International Journal of Speech Technology
https://doi.org/10.1007/s10772-022-09988-3

Building a neural speech recognizer for quranic recitations

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

Received: 1 May 2021 / Accepted: 22 June 2022

Abstract
This work is an effort towards building Neural Speech Recognizers system for Quranic recitations that can be effectively used by anyone regardless of their gender and age. Despite having a lot of recitations available online, most of them are recorded by professional male adult reciters, which means that an ASR system trained on such datasets would not work for female/child reciters. We address this gap by adopting a benchmark dataset of audio records of Quranic recitations that consists of recitations by both genders from different ages. Using this dataset, we build several speaker-independent NSR systems based on the DeepSpeech model and use word error rate (WER) for evaluating them. The goal is to show how an NSR system trained and tuned on a dataset of a certain gender would perform on a test set from the other gender. Unfortunately, the number of female recitations in our dataset is rather small while the number of male recitations is much larger. In the first set of experiments, we avoid the imbalance issue between the two genders and down-sample the male part to match the female part. For this small subset of our dataset, the results are interesting with 0.968 WER when the system is trained on male recitations and tested on female recitations. The same system gives 0.406 WER when tested on male recitations. On the other hand, training the system on female recitations and testing it on male recitation gives 0.966 WER while testing it on female recitations gives 0.608 WER.

Keywords Quran · Speech · ASR · DeepSpeech · WER · Dataset

1 Introduction

The Quran or Qur'an, in other spelling, is the most important holy book in the Muslim world. It is the words of ALLAH (the only GOD) that were revealed through the angel Gabriel, over a period of 23 years, to prophet Mohammad (peace be upon him). The Quran supersedes any other writing. The scribes of prophet Muhammad wrote down the words of the Quran as prophet Muhammad was never taught to write or read.

In the Quran, a riches Muslim finds guidance, encouragement, admonishment, kindness, promises of righteous mercy, and eternal happiness. At the same time, the Quran threats the bad with punishments and eternal torment. It consists of 114 different Surahs, where each Surah consists of a specific number of verses (Ayah). The Quran is also divided into 30 equal parts (called Juz's) in terms of the number of pages (Alhawarat et al., 2015).

The first five verses of the Quran that were revealed are: *"Recite in the name of your Lord who created, created man from a blood clot. Recite, for your Lord is most*

✉ Osama Al-Khaleel
oda@just.edu.jo

Suhad Al-Issa
suhad.al.essa@gmail.com

Mahmoud Al-Ayyoub
maalshbool@just.edu.jo

Nouh Elmitwally
Nouh.elmitwally@bcu.ac.uk; Nouh.sabri@fci-cu.edu.eg

1 Department of Computer Engineering, Jordan University of Science and Technology, Irbid 22110, Jordan

2 Department of Computer Science, Jordan University of Science and Technology, Irbid 22110, Jordan

3 School of Computing and Digital Technology, Birmingham City University, Birmingham B4 7XG, UK

4 Department of Computer Science, Faculty of Computers and Artificial Intelligence, Cairo University, Giza 12613, Egypt

Published online: 05 August 2022

magnanimous—who taught by the pen; taught man that which he did not know. (Qur'an 96:1–5)"

The Quran is written in the language of Classical Arabic. The number of words in the entire Quran is 77,794 words (with AlBasmalah). Therefore, the Quran is considered a source of specific vocabulary with a considerable number of Arabic words. These Arabic words comprise the 28 (29 with the hamza) letters of the Arabic alphabet.

In the Arabic language, diacritics marks are placed on the letters so that they are pronounced correctly because the Arabic language is characterized by the presence of identical words in letters but differ in meaning. The meaning is determined by the diacritics that are represented by the difference in pronunciation. The main diacritics marks include Damma, Fatha, Kasra, Tanween al-Dam, Tanween al-Fath, Tanween al-Kasr, Sukoon, and Shaddha. Hence, the Arabic text can be written in more than one form, depending on the number of symbols and diacritics marks that are used.

The Quran is read according to the rules of recitation and the provisions of intonation to be within the correct way according to the commands of Islam. Tajweed, which is considered an improvement on the recitation, articulates each letter and gives it the correct attributes (Tajweed, 2006). Tajweed is the science that concerns with applying the rules of pronunciation to the letters during the recitation of the Quran within specific provisions that include Alethhar, Idgham Iqlab, Ikhfa', pauses, and Madd. Wasting any of the intonation rules is considered a mistake that does not change the meaning.

Changing or deleting a word or letter or changing the diacritic of a vowel is a type of a mistake that probably changes the meaning. This error is considered one of the errors that are popular in the Arabic language. This error is considered more important than the mistake of intonation and must be corrected.

The recitation of the Quran is organized through different schools, and it is called Al-Qiraat. The number of these schools is ten, and the differences between them are related to changing some of the diacritics of a vowel or changing letters. The Holy Quran 10 Readings (Qiraat) also differ in Tajweed rules, which constitute the major variation reason.

Due to the importance of the Holy Quran in the life of Muslims and due to the heavy and thorough use and the interaction between Muslims and the Holy Quran in their daily life, building an ASR system for Quranic recitation would tremendously help people in the Muslim world. For example, Muslims would benefit from such a system in reciting the verses of the Quran correctly and in a professional way. The system might be of interest to a wide range of Muslims who would like to memories the Holly Quran verses and be able to recite them professionally. Such a system can be one of the means, to memorize the Holy Quran verses, for blind people, kids, and adults who can not read.

Most of the existing ASR systems for Quranic recitations are built to work for recitations by male professional reciters. There exist a lack of work that targets recitation by female reciters in general (professional and none professional) or recitation by male none professional reciters. Furthermore, the age of the reciters has not been taken into consideration in those proposed systems; especially when it comes to female reciters.

The objective is to build an ASR that addresses this gap by adopting benchmark datasets of audio records of Quranic recitations that consist of recitations by both genders from different ages. This work proposes a speaker-independent neuro speech recognizer (SR) that can be easily and efficiently used by any reciter of the Quran regardless of the gender or the age. As illustrated in Fig. 1, the system takes the Quranic recitation voice as input and transforms it into a readable text that matches the sound. The performance of the SR is measured using the word error rate (WER) and the character error rate (CER).

The rest of this paper is organized as follows: Sect. 2 provides a general background about the process of speech recognition, and its essential elements. Moreover, relevant works from literature and their main ideas are presented. Section 3 describes the language model including the decoding, training, and the evaluation. A training method is applied to the independent systems for speakers to increase effectiveness. The experimental results and the discussion are provided in Sect. 4. Finally, the conclusions are given in Sect. 5.

2 Background and literature review

Speech is recognized by treating it as a signal that has specific characteristics and properties. In theory, speech is represented as an analog wave that changes over time. Digitally, this continuous-time signal is stored as a series of separate samples (discrete). Each sample is a discrete number that represents the amplitude information corresponding to the sound wave during a sampling period (Rabiner & Juang, 1993).

The time interval for sampling is the distance between successive samples with time. Thus, each digital speech has a fundamental characteristic analogous to the sampling

Fig. 1 Quranic ASR system

period's reciprocal, which is called the Sampling Rate. Its measurement unit is Hertz, where Hertz is defined as the number of samples per second. To reproduce the sound accurately and efficiently, the sampling frequency of recording must be carefully chosen. The most common sampling rates for digital audio recording are 8K, 16K, 22.05K, 44.1K, 48K, 96K, and 192K samples per second. The sampling rate of 44.1 KHz is the standard among all.

The audio recording might also contain background sounds and noise that are captured during the recording process. These are present as other information in the digital representation of the audio signal and have negative impact on the sound quality.

Bit Depth specifies the number of amplitude values that can be specified for each sample. The most common bit depths are 16 bits, 24 bits, and 32 bits. The higher the bit depth, the better the accuracy and the quality (Rabiner & Juang, 1993).

In addition, there may be more than one channel contained in the audio signal. This information (known as Channel Count) represents the position of the source of the sound in the audio signal. In fact, each channel includes a sample representing the amplitude of the sound produced from the source at a specific moment in time. For example, there are two channels in the stereophonic (stereo) sound, as there are two sources for the sound (left and right speakers). On the other hand, there is only one channel in the monophonic (mono) as there is one source for the sound (Rabiner & Juang, 1993).

There are two domains for representing the sound signal: the time domain and the frequency domain. The time domain deals with the variation of the amplitude of the signal with time. While the frequency domain is the representation of the time domain signal using the Fourier theory (Rabiner & Juang, 1993).

In linguistics science, speech is treated as a sequence of a group of spoken words. Each spoken word is a group of small phonemic units called Phones. There is tremendous diversity in the formation of speech sounds for a single word, as phones associated with words are distinguished into Phonemes where a phoneme replaces another. This leads to the formation of a different word. For example, placing an "s" instead of "b" in 'bad: /b...d/' word leads to another word 'sad:/s...d/' with a different meaning. These phonemes were written using the internationally recognized alphabet letters in the English language. Every language has its own alphabet that has its phonemes as well (Rabiner & Juang, 1993).

There are different types of the SR systems depending on the target to be recognized and identified. The factors that play a role in influencing any SR are Vocabulary Size, Speaker Dependency, Type of Speech, and Grammar (Lee et al., 1990). A SR system usually consists of signal processing, acoustic model, pronunciation model, language model, and decoding (Huang & Deng, 2010).

The processing of the signal might involve filtering, normalization, mean subtraction, framing, and windowing (Ibrahim et al., 2017). The associated information is extracted, from the input speech, and is used to distinguish between pronouncement units of the speech. Irrelevant information such as noise, disruption, and channel distortion are removed to increase the effectiveness.

The Acoustic model function is to make a mapping between the features extracted from the speech signal and the linguistic units. Statistical models like neural networks or hidden Markov models are used in most SR systems to represent phonemes and words, which is necessary for the recognition process. The pronunciation model matches between words and the acoustic models and assigns each word to the corresponding pronunciation in the recognition system. It represents the word to be pronounced as a sequence of phonemes. This phoneme group is mapped from existing speech sounds to train the phonemic model. Moreover, this is done using the pronunciation dictionary. The pronunciation model is usually an optional component, especially when using the end-to-end ASR system.

The language model supplies the system's grammatical rules of the language that are used by humans. These language models serve to determine the probabilities of the word sequence. There are two main models for the language model: the stochastic and the grammar rules. In the stochastic model, the probability is estimated based on an appropriately sized text set using n-grams. It is possible to use several forms of the linguistic model such as unigrams (one word), bigrams (2 g), triples (3 g), and can be any other arrangement of n-grams. In the 2-g language model, the current word's probability is based on the previous word only. Also, in the 3-g model, it is a string of three words. The language model can use context to resolve ambiguity and mixing between identical speech sounds. The language models are based on grammar rules, through the confirmed grammatical rules, by defining the sequence of the defined words clearly and with high certainty. Hence, there is one possibility for each proper sequence.

The decoding function is to search for and find the most probable word sequence by monitoring the acoustic input. The search problem is solved using stack decoding (Wang & Waibel, 1997) or through a graph search algorithm (such as Viterbi search) (Lou, 1995). When the vocabulary size is large, the search process is expensive. Thus, it is possible to reduce the search area using a restricted language model in the decoding process.

To evaluate the functioning and performance of the SR system, the accuracy is measured by comparing the transcriptions version that is identified through the SR system and the reference transcriptions copy. One of the

most popular evaluation criteria used is the word error rate (WER). It is a derivation from Levenshtein distance at the word level, and it is calculated by taking insertions, substitutions, and deletions in the evaluation. The character error rate (CER) is also used in some systems to evaluate the character level error rate rather than the word level. It is also possible that, in some systems, the complement of the WER is used instead of the WER as a tool for the performance measurement. The complement of the WER is called word recognition rate (WRR) (Radha, 2012). For example, in isolated SR systems, the WRR measuring tool is used to measure the amount of accuracy.

Most traditional SR systems use the hidden Markov model (HMM) (Juang & Rabiner, 1991) for the phonemic model in most published literature. HMM is a statistical model that uses the Markov process, which includes a fixed number of hidden states, a probability matrix of transition between potential states. Each of the possible states has a probability distribution over the potential output values. With the training data transcript, these transition probabilities and monitoring of the HMM training can be calculated using the Baum-Welch algorithm. By observing the model parameters, the hidden states can then be predicted using the Viterbi algorithm (Lou, 1995) or through the use of other dynamic programming methods at the time of inference. The HMM was used on SR and time-series predictions since its first appearance in the mid-1970s. As for speech-to-text applications, HMMs are used to place (approximate) the probability of a word sequence while providing a specific and particular sequence of feature vectors for the observed speech.

In the field of Quran recitation analysis, the work in (Tabbal et al., 2006) used SR techniques through the open-source Sphinx framework to introduce a SR system for the Arabic language, which was then expanded to deal with the recitation of the Quran. In fact, much speech research on the Arabic language recognition have been done based on the Quran using the Sphinx framework from Carnegie Mellon University with the hidden Markov model toolkit (HTK) (Abushariah et al., 2017).

Sphinx-4 is the latest version of CMU Sphinx and is supported by Java. Through its toolkit, it is possible to train a continuous SR system independently of the speaker based on HMM with Gaussian mixtures model (GMM) for output possibilities (Lamere et al., 2003). The HTK supports many templates, written in the C language, that are free for non-commercial use (Young, 1994).

The system in (Tabbal et al., 2006) recognized the Quran through the use of the Sphinx tool. The word recognition rate reached 90% with 20 males reciters in the data that is based on Surat Al-Ikhlas only. On the other hand,they achieved 85% recognition rate with 20 females reciters in the data that is also based on Surat Al-Ikhlas. They measured the performance of their system by the number of words in the text that are correctly recognized.

The work in (Hyassat & Abu Zitar, 2006) developed a system to recognize and teach the Quran through the Sphinx tool. The word error recognition was 46%. One of the most important components of using the Sphinx framework is the pronunciation dictionary for building the traditional SR system. Many different languages have dictionaries and are available online for speech recognition. For example, CMU English (American) provides a hand-built dictionary supported in the Sphinx tool. Unfortunately, for the Arabic language, there is no available dictionary. This is one of the limitations of using the Sphinx in SR for the Arabic language. To address this, the work in (Hyassat & Abu Zitar, 2006) built several tools to build the Arabic language dictionary and the Holy Quran dictionary automatically from a group of manually derived grammar. The tool ultimately converts the formed text into sequential sounds made of 44 easy-to-read Arabic symbols. They created their own set of the Holy Quran (called HQC-1) that includes 18.5 hours of short speech and 25,740 unique words. Then they worked on extracting feature vectors from the audio recordings. Furthermore, through this developed dictionary and their data, they were able to build an SR system through CMU SPHINX-4. They could achieve an accuracy of 46.182% in terms of the WER scale. They also worked on training two Arabic language models on numerical data by using five-state HMMs with 8 Gaussian mixtures to give better results.

For the HMM-GMM to be trained, the training data must be identical to the corresponding phonograms. The transcriptions (words) on which the phonemic model is working must be temporally synchronized with the speech segment. Manual alignment of training data with experts is the best method for accurate synchronization, but it is very time-consuming. Alternatively, transcription is automatically synchronized with time by using forced Viterbi alignment in conventional models for more efficient training.

Some alternative automatic time synchronization tools are presented in the literature. However, the efficiency of the SR audio model is affected by the quality of the alignment. Thus, data segmentation and alignment are some of the traditional SR models (Wang et al., 2019).

For the Arabic SR systems, which are based on the traditional method, a detailed and comprehensive review is found in (Al-Anzi & AbuZeina, 2018).

In memorizing the Quran without human intervention, the work in (Abro et al., 2012) presented an automatic Quran recognizer and identified the main differences between a simple SR system and a language teacher using SR. The SR was implemented on speaker-dependent by including pre-processing speech signals, feature extraction, and pattern matching. Once the features are extracted, they are used for acoustic model and classification.

The work in (Khalaf et al., 2014) proposed a new way to extract the appropriate features for Arabic recognition, where the wavelet packet transformation (WPT) (Khalaf et al., 2011b; Lei et al., 2005; Kirchhoff et al., 2003) was studied with standard modular arithmetic and neural network to identify Arabic vowels. They gave 266 Probabilistic Neural Network (PNN) coefficients for classification. Their results showed that the proposed modular wavelet packet and neural networks (MWNN) (Khalaf et al., 2011a) system achieved the best recognition rate.

The authors of (Mohammed et al., 2015) examined the challenges and the solutions for building a successful system for verifying Quran verses on the Internet. It examined the techniques used to deal with finite vocabulary and how modeling can avoid some complexities in the language and dictionary model's phonetic domain. The work proposed a system to identify the errors in the recitation of the Quran and show where errors have occurred exactly. They used feature extraction, HMMs for speech recognition, and the phonetic search engine (PSE) technique for searching through a Quran database.

In Quranic verse recitation recognition with Tajweed rule's function, the work in (Ibrahim et al., 2013) proposed an automated Tajweed verification engine dedicated to the Quran learn. It was carried out and tested towards j-QAF students in elementary school in Malaysia. They achieved a 91.95% recognition at the Ayah level. The engine was only tested on Surat Al-Fatihah.

The use of a simplified set of Arabic phonetics in the Arabic SR system applied to the Holy Quran was examined in (El Amrani et al., 2016). CMU Sphinx 4 was used to train and evaluate the language model of the Hafs novel about the Quran. The language model was built using a simplified list of Arabic phonetics to simplify creating the language model. They were able to achieve a 1.5% WER using a very small set of audio files during the training phase when using all the audio data for the training and testing phases. Moreover, they achieved a 50.0% WER when using a 90% of their audio files for training. Their dataset includes Surat Al- Fatihah, Al-Ikhlas, Al-Falaq, and Al-Nass for 22 different famous reciters.

The work in (Shafie et al., 2017) aimed to develop a technological application model to evaluate the recitation of the Quran. Scientific methods are applied in the analysis of correct recitation based on the appropriate rules. It examined the practice of SR to detect the error of recitation. It addressed the difficult issues of character representation and classification based on digital speech processing (DSP) techniques (Rabiner & Schafer, 2007), which automatically identified, categorized, and recognized the Quran recitation speech for the representation function.

An intelligent tutoring system (ITS) was developed by (Akkila & Abu-Naser, 2018). A computer program provided direct training or response to students without a human teacher's intervention. The aim was to facilitate the learning process through extensive facilities of the computer. The proposed system was implemented using the Intelligent Tutoring System Builder (ITSB) authoring tool, which provides an intelligent educational system for teaching Quran reading and "Tajweed" with the Hafs novel. Teachers and students in recitation school evaluated the system, and the result of the evaluation was promising.

Recently, researchers have built and developed an ASR system based on deep learning (DL) (Hannun et al., 2014; Amodei et al., 2015; Battenberg et al., 2017; Chan et al., 2016; Dahl et al., 2012). In 2011 the CLDNN-HMM model (Dahl et al., 2012) was presented by Microsoft Research Institute researchers. The CLDNN-HMM model consists of a DNN and an HMM in which the DNN outputs are used in place of the GMM in the HMM-GMM, where these outputs are used to estimate the subsequent output probabilities required for the hidden HMM cases. This model was introduced to achieve a significant improvement in the SR system by solving the problem of recognizing large vocabulary in speech greater than that achieved by the HMM-GMM model.

The authors of (Santosh et al., 2010), provided a quick review of different ASR models, where they discussed the advantages and disadvantages of the different models and compared them to the traditional model based on HMM. They found in their experiments that the accuracy of recognition achieved by DL technology is much better than that achieved by HMM. Then they presented the comprehensive models and their ability to be an alternative to the HMM-DNN model, as it is characterized by its simplicity and ability to solve the problem of data segmentation and its appropriate time synchronization. According to their study, the most common models for SR, ranked from the most to the least accurate, are attention-based sequence-to-sequence, Recurrent Neural Network (RNN)-Transducer, and Connectionist temporal classification (CTC). Although the CTC model is the least accurate among all models, it is the best for the decoding stage and training time. To complete the SR process, a system that based on CTC and HMM uses LM built from a large group of words in a particular language. In the end-to-end systems, the outputs labels are sequential and dependent on each other. Therefore, the learning of the LM is implicit through the data being trained. Also, this model solves the alignment problem in SR by calculating the CTC loss.

The researchers in (Hannun et al., 2014) developed a DeepSpeech CTC model that transforms speech into text by a DL method. They achieved better results, comparing previous work, with the presence of noise. The model uses (DNNs) with computing resources and a large amount of data to use only one phase rather than using the

sophisticated multiple stages pipelines in previous works. Also, there is no phoneme dictionary in this model for converting training data into phonemes. It can directly convert phonemes into words. It predicts transcription using voice input, where the language model and the CTC decoder are used for word-level transcription production. The DeepSpeech engine components consist of five layers. The researchers trained several models by reading different datasets or a conversation to compare the performance with the performance of the DeepSpeech model. The best result of the DeepSpeech model was achieved by training with the general Fisher and Switchboard data, which are about 2,300 hours. It was tested using CallHome data and Switchboard Hub5'00 data. The WER ratio was 12.6% and 19.3%, respectively. This model was able to outperform all previous SR models for the English language.

Version 2 of DeepSpeech was developed in (Amodei et al., 2015), which is very similar to the first version DeepSpeech1. Both versions are RNN and are supported by CTC. The authors did the same training method, except that version 2 used SortaGrad, through which training data is supplied with increasing lengths of speech duration. DeepSpeech1 and DeepSpeech2 have gained widespread use in SR, but the researchers published DeepSpeech2, relying on PaddlePaddle with its implementation. To facilitate the process of using it, a group of technologies appeared in multiple fields in DL. Several researchers tested the DeepSpeech model to build an SR system for Russian and German languages (Agarwal & Zesch, 2019; Panaite et al., 2019; Iakushkin et al., 2018).

Researchers continued to develop end-to-end speech recognition, and in 2017 DeepSpeech researchers announced and improved version, DeepSpeech3 (Battenberg et al., 2017), using the loss of RNN-Transducer to replace CTC. The RNN transducer is a loss function that supports a neural decoder, eliminating the decoding with a language model during inference time.

In 2016, Google introduced an ASR model based on attention LAS (listening, attending, and spelling) (Chan et al., 2016). The system consists of three components that form an end-to-end system and is trained through sounds and the corresponding text. These components are the encoder that listens and converts the sound into a higher representation. The encoder consists of bidirectional long short-term memory (LSTM) recurrent layers hierarchically. The outputs of each layer outputs are fed into the consecutive LSTM layer with a time resolution decrease of 2, followed by an attention component that explicitly aligns the character output with the input features. Then comes the decoder component that defines the probability distribution over the character sequence. To assess performance of the model, it was trained using the Google Voice Search data that are about 2000 hours and tested using data of about 16 h. The WER was differentiated by 2.3% from Google's model.

Recently Facebook AI Research introduced an ASR system called Wav2Letter (Collobert et al., 2016). In this open-source model, the goal was to achieve an improvement over the RNN model used in DL in terms of the amount of data needed for training and the computing power. The Wav2Letter model is based entirely on convolutional neural networks (CNNs). Later on, Wav2Letter++ was introduced in (Pratap et al., 2018).

Most of the existing SR research that is based on DL are limited to the English language and some other languages to some small extent. However, few existing studies that use DL are concerned with the Arabic language and the Quran.

A recent study that provides a review in the field of DL for ASR for the Arabic language in presented in (Algihab et al., 2019). This comprehensive study presents 17 different studies in the field of ASR from the isolated word, continuous word, to automatic speech. It also presents the tools and models used to employ DL Arabic language SR system.

The study in (Al-Ayyoub et al., 2018) help people to correctly read the Quran according to the rules and provisions. The authors used eight rules for reciting the Quran. The model was an extended version of the Deep Belief Network (called the DBN convolutions). They used the supporting vector machine to train their data in the Weka tool as a classifier. The model achieved an accuracy of 97.7% when tested using unseen records.

The work in (Alkhateeb, 2020) aimed at recognizing the reciter of Quran. A dataset that includes a reading often for the imams of the holy mosques in Mecca and Medina for Surahs 18 and 36 only was collected. After extraction, the features are mapped to the Artificial Neural Networks (ANN) and K Nearest Neighbor (KNN) classifiers for training. When tested using unseen records, the model achieved an accuracy of 97.6% for Surah 18 and 96.7% for Surah 36 through using the ANN classifier. In comparison, the accuracy was 97.03% for Surah 18 and 96.08% for Surah 36 by using the KNN classifier.

Among the recent research that was conducted in the field of the Tajweed of the Quran is (AlKhatib et al., 2020). The work aimed to teach adolescents to recite the Quran correctly and to familiarize them with the provisions of reciting the Quran through an entertaining method that adds fun to them by using virtual reality game using the HTC Vive device. The work encourages adolescents and increase their level of enthusiasm through the most technological means and techniques that adolescents are exposed to. The game was tested and evaluated by 20 teenage participants with positive results.

The work in (Abdelhamid et al., 2020) presented a comprehensive review of the latest technologies to recognize Arabic speech and guide researchers interested in working

on the Arabic language. This review focuses on machine learning and DL techniques in building ASR systems using the hybrid HMM-Deep Neural Network (DNN) models, the CNN model, the RNN model, and the end-to-end DL models. The review also presents and focuses on the end-to-end model on Arabic speech and the Arabic language, as the end-to-end model is essential and vital in speech recognition. The review also presents the latest services and toolkits currently available and necessary for building comprehensive models for speech recognition.

Recently, a Quranic dataset called QDAT was publicly published on the Kaggle site (Mustafa). QDAT includes more than (1500) Quranic audio records files in Arabic speech with Tajweed. Audio files cover recitation from both genders (Male and Female), where 165 readers participated.

The work of (Bettayeb, 2020) includes developing a text-to-speech (TTS) system where speech is a recitation from the Quran. It aims to help readers and facilitate reading the Quran. They used the unit selection method to improve speech quality in their work. The work consists of two steps. The first integrates the expert system unit (ES) by using the Arabic phonetic features and the Quran's language. The second step reduces the cost of concatenation cost function through the final selection of modules. They achieved correct recitation of the Holy Quran with Al-Tajweed reading rules with a percentage of 97%.

The DeepSpeech framework was used by (Eldeeb) to develop an ASR called 'DeepSpeech-Quran' on the Quran recitations to help reading the Quran. The work builds an ASR model from a set of professorial reciters called 'Imam-Recitations'. Then an ASR model from a set of professorial reciters and semi-professionals called 'Imam-Tusers-Recitations was build.

3 Methodology

3.1 Datasets

Two Quranic recitations datasets for reciters from both genders (Male and Female) are adopted in this work. They are huge and rich datasets of audio recordings for Quranic recitations. The first dataset is a male's recitations dataset, where those males are professional and adult reciters of Arab Muslim and non-Arab Muslim origin from different countries. The records by professional reciters covering the entire Holy Quran.

The second dataset is a female recitations dataset. Those females are adult and non-adult reciters of Arab Muslim origin from different countries. The records by female's memorizer cover one Surah or more. The audio recordings are WAV files. Each record contains only one verse from the Quran "one Ayah" and for each record, there is the corresponding Quranic text of each verse.

The Hafs narration for audio recordings and the Uthmani style for the Quran text was adopted at the beginning to ensure obtaining an error free copy of the text. Then we converted the Uthmani style to the Orthographic style to be the text adopted in our datasets. The records by males have high quality, the length of the records is much larger, and it is more diverse. The female records are of less quality, the length of the records is short because it is one Ayah or part of Ayah, and less diversities exist. Table 1 provides a summary of the adopted datasets.

In this work, the widely used and robust DeepSpeech model is adopted as a DL-based system for speech recognizer on Quranic recitation. DeepSpeech is an end-end speech recognition system that performs a deep learning technique developed in (Davis & Mermelstein, 1980). It provides simple design for the architecture comparing to the traditional systems which give poor performance in noisy environments. At the same time, DeepSpeech does not need manual tools to model background noise and there is no need for an audio dictionary.

DeepSpeech is a multi-layer character-level of Recurrent Neural Network (RNN). This network's input is a raw audio spectrogram, and the output is a sequence of characters of the corresponding text transcription, where the spectrogram is a time-frequency representation.

The RNN model consists of seven layers, the input layer, five hidden layers, and the standard SoftMax output layers. The first three layers of the five hidden units are non-recurrent, the fourth layer is a bidirectional vanilla recurrent layer, and the fifth layer is non-recurrent. The fourth layer contains two RNN groups: a forward recurrence group and a group with backward recurrence (Schuster & Paliwal, 1997). A clipped ReLU activation function follows each hidden layer, as shown in Fig. 2.

The input sound was divided into staggered windows of equal sizes in the feed-forward phase. The extracted feature vector is sequentially fed into the first fully connected (FC) layers for each time-slice window. Their RNN encoded memory and output are the inputs of each of the recurrent units. Since the backward recurrent unit requires the next

Table 1 The dataset information

Dataset	Males	Females
Num. of Audio Records	257,705	5744
Num. of Reciters	42	21
Num. of Hours	1147.65	14.36
MIN Time (s)	0.0451389	1:01
MAX Time (s)	459.46	273.2
Average Time (s)	16.03	9.25

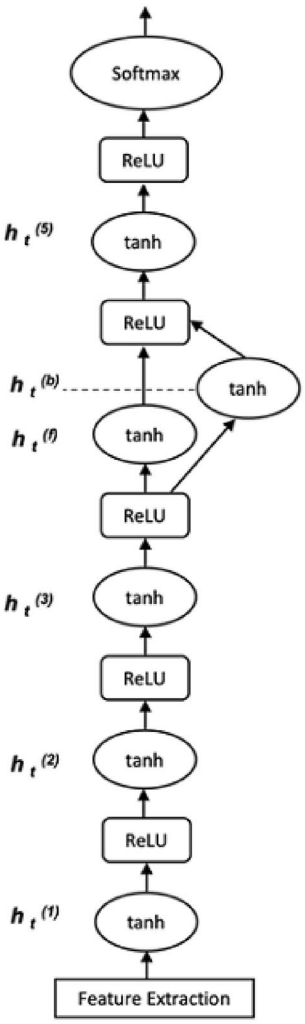

Fig. 2 The RNN's layers

step's output to begin its calculation, the network needs to wait until the end of a full voice is reached.

The output is created by RNN, beginning with the last phase in the time phase and going back in time. Next, the two output recurrences are summed up by the last fully connected layer, and the flow to the final output layer continues. The final output at each time stage is a vector of the character's transcription probabilities, and one probability for each character in the alphabet predicts the probability of a voice corresponding to that character timestamp.

To calculate the CTC loss, which is essential for training and evaluation, probabilities are used. In Fig. 3, the bidirectional RNN structure is shown, which we drew by imitating the original Figure of DeepSpeech (Hannun et al., 2014). The CTC loss is determined by aggregating the likelihood of all possible alignments of its target transcription when the output matrix of an input voice is set. Similarly, to find the optimum decoding alignment for inference, it uses the output likelihood matrix for beam scoring in the beam search algorithm. The final output transcription is then obtained by removing all blanks and replicated labels from the found path. A character-level language model will combine with the beam search decoding algorithm (Hannun et al., 2014).

3.2 Mozilla's DeepSpeech implementation

Mozilla introduced DeepSpeech as an open-source project that seeks to provide the public with business-quality SR services. Centered on Baidu's DeepSpeech, their free speech-to-text engine is constructed and built through contributions from a large group of developers and researchers. A public Fisher, SwitchBoard, and Librispeech datasets are trained by Mozilla to provide an American English pre-trained model in their 0.5.1 release in 2019. They obtained a common word error rate of 8.22% on the LibriSpeech clean test dataset, which means that the accuracy rate was high. DeepSpeech from Mozilla is a TensorFlow-based implementation, and because the TensorFlow framework is characterized by power and effectiveness, we chose it as the primary source code in our search (Mozilla). We used 0.7.0-alpha.1 release in our search. The Mozilla engine varies from the original DeepSpeech in many aspects; DeepSpeech (Hannun et al., 2014) different from Mozilla DeepSpeech, where in version 0.5.1 (Mozilla), they substitute bidirectional recurrent cells with unidirectional Long Short-Term Memory (LSTM) cells. LSTM is an evolutionary form of RNN that incorporates the cell states principle. Long-term dependencies can be learned, and information can be related from past to present by cell states (Hochreiter & Schmidhuber, 1997), as shown in Fig. 4.

The feeding input differs in a unidirectional LSTM, as shown in Fig. 5, which we drew by imitating the original Figure of DeepSpeech (Hannun et al., 2014). In the changed architecture, each time step relies only on the output of the first FC layers at that time, besides the LSTM state of the previous time step. Furthermore, Mozilla uses a more famous optimization algorithm called Adam (adaptive moment estimation), compared to the original DeepSpeech. The Adam optimizer in the training process requires less fine-tuning of hyper-parameters than the Nesterov process. Moreover, Mozilla uses MFCCs for feature extraction instead of the spectrogram feature. In the Mozilla GitHub repository, the hyper-parameters that were used are present.

3.3 Mel-frequency cepstral coefficients (MFCC)

A key component of SR is feature extraction. MFCC features are used extensively in this field. The MFCC was built on the human peripheral auditory system (Davis & Mermelstein, 1980). MFC representation is the product of a cosine transformation on a Mel-frequency scale of the

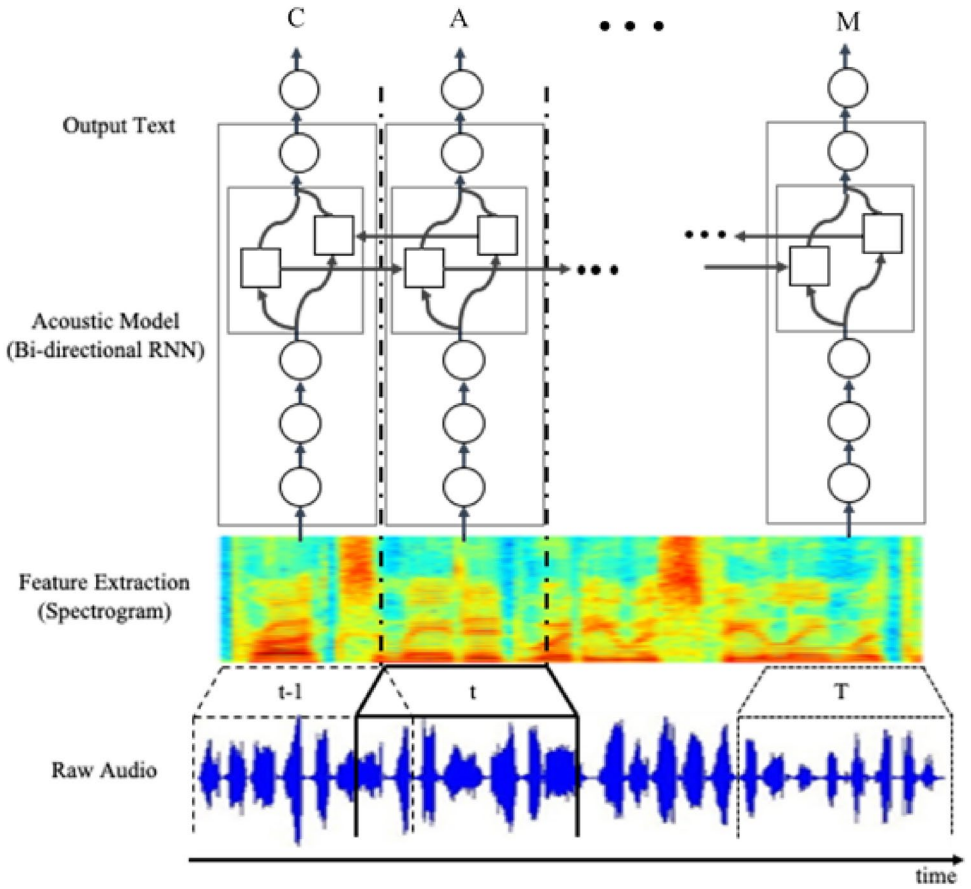

Fig. 3 The bidirectional RNN structure over the time

short-term power spectrum's actual logarithm. To clarify, for low frequencies (up to 1 kHz), the Mel scale is roughly linear, and for higher frequencies, it is logarithmic. The extraction process of the MFCC feature from audio data includes the following key components:

(1) Framing blocking and windowing.
(2) Discrete Fourier Transform (DFT) spectrum.
(3) Mel-frequency warping.
(4) Logarithmic operation.
(5) Discrete Cosine Transform (DCT).

The coefficients producing from the DCT are the MFCCs. Most of the information is maintained by the first few coefficients, so it is used to represent an audio frame. For example, the HTK toolkit uses 13 MFCCs by default, including the zeroth coefficient that reflects the average energy of the spectrum (Young, 1994). Higher-order coefficients represent rising spectral information levels. The MFCC feature vector provides a smooth version of the spectrum of log energy. Therefore, the speech signal is transformed into a low- dimensional and compact representation (Davis & Mermelstein, 1980).

3.4 Model training phase

The way a neural network learns, the amount of data available, and the quality of the underlying truth labels significantly affect the recognition errors. Although DeepSpeech can be trained in many approaches like (supervised learning, unsupervised learning, semi-supervised learning). In our case, since supervised learning is best suited because of the availability of a set of truth labels to train the algorithm with, we will suffice with this approach as our primary model.

SR is regarded as a classification activity; a supervised training approach is useful for classification problems and best suited to train a neural network of DeepSpeech. Where the data in training is fully labeled, whereas the labeled input and the expected output are provided, and the model is learned in the training phase to detect the relationship and in the result gets the mapping the wanted output to its input (Riesen & Bunke, 2010). When our network initialization is complete, the learning phase starts. The initial weights are selected randomly in a supervised approach. In neural networks, the strength of connections between units in adjacent layers is represented by weights. The initialization of weights is a significant step that can impact the learning

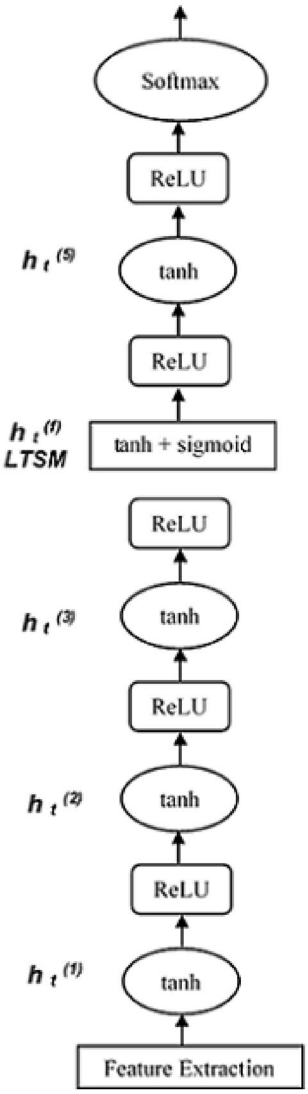

Fig. 4 Mozilla's layers

process and, ultimately, the model's convergence (Hayou et al., 2019).

There are several methods for initialization. Mozilla's DeepSpeech implementation is TensorFlow-based, which is used in our approach; after it uses the TensorFlow's default weight initializer, the process of training starts. Mozilla's DeepSpeech model is equipped to minimize the CTC loss function by supplying it with labeled training data and optimizing the network's weights and biases parameters, where the optimization is performed repeatedly for a specific number of epochs. In the actual training process, the labeled train for our datasets in male and female which is (1147.65) hours and (14.36) hours, respectively is passed on to the model in batches of a specific size in each epoch.

The way the training data is passed into the model is in an arranged order by audio file time length (in seconds), which is a reflection of the audio file size (in bytes) (Hannun et al., 2014). After the non-recurrent layers, dropout regularization is performed through training to reduce the overfitting and improve generalization. There are two stages of DeepSpeech: an acoustic model and a decoder that follows it. For each time, the softmax final layer of RNN acoustic model outputs slices each character's probabilities into the list of phonemes and the blank probability. The function of the decoder is to turn the probabilities into textual transcripts.

3.5 Language model building phase

The authors in (Hannun et al., 2014) used a graphemic n-gram language model (LM) to decode their model results and showed that it improved their model's efficiency. To create the n-grams language model, the KenLM tool was used (Heafield, 2011). So, to perform this phase, we downloaded the KenLM source code's stable release and then compiled it using a CMake process. Where this, in turn, adds the KenLM built binaries like lmplz, build_binary, and filter. To create a language model, we wrote a shell script provided with raw text files. The steps involved in the creating of a language model are as follows:

(1) Creating a plain text corpus, where one sentence per line is the same as the text in the description column of CSV files, we used in the training process.
(2) Creating a LM of a particular order by creating an ARPA (Advanced Research Projects Agency) file using the program lmplz.
(3) Creating a trie binary using a build_binary program to quantize the ARPA file because the loading time is faster with a binary file.
(4) Generate tries from the language model's vocabulary provided the binary file and phoneme list using Mozilla's generate_trie program.

Each time a language model is created, the list of phonemes provided to the script must be in the same order. Also, a phonemes list must be in the same order used in training an acoustic mode to synchronize with the probabilities of generating by the softmax layer. The text corpus used in creating a language model has a significant effect on its decoding effectiveness and, therefore, the accuracy of model recognition. We decided to use two phonemic language models covering the entire Holy Quran text to analyze and compare the results.

We created two language models based on the entire Quran text, where we created transcribing of each Ayah found in the Quran text to build our phonemic corpus where there is one complete Ayah per line. We created the first language model based on the entire cleaned Quran text. We called it cleaned because we removed diacritics marks, stop

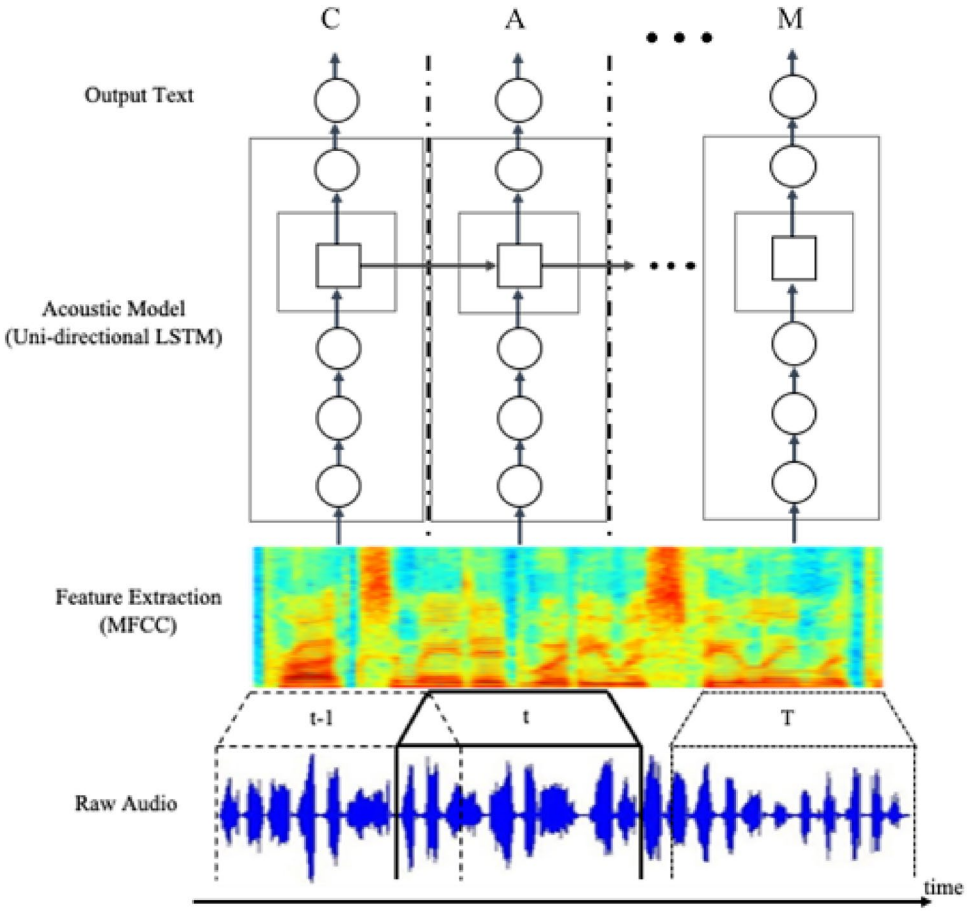

Fig. 5 Mozilla's DeepSpeech model structure over the time

signs, control and adjustment marks, and other accent marks. It is entirely free of any marks; it includes only the Arabic alphabet characters. We created the second language model based on the entire diacriticked Quran text. We called it diacriticked because it includes only diacritics marks. It is also entirely free of stop signs, control, adjustment marks, and other accent marks. The flow chart of creating two versions of language models are depicted in Figs. 6 and 7.

3.6 Metrics for evaluating performance phase

Our SR task evaluation is carried out on a dedicated, independent set of audio files that have never been applied in the training phase. These unseen audio files are used because, in the training phase, the device usually works much better on data that has already been used. This problem, called over-fitting, is avoided. For evaluation, the male test set and female test set, which includes unseen reciters, are used.

Evaluating an SR model's performance and effectiveness is performed through appropriate evaluation metrics that must be specified. In the development and optimization process, these measures also are applied. When the change is in the test set or model, the evaluation phase is carried out, and this change is at any time. We base our evaluation phase on the Word Error Rate (WER) metric because it is the most relevant and most applied in the literature for continuous SR systems. WER is derived from Levenshtein distance.

The minimum modifications enter a ground-truth for a decoded transcription calculate. WER compares the sequence of known words with words spoken; this means that WER indicates the number of words in the speech that did not read correctly. A standard definition of WER given in Eq. 1 on the next page, combining the three forms of editing (insertion, deletion, and substitution) over the ground-truth string length (Radha, 2012). So, we can get the number of correctly recognized words is defined in Eq. 2. The higher the WER, the more differencing between the two transcriptions.

In contrast, the minimum the word error rate, the higher the efficiency of recognition. Since the insertion number words are used in the equation's expression, this means that the WER will reach more than 100%. Also, we used the Character Error Rate (CER) metric, which is usually used, and there is a large correlation between the CER and the WER. The CER is applied to the character level. CER calculates the error rate on the recognized character in SR (Radha, 2012). A standard definition of CER is defined in a similar way to WER as in Eq. 3. It is worth noting that

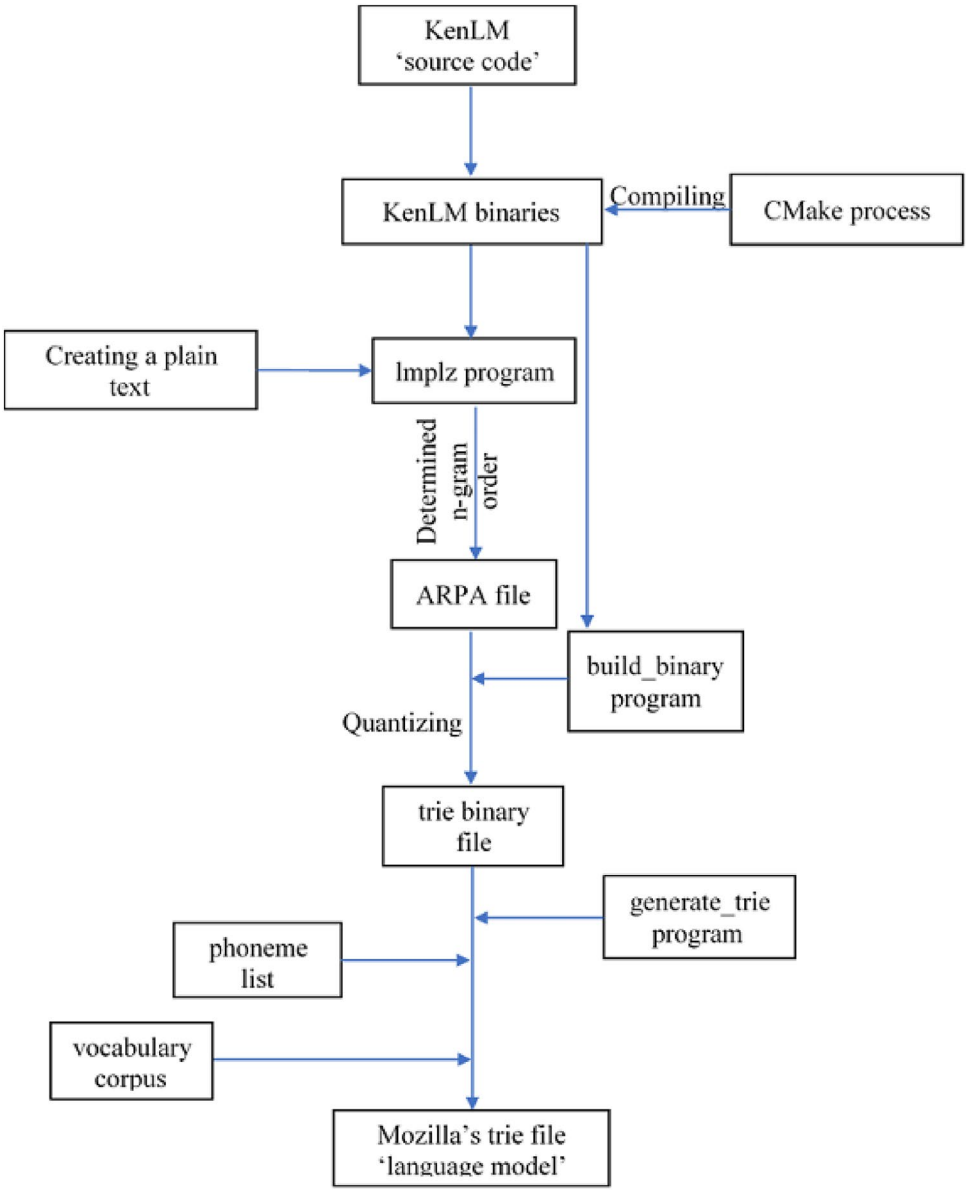

Fig. 6 Language model based on the entire cleaned Quran text

word error rates are typically higher than error rates for characters. CER compares the sequence of known characters with characters that are being spoken; this means that CER indicates the number of characters in the speech that did not read correctly, including letters, punctuations, diacritics, and spaces. For example, a CER of (5%) means that every fifth character was not correctly recognized.

Hence WER and CER reflect the accuracy. For example: when a WER of (10%), therefore, the rate of accuracy will be (90%). The word error rate can be high, even with a good CER. The WER reveals how good is the accurate recognition of the words in the speech.

$$WER = \frac{I_W + D_W + S_W}{N_W} 100\% \tag{1}$$

where N_W = The total number of words in the reference, I_W = The number of words inserted, D_W = The number of words deleted, S_W = The number of words substituted.

$$C_W = N_W - D_W - S_W \tag{2}$$

where C_W = The number of corrected words.

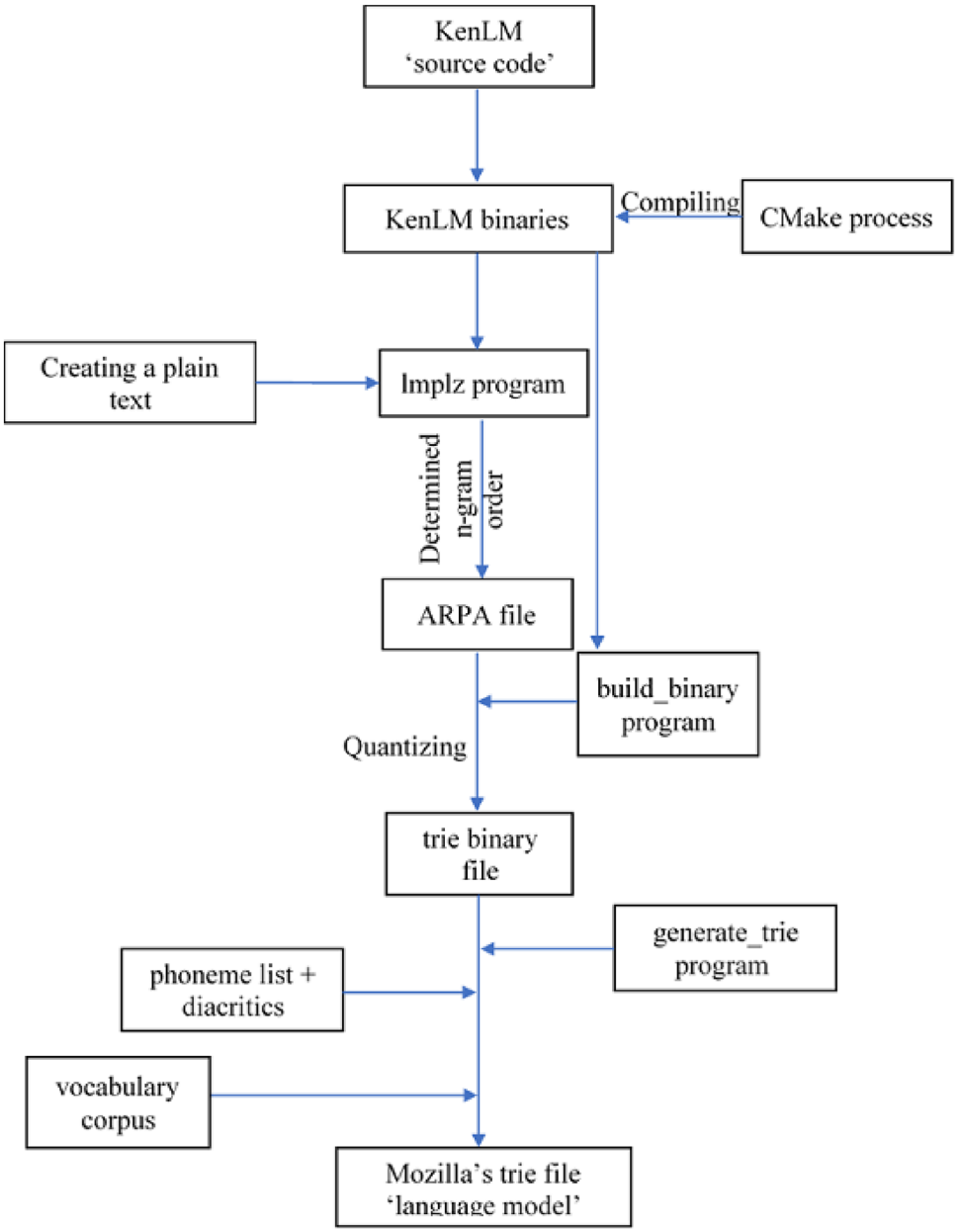

Fig. 7 Language model based on the entire diacriticked Quran text

$$CER = \frac{I_C + D_C + S_C}{N_C} 100\% \tag{3}$$

where N_C = The total number of characters in the reference, I_C = The number of characters inserted, D_C = The number of characters deleted, S_C = The number of characters substituted.

4 Results and discussions

We carried out all the experiments on a HP Workstation with Intel Xeon®W-2125 4.00GHz 8 Cores CPU, 32G RAM, two Hard Disk storages (256 GB SSD and 1 TB HDD), and one NVIDIA TITAN XP GPU with 12 GB memory and 3840 CUDA Cores. The operating system is 64-bit Ubuntu Linux 18.04.4 LTS. We used Mozilla's DeepSpeech v0.7.0-alpha.1 for the main model implementation. We developed a Python 3.6 virtual environment with TensorFlow-GPU 1.15 installed and built with CUDA 10.1 and CuDNN 7.6.5. We have installed and obtained many requirements that are necessary for sound processing such as wave, SoundFile, SoX, WebRTCVAD, and LibROSA. Pandas has been used for data manipulations and NumPy for numerical calculations.

We suffice with a typical approach to perform an SR task and get the best possible performance of this model on a Quranic record. We applied the same parameters used in the Mozilla DeepSpeech project for the English model

Table 2 Hyperparameters for the features extraction in DeepSpeech

DeepSpeech's hyperparameters	Value
The number of MFCC features	26
The length of audio window (milliseconds)	32
The step of audio window (milliseconds)	20
The step of audio window (samples)	320
The length of audio window (samples)	512
The sample rate (Hertz)	16,000

(Mozilla). As shown in Table 2, the neural network displays each audio record as a sequence of time slices, 26-dimensional MFCC vectors with a step of 20ms are calculated over windows of 32ms. 12 MFCCs, the first derivative, and the zeroth coefficient, all of these steps make up the 26-MFCC function vector. Our audio files are sampled at 16KHz, therefore, there are 512 samples and 320 samples step size for each audio windows.

For the training step, we applied the same parameters used in the Mozilla DeepSpeech project for the English model (Mozilla) as shown in Table 3. Also, we used the same epochs number which is equal to 75. The rest of the settings for speech features and training remained the same.

We evaluated the development model learned in the training process using the best English decoding parameters (α, β, beam width) from Mozilla with our language models (LM) was built, which covered the whole of the Holy Quran on the test set in the testing step. As shown in Table 4, we applied the same parameters for the evaluation and testing step in the Mozilla DeepSpeech project for English (Mozilla).

An increase in the beam width value is supposed to raise the outcomes, but this will increase the time for tests and the inference time in user applications when the model is used will increase as well.

The maximum order of the KenLM tool is 6, and for a small corpus, the n-gram order value will be small. So, in our LM, we used n-gram LM of order 5 built with 77,799 words vocabulary that covering all the words of the Holy Quran, including (AlEstiathah and AlBasmalah), where the

Table 3 Hyperparameters for the training in DeepSpeech

DeepSpeech's hyperparameters	Value
The number of hidden neurons	2048
The 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 Hyperparameters for the evaluation and testing in DeepSpeech

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

number of unique words in the holy Quran without repetition that were used in creating the two versions of our LM is 18,205 unique words.

A summary of the experiments that have been conducted in this work is provided in Table 5, Table 7 and Table 10. The variations in the experiments are based on the number of records that we used for the training, the development, and the testing. Also, the experiments vary based on the number of speakers and the gender of the speakers. Generally, the first 6 experiments use small set of records and explore the impact of the variation of the gender of the speakers. Whereas, the next 5 experiments use much larger set of only records by male reciters. Finally, the last 3 experiments are based on mixed gender sets of audio records for the Quarnic recitation.

In Experiment #1, a male model is created to measure the effectiveness of the DeepSpeech in SR on males' recitation records. We selected a small group of male records made up of 5,660 audio files of 21 male reciters that are divided into the training, the development, and the testing sets as shown in Table 5. The WER is 0.406, the CER is 0.232, and the model size is 181M. We have trained this model over 75 epochs where it took 189 steps/epoch.

To check the performance of the male model when tested on female recitations records, we had Experiment #2 which is similar to Experiment #1 except that we used female records for testing. As expected, the performance is less with a WER of 0.968 and CER of 0.758.

A female model was investigated in Experiment #3 which is also similar to experiment #1 with all records in the training, the development, and the testing are for female reciters. In this case, the WER is 0.608 and the CER is 0.396.

The DeepSpeech model perform the SR task on a male dataset better than on female dataset, this is because the quality of the record and the quality of recitation in a male dataset are better than in female dataset. Also, because the duration of the female records is less as can be figured out from the average time of the records which intern affect the training of the DeepSpeech.

To enhance the results of Experiment #3 (the female model), we had to play around with the the values of the hyperparameters in DeepSpeech. In the DeepSpeech model, the outputs are generated by the RNN model as

Table 5 Summary of the first 6 experiments

Experiment	Set	# of Records	Time range (s)	Avg. Time (s)	# of speakers	Gender	WER	CER
Experiment #1	Train	4541 (80.22 %)	[1.38–45.89]	16.88	13	M	0.406	0.23
	Dev	559 (9.87 %)	[1.09–45.79]	9.58	3	M		
	Test	560 (9.89 %)	[1.14–45.68]	8.55	5	M		
Experiment #2	Train	4541(80.22%)	[1.38–45.89]	16.88	13	F	0.968	0.758
	Dev	559 (9.87 %)	[1.09–45.79]	9.58	3	F		
	Test	560 (9.89 %)	[1.1–45.67]	6.78	5	F		
Experiment #3	Train	4541(80.22 %)	[1.4–45.60]	8.55	13	F	0.608	0.396
	Dev	559 (9.87 %)	[1.74–45.18]	7.72	3	F		
	Test	560 (9.89 %)	[1.1–45.67]	6.78	5	F		
Experiment #4	Train	4541(80.22 %)	[1.4–45.60]	8.55	13	M	0.966	0.664
	Dev	559 (9.87 %)	[1.74–45.18]	7.72	3	M		
	Test	560 (9.89 %)	[1.14–45.68]	8.55	5	M		
Experiment #5	Train	45,900 (80.21 %)	[1.38–45.89]	13.72	8	M	0.842	0.595
	Dev	5660 (9.89 %)	[1.20–45.74]	15.67	8	M		
	Test	5660 (9.89 %)	[1.1–45.67]	8.29	21	F		
Experiment #6	Train	4992(78.88 %)	[1.4–45.60]	8.5	17	F	0.857	0.601
	Dev	668 (10.55 %)	[1.1–45.67]	6.69	4	F		
	Test	668 (10.55 %)	[4.3–45.71]	19.16	4	M		

phonemes in the form of phoneme by phoneme. While, the language model interprets the phoneme sequence as words. To reduce the word error rate, we need to improve the work of the objective function of decoding in particular, to get the best sequence of phonemes that directly replace the coming out vocabulary of RNN.

Authors of (Hannun et al., 2014) defined this primary function in Eq. 4 which is a weighted combination of the LM score *Plm* and RNN score $P(c|x)$, where x represents the input utterance and c is the possible output character sequence which represents the phonemes in our work. So, α is matched with the language model weight, and β is the word insertion weight.

$$Q(c) = \log(P(c|x)) + \alpha \log(Plm(c)) + \beta \; word\_count(c) \quad (4)$$

To achieve the best value for the coefficients α and β, DeepSpeech's original search uses a beam search algorithm that yields the best results. While Mozilla uses its method from the CTC beam search decoder.

The ideal values pair of coefficients α and β in Mozilla' DeepSpeech are $(\alpha, \beta) = (0.75, 1.85)$ with a beam size of 1,024 used for the English Language test dataset. Therefore, we have worked practically to choose the best values for the coefficients α and β that achieve the lowest WER through many attempts. However, the lowest WER we got was when the value of the coefficients α and β changed together so that the value of $(\alpha, \beta) = (0.931289039, 1.183413758)$, where these values are the same as those used in DeepSpeech release 0.9.3 in English data (Mozilla, 2020).

Moreover, in our attempts to enhance our female model and get a lower WER value, we tried to change the number of hidden neurons. Since the number of hidden neurons determines the neural network size (small or big), where it is preferred if the amount of data is small as in our female data, the network size is also small. Also, the smaller value of hidden neurons will save memory and CPU. Hence, we tried the value 1024 of several hidden neurons instead 2,048 in Mozilla' DeepSpeech.

Also, we tried to get more improvement through changing the dropout rate. Dropout provides a remarkably effective regulation method to reduce overfitting and improve generalization error in all neural networks generally. We made many attempts and could achieve better results when the dropout rate value is 0.5 instead 0.15 in Mozilla' DeepSpeech. All of these attempts are listed in Table 6. Each attempt is a modification to a previous attempt as in the column with title BA#(MP) which is the abbreviation for "Base Attempt # (Modified Parameter)". For example, Attempt #10 is based on Attempt #8 with β has been changed. Attempt #1 is based on the original Experiment #3. As shown by Table 6, the best WER and CRE was achieved by Attempt #24 (WER=0.608 and CER=0.292).

In Experiment #4, the female model is tested with male recitations. We used the same training and development sets as in Experiment #3, but testing is done using male

Table 6 The attempts to enhance results of Experiment #3

Attempt	n-hidden	α	β	Dropout	lr	Epoch	BA # (MP)	WER
1	2048	0.75	1.85	0.15	0.0001	75	Experiment #3	0.608
2	2048	0.6	1.85	0.15	0.0001	75	1 (α)	0.633
3	2048	0.83	1.85	0.15	0.0001	75	1 (α)	0.587
4	2048	0.84	1.85	0.15	0.0001	75	1 (α)	0.617
5	2048	0.85	1.85	0.15	0.0001	75	1 (α)	0.579
6	2048	0.86	1.85	0.15	0.0001	75	1 (α)	0.604
7	2048	0.9	1.85	0.15	0.0001	75	1 (α)	0.598
8	2048	1	1.85	0.15	0.0001	75	1 (α)	0.67
9	2048	0.75	5.5	0.15	0.0001	75	1 (β)	0.606
10	2048	1	5.5	0.15	0.0001	75	8 (β)	0.658
11	2048	0.85	5.5	0.15	0.0001	75	10 (α)	0.635
12	2048	0.75	1.1834138	0.15	0.0001	75	1 (β)	0.593
13	2048	0.85	1.1834138	0.15	0.0001	75	12 (α)	0.5813
14	2048	0.931289	1.1834138	0.15	0.0001	75	12 (α)	0.578
15	2048	0.931289	1.1834138	0.5	0.0001	75	14 (dropout)	0.75
16	2048	0.85	1.1834138	0.5	0.0001	75	15 (α)	0.5818
17	2048	0.931289	1.85	0.5	0.0001	75	15 (β)	0.692
18	2048	0.931289	1.85	0.15	0.0001	75	1 (α)	0.644
19	2048	0.931289	1.1834138	0.15	0.00001	75	14 (lr)	0.669
20	2048	0.6940122	1.85	0.15	0.0001	75	1 (α)	0.588
21	1024	0.75	1.85	0.15	0.0001	75	1(n-hidden)	0.5644
22	1024	0.75	1.85	0.3	0.0001	75	21 (dropout)	0.592
23	1024	0.931289	1.1834138	0.15	0.0001	75	14(n-hidden)	0.609
24	1024	0.931289	1.1834138	0.5	0.0001	75	23 (dropout)	0.498

BA# base attempt #, *(MP)* (modified parameter)

recitations. The WER and the CER in this case are 0.966 and 0.664, respectively.

All previous experiments confirm that training the DeepSpeech model on data belonging to a particular gender and testing it on data belonging to the other gender will result in poor performance. However, an improvement on the performance is achieved in Experiment #5. Experiment #5 is similar to Experiment #2 with the number of the male records is increased to 51,560 records for the training and the development and all of the available 5660 female records are used for the testing. The WER becomes 0.842 and the CER becomes 0.595 comparing to 0.968 and 0.758 in Experiment #2. This is obviously means that widening the dataset would improve the performance of the ASR system even different genders are involved in the training, the development, and the testing processes.

In Experiment #6, we used all of the 5660 female records for the training and the development. While, we used male records for the testing. The WER in this case is 0.857 and the CER is 0.601. It is noticeable that performance in this case is better that the case of Experiment #4 which is similar to Experiment #6 but with less number of records. Again, this is consistent with the fact that the larger the data set, the better the performance.

In the next 5 experiments, our aim is to study the effect of diacritics in the text on the recognition errors (when building language models based on the entire Quran text). We have created two versions of the language model. The first is the cleaned version, and the second is the diacriticked version. We evaluated the ASR task on these two versions. Also, we only used the male dataset as it covers all of the Holy Quran. A summary of these experiments is shown in Table 7.

In Experiment #7, all records duration is equal to or less than 10 s and the text is cleaned. A total of 112,831 records by 41 male reciters were used. A low WER and CER values of 0.046 and 0.025 were achieved. The model was trained over 75 epochs where it took 3,760 steps /epoch. The WER has little increase when the text is diacriticked as foumded by Experiment #8 which is similar to Experiment #7 with the text is diacriticked instead of being cleaned. This means that the effectiveness of the DeepSpeech decreases with the existence of diacritics in the transcript. An increase in the number of characters in the recognition process would increase in the degree of complexity and thus an increase in the number of errors.

In Experiment #9, the model was evaluated differently. Records (of duration less than 11 s) by male reciters were used for the training and the development. Whereas, the

Table 7 Summary of the next 5 experiments

Experiment	Set	# of records	Time range (s)	# of speakers	Gender	Clean or diacriticked text	WER	CER
Experiment #7	Train	90,257 (79.99 %)	≤ 10	34	M	Clean	0.046	0.025
	Dev	11,289 (10.005 %)	≤ 10	4	M			
	Test	11,285 (10.001 %)	≤ 10	4	M			
Experiment#8	Train	90,257 (79.99 %)	≤ 10	34	M	Diacriticked	0.049	0.025
	Dev	11,289 (10.005 %)	≤ 10	4	M			
	Test	11,285 (10.001 %)	≤ 10	4	M			
Experiment #9	Train	92,964 (80.40 %)	< 11	34	M	Diacriticked	0.128	0.086
	Dev	11,642 (10.06 %)	< 11	4	M			
	Test	11,014 (9.52 %)	≤ 30	4	M			
Experiment #10	Train	183,690 (80.29 %)	≤ 30	34	M	Diacriticked	0.295	0.251
	Dev	23,022 (10.06 %)	≤ 30	4	M			
	Test	22,057 (9.64 %)	≤ 30	4	M			
Experiment #11	Train	200,213 (80.54 %)	≤ 45	34	M	Diacriticked	0.16	0.107
	Dev	24,199 (9.73 %)	≤ 45	4	M			
	Test	24,165 (9.72 %)	≤ 45	4	M			

records (also by male reciters) used for testing were of duration less than or equal to 30 s and the text is diacriticked. The outcomes of this experiment were 0.128 for the WER and 0.086 for the CER. The WER value is greater than that of Experiment #8. This is expected because, in Experiment #9, we trained the model with records of length less than the length of the records used for the testing. The number of words in the test process is increased. Thus, an increase in the number of errors happened.

Another variation is investigated in Experiment #10. A total of 228,769 records by 42 male reciters were used. The text was diacriticked and the duration of the records were less than or equal to 30 s for the training, the development, and the testing. The WER and CER were 0.295 and 0.251, respectively.

In Experiment #11, we imitated Experiment #10 with the number and the duration of the records are set to the maximum. The number of the records was 248,577 and the records were less than or equal to 45 s in duration. In this case, the achieved WER and CER were better than those of Experiment #10 with values of 0.160 and 0.107, respectively.

The trend of the WER in Experiments #8 through #10, which were conducted on the diacriticked LM, indicate that whenever the amount of data being processed through the DeepSpeech model increased, the WER value increases. However, this is not true for Experiment #11 where we justify a lower WER is because the model has been over fitted with the 45 s audio length through 75 epochs. The 45 s is the limit of audio length that can be processed by our experimental environment.

Table 8 Summary of the recitation set by male reciters in the mixed-gender data

Set	# of records	Time range (s)	Avg. time (s)	# of speakers	Gender
Train	4224	[1.38–45.89]	17.12	12	M
Dev	471	[1.09–40.28]	7.39	3	M
Test	1030	[1.14–45.68]	11.10	5	M

Table 9 Summary of the recitation set by female reciters in the mixed-gender data

Set	# of records	Time range (s)	Avg.time (s)	# of speakers	Gender
Train	4006	[1.92–45.60]	8.84	12	F
Dev	559	[1.74–45.18]	7.72	3	F
Test	1030	[1.1–45.67]	6.52	6	F

In Experiments #12 and #13, a mixed gender model is created to measure the effectiveness of the DeepSpeech in SR when mixing the recitations records such that some are by male reciters and some by female reciters. We selected a group of recitation records that are recorded by 20 males reciters. A total of 5,725 audio files are distributed among the training, the development, and the testing as shown in Table 8. Similarly, we selected another group of recitation records that are recorded by 21 female reciters in this case.

The total number of audio records in this case are 5,595. They are distributed among the training, the development, and the testing as shown in Table 9. We then mixed audio records from Tables 8 and 9 to form a set of 10,290 audio files that are recorded by 41 males and females. We used this mixed-gender set to conduct Experiments #12 and #13.

In In Experiments #12, the audio records from the mixed gender set were used in the training and the development sets as shown in Table 10. Whereas, the testing is done using audio records by male recitations only. The WER and the CER in this case are 0.175 and 0.097, respectively.

In Experiment #13, the same training and development sets of Experiment #12 were used for the training and the development. However, the testing is done using audio records by female recitations in this case. The mixed gender model in this case achieved a WER of 0.47 and CER of 0.29.

The performance obtained in Experiment #13 is less than the performance obtained in Experiment #12. In fact, the quality of the records and the quality of the recitation that are recorded by the male reciters are better than those recorded by the female reciters. The testing in Experiment #12 was done using records that are recorded by male reciters. While the testing in Experiment #13 was done using records that are recorded by female reciters. Moreover, the duration time of the records that are recorded by female reciters is less than those recorded by the male reciters as can be figured out from the average time listed in Tables 8 and 9. Putting these facts together and knowing that the testing in Experiment #12 is performed using the records that are recorded by male reciters, can be a justification for getting better WER and CER in Experiment #12. Generally, one can conclude that training the DeepSpeech model using mixed data belonging to both genders and testing it on data belonging to a specific gender will result in acceptable performance.

Finally, in Experiment #14, we tried to measure the performance when testing with mixed gender data. We mixed the same testing sets that were used in experiment #12 and #13 to form a testing sets in this experiment. Hence, a total of 2060 audio files of 11 reciters (5 males and 6 females) were used for testing. For training and development, we used same data sets that were used in the training and development in Experiment #12 and #13. A WER of 0.361 and a CER of 0.213 were obtained.

Approximately, the WER obtained by Experiment #14 equals the average of WER values obtained in Experiment #12 and #13. The justification for this can be that the training and development used the same data that used in either of Experiments #12 or Experiment #13. While, the testing mixed both of the testing used in Experiment #12 and #13. Similarly, it can be concluded that training the DeepSpeech model on mixed data belonging to both genders and testing it using mixed data also results in acceptable performance. It should be pointed out that in the last three Experiments (Experiment #12, Experiment #13, and Experiment #14), we used the same hyperparameters values that we used to enhance the results of Experiment #3.

In all of our experiments on the male dataset, we fixed the reciters splitting to use the same reciters' id in the train set, in the development set, and the test set.

We compare our results with the work in (Eldeeb) which used Mozilla's DeepSpeech model version 0.7.1 and built two SR models. The first model, called Imam-Recitations, used full Quran audio records for 7 Imam reciters, with language model built based on Uthmani style text. They did their training and evolution phases through Nvidia GeForce GTX 1070, 8 GB GPU and 16 GB RAM. They achieved a positive result on this model with WER of 0.056. Default Mozilla' hyperparameters were used except that the value of n-hidden was changed to 1024 through 30 epochs. The second model, called Imam -Recitations and T users- Recitations, used another filtered dataset that consists of 25,000 Quran audio records from Tarteel user's reciters. To have clean and useful training data, the noisy, wrong, and irrelevant audios were filtered based on the Imam reciters dataset. The difference in the text of the Quran of Tarteel user's records and the texts of the Imam's recitations is also considered. The outputs record from the filtering phase were added to the Imam reciters dataset and the total audio records

Table 10 Summary of the last 3 experiments

Experiment	Set	# of records	Time range (s)	Avg. time (s)	# of speakers	Gender	WER	CER
Experiment #12	Train	8230 (80 %)	[1.83-45.89]	13.09	24	M+F	0.175	0.097
	Dev	1030 (10 %)	[1.09-45.18]	7.57	6	M+F		
	Test	1030 (10 %)	[1.14-45.68]	11.10	5	M		
Experimen #13	Train	8230 (73 %)	[1.83-45.89]	13.09	24	M+F	0.470	0.290
	Dev	1030 (9 %)	[1.09-45.18]	7.57	6	M+F		
	Test	1030 (10 %)	[1.1-45.67]	6.52	6	F		
Experiment #14	Train	8230 (80 %)	[1.83-45.89]	13.09	24	M+F	0.361	0.213
	Dev	1030 (10 %)	[1.09-45.18]	7.57	6	M+F		
	Test	2060 (18 %)	[1.1-45.68]	8.81	11	F+M		

were entered into their training and evolution phases. They achieved WER of 0.099 through 30 epochs with the same device and hyperparameters used in the first model.

The work of (Eldeeb) and ours were done concurrently and we only discovered (Eldeeb) when we were in the final stages of our work. Nonetheless, there is a group of differences in the two works. The first difference is related to the used dataset where the size was greater and more diversity in our study. Secondly, the Uttoman text and all its drawings and symbols were used in their work, while we used the orthographic style text and the basic diacritics. This is reflected in the language model that is being built, which affects the work of the decoder. Thus, the results of the prediction of words will be different, and this means the difference in the value of the WER. The third difference is in the values of the hyperparameters coefficients on which the model was trained in their work from the values in our work. This lead to the difference in the geometric constants. Fourth and most importantly is that the basic idea in our work is to build a neural Quranic model for females, while their goal was to build a neural Quranic model based on the professional recitations mainly. At the end, the two models that were obtained were different. Therefore, if we test our data on their model, the result will not be satisfactory. We have proven this experimentally. We used our audio files, which were used in our Experiment #2, with their "Imam -Recitations" and "T users- Recitations" model that they developed in the second case. We also set the hyperparameters to the same values they used. The resulted WER was 0.767, which is worse than what they got in the second model.

5 Conclusions

In this work, we introduced a speaker-independent neuro speech recognizer structure based on the DeepSpeech model. The SR can be effectively used by any reciter of the Holy Quran regardless of the gender and the age. Training the DeepSpeech model on data belonging to a specific gender and testing it on data belonging to the other gender result in poor performance. We got 0.968 WER when we tested audios by females on the male model, while we got 0.406 WER when we tested audios by males on the male model. Similarly, testing audios by males on the female model resulted in high WER with a value of 0.966. Testing audios by females on the female model resulted in 0.608 WER. An optimization on the female model was performed and the WER was reduced from 0.608 to 0.498. The diacritics has minor negative effect on the transcript on data by males. Increasing the number of records with audio length less than 30 s in the male DeepSpeech model resulted in an increase in the WER. Whereas, the WER decreased at 45 s audio length due to overfitting issue as the 45 s value is the maximum length of the audio that can be processed by our experimental environment.

References

Abdelhamid, A., Alsayadi, H., Hegazy, I., & Fayed, Z. (2020). End-to-end Arabic speech recognition: A review. In *The 19th conference of language engineering (ESOLEC'19)*.

Abro, B., Naqvi, A.B., & Hussain, A. (2012). Qur'an recognition for the purpose of memorisation using speech recognition technique. In *2012 15th International multitopic conference (INMIC)* (pp. 30–34). https://doi.org/10.1109/INMIC.2012.6511440

Abushariah, M. A. M. (2017). Tameem v1.0: Speakers and text independent Arabic automatic continuous speech recognizer. *International Journal of Speech Technology, 20*(2), 261–280.

Agarwal, A., & Zesch, T. (2019). German end-to-end speech recognition based on deepspeech. In *Proceedings of the 15th conference on natural language processing (KONVENS 2019)*.

Akkila, A.N., & Abu-Naser, S. S. (2018). In *Rules of Tajweed the Holy Quran Intelligent Tutoring System*.

Al-Anzi, F., & AbuZeina, D. (2018). Literature survey of Arabic speech recognition. In *2018 International conference on computing sciences and engineering (ICCSE)*, (pp. 1–6).

Al-Ayyoub, M., Damer, N. A., & Hmeidi, I. (2018). Using deep learning for automatically determining correct application of basic quranic recitation rules. *International Arab Journal of Information Technology, 15*, 620.

Algihab, W., Alawwad, N., Aldawish, A., & AlHumoud, S. (2019). Arabic speech recognition with deep learning: A review. In G. Meiselwitz (Ed.), *Social computing and social media. Design, human behavior and analytics* (pp. 15–31). Springer.

Alhawarat, M., Hegazi, M. O., & Hilal, A. (2015). Processing the text of the Holy Quran: A text mining study. *International Journal of Advanced Computer Science and Applications, 6*, 262–267.

Alkhateeb, J. (2020). A machine learning approach for recognizing the Holy Quran reciter. *International Journal of Advanced Computer Science and Applications*. https://doi.org/10.14569/IJACSA.2020.0110735

AlKhatib, H., Mansor, E., Alsamel, Z., & AlBarazi, J. (2020). A study of using VR game in teaching Tajweed for teenagers (pp. 244–260). https://doi.org/10.4018/978-1-7998-2637-8.ch013

Amodei, D., Anubhai, R., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Chen, J., Chrzanowski, M., Coates, A., Diamos, G., Elsen, E., Engel, J.H., Fan, L., Fougner, C., Han, T., Hannun, A.Y., Jun, B., LeGresley, P., Lin, L., …, Zhu, Z. (2015). Deep speech 2: End-to-end speech recognition in English and Mandarin. CoRR. http://arxiv.org/abs/1512.02595

Battenberg, E., Chen, J., Child, R., Coates, A., Li, Y.G.Y., Liu, H., Satheesh, S., Sriram, A., & Zhu, Z. (2017). Exploring neural transducers for end-to-end speech recognition. In *2017 IEEE automatic speech recognition and understanding workshop (ASRU)* (pp. 206–213). https://doi.org/10.1109/ASRU.2017.8268937

Bettayeb, N. (2020). Speech synthesis system for the Holy Quran recitation. *The International Arab Journal of Information Technology, 18*, 8–15. https://doi.org/10.34028/iajit/18/1/2

Chan, W., Jaitly, N., Le, Q., & Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 4960–4964). https://doi.org/10.1109/ICASSP.2016.7472621

Collobert, R., Puhrsch, C., & Synnaeve, G. (2016). Wav2letter: An end-to-end convnet-based speech recognition system. CoRR. http://arxiv.org/abs/1609.03193

Czerepinski, K. C. (2006). Tajweed rules of the Quran. DAR-AL-KHAIR ISLAMIC BOOK.

Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing, 20*(1), 30–42. https://doi.org/10.1109/TASL.2011.2134090

Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 28*(4), 357–366. https://doi.org/10.1109/TASSP.1980.1163420

El Amrani, M. Y., Rahman, M. H., Wahiddin, M. R., & Shah, A. (2016). Building Cmu sphinx language model for the Holy Quran using simplified Arabic phonemes. *Egyptian Informatics Journal, 17*(3), 305–314. https://doi.org/10.1016/j.eij.2016.04.002

Eldeeb, T. Deepspeech-quran. (2021) https://github.com/tarekeldeeb/DeepSpeech-Quran

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., & Ng, A.Y. (2014). Deep speech: Scaling up end-to-end speech recognition.

Hayou, S., Doucet, A., & Rousseau, J. (2019). On the impact of the activation function on deep neural networks training. https://doi.org/10.48550/ARXIV.1902.06853

Heafield, K. (2011). KenLM: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation* (pp. 187–197). Association for Computational Linguistics, Edinburgh, Scotland. https://www.aclweb.org/anthology/W11-2123

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Huang, X., & Deng, L. (2010). An overview of modern speech recognition. In *Handbook of Natural Language Processing, Second Edition* (pp. 339–366)

Hyassat, H., & Abu Zitar, R. (2006). Arabic speech recognition using sphinx engine. *International Journal of Speech Technology, 9*(3), 133–150.

Iakushkin, O., Fedoseev, G., Shaleva, A., Degtyarev, A., & Sedova, O. (2018). Russian-language speech recognition system based on deepspeech. In *Proceedings of the VIII international conference "Distributed computing and grid-technologies in science and education"*.

Ibrahim, N. J., Idris, M., Razak, Z., & Rahman, N. (2013). Automated Tajweed checking rules engine for quranic learning. *Multicultural Education & Technology Journal, 7*, 275–287. https://doi.org/10.1108/metj-03-2013-0012

Ibrahim, Y. A., Odiketa, J. C., & Ibiyemi, T. S. (2017). Preprocessing technique in automatic speech recognition for human computer interaction: An overview. *The journal Annals. Computer Science Series, XV*, 186–191.

Juang, B. H., & Rabiner, L. R. (1991). Hidden Markov models for speech recognition. *Technometrics, 33*(3), 251–272.

Khalaf, E., Daqrouq, K., & Morfeq, A. (2014). Arabic vowels recognition by modular arithmetic and wavelets using neural network. *Life Science Journal, 11*, 33–41.

Khalaf, E., Daqrouq, K., & Sherif, M. (2011a). Modular arithmetic and wavelets for speaker verification. *Journal of Applied Sciences*. https://doi.org/10.3923/jas.2011.2782.2790

Khalaf, E., Daqrouq, K., & Sherif, M. (2011b). Wavelet packet and percent of energy distribution with neural networks based gender identification system. *Journal of Applied Sciences, 11*, 2940.

Kirchhoff, K., Bilmes, J., Das, S., Duta, N., Egan, M., Ji, G., He, F., Henderson, J., Liu, D., Noamany, M., Schone, P., Schwartz, R., & Vergyri, D. (2003). Novel approaches to arabic speech recognition: Report from the 2002 johns-hopkins summer workshop (pp. I–344). https://doi.org/10.1109/ICASSP.2003.1198788

Lamere, P., Kwok, P., Gouvêa, E., Raj, B., Singh, R., Walker, W., Warmuth, M., & Wolf, P. (2003). The cmu sphinx-4 speech recognition system.

Lee, K. F., Hon, H. W., & Reddy, R. (1990). An overview of the sphinx speech recognition system. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 38*(1), 35–45. https://doi.org/10.1109/29.45616

Lei, Z., Jiandong, L., Jing, L., & Guanghui, Z. (2005). 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, 24*(3), 287–302.

Lou, H. L. (1995). Implementing the viterbi algorithm. *IEEE Signal Processing Magazine, 12*(5), 42–52. https://doi.org/10.1109/79.410439

Mohammed, A., Sunar, M. S., & Salam, M. S. (2015). Quranic verses verification using speech recognition techniques. *Journal Teknologi*. https://doi.org/10.11113/jt.v73.4200

Mozilla: Deepspeech. (2021) https://github.com/mozilla/DeepSpeech

Mozilla: Deepspeech 0.9.3. (2020) https://github.com/mozilla/DeepSpeech/releases

Mustafa, B.S. Qdat. (2020) https://www.kaggle.com/annealdahi/quran-recitation

Panaite, M., Ruseti, S., Dascalu, M., & Trausan-Matu, S. (2019). Towards a deep speech model for Romanian language. In *2019 22nd International Conference on Control Systems and Computer Science (CSCS)* (pp. 416–419). https://doi.org/10.1109/CSCS.2019.00076

Pratap, V., Hannun, A., Xu, Q., Cai, J., Kahn, J., Synnaeve, G., Liptchinsky, V., & Collobert, R. (2018). wav2letter++: The fastest open-source speech recognition system. CoRR. http://arxiv.org/abs/1812.07625

Rabiner, L., & Juang, B. H. (1993). *Fundamentals of speech recognition*. Prentice-Hall Inc.

Rabiner, L.R., & Schafer, R.W. (2007). An introduction to digital speech processing. *Foundations and Trends*.

Radha, V. (2012). Implementing the Viterbi algorithm. *World of Computer Science and Information Technology Journal (WCSIT), 2*(1), 1–7.

Riesen, K., & Bunke, H. (2010). *Graph classification and clustering based on vector space embedding*. World Scientific Publishing Co.

Santosh, K., Bharti, W., & Yannawar, P. (2010). A review on speech recognition technique. *International Journal of Computer Applications*. https://doi.org/10.5120/1462-1976

Schuster, M., & Paliwal, K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing, 45*(11), 2673–2681. https://doi.org/10.1109/78.650093

Shafie, N., Adam, M., & Abas, H. (2017). The model of al-quran recitation evaluation to support in da'wah technology media for self-learning of recitation using mobile apps (2017). https://doi.org/10.13140/RG.2.2.29744.87041

Tabbal, H., El Falou, W., & Monla, B. (2006). Analysis and implementation of a "quranic" verses delimitation system in audio files using speech recognition techniques. In *2006 2nd international conference on information communication technologies* (vol. 2, pp. 2979–2984). https://doi.org/10.1109/ICTTA.2006.1684889

Wang, Y.Y., & Waibel, A. (1997). Decoding algorithm in statistical machine translation. In *Proceedings of the 35th annual meeting of the association for computational linguistics and eighth conference of the European Chapter of the Association for Computational Linguistics, ACL '98/EACL '98* (pp. 366-372). association

for computational linguistics, USA. https://doi.org/10.3115/976909.979664

Wang, D., Wang, X., & Lv, S. (2019). An overview of end-to-end automatic speech recognition. *Symmetry, 11*(8), 1018.

Young, S. (1994). The htk hidden markov model toolkit: Design and philosophy (vol. 2, pp. 2–44). Entropic Cambridge Research Laboratory, Ltd.