6)} h_t^{(5)})_k$ the k-th element of the matrix product. When we have calculated a prediction for

$$\hat{y}_{t,k} = \mathbb{P}(c_t | x) = \frac{\exp(W_k^{(6)} h_t^{(5)} + b_k^{(6)})}{\sum_j \exp(W_j^{(6)} h_t^{(5)} + b_j^{(6)})}, \quad (7)$$

we calculate the connectionist temporal classification (CTC) loss [38] $\mathcal{L}(\hat{y}, y)$ to calculate the error in prediction. Through the training phase, we perform the gradient $\nabla\mathcal{L}(\hat{y}, y)$ regarding the network outputs given the truth reference character sequence y, where calculating the gradient regarding all the model parameters done through back-propagation through the rest of the network and using the Adam [39] algorithm for optimizing the training.

5. EXPERIMENTAL RESULTS

In this section, we present the experiments we conduct, and discuss the obtained results.

5.1. Experimental Setting

All experiments in this work are carried out on an HP workstation with an Intel Xeon® W-2125 4.00GHz CPU with 8 Cores, 32 GRAM, two hard disk storage (256 GB SSD and 1 TB HDD), and one NVIDIA TITAN XP GPU, with 12 GB memory and 3840 CUDA Cores. The operating system running on the workstation is 64-bit Ubuntu Linux 18.04.4 LTS. We use Mozilla's DeepSpeech v0.7.0-alpha.1 implemented with Python using TensorFlow DL framework [32]. For our work, we use a Python 3.6 virtual environment with TensorFlow-GPU 1.15 installed and built with CUDA 10.1 and CuDNN 7.6.5.

5.2. Experiments

Four different experiments are conducted. The parameters that are used for the feature's extraction, the training and the testing are exhibited in Tables 3, 4 and 5, respectively. These parameters are similar to those used in the Mozilla DeepSpeech project for the English language.

Table 3. DeepSpeech's hyperparameters used for the features extraction.

DeepSpeech's Hyperparameters	Value
Number of Hidden Neurons	2048
Number of Output Neurons	54
The learning rate	0.0001
The dropout rate	0.15
The ReLU clipping value	20
The train batch size	24
The development batch size	48

Table 4. DeepSpeech's hyperparameters used for the training.

DeepSpeech's Hyperparameters	Value
Number of MFCC features	26
Length of the audio window [ms]	32
Step of the audio window [ms]	20
Step the audio window [samples]	320
Length the audio window [samples]	512
The sample rate [Hz]	16000

Table 5. DeepSpeech's hyperparameters used for the evaluation and testing.

DeepSpeech's Hyperparameters	Value
The beam width	1024
The language model decoding 'α'	0.75
The language model decoding 'β'	1.85
The test batch size	48

In the first experiment (which we call 'MMM'), we trained our male model using 4,541 audio files by 13 male reciters. The length of the audio files ranges from 1.386 to 45.897 s with an average of 16.883 s. In this case, the data represents 80.229% of the total audio files. For validation, we formed a development set of 559 audio files by 3 male reciters with an audio length that ranges from 1.097 to 45.792 s and an average of 9.5885 s. The development data represents 9.876% of the total audio files. The male model was tested using the rest 9.893% audio files that form 560 audio files by 5 male reciters with an audio length ranging from 1.149 to 45.688 s and an average of 8.552 s. We got a 0.406 WER and a 0.232 character error rate (CER).

In the second experiment (which we call 'MMF'), we used the same training and development sets of the first experiment. However, the male model was tested using 560 audio files of 5 female reciters. In this case, the testing data represents 9.893% and the audio length ranges from 1.1 to 45.672 s with an average of 6.781 s. A WER of 0.968 and a CER of 0.758 were obtained.

We notice that the performance of the model in the first experiment is better than that of the second one. This indicates that testing a female recitation on a male model results poor performance.

In the third experiment (which we call 'FFF'), we trained our female model using 4,541 audio files by 13 female reciters. The length of the audio files ranges from 1.4 to 45.60 s with an average of 8.55 s. In this case, the data represents 80.22% of the total audio files. For validation, we formed a development set of 559 audio files by 3 female reciters with an audio length ranging from 1.74 to 45.18 s and an average of 7.72 s. The development data represents 9.87% of the total audio files. The female model was tested using the rest 9.89% audio files that form 560 audio files by 5 female reciters with audio length ranges from 1.1 to 45.67 s and an average of 6.78 s. We got a 0.608 WER and a 0.396 CER.

Comparing the WER in the first experiment and the WER in the third experiment shows that the male model performs better than the female model. The justification for this is that the quality of the records and the quality of the recitation in dataset by the male reciters are better than those in the dataset by the female reciters.

In the fourth experiment (which we call 'FFM'), we used the same training and development sets as in the third experiment. However, the testing is performed using 560 audio files by 5 male reciters. These 560 audio are of recording length that ranges from 1.149 to 45.68 s with an average of 8.55 s. The testing audio files represent 9.89% of all audio files. The WER was 0.966 and the CER was 0.664. All of the WER resulted from the conducted experiments are listed in Table 6.

Table 6. Summary of the WER experimental results.

Experiment	WER
MMM	0.406
MMF	0.968
FFF	0.608
FFM	0.966

From all the experiments, one can conclude that training the DeepSpeech model on data that belong to a certain gender and testing it using data that belong to the other gender results poor performance.

Fig. 3 illustrates examples from the 'MMM' experiment for a correct prediction case and a wrong prediction case.

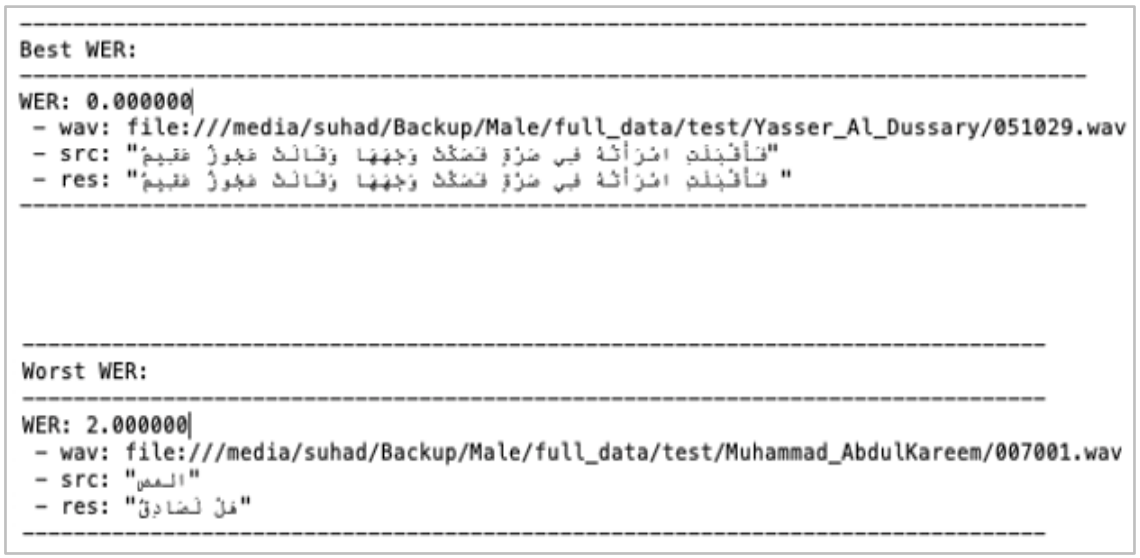

Fig. 3. Examples from the 'MMM' experiment.

We compare our work with the work presented in [40], which introduces an ASR engine-based Arabic recognizer using Sphinx framework and MFCC algorithm for feature vectors. We achieved a comparable result with this work for the 'MMM' experiment where we achieved 0.406 WER compared to 0.460 WER in [40]. We could not compare other experiment results since, up to our knowledge, there is no work that mixes genders in the training and testing.

6. CONCLUSIONS

In this work, a Quranic recitations dataset by both genders (Male and Female) was created. The recitations by males were collected from the net. While the recitations by

females were manually collected by approaching female reciters locally and internationally. The dataset has been restructured, cleaned, transferred into a specific form and split into subsets with suitable and specific percentages to be used for training and testing any Arabic speech recognition system. Mozilla's DeepSpeech v0.7.0-alpha.1 was used as the main implementation model that is implemented with Python using TensorFlow. Four different experiments were conducted using the audio files by both genders. In two of the experiments, we trained using audio files by one gender and tested using audio files from the other gender. The results show that training the DeepSpeech model on data that belong to a certain gender and testing it using data that belong to the other gender results poor performance.

REFERENCES

[1] S. Gaikwad, B. Gawali, P. Yannawar, "A review on speech recognition technique," *International Journal of Computer Applications*, vol. 10, no. 3, pp. 16–24, 2010.

[2] W. Ghai, N. Singh, "Literature review on automatic speech recognition," *International Journal of Computer Applications*, vol. 41, no. 8, 2012.

[3] T. Shanthi, C. Lingam, "Review of feature extraction techniques in automatic speech recognition," *International Journal of Scientific Engineering and Technology*, vol. 2, no. 6, pp. 479–484, 2013.

[4] R. Dixit, N. Kaur, "Speech recognition using stochastic approach: a review," *International journal of innovative research in science, engineering and technology*, vol. 2, no. 2, pp. 356–361, 2013.

[5] M. Gamit, K. Dhameliya, N. Bhatt, "Classification techniques for speech recognition: a review," *International Journal of Emerging Technology and Advanced Engineering*, vol. 5, no. 2, pp. 58–63, 2015.

[6] K. Lee, H. Hon, R. Reddy, "An overview of the SPHINX speech recognition system," *IEEE Transactions of Acoustics*, vol. 38, no. 1, pp. 35–45, 1990.

[7] R. Djemili, M. Bedda, H. Bourouba, "Recognition of spoken arabic digits using neural predictive hidden markov models," *International Arab Journal of Information Technology*, vol. 1, no. 2, pp. 226–233, 2004.

[8] H. Bourlard, N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, vol. 247, Springer Science and Business Media, 2012.

[9] L. Burget, P. Schwarz, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, D. Povey, A. Rastrow, "Multilingual acoustic modeling for speech recognition based on subspace Gaussian mixture models," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4334–4337, 2010.

[10] D. Su, X. Wu, L. Xu, "GMM-HMM acoustic model training by a two level procedure with Gaussian components determined by automatic model selection," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4890–4893, 2010.

[11] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, A. Ng, "Deep speech: scaling up end-to-end speech recognition," *arXiv Preprint arXiv1412.5567*, 2014.

[12] B. Álvarez, P. Quirós, B. Fernández, "Semi-supervised learning for spanish speech recognition using deep neural networks," *APPIS*, pp. 19–29, 2018.

[13] H. Tabbal, W. El Falou, B. Monla, "Analysis and implementation of a" Quranic" verses delimitation system in audio files using speech recognition techniques," in *2006 2nd International Conference on Information and Communication Technologies*, vol. 2, pp. 2979–2984, 2006.

[14] A. Mohammed, M. Sunar, M. Salam, "Quranic verses verification using speech recognition techniques," *Jurnal Teknologi*, vol. 73, no. 2, 2015.

[15] B. Abro, A. Naqvi, A. Hussain, "Qur'an recognition for the purpose of memorisation using speech recognition technique," in *2012 15th International Multitopic Conference*, pp. 30–34, 2012.

[16] E. Khalaf, K. Daqrouq, A. Morfeq, "Arabic vowels recognition by modular arithmetic and wavelets using neural network," *Life Science Journal*, vol. 11, no. 3, pp. 33–41, 2014.

[17] E. Khalaf, K. Daqrouq, M. Sherif, "Wavelet packet and percent of energy distribution with neural networks based gender identification system," *Journal of applied Sciences*, vol. 11, no. 16, pp. 2940–2946, 2011.

[18] Z. Lei, L. Jiandong, L. Jing, Z. Guanghui, "A novel wavelet packet division multiplexing based on maximum likelihood algorithm and optimum pilot symbol assisted modulation for Rayleigh fading channels," *Circuits, Systems and Signal Processing*, vol. 24, no. 3, pp. 287–302, 2005.

[19] K. Kirchhoff, J. Bilmes, S. Das, N. Duta, M. Egan, G. Ji, F. He, J. Henderson, D. Liu, M. Noamany, P. Schone, "Novel approaches to Arabic speech recognition: report from the 2002 Johns-Hopkins summer workshop," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing Proceedings*, vol. 1, pp. I–I, 2003.

[20] E. Khalaf, K. Daqrouq, M. Sherif, "Modular arithmetic and wavelets for speaker verification," *Journal of Applied Sciences*, vol. 11, no. 15, pp. 2782–2790, 2011.

[21] M. El Amrani, M. Rahman, M. Wahiddin, A. Shah, "Building CMU Sphinx language model for the Holy Quran using simplified Arabic phonemes," *Egyptian Informatics Journal*, vol. 17, no. 3, pp. 305–314, 2016.

[22] A. Akkila, S. Abu-Naser, "Rules of Tajweed the Holy Quran intelligent tutoring system," *International Journal of Academic Pedagogical Research*, vol. 2, no. 3, pp. 7–20, 2018.

[23] N. Shafie, M. Adam, H. Abas, "The model of Al-Quran recitation evaluation to support in Da'wah technology media for self-learning of recitation using mobile apps," *In 3rd International Seminar on Da'wah, National University of Malaysia*, 2017.

[24] L. Rabiner, R. Schafer, *Introduction to Digital Speech Processing*, vol. 1, Now Publishers Inc., 2007.

[25] H. Osman, B. Mustafa, Y. Faisal, "QDAT:a data set for reciting the Quran," *International Journal on Islamic Applications in Computer Science And Technology*, vol. 9, no. 1, pp. 1-9, 2021. <https://www.kaggle.com/annealdahi/quran-recitation>

[26] R. Raja-Yusof, F. Grine, N. Ibrahim, M. Idris, Z. Razak, N. Rahman, "Automated tajweed checking rules engine for Quranic learning," *Multicultural Education and Technology Journal*, vol. 7, no. 4, pp. 275–287, 2013.

[27] H. AlKhatib, E. Mansor, Z. Alsamel, J. AlBarazi, "A study of using VR game in teaching Tajweed for teenagers," *in Interactivity and the Future of the Human-Computer Interface*, IGI Global, pp. 244–260, 2020.

[28] A. Abdelhamid, H. Alsayadi, I. Hegazy, Z. Fayed, "End-to-end Arabic speech recognition: a review," *In Proceedings of the 19th Conference of Language Engineering*, Alexandria, Egypt, pp. 26-30, 2020.

[29] A. Alqudah, M. Alshraideh, A. Sharieh, "Arabic disordered speech phonetic dictionary generator for automatic speech recognition," *Journal of Theoretical and Applied Information Technology*, vol. 98, no. 4, pp. 571–586, 2020.

[30] EveryAyah. < https://everyayah.com/) >

[31] K. Al-Juhani, "The difference between the Ottoman drawing and the spelling drawing that was customary," *Alukah Network*, 2013.

[32] GitHub, "DeepSpeech: DeepSpeech is an open source embedded (offline, on-device) speech-to-text engine which can run in real time on devices ranging from a Raspberry Pi 4 to high power GPU servers." <https://github.com/mozilla/DeepSpeech>

[33] S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] K. Heafield, "KenLM: Faster and smaller language model queries," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pp. 187–197, 2011.

[35] S. Davis, P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[36] S. Young, S. Young, "The HTK hidden Markov model toolkit: Design and philosophy," 1993.

[37] Wikipedia , "Recurrent neural network." <https://en.wikipedia.org/wiki/Recurrent_neural_network>

[38] K. Kawakami, *Supervised Sequence Labelling with Recurrent Neural Networks*, Doctoral Dissertation, Technical University of Munich, 2008.

[39] D. Kingma, J. Ba, "Adam: a method for stochastic optimization," *arXiv Prepr. arXiv1412.6980*, 2014.

[40] H. Hyassat, R. Zitar, "Arabic speech recognition using SPHINX engine," *International Journal of Speech Technology*, vol.9, no. 3, pp. 133–150, 2006.