International Journal of Speech Technology
https://doi.org/10.1007/s10772-021-09853-9

A deep learning approach for automatic speech recognition of The Holy Qur'ān recitations

Imad K. Tantawi[1] · Mohammad A. M. Abushariah[2] · Bassam H. Hammo[2]

Received: 3 November 2020 / Accepted: 20 May 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract
Being the main spiritual source and reference for Muslims, The Holy Qur'ān can be recited in ten recitations (Qiraat). Each recitation (Qiraah) possesses certain features and characteristics that can be discriminated using Tajweed rules, which can best be defined as the elocution rules for reciting The Holy Qur'ān. This paper describes our efforts towards preparing, designing, developing, and evaluating a large-vocabulary speaker-independent and continuous speech recognizer for The Holy Qur'ān based on the narration of Hafs from A'asim by utilizing the state-of-the-art Automatic Speech Recognition (ASR) evolutionary approaches. Several Tajweed rules as depicted from the narration of Hafs from A'asim have been addressed and embedded in the development of the speech recognizer in our work. In addition, this paper presents the preparation process of The Holy Qur'ān speech corpus, which was used to train and test the speech recognizer. For training the acoustic model in our speech recognizer, four experimental setups were used within KALDI toolkit that are different in terms of dataset size and Tajweed rules. The best experimental setup is based on Time Delay Neural Networks (TDNN) with sub-sampling technique and obtained a Word Error Rate (WER) in the range of (0.27–6.31%) and a Sentence Error Rate (SER) in the range of (0.4–17.39%). Therefore, the experimental results are very promising and they indicate that the speech recognizer is able to recognize The Holy Qur'ān based on the narration of Hafs from A'asim.

Keywords The Holy Qur'ān · Recitations · Hafs from A'asim · ASR · KALDI · Tajweed rules · The Holy Qur'ān elocution rules

1 Introduction

Qur'ān or best known as The Holy Qur'ān is the spiritual and main regulatory book that contains the teachings of Islam, which was revealed and written in Arabic language to Prophet Muhammad (P.B.U.H.). Muslims refer and recite The Holy Qur'ān in various occasions pertaining Muslims' life and worships such as the five daily prayers. The Holy Qur'ān contains 323,015 letters, 77,439 words, more than 6000 verses and 114 chapters or Sūras, where each Chapter (Sūra) contains a definite number of Verses (Ayat), and each verse contains one sentence or more in general (Mcauliffe, 2006; Nasr et al., 2015).

The Holy Qur'an has various narrations (Rewayat) such as Hafs from A'asim, Warsh, Qalun, and others. Every narration of the aforementioned narrations is famous and widely recited in specific geographical locations. For instance, Hafs from A'asim narration is widely spread all over the world, whereas Warsh narration is widely spread in Algeria, Morocco, Mauritania, and many other countries in West Africa. On the other hand, Qalun narration is recited mainly in Libya, Tunisia, east parts of Algeria, and west parts of Egypt (Mcauliffe, 2006).

There are several other narrations that exist, but they are not publicly recited. However, they are recited by famous reciters and used by Muslim scholars in The Holy Qur'an interpretation (Tafseer) and jurisprudence (Fiqh). The total

✉ Mohammad A. M. Abushariah
m.abushariah@ju.edu.jo

Imad K. Tantawi
itantawi@hotmail.com; ama9150243@fgs.ju.edu.jo

Bassam H. Hammo
b.hammo@ju.edu.jo

[1] Department of Computer Science, King Abdullah II School of Information Technology, The University of Jordan, Amman, Jordan

[2] Department of Computer Information Systems, King Abdullah II School of Information Technology, The University of Jordan, Amman, Jordan

Published online: 14 June 2021

number of all The Holy Qur'an narrations is twenty, however, Hafs from A'asim, Warsh, and Qalun are the most publicly recited narrations all over the world. Table 1 shows a list of the first seven readers (Qurra'a) along with the two most famous transmitters or narrators (Rowat) (Al-Imam, 2006). In addition, there are three more readers who are less famous as shown in Table 2.

There are several acoustic feature differences that coexist between these narrations. This article focuses on Hafs from A'asim narration.

The action of repeating the Qur'ānic verses loudly from human memory is known as Qur'ān recitation. The recorded recitations of The Holy Qur'ān are the audio counterparts of its scripts. They are of great interest to several hundreds of millions of Muslims all over the world. However, there are certain rules that should be followed while reciting The Holy Qur'ān. They are called the Tajweed rules. These rules affect the Automatic Speech Recognition (ASR) process and its components with regard to recitation recognition of The Holy Qur'ān. For instance, they might affect the ASR's pronunciation dictionary, the acoustic model, the language model, and the transcription method.

The main objectives of this study are as follows: (1) to develop a large vocabulary, speaker-independent, and continuous speech recognition engine for The Holy Qur'ān recitations, (2) to provide a newly developed written corpus representing the scripts of The Holy Qur'ān covering the issues related to this study, (3) to build a phonetic dictionary for The Holy Qur'ān recitations to address the issues of this study, (4) to run several experiments to test the newly developed engine using KALDI ASR toolkit, and (5) to study the impact of tuning some KALDI hyper parameters on the overall performance of the large vocabulary, speaker-independent, and continuous speech recognition engine.

Table 2 Three less famous readers and their narrators (Al-Imam, 2006)

Qari' (*Reciter*)		Rawi (*Narrator*)	
Name	Death	Name	Death
Abu Ja'far	130 AH	'Isa Ibn Wirdan	160 AH
		Ibn Jummaz	170 AH
Ya'qub Al-Hadrami	205 AH	Ruways	238 AH
		Rawh	234 AH
Khalaf	229 AH	Ishaq	286 AH
		Idris	292 AH

The rest of the paper is organized as follows. Section 2 presents the related works. Section 3 presents the transcription and recitation background of The Holy Qur'ān, including the impact of Tajweed rules on the ASR process and its components. Section 4 addresses the design and implementation of the ASR engine for The Holy Qur'ān recitations, including a comprehensive description of the speech corpus used, and the pronunciation dictionary. Section 5 discusses the usage of KALDI deep learning ASR toolkit in the development of the ASR engine for The Holy Qur'ān recitations. Section 6 presents setting up the experiments and running them. Section 7 presents the experimental results and discussion. The conclusions and future work are finally presented in Sect. 8.

Table 1 List of the first seven readers (Qurra'a) along with the two most famous transmitters/narrators (Rowat) (Al-Imam, 2006)

Qari' (reciter)		Rawi (narrator)		
Name	Death	Name	Death	Recitation region
Nafi'Al-Madani	169 AH—785 CE	Qalun	220 AH—835 CE	Libya, Tunisia
		Warsh	197 AH—812 CE	Algeria, Morocco, Mauritania
Ibn Kathir Al-Makki	120 AH—738 CE	Al-Bazzi	250 AH—864 CE	–
		Qunbul	291 AH—904 CE	–
Abu 'Amr Ibn Al-'Ala'	154 AH—770 CE	Addouri	246 AH—860 CE	Parts of Sudan and West Africa.
		Al-Susi	261 AH—874 CE	–
Ibn Amir Ad-Dimashqi	118 AH—736 CE	Hisham	245 AH—859 CE	–
		Ibn Dhakwan	242 AH—856 E	–
Aasim Ibn Abi Al-Najud	127 AH—745 CE	Shu'bah	193 AH—809 CE	–
		Hafs from A'asim	180 AH—796 CE	Muslim world in general.
Hamzah Az-Zaiyyat	156 AH—773 CE	Khalaf	229 AH—844 CE	–
		Khallad	220 AH—835 CE	–
Al-Kisa'i	189 AH - 804 CE	Abu-Al Hareth	240 AH—854 CE	–
		Addouri	246 AH—860 CE	–

2 Related works

Arabic ASR research efforts vary according to the selected Arabic language form, some researchers worked on Classical Arabic (CA) by recognizing The Holy Qur'an speech, others worked on Modern Standard Arabic (MSA) such as recognizing phonetically rich and balanced sentences, proverbs, questions, broadcast news, telephone conversations and many more, and finally other researchers worked on Dialectal Arabic (DA) such as recognizing the spoken dialect of a certain country (Abushariah, 2017). Our research work focuses on CA since it is the Arabic language form that represents the language used to reveal and write The Holy Qur'an. Therefore, this related works section highlights mainly the research efforts for recognizing the speech and recitations of The Holy Quran, and some research attempts pertaining ASR for MSA and DA forms.

In the past two decades, several research efforts have been attempted to build efficient Arabic ASR systems to recognize MSA and DA (Abushariah et al., 2012; Kirchhoff & Vergyri, 2005; Malmasi & Zampieri, 2017). However, Arabic language is considered as an under-resourced language and the research on Arabic ASR is an open research area that needs a plethora of publicly available language resources and tools.

Few attempts were made to tackle the problem of Arabic ASR through building corpora, lexicons and other necessary tools to enrich the Arabic ASR environment (Alsulaiman et al., 2017; Abushariah et al., 2010; Elrefaei et al., 2019). Several ASR applications were built to serve multiple purposes. For instance, the work of (Alsunaidi et al., 2018; Khan et al., 2013) used ASR for pedagogical purposes, while the work of (Abushariah, 2017) used ASR for technical applications such as general purpose Arabic speech recognizer.

In addition, several efforts were devoted to the purpose of speech recognition of The Holy Qur'ān recitations. One of the earliest attempts was the work of (Tabbal et al., 2006). The main objective of their ASR system was the use of common speech recognition techniques to automatically find and delimit verses in audio recitations independent of the reciter. They used CMU Sphinx 4 ASR toolkit with a small volume speech corpus. However, building an ASR for Qur'ān recitations was not a target of their work.

The work of (El Amrani et al., 2016) addressed the Qur'ānic recitations using the CMU Sphinx 4 ASR toolkit. They used their system to train and evaluate a relatively small language model made of four short Chapters (Sūras). Furthermore, few Arabic and Tajweed rules were incorporated in their work.

On the other hand, the work of (Hafeez et al., 2014) developed a speaker-dependent speech recognition system to recognize and evaluate the accuracy of the recitation of some selected Qur'ānic verses. The system was developed using CMU Sphinx 4 ASR toolkit based on Hidden Markov Models (HMMs). However, the volume of the speech corpus used in their work was also relatively small.

The work of (Khelifa et al., 2017) addressed the subject matter from Quranic sounds recognition point of view, and modeling the Quranic sounds' durations. It does not use different Narration, the main issue is the duration, and also the proposed ASR system is speaker dependent. The system uses the statistical approach of Hidden Markov Models (HMMs) for modeling the Quranic sounds, and the Cambridge HTK tools as a development environment.

Regarding the research efforts on Tajweed rules detection using ASR approaches, the number of published papers is less than that of published papers devoted to the ASR of The Holy Qur'ān. For instance, the work of (Yousfi et al., 2018) intended to distinguish between two types of Madd (prolongation) as contained in 10 verses only. The error rate ranges from 30 to 50%, which is not good for a relatively small volume speech corpus. The work of (Mahmod, 2016) provides a review of the techniques and methods used for building systems that are able to check Qur'ānic Tajweed rules through recitation recognition.

3 The Holy Qur'ān transcription and recitation

Reading and writing the scripts of The Holy Qur'ān are different from any other Arabic script. This is due to the very strict pronunciation and orthography Qur'ānic rules, which are summarized and elaborated in this section.

3.1 The Holy Qur'ān transcription

Historically, the Qur'ānic text went through several stages until it reached the final stage, which constitutes the original set for the Qur'ān copies we have today. This section describes the three main stages that were developed to evolve the Qur'ānic text, namely, the Othmanic manuscripts, the inclusion of some diacritical points and last letter marks, the inclusion of Tajweed signs, stop and resumption signs, and other marks, finally the Digital Qur'ān Computing (DQC) Theme, will be briefly presented.

3.1.1 The othmanic manuscripts

The origin of the millions of copies of The Holy Qur'ān in use through the centuries up to these days, can be traced back to the manuscripts written in the period of the third Muslims Caliph, Othman Ibn Affan (Al-Imam, 2006). These manuscripts take into consideration the several readings of

The Holy Qur'ān. At this stage, diacritical marks were not used and several letters such as (ﺑ / ﺗ / ﺛ / ﻧ / ﻳ) (b / n / θ / t / j) have the same orthographic representation with the absence of the dots above and beneath the letters. However, this was not a problem for ancient Arabs to recite The Holy Qur'ān correctly (Mcauliffe, 2006).

3.1.2 Diacritic points and last letter marks

When several non-Arab nations converted to Islam such as Persians and the Copts among others, people of those nations had difficulties reciting the Qur'ān correctly. Their recitations were criticized by making a lot of spelling and grammatical mistakes, which affect the meaning of Qur'ān verses. Muslims' Caliphs during the first century of Islam were aware of this problem and, hence, they commanded Muslim scholars to introduce grammatical marks (last letter marks) and diacritical marks (tashkeel) to adjust the methods of the Qur'ān recitation among all Muslims (Al-Imam, 2006).

3.1.3 Tajweed signs, stop and resumption signs, and other marks

Tajweed rules (discussed in Sect. 3.2), are related to the correct recitation of The Holy Qur'ān. Several Tajweed rules are indicated by special signs (i.e. superscript letters) placed above some letters of a Qur'ānic verse. For instance, the superscript letter ' م ' /m/ indicates a Tajweed rule of Iqlab when a non-vowelized Noon ' نْ ' /n/ is followed by a Ba ' ب ' /b/, then the Noon ' نْ ' /n/ is pronounced as the letter Meem ' م ' /m/.

The stop and resumption signs are used to indicate whether a stop on a word of a verse is permissible or prohibited. The signs are placed according to context and meaning. For instance, the superscript letter ' ج ' /ʤ/ above a word at the end of a verse indicates that stopping is permissible, while the superscript letter ' لا ' /læ:/ indicates that stopping at that position is prohibited.

Several signs were also used for other purposes, such as boundaries between Chapters (Sūras) and a mark of a verse in a chapter followed by a number. The Holy Qur'ān is divided into thirty parts (Juzu'), where each Juzu' is divided into two parts that are called Hizb. Thus, the Qur'ān has 60 Hizbs, then each Hizb is divided into four quarters, where each quarter is called (Rubu' Hizb). Accordingly, this division introduced special signs to mark the boundaries of Qur'ānic organization. Other signs indicating a position of Sajdah or a position of Sakt were also introduced (Abudena, 2015).

Tajweed signs are not used at this stage of preliminary work and they might be addressed in future work. Although Tajweed signs affect the recitation process, the underlying mapping rules are difficult to implement. Some Tajweed rules are optional in some situations. For instance, the range of Madd Monfasel ' المد المنفصل ' in the narration of Hafs, spans from 2 to 5 counts. In addition, stop and resumption signs are also not implemented at this stage. The reason behind this decision is because of their semantic dependability, which is not required for the ASR process.

3.1.4 The digital Qur'ān computing (DQC) theme

The wide spread usage of digital media content has fueled the process of digitizing printed documents, which results in increased efficiency of the publishing process, and increases the accessibility of these documents for more people. This process in the Qur'ānic context was fueled by the willingness of several parties and persons to provide any needed funds or resources to bring the emerging IT services to The Holy Qur'ān domain. DQC was introduced to conduct conferences and symposia during the last few years, and build applications, tools, and algorithms that were dedicated to meet its theme and serve its objectives (Zakariah et al., 2017).

Similarly, Natural Language Processing (NLP), ASR, and many other applications require language resources that are machine/computer readable in order to achieve their objectives. In our work, the scripts of The Holy Qur'ān were transformed into a format that can be used to create the acoustic model, the language model, the pronunciation dictionary, and many other recipes of the ASR engine. In other words, the aforementioned three stages were taken into consideration and were normalized to produce a machine/computer readable text corpus of The Holy Qur'ān that can be used for various applications. In our work, we downloaded a simple version of The Holy Qur'ān text from Tanzil, which is an international Quranic project that provides the Qur'ān text in Unicode. The simple version contains 6236 lines that represent the verses of The Holy Qur'ān. However, it does not include the special signs to mark the boundaries of Qur'ānic organization and the superscript ' ا ' /æ:/ (Tanzil, 2020).

3.2 The Holy Qur'ān recitation

3.2.1 The Tajweed rules

Reading The Holy Qur'ān is different from reading any other Arabic scripts. Qur'ān has very strict pronunciation rules, known as Tajweed rules. Each narration of Qur'ān such as Warsh, Hafs from A'asim, and others has its own Tajweed rules. However, there are several rules that are common between all narrations such as the rule of compulsory prolongation (اللازم المد). The main objective of using Tajweed is to adjust the process of Qur'ān recitation, thus the words of

Qur'ān are pronounced correctly among all Muslims (Elhadj et al., 2012). To learn more about Tajweed and its rules, the readers can refer to (Elhadj et al., 2012) and (Czerepinski & Swayd, 2006).

3.2.2 The impact of Tajweed rules on The Holy Qur'ān ASR systems

Several Tajweed rules might affect The Holy Qur'ān ASR systems differently, which might depend on the letter itself, its diacritics, the letter preceding it and its diacritics, the letter following it and its diacritics, and the word that contains the letter might also have a special case of some rules. For instance, in the category of velarization and attenuation (التفخيم والترقيق), the same letter may be read velarized like the letter ' ر ' /r/. According to Hafs from A'asim narration, this letter is attenuated(ترقق)) when its diacritic is Kasrah (ـِ) /i/ as in the Arabic word ' غَيْرِ ' /ɣajr/ or when its diacritic is Sukoon (i.e. it is not vowelized) and it is preceded by a letter with Kasrah (ـِ) /i/ diacritic as in the Arabic word ' فِرْعَوْنُ ' /firʕawnu/. An exception for this case when it is followed by a velarized letter, it should be velarized even if its diacritic is Sukoon and it is preceded by a letter with Kasrah (ـِ) diacritic, like the word ' قِرْطَاسٍ ' /qirtˁa:si:/.

This exceptional rule generates a new rule, which states that if the diacritic of the velarized letter is Kasrah (ـِ) /i/, then the ' ر ' /r/ can either be velarized or attenuated such as the word ' فِرْقٍ ' /firqi:/. In addition, it should be velarized when it is preceded by a letter with Kasrah (ـِ) /i/ diacritic, but that Kasrah (ـِ) /i/ is preceded by the conjunctive hamzah (همزة الوصل) has a casual Kasrah (ـِ) /i/ (i.e. the Kasrah (ـِ) /i/ is not the original diacritic) like the word ' ارْجِعِي ' /rdʒiʕij/. Otherwise, it will be velarized if its diacritic is Fathah (ـَ) /a/ or Dammah (ـُ) /u/, like the word ' رُسُلِي ' /rusulij/, or it is not vowelized and the diacritic of the preceding letter is Fathah (ـَ) /a/ or Dammah (ـُ) /u/, like the word ' أَنذَرْتَهُمْ ' /ʔanðartahum/ (Bellegdi & Al-Muhtaseb, 2015).

This problem will be exaggerated for the letter Noon Sakinah (i.e. non vowelized Noon) ' نْ ' /n/ or Nunation (Tanween) (ـٍ) /i:/ (ـٌ) /u:/ (ـً) /a:/, which has four rules according to the letter succeeding it. Furthermore, some of these rules will also generate new rules according to the characteristics of the succeeding letter (Abalkheel, 2016), which also applies to Meem Sakinah ' مْ ' /m/(Czerepinski & Swayd, 2006).

Prolongation (Madd) introduces several length types of the vowels that can be of 4, 5 and 6 counts for each of the three long vowels namely, (ا) /a:/, (و) /w/, and (ي) /j/. If optionality is introduced such as the case for the temporary prolongation (مد عارض للسكون), every word with this rule applicability should be represented 3 times in the pronunciation dictionary, which will cause that dictionary to grow exponentially (Czerepinski & Swayd, 2006).

The recitation speed makes the situation even more complex, which can either be fast recitation (حدر), medium recitation (تدوير), and slow recitation (تحقيق). The major effect of the recitation speed is timing and prolongation, where more time and prolongation counts are caused as we move from fast recitation through slow recitation. In addition, the dependency of several Tajweed rules on stop and continuation of recitation creates new situations. In the case of continuation of recitation, the reciter utters a new syllable that combines the last letter of the word that ends with Noon Sakinah ' نْ ' or Nunation (Tanween) (ـٍ) /i:/ (ـٌ) /u:/ (ـً) /a:/ with the first letter of the word succeeding it, which results a case of assimilation (إدغام), thus, assimilation (إدغام) rules are applied (Czerepinski & Swayd, 2006).

Repetition of some words due to resumption rules needs attention to include the repeated words in the transcription of the audio segment. In a narration such as Warsh, new phonemes for cases of major deflection (Emaalah), minor deflection (Taqleel), simplification (Tas-heel) are needed (Yousfi & Zeki, 2017; Yousfi et al., 2018).

4 ASR design and implementation for The Holy Qur'ān recitations

ASR converts a speech signal into a textual representation, (i.e. sequence of said words) in a specific natural language by means of an algorithm implemented as a software or hardware module (Besacier et al., 2014). In this work, we aim to design, develop, and evaluate an ASR engine for The Holy Qur'ān recitations using a deep learning approach in KALDI toolkit. In this paper, the ASR engine for The Holy Qur'ān recitations focuses mainly on Hafs from A'asim narration, which will be extended in future versions of the ASR engine that are currently under development by including narrations such as Warsh and Qaloon. Therefore, this paper presents our efforts towards designing, developing, and evaluating the first version of the ASR engine for The Holy Qur'ān recitations using Hafs from A'asim narration. The following sub-sections describe the design and implementation procedures and components including the speech corpus collection and preparation efforts.

4.1 Architectural design of the ASR engine for The Holy Qur'ān recitations

The main components for developing and evaluating the Qur'ānic ASR engine are shown in Fig. 1 (Abushariah, 2017), which includes the speech and text corpora as inputs to other succeeding processes, feature extraction that produces feature vector (Y) as extracted from the speech

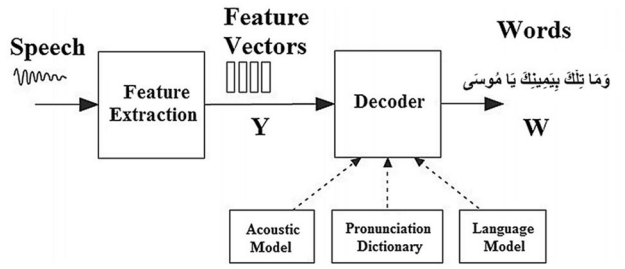

Fig. 1 Components of a typical ASR System (Abushariah, 2017)

corpus using some feature extraction algorithm, acoustic model using a deep learning approach, phonetic dictionary also known as pronunciation dictionary, language model, and decoder. The output of the Qur'ānic ASR engine is a sequence of words (W) that represents a verse or verses from The Holy Qur'ān.

4.2 Speech corpus collection and preparation of The Holy Qur'ān recitations

Several Qur'ān recitations are available and distributed for free for many famous reciters from around the globe. In this study, we used Chapter 20 (Sūrat Taha) from the The Holy Qur'ān, which is a medium sized Chapter that consists of 135 verses, contains 1353 words with 875 unique words, and makes up nearly 10 pages that is represented as one Hizb in The Holy Qur'ān. The main reason for choosing this Sūra refers to its acoustic richness in terms of many acoustic features such as the minor deflection (Taqleel) in accordance to Warsh narration. Although, we used Hafs from A'asim narration in this research, we intend to extend this work in future to accommodate Warsh and other narrations for comparison purposes. The preparation process was compliant with KALDI ASR toolkit recipe and requirements.

4.2.1 The collection of the data set

The recitations of Sūrat Taha in (.mp3) format were collected for a set of reciters according to Hafs from A'asim narration. We collected 50 recitations of Sūrat Taha, where each recitation of the entire Sūra is stored in one (.mp3) file. However, due to noise and echo in some of the collected recitations, the final number of recitations that is used in this work is reduced to 32 recitations for Sūrat Taha according to Hafs from A'asim narration. The file duration of each recitation was about 25 min. The major difference among reciters was the recitation speed, where most reciters recite in a medium style 'تدوير', few recite in fast style 'حدر', and few others recite in slow style 'تحقيق'. The other difference was related to the count of Separated Prolongation 'المد المنفصل', where most reciters were using four counts, while few of them were using two counts.

Table 3 Training and testing data sets

Dataset ID	Training			Testing	
	Speakers	Utt.	Duration	Speakers	Utt.
Set # 1	5	870	1.74 h	2	369
Set # 2	7	1141	2.40 h	2	363
Set # 3	26	4035	10.40 h	6	1000
Set # 4	32	5035	12.50 h	0	0

4.2.2 Audio files segmentation, conversion and transcription

The collected audio files were checked for noise and echo. Several files were excluded from this study due to large noise or echo, where some of these effects were natural since some recitations were recorded during prayer time, or the existence of audience with some background noise. On the other hand, the use of time-based audio effects like delay, echo and reverb as enhancements in some recorded recitations introduce noise intentionally. In order for these recitations to be usable, echo cancellation techniques should be used such as the research efforts conducted by (Kamarudin et al., 2015, 2016).

Each of the selected (.mp3) recitation files went through a segmentation process. Some were segmented manually using some audio editing tool, where each file was listened to and segmented on stop position. While in other cases, the segmentation was automated using an audio splitting tool. However, the tool suffered from several problems, including the need to calibrate the silence level threshold, determine the minimum length of silence and none-silence periods, and the creation of long utterances, which may degrade the accuracy of the ASR process. Therefore, we had several trials before the final segmentation process was succeeded. The results of auto segmented recitation files were revised and manual corrections were done if needed using audio editing tools. Finally, every segmented file was converted to (.wav) format to meet the requirements of KALDI ASR toolkit.

The transcription phase of the audio segments (i.e. the wav files of the audio segmentation process), were carried carefully using Tanzil version of The Holy Qur'ān text as mentioned earlier, which is used to build the transcription file. Each (.wav) file was carefully examined and the transcription of all audio segments' files of recitations were produced. Both segmentation and transcription were checked by a subject matter for validation. Due to the requirements of the KALDI ASR toolkit pertaining

the naming rules, both sets of (.wav) files containing utterances and text files containing transcriptions, were modified.

4.2.3 Constructing experimentation data sets

The collected speech corpus is used to construct four datasets that can be used in conducting the training and testing of The Holy Qur'ān ASR engine as shown in Table 3. Description of the newly four constructed datasets are as follows:

(a) The first dataset (Set # 1) has Separated Prolongation المد (المنفصل) with two counts long as the main Tajweed feature. Seven recitations of Sūrat Taha using Hafs from A'asim narration Tajweed rules with Separated Prolongation length equals to two counts were chosen, where five of them were chosen for training and two were chosen for testing.
(b) The second dataset (Set # 2) has Separated Prolongation (المد المنفصل) with four counts long as the main Tajweed feature. Eight recitations of Sūrat Taha using Hafs from A'asim narration Tajweed rules with the Separated Prolongation length equals to four counts were chosen, where six of them were chosen for training and two were chosen for testing.
(c) The third dataset (Set # 3) was the result of combining the recitations of the previous two datasets with another seventeen recitations to construct a larger dataset with thirty-two recitations. In this dataset, there was no concentration on specific Tajweed rules, but only training and testing of The Holy Qur'ān ASR engine regardless of any Tajweed rules. The dataset was split into twenty-six recitations for training and six for testing.
(d) The fourth (Set # 4) was constructed to build a simple ASR application that takes an audio file containing a piece of recitation from Sūrat Taha in Hafs from A'asim narration. The audio file is then recognized and converted to text. This dataset used all thirty-two recitations from (Set # 3) to train the acoustic model.

The dataset ID along with the number of reciters (speakers), number of utterances (verses) that is referred to as (Utt.), and dataset duration in hours are shown in Table 3.

4.3 Feature extraction of The Holy Qur'ān recitations

Raw speech signals cannot be used directly in the ASR process. Instead, they should be transformed by some technique to a form that is usable by other ASR components in a process that is referred to as feature extraction process. Several feature extraction techniques are available such as Linear Predictive Coding (LPC), Perceptual Linear Prediction (PLP), and Local Discriminant Bases (LDB). However, Mel-Frequency Cepstral Coefficients (MFCC) feature extraction technique is proved more efficient and considered the dominating technique that is widely used in many of the-state-of-the-art ASR toolkits (Bezoui et al., 2016).

MFCC follows the human peripheral auditory system, which follows a special scale (i.e. Mel Scale), which uses linear frequency spacing for frequencies below 1000 Hz, and logarithmic spacing above 1000 Hz. MFCC has seven computational steps as shown in Fig. 2 (Davis & Mermelstein, 1980; Magre et al., 2013).

1. Preprocessing (or Pre-Emphasis): It involves the conversion of analog speech signal into digital form using sampling and quantization techniques. The produced signal then undergoes a high pass filter, consequently emphasizing higher frequencies.
2. Framing: The speech signal is segmented into frames, where the length of each frame ranges from 20 to 40 ms (ms), at a rate of 100 frames per second. This will enable the non-stationary original speech signal to be segmented into quasi-stationary frames, and enables Fourier transformation of the speech signal.
3. Windowing: It is performed in order to avoid any unnatural discontinuities in the speech segment or distortions in the underlying spectrum. In order to conduct the windowing process, Hamming window is used, since it is considered the most commonly used window shape.
4. Discrete Fourier Transform (DFT): The speech signal has finite energy; therefore, it can be converted using Fourier theorem into a series of sinusoidal functions, which converts the speech signal from time domain into frequency domain. This will aid in subsequent steps.
5. Mel Filter Bank: The information carried by low frequency components of human speech signal is more important than the high frequency components. Therefore, this filter bank will place more emphasis on the low frequency components of the signal from the previous step (i.e. DFT).

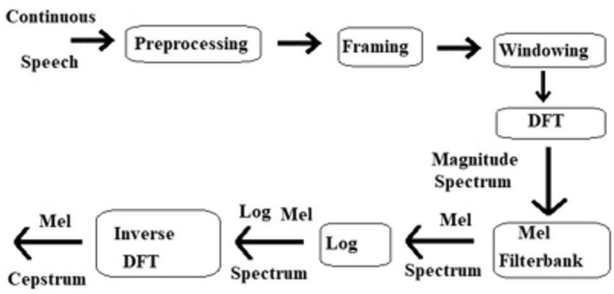

Fig. 2 MFCC computational processes (Davis & Mermelstein, 1980; Magre et al., 2013)

6. Logarithm: Humans are less sensitive to slight changes in amplitude at high amplitudes than at low amplitudes. To follow this phenomenon, this log process is introduced to also make the feature extraction less sensitive to input variations.
7. Inverse DFT: It extracts (using deconvolution) the slowly varying vocal tract impulse response, and discards less relevant information of glottal pulse.

The Fourier series is infinite, therefore the first twelve coefficients are used, and the rest are discarded. The energy in the frame will be added to this set of elements. In addition, for taking the context into consideration, the delta set is added that represents the difference between corresponding coefficient values of consecutive frames, and delta–delta is also added that represents the difference between corresponding first order delta values, which results in 39 real number values. These values will be grouped into a single vector called the feature vector, and used in subsequent stages for the training and testing of The Holy Qur'ān ASR engine.

4.4 Pronunciation dictionary of The Holy Qur'ān recitations

Pronunciation dictionary also known as phonetic dictionary plays a crucial role in the development of ASR systems, which is treated as a mediator and intermediary link between the acoustic model and the language model (Abushariah, 2017; Ali et al., 2008) as shown in Fig. 3.

Based on Fig. 3, it is clear that the development of any ASR system requires primary language resources including texts and transcriptions for training the language model, the audio or speech data for training the acoustic model, and the pronunciation dictionary that links both the acoustic model and the language model.

The pronunciation dictionary contains a mapping between the words of the intended language as contained in the text corpus and transcriptions and their corresponding phoneme sequences. It can be generated completely manual, manually supervised, using grapheme-to-phoneme rules, and/or manually supervised grapheme-to-phoneme rules (Alqudah et al., 2020; Abushariah, 2017; Adda-Decker & Lamel, 2006).

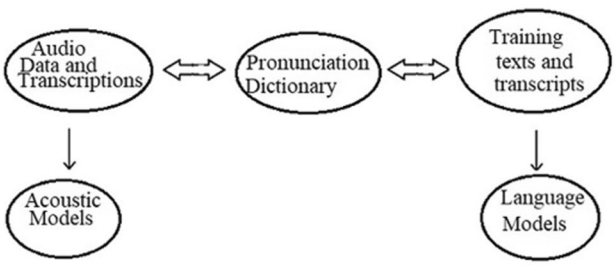

Fig. 3 Language-dependent resources for ASR (Adda-Decker & Lamel, 2006; Alqudah et al., 2020)

The pronunciation dictionary acts as a lookup table used to map each word in the Qur'ānic corpus (or any other corpus used in the ASR system) to the sequence of phonemes that represents the articulation of that word, and transliteration is the process used in the generation of such dictionary (Ali et al., 2008).

Several transliteration schemes were used in grapheme to phoneme mapping (i.e. pronunciation process) for Arabic (Lawson, 2008). One of these schemes is the Buck Walter transliteration. However, it does not consider diacritics, and hence, it would not be suitable for the transliteration of The Holy Qur'ān scripts. According to (Kirchhoff & Vergyri, 2005) most available acoustic data collections are transcribed without diacritics. Such a transcription omits essential pronunciation information about a word, such as short vowels.

For the purpose of creating the pronunciation dictionary for The Holy Qur'ān recitations, the transcription of The Holy Qur'ān is fully vowelized, which was used as input to an automated tool created in (Alqudah et al., 2020) based on previous attempts in (Abushariah, 2017; Ali et al., 2008). The tool is a rule-based phonetic dictionary generator that is developed using Java programming language. It receives the vowelized transcription of The Holy Qur'ān and transforms it into the corresponding phoneme sequences using pre-defined grapheme-to-phoneme and phonological rules as presented in (Ali et al., 2008; Al-Ghamdi et al., 2004; Elshafei, 1991).

In developing the ASR engine for The Holy Qur'ān recitations, the transcription file contains 77,805 words and the vocabulary list contains 18,202 unique words. The number of pronunciations in the developed phonetic dictionary is 21,396 entries. Figure 4 shows a sample of the pronunciation dictionary for The Holy Qur'ān recitations.

4.5 Language model of The Holy Qur'ān recitations

Language model is constructed from the written set of all verses or sentences used in the transcription process (i.e. the language). The occurrence probability of the n-gram substrings (i.e. sets of consecutive n words) in this language is computed using probability algorithms. For most Large Vocabulary Continuous Speech Recognition (LVCSR) systems the value of n is 3 (tri-gram). For developing the language model of The Holy Qur'ān recitations, SRI Language Modeling (SRILM) toolkit is used (Stolcke, 2002), for the transcription of Sūrat Taha.

For the ASR engine of The Holy Qur'ān recitations, the number of uni-grams is 891, whereas the number of

آبَاءَكُمْ E AE: B AE: E AE K UH M
آبَاءَنَا E AE: B AE: E AE N AE:
آبَاءَهُمُ E AE: B AE: E AE H UH M UH
آبَاءَهُمْ E AE: B AE: E AE H UH M
آبَاءِ E AE: B AE: E IH
آبَاؤُكُم E AE: B AE: E UH K UH M
آبَاؤُكُمْ E AE: B AE: E UH K UH M
آبَاؤُنَا E AE: B AE: E UH N AE:
آبَاؤُهُم E AE: B AE: E UH H UH M
آبَاؤُهُمْ E AE: B AE: E UH H UH M
آبَائِكَ E AE: B AE: E IH K AE
آبَائِكُمُ E AE: B AE: E IH K UH M UH
آبَائِكُمْ E AE: B AE: E IH K UH M
آبَائِنَا E AE: B AE: E IH N AE:
آبَائِهِمْ E AE: B AE: E IH H IH M
آبَائِهِنَّ E AE: B AE: E IH H IH N AE
آبَائِي E AE: B AE: E IY

Fig. 4 Sample of the Arabic pronunciation dictionary for The Holy Qur'ān

bi-grams and tri-grams is 1697 and 1238, respectively for Sūrat Taha.

4.6 Acoustic model of The Holy Qur'ān recitations

This component of the ASR system is built during the training phase of the ASR process using the training data set audio and text files as presented in Table 3. The utterances are stored in audio files using standard format (.wav) type. The text files contain the transcription of the utterances. The main functionality is to represent the relationship between an audio signal using feature vectors and the phonemes or other linguistic units that make up speech.

Both sets of files (i.e. audio and transcription files) of the test data set are used in the testing process to test the accuracy of the whole ASR engine of The Holy Qur'ān.

The ASR process, using classification techniques, will infer from the acoustic model and the feature vectors of test dataset, the sequence of phonemes correlated to this set of input vectors, then using the pronunciation dictionary, the phoneme sequence will be transformed into a sequence of words. Finally using the language model, the ASR will choose the most probable sentence or verse, for that sequence of words.

In our work, KALDI toolkit is used to train the acoustic model using traditional approaches and Deep Neural Networks (DNN) approach as presented in detail in Sect. 6.2.

4.7 The decoder

The decoder is the heart of any ASR system that takes the set of feature vectors, as input, then with the aid of other components of the ASR system, it will produce a string of words. This string of words can be thought of as the answer to the following question: "Given a string of acoustic observations, how should we choose the string of words which has the highest probability of being the string uttered by the utterance generated these acoustic observations" (Jurasky & Martin, 2000).

Several open source toolkits are available for ASR researchers, HTK, CMU SPHINX and JHU KALDI (Sahu and Ganesh, 2015). The most recent one and the state-of-the-art is KALDI toolkit, which introduces DNN to the open source ASR technology. Therefore, our research uses KALDI toolkit to evaluate the ASR engine for The Holy Qur'ān recitations using Hafs from A'asim narration. Sufficient details are presented in Sect. 5.1.

5 Developing the ASR engine of The Holy Qur'ān using KALDI toolkit

5.1 Overview of KALDI toolkit

According to (Povey et al., 2011), KALDI is an open-source toolkit for speech recognition written in C++ and licensed under the Apache License v2.0. One of the main reasons for using this toolkit is its license type. In addition, the following reasons make KALDI an attractive toolkit for ASR according to (Cosi, 2015):

(1) Ease of use, where the user just needs to learn the basics and understand the ASR concepts.
(2) KALDI can be redistributed due to its license.
(3) It is supported by a wide community that is open to establish team cooperation.
(4) KALDI also supports speaker recognition, and speaker diarisation (the process of partitioning an input audio stream into homogeneous segments according to the speaker identity).

An important aspect of using KALDI toolkit is the availability of many out of the shelf solutions, which can be reused or customized according to the researcher's needs.

A schematic overview of the KALDI toolkit is shown in Fig. 5 (Povey et al., 2011). Based on Fig. 5, KALDI toolkit depends on freely available libraries. The first one is linear algebra library (the standard "Basic Linear Algebra Subroutines" (BLAS) and "Linear Algebra PACKage" (LAPACK)) both of which are maintained by Netlib repository of software for scientific computing.

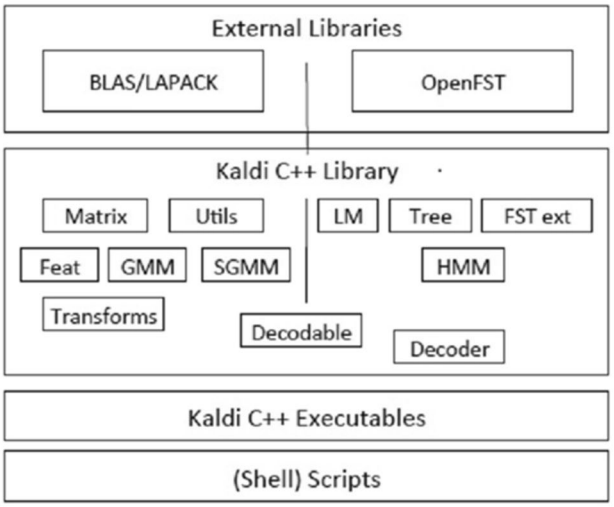

Fig. 5 A simplified view of the different components of KALDI

KALDI uses several implementations and interfaces for these libraries. The most recent and currently the preferred CBLAS/CLAPACK provider for KALDI, is Intel Math Kernel Library (MKL), which provides a freely available C-language interface to a high-performance implementation of the BLAS and LAPACK routines. The MKL provides a very highly optimized implementation of linear algebra routines, and especially on Intel CPUs. Thus with MKL you will automatically benefit from all features and instruction sets, if they are available on your CPU, without any additional configuration. These instructions accelerate linear algebra operations on Intel CPUs significantly (Kaldi_Team_MKL, Braun et al., 2019).

KALDI toolkit added NVIDIA CUDA matrix library in 2017, which provides access to GPU-based matrix operations with an interface similar to the original KALDI Matrix library. This addition enables KALDI to introduce heavy computation DNN implementations like Time Delay Neural Networks (TDNN), Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM), and enables KALDI users to achieve superior performance over other ASR toolkits and non GPU KALDI configurations (Kaldi_Team_CUDA).

The other external library is OpenFst for the finite-state framework, which is an open-source library for weighted finite-state transducers (WFSTs) (Allauzen et al., 2007).

The second layer is a set of C++ library modules, grouped into two distinct halves, each depending on only one of the external libraries. A single module, the Decodable-Interface bridges these two halves.

The third layer is a set of command line executables written in C++. They provide library functionalities access to the fourth layer components, which are a set of shell scripts.

These shell scripts are usually grouped into recipe main shell script (i.e. the run.sh script), which executes the actual ASR session.

The KALDI toolkit will take from the user the set of files described in the next section, and do the rest of the training and testing work. KALDI will perform validation of user supplied files, and fixing some problems like sorting for example or reporting any warnings or errors detected. It will then perform feature extraction (i.e. calculate the MFCC for each frame) for all datasets (i.e. training and testing), build the language model, perform several training and aligning phases, build the acoustic model from the training dataset, and build the phonetic decision trees.

The testing phase (or the decoding phase to be more general) as previously shown in Fig. 1. (Abushariah, 2017), the feature extraction is done on each frame of the testing dataset of utterances, the decoder will then be invoked. Using the feature vectors and the acoustic model, which were developed during the previous training phase, a set of phonemes will be generated. With this set of phonemes and the pronunciation dictionary, a set of words will be generated. Finally, this set of words with the language model will generate the most probable sentence for the transcription of the input utterance. After decoding the input utterance(s) into sentence(s), scoring will be done by comparing word to word, the decoded sentence(s) with the real transcription of the test utterance, consequently the Word Error Rate (WER) and the Sentence Error Rate (SER) will be calculated for each utterance, each speaker, and the whole testing dataset.

There are several types of DNN implementations in KALDI, where the most recent and the-state-of-the-art at the time of conducting this research is called nnet3 (i.e. neural networks release 3). It is intended to support more general kinds of networks than simple feed-forward networks (e.g. things like RNNs and LSTMs) in a natural way that does not require the user to do any real coding.

Further, the nnet3 was enhanced by TDNN, which uses shift invariance pattern classification. For ASR, this technology eliminates the need for determining the beginning and ending points of utterances before classifying them (Waibel et al., 1989), but this gain of time invariance comes at a large computation cost, which is due to the need of computing hidden activations for all time steps in the entire temporal context. This motivates the inventors of KALDI to improve the traditional TDNN model using a sub-sampling technique, where hidden activations at only few time steps are computed at each level. Through a proper selection of time steps at which activations are computed, computational cost can be reduced, while ensuring that information from all time steps in the input context is processed by the network as shown in Fig. 6 (Peddinti et al., 2015).

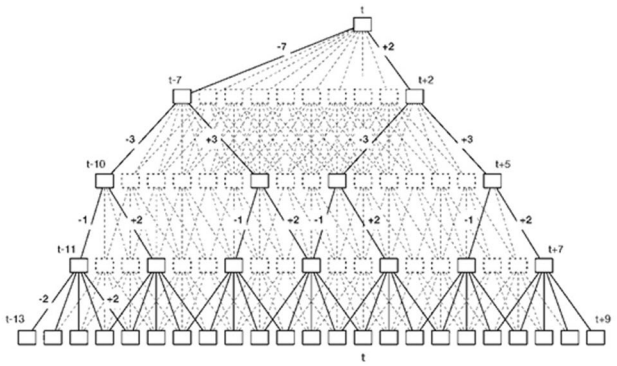

Fig. 6 Computation in TDNN with sub-sampling (solid) and without sub-sampling (Solid + Dotted)

5.2 Development of the ASR system using KALDI

After constructing the audio and text files, several steps should be taken before running the experiments, which are described as follows:

(1) Building the files that will be used by KALDI toolkit to run the experiments. We have two datasets of files; one dataset for training and one dataset for testing. These files have special syntax and convention and they include: the file (wav.scp) that specifies the location of the audio files' utterances (.wav) files, a text file that maps each utterance to its transcription (text), speaker to gender file (spk2gender), and utterance to speaker mapping file (utt2spk).
(2) Preparing the language model, which requires a text file that contains at least all transcriptions used in the experiment. In this study, this was done using Tanzil version of The Holy Qur'ān text. This file is referred to as the (corpus.txt) file.
(3) Preparing the acoustic set of files. These files include: The pronunciation dictionary (lexicon.txt), the following phoneme files; (nonsilence_phones.txt) file that contains all non-silence phonemes used in this research, (silence_phones.txt) file, and (optional_silence.txt) file.
(4) Preparing the configuration files for feature extraction and decoding steps. These files will specify some values of certain parameters, which can be used in performance tuning. Some of them might affect some stages of the ASR process.
(5) Preparing the execution environment and building the master shell script (run.sh), which will handle the whole ASR process. The shell script is executed sequentially, in order to achieve an incremental execution style, which will simplify code debugging (Netzer & Weaver, 1994). Another interesting reason for using this script file is to do parallel execution of some independent steps or some sub steps of a complex step such as training on a cluster of machines. In our research work, we used a customized version of the Wall Street Journal Corpus (wsj) KALDI recipe environment and (run.sh) script.

6 Running the experiments

6.1 Experimentation setup

Four different experiments were conducted to evaluate The Holy Qur'ān ASR system. All experiments were conducted using the steps described in Sect. 5.2 above, and use the same shell script file (run.sh), the same environment variables and the same hyper parameter, but with different dataset for training and testing as shown in Table 3 above.

6.2 Training methods applied

The training methods used in our research are discussed and explained in this section. It is important to highlight that the training phase has undergone five main passes, which are explained as follows:

(1) Monophone training pass: this is the first pass in the training sequence. The input for this pass is a set of feature vectors, which contains the Mel-frequency cepstral coefficients (MFCC) of sampled frames, Delta and Delta–Delta features along with the rest of files described before (transcription files, lexicon file, and phoneme files). The output of this pass is a monophone model, which is an acoustic model that does not contain any contextual information about the preceding or succeeding phonemes. This model is used as a starting point for the succeeding triphone models. An alignment pass is done on the output of this monophone training pass to correlate each audio frame (i.e. feature vector) in the input utterance to an HMM state (i.e. phoneme) in the transformed transcription (word to phoneme set transform) of that input utterance. Several alignment passes took place after the training process. In our research, 40 alignment passes were done by the monophone training shell script. Since the process of building the acoustic model is iterative, the subsequent training passes (i.e. the tri-phone passes) use the output of the monophone training pass, which make the output of the monophone training pass as the backbone and the starting pass of the succeeding training passes.
(2) First triphone training pass: this pass uses a pattern containing the current phone, the preceding and the succeeding phones instead of a single phone. This pass uses the output of the previous monophone alignment

pass. It requires specifying the number of HMM states (or leaves) on the decision tree parameter and the number of Gaussians parameter in the Gaussian Mixture Model (GMM). According to (Chodroff, 2018), the values of those two parameters are crucial to the experimental results and overall performance of the experiment. The suitable values for the two parameters are often chosen based on heuristics, which are related to the size of the training dataset. Hence, for every pass in each of the experiments, several training and testing sessions were attempted with different pairs of values for these critical parameters. The values of the parameters that achieved the best WER results were chosen for the final trial of the experiment for each dataset.

The training method in this pass uses Delta and Delta–Delta training algorithm, which captures dynamic properties of speech (Trabelsi & Ayed, 2012). After this pass another alignment pass will take place as explained in the second triphone training pass.

(3) Second triphone training pass: this pass uses Linear Discriminate Analysis (LDA)-Maximum Likelihood Linear Transform (MLLT) as a training method. The LDA takes the MFCC and builds HMM states. The feature space is reduced using dimensionality reduction capabilities of LDA (Erdogan, 2005). The MLLT takes the reduced feature space from the LDA and applies the linear simple transformation to obtain a unique transformation for each speaker. MLLT promotes speaker normalization through minimizing the differences among speakers (Axelrod et al., 2003). In addition, each frame is spliced with three preceding frames and three succeeding frames. Another alignment pass will be done after this training pass as explained in the next pass.

(4) Speaker Adaptive Training (SAT) pass: it is a triphone technique, which performs speaker and noise normalization to obtain ASR systems with high recognition accuracy (Miao et al., 2014). This pass actually applies LDA, MLLT, and SAT training methods. It is important to highlight that after this pass, any succeeding training processes should use the speaker's normalized features as input for the training process and not the original features. Consequently, there is a need for removing speaker identity from the set of features before starting any alignment process. This removal can be accomplished by estimating the speaker identity with the inverse of the Feature Space Maximum Likelihood Linear Regression (fMLLR) matrix, then removing it from the model by multiplying the inverse matrix with the feature vector.

(5) TDNN with Sub-sampling pass: One of the major differences between this pass and the previous passes is the need for a CUDA enabled Graphical Processing Unit (GPU) to run this pass in a reasonable time. In our research, the GPU used was Nvidia GeForce GTX 1660 TI, which is capable of producing more than 5 teraflops (TechPowerUp, 2019). Several parameters are passed as arguments to the main training script of this pass. In this research paper, the number of epochs parameter is chosen for further investigation and tuning purposes. Several trials took place for the third dataset by changing the value of the number of epochs parameter for each trial where the values range from 1 to 32. In different steps, WER, SER, and running time length were collected, tabulated, and presented in this paper. Although this pass takes the longest time period to complete that mainly depends on the number of epochs used, it achieves the best performance in terms of WER and SER. The model built as an output of this pass is the one we used in our ASR application for The Holy Qur'ān recitations.

7 Experimental results and discussion

Each of the aforementioned training passes is followed by a decoding graph building pass, which builds a decoding lattice for the testing dataset. Consequently, the decoding pass runs a scoring script to evaluate the WER and the SER performance measures for the whole experiment, and the WER is measured for each speaker and utterance separately. Several scoring passes took place and the best WER and SER results were obtained for the whole decoding process of each training pass, which are presented in this section.

To evaluate the performance of an ASR engine for The Holy Qur'ān recitations, two main measures are used. The first measure is the WER, which is calculated using Eq. (1) as follows (Bahl and Jelinek, 1975; Morris et al., 2004):

$$\mathrm{WER} = 100 * (\mathrm{I} + \mathrm{S} + \mathrm{D})/\mathrm{N} \tag{1}$$

where I is the insertion error, S is the substitution error, D is the deletion error, and N is the total number of words in the testing transcriptions.

The second measure is the SER, which is calculated using Eq. (2) as follows (Evermann, 1999):

$$\mathrm{SER} = 100 * (\mathrm{SE}/\mathrm{ST}) \tag{2}$$

where SE is the number of sentences that contain one or more I, S, or D errors, and ST is the total number of sentences in the testing transcriptions.

The experimental results of the first experiment using the first dataset (Set # 1) that has Separated Prolongation with

two counts long as the main Tajweed feature are shown in Table 4. However, the dataset used in this experiment was relatively small as shown earlier in Table 3. Therefore, the experimental results in terms of WER of the different passes were close to each other. Several trials took place before reaching the suitable parameters values. It was noticed that these values were relatively small. The TDNN pass was characterized by achieving the best WER and SER.

In addition, it was noticed that when the WER was close to all passes, the SER of the TDNN pass was the best among all as shown in Table 4.

The experimental results of the second experiment using the second dataset (Set # 2) that has Separated Prolongation with four counts long as the main Tajweed feature are shown in Table 5. However, the dataset used in this experiment was relatively small as shown earlier in Table III, but slightly larger than the dataset used in the first experiment. Thus, the experimental results in terms of WER was the best in the fourth pass using LDA + MLLT + SAT followed by the fifth pass using TDNN. Several trials took place before reaching the suitable values of the parameters. It was noticed that these values were relatively small but larger than those of the first experiment. The larger WER and SER were due to the existence of more noise in the recordings of this dataset. The TDNN pass in this experiment was also distinguished by achieving the best SER.

Table 6 shows the experimental results of the third experiment using the third dataset (Set # 3), which was the result of combining the recitations of (Set # 1) and (Set # 2) with seventeen other recitations to make up a relatively larger dataset with thirty-two recitations as shown earlier in Table III. However, the third dataset focused on training and testing of The Holy Qur'ān ASR engine and did not concentrate on Tajweed rules. Therefore, the experimental results of the different passes in terms of both WER and SER were clearly better. Several trials took place before reaching the suitable values of the parameters. It was noticed that these values were relatively larger than those of the previous experiments.

In addition, the obtained WER for all passes were low and acceptable due to the large size of this dataset. Furthermore, TDNN pass was distinguished by achieving the best WER and SER in this experiment.

Table 7 presents the experimental results of the fourth and last experiment using the fourth dataset (Set # 4) as shown in Table III. The objective of this experiment was to build an ASR engine for The Holy Qur'ān recitations. In this experiment, almost all available data was used for training and a small set was used for testing purposes. Hence, the experimental results of this experiment were the best for all passes. The ASR engine for The Holy Qur'ān recitations was tested and the experimental results in terms of both WER

Table 4 Experimental results of the separated prolongation with two counts long dataset

Training method	Best WER (%)	Best SER (%)	HMM states	Gaussians
Monophone	6.11	22.28	NA	NA
Triphone	6.39	24.73	80	350
LDA + MLLT	7.70	31.25	100	900
LDA + MLLT + SAT	6.32	26.63	65	650
TDNN	6.31	17.39	NA	NA

Table 5 Experimental results of the separated prolongation with four counts long dataset

Training method	Best WER (%)	Best SER (%)	HMM states	Gaussians
Monophone	13.03	38.12	NA	NA
Triphone	13.72	39.23	200	900
LDA + MLLT	11.28	35.64	250	2000
LDA + MLLT + SAT	4.23	19.61	200	2000
TDNN	5.40	15.19	NA	NA

Table 6 Experimental results of the THIRD DATASET

Training method	Best WER (%)	Best SER (%)	HMM states	Gaussians
Monophone	3.26	14.90	NA	NA
Triphone	2.81	12.30	700	7000
LDA + MLLT	2.79	13.20	500	5000
LDA + MLLT + SAT	1.69	8.90	800	8000
TDNN	0.27	0.40	NA	NA

Table 7 Experimental results of the FOURTH DATASET

Training method	Best WER (%)	Best SER (%)	HMM states	Gaussians
Monophone	2.24	11.30	NA	NA
Triphone	1.46	6.90	1000	6000
LDA + MLLT	1.18	5.80	500	5000
LDA + MLLT + SAT	0.44	1.30	3500	35,000
TDNN	0.27	0.40	NA	NA

Table 8 Number of epochs tuning results

No. of epochs	WER (%)	SER (%)	Time (MI)
1	0.61	2.10	85
2	0.29	0.60	95
3	0.28	0.50	101
4	0.27	0.40	118
5	0.27	0.40	132
6	0.27	0.40	137
8	0.27	0.40	159
16	0.27	0.40	245
32	0.27	0.40	417

and SER were excellent and were able to show the potential to achieve better performance in future attempts.

Several recitations of Sūrat Taha were collected from different sources and some were obtained from ordinary persons. These recitations were fed into ASR engine of The Holy Qur'ān recitations. The output was placed in a (.html) file and was displayed in a browser window.

Table 8 presents the experimental results of the 'Number of Epochs' parameter tuning trials. Initially, the performance in terms of both WER and SER were enhanced gradually. However, at a later stage the performance reached a steady state regardless of the number of epochs used. Readings of trial time in minuets were also collected and presented in Table 8.

8 Conclusions and future work

The Holy Qur'ān is considered as the main spiritual source and reference for more than 1.5 billion Muslims all over the world. The Holy Qur'ān can be recited in ten recitations (Qiraat). Each recitation (Qiraah) possesses certain acoustic features and characteristics that can be discriminated using Tajweed rules.

In this paper the effects of Tajweed rules on the ASR process and its components were addressed and discussed. This paper described the process of developing a large-vocabulary speaker-independent continuous speech recognizer for The Holy Qur'ān recitations based on the narration of Hafs from A'asim, using state-of-the-art ASR evolutionary approaches. In addition, this paper presented the process of preparing The Holy Qur'ān speech corpus, which was used to construct the datasets used in the experimental work. Four experimental setups were evaluated using the deep learning KALDI ASR toolkit, where these setups were different in terms of dataset size and Tajweed rules. Based on our experimental work, the best experimental setup was based on TDNN with sub-sampling technique, which obtained a WER in the range of (0.27–6.31%) and a SER in the range of (0.4–17.39%). Therefore, the experimental results are very promising and they indicate that the newly developed speech recognizer is able to recognize The Holy Qur'ān recitations based on the narration of Hafs from A'asim.

Tuning the training parameters for the different stages of the training process took place, where most of these parameters are of heuristic nature. Therefore, several experiments were conducted to find the optimal values of these parameters. These optimal values were used to train a simple speech-to-text application, which was tested and was able to achieve acceptable results and performance.

It is worth mentioning that the training process builds a model, which will be used in the decoding process of several utterances (i.e. speech-to-text conversion). Therefore, it is a build once and use many times process. This fact can justify the time spent in building this model, the use of expensive hardware components like GPUs, and the time spent in ASR hyper parameters tuning process. In addition, for relatively short utterances, the decoding process using this pre-built model does not need hardware resources with large capabilities. It can be achieved in reasonable time (i.e. few seconds) even on virtual machines.

Future research direction will focus on expanding the corpus to incorporate more Chapters (Sūras) of The Holy Qur'ān and consider other Qur'ānic narrations such as Warsh and Qaloon. We also intend to conduct experiments to compare the different narrations and differentiate between them based on acoustic features.

References

Abudena, M. A. (2015). Proposal to encode Quranic marks used in Quran published in Libya. L2/15-329, Complete UTC Document Register.

Abushariah, M. A. (2017). TAMEEM V1. 0: Speakers and text independent Arabic automatic continuous speech recognizer. *International Journal of Speech Technology, 20,* 261–280.

Abushariah, M. A., Ainon, R. N., Zainuddin, R., Elshafei, M., & Khalifa, O. O. (2012). Phonetically rich and balanced text and speech corpora for Arabic language. *Language Resources and Evaluation, 46,* 601–634.

Abushariah, M. A., Ainon, R. N., Zainuddin, R., Khalifa, O. O. & Elshafei, M. (2010). Phonetically rich and balanced Arabic speech corpus: an overview. In *International conference on computer and communication engineering* (ICCCE'10) (pp. 1–6). IEEE.

Adda-Decker, M. & Lamel, L. (2006). Multilingual dictionaries. In Schultz, T., Kirchhoff, K. (Eds.), *Multilingual Speech Processing* (pp. 123–168).

Al-Ghamdi, M. M., Al-Muhtasib, H., & Elshafei, M. (2004). Phonetic rules in arabic script. *Journal of King Saud University-Computer and Information Sciences, 16,* 85–115.

Ali, M., Elshafei, M., Al-Ghamdi, M., Al-Muhtaseb, H. & Al-Najjar, A. (2008). Generation of Arabic phonetic dictionaries for speech recognition. In *2008 International conference on innovations in information technology* (pp. 59–63). IEEE.

Al-Imam, A. A. (2006). *Variant readings of the Qur'an: A critical study of their historical and linguistic origins*, International Institute of Islamic Thought (IIIT).

Allauzen, C., Riley, M., Schalkwyk, J., Skut, W. & Mohri, M. (2007). OpenFst: A general and efficient weighted finite-state transducer library. In *International conference on implementation and application of automata* (pp. 11–23). Springer.

Alqudah, A. A. M., Alshraideh, M. A. M., & Sharieh, A. A. S. (2020). Arabic disordered speech phonetic dictionary generator for automatic speech recognition. *Journal of Theoretical and Applied Information Technology, 98,* 571–586.

Alsulaiman, M., Mahmood, A., & Muhammad, G. (2017). Speaker recognition based on Arabic phonemes. *Speech Communication, 86,* 42–51.

Alsunaidi, N., Alzeer, L., Alkatheiri, M., Habbabah, A., Alattas, M., Aljabri, M., et al. (2018). Abjad: Towards interactive learning approach to arabic reading based on speech recognition. *Procedia Computer Science, 142,* 198–205.

Axelrod, S., Gopinath, R., Olsen, P. & Visweswariah, K. (2003). Dimensional reduction, covariance modeling, and computational complexity in ASR systems. In 2003 IEEE international conference on acoustics, speech, and signal processing, 2003. Proceedings.(ICASSP'03) (pp. I–I). IEEE.

Bahl, L., & Jelinek, F. (1975). Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition. *IEEE Transactions on Information Theory, 21,* 404–411.

Bellegdi, S. A. & Al-Muhtaseb, H. A. (2015). Automatic rule based phonetic transcription and syllabification for quranic text. Unpublished.

Besacier, L., Barnard, E., Karpov, A., & Schultz, T. (2014). Automatic speech recognition for under-resourced languages: A survey. *Speech Communication, 56,* 85–100.

Bezoui, M., Elmoutaouakkil, A. & Beni-Hssane, A. (2016). Feature extraction of some Quranic recitation using mel-frequency cepstral coefficients (MFCC). In *2016 5th international conference on multimedia computing and systems (ICMCS)* (pp. 127–131). IEEE.

Braun, H., Luitjens, J. & Leary, R. (2019). GPU-accelerated Viterbi exact lattice decoder for batched online and offline speech recognition. arXiv preprint arXiv:1910.10032.

Chodroff, E. (2018). Corpus phonetics tutorial. arXiv preprint arXiv:1811.05553.

Cosi, P. (2015). A kaldi-dnn-based asr system for italian. In *2015 international joint conference on neural networks (IJCNN)* (pp. 1–5). IEEE.

Czerepinski, K. C., & Swayd, A.-S. D. A. R. (2006). *Tajweed rules of the Qur'an*. Dar Al-Khair Islamic Books Publisher.

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,* 357–366.

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,* 305–314.

Elhadj, Y. O. M., Aoun-Allah, M., Alsughaiyer, I. A. & Alansari, A. (2012). In A. Silva & E. Pontes (Eds.), A new scientific formulation of Tajweed rules for E-learning of Quran phonological rules (p. 197).

Elrefaei, L. A., Alhassan, T. Q., & Omar, S. S. (2019). An Arabic visual dataset for visual speech recognition. *Procedia Computer Science, 163,* 400–409.

Elshafei, M. (1991). Toward an Arabic text-to-speech system. *The Arabian Journal for Science and Engineering, 16,* 565–583.

Erdogan, H. (2005). Regularizing linear discriminant analysis for speech recognition. In Ninth European conference on speech communication and technology.

Evermann, G. (1999). *Minimum word error rate decoding* (pp. 45–67). Cambridge University.

Hafeez, A. H., Mohiuddin, K. & Ahmed, S. (2014). Speaker-dependent live quranic verses recitation recognition system using Sphinx-4 framework. In 17th IEEE international multi topic conference 2014 (pp. 333–337). IEEE.

Jurasky, D., & Martin, J. H. (2000). *Speech and language processing: An introduction to natural language processing*. Computational Linguistics and Speech Recognition.

KALDI_TEAM_CUDA. (2020). *The CUDA matrix library* [Online]. Retrieved April 28, 2020, from https://kaldi-asr.org/doc/cudamatrix.html.

KALDI_TEAM_MKL. (2020). *External matrix libraries* [Online]. Retrieved April 28, 2020, from https://kaldi-asr.org/doc/matrixwrap.html.

Khan, A. F. A., Mourad, O., Mannan, A. M. K. B., Dahan, H. B. A. M., & Abushariah, M. A. (2013). Automatic Arabic pronunciation scoring for computer aided language learning. 2013 1st international conference on communications, signal processing, and their applications (ICCSPA) (1–6). IEEE.

Khelifa, M. O., Elhadj, Y., Abdellah, Y., & Belkasmi, M. (2017). Strategies for implementing an optimal ASR system for quranic recitation recognition. *International Journal of Computer Applications, 172,* 35–41.

Kirchhoff, K., & Vergyri, D. (2005). Cross-dialectal data sharing for acoustic modeling in Arabic speech recognition. *Speech Communication, 46,* 37–51.

Lawson, D. R. (2008). An evaluation of arabic transliteration methods.

Magre, S. B., Deshmukh, R. R. & Shrishrimal, P. P. (2013). A comparative study on feature extraction techniques in speech recognition. In International conference on recent advances in statistics and their application.

Mahmod, M. A. (2016). Automated quranic Tajweed checking rules system through recitation recognition: a review.

Malmasi, S. & Zampieri, M. (2017). Arabic dialect identification using iVectors and ASR transcripts. In *Proceedings of the fourth workshop on NLP for similar languages, varieties and dialects (VarDial)* (pp. 178–183).

Mcauliffe, J. D. (2006). *The Cambridge companion to the Qur'ān*. Cambridge University Press.

Miao, Y., Zhang, H., & Metze, F. (2014). Towards speaker adaptive training of deep neural network acoustic models. In Fifteenth annual conference of the international speech communication association.

Morris, A. C., Maier, V. & Green, P. (2004). From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition. In *Eighth international conference on spoken language processing*.

Nasr, S. H., Dagli, C. K., Dakake, M. M., Lumbard, J. E. & Rustom, M. (2015). The study Quran. A new translation and commentary. New York.

Netzer, R. H., & Weaver, M. H. (1994). Optimal tracing and incremental reexecution for debugging long-running programs. *ACM SIGPLAN Notices, 29,* 313–325.

Peddinti, V., Povey, D. & Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. In *Sixteenth annual conference of the international speech communication association*.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y. & Schwarz, P. (2011). The Kaldi speech recognition toolkit. IEEE 2011 workshop on automatic speech recognition and understanding. In *IEEE Signal Processing Society*.

Stolcke, A. (2002). SRILM-an extensible language modeling toolkit. In *Seventh international conference on spoken language processing*.

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* (pp. 2979–2984). IEEE.

Tanzil. (2020). *Tanzil documents* [Online]. Retrieved May 29, 2020, from http://tanzil.net/docs/download.

Techpowerup. (2019). *NVIDIA GeForce GTX 1660 Ti* [Online]. Retrieved October 27, 2019, from https://www.techpowerup.com/gpu-specs/geforce-gtx-1660-ti.c3364.

Trabelsi, I. & Ayed, D. B. (2012). On the use of different feature extraction methods for linear and non linear kernels. In *2012 6th international conference on sciences of electronics, technologies of information and telecommunications (SETIT)* (pp. 797-802). IEEE.

Yousfi, B. & Zeki, A. M. (2017). Holy Qur'an speech recognition system Imaalah checking rule for warsh recitation. In *2017 IEEE 13th international colloquium on signal processing & its applications (CSPA)* (pp. 258–263). IEEE.

Yousfi, B., Zeki, A. M., & Haji, A. (2018). Holy Qur'an speech recognition system distinguishing the type of prolongation. *Sukkur IBA Journal of Computing and Mathematical Sciences, 2,* 36–43.

Zakariah, M., Khan, M. K., Tayan, O., & Salah, K. (2017). Digital Quran computing: review, classification, and trend analysis. *Arabian Journal for Science and Engineering, 42,* 3077–3102.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.