IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

qArI: A Hybrid CTC/Attention-Based Model for Quran Recitation Recognition using Bidirectional LSTMP in an End-to-End Architecture

SUMAYYA ALFADHLI[1,2], HAJAR ALHARBI[3], and ASMA CHERIF[4,5]

[1]Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia (e-mail: shassenalfadhli@stu.kau.edu.sa)
[2]Department of Computer Science, Adham University College, Umm Al-Qura University, Makkah, Saudi Arabia(e-mail: safadly@uqu.edu.sa)
[3]Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia (e-mail: hmsalharbi@kau.edu.s)
[4]Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia (e-mail: acherif@kau.edu.sa)
[5]Center of Excellent in Smart Environment Research, King Abdulaziz University, Jeddah, Saudi Arabia

Corresponding author: First A. Sumayya Alfadhli (e-mail: shassenalfadhli@stu.kau.edu.sa).

ABSTRACT The accurate speech recognition of the Holy Quran is crucial for maintaining the traditional recitation styles and pronunciations, which helps in preserving the authenticity of the Quranic teachings and ensuring their accurate transmission across generations. Though the application of freshly developed models to spoken and written Arabic and non-Arabic speech recognition has yielded highly accurate results, research on Holy Quran is still in its early levels. Indeed, speech recognition of the Holy Quran presents several challenges, including language complexity and the absence of a comprehensive dataset. This research aims to improve the accuracy of speech recognition models for the recital of the Holy Quran. A new dataset called comprehensive Quranic dataset version 1 (CQDV1) is created to serves the HQSR field. The dataset is publicly available for use by other researchers and includes recitations of the entire Quran (114 sura, recited by 35 reciters), based on Hafs from Asim narrative.The study explores the development of a speech recognition model for the accurate recital of the Holy Quran. The model combines a connectionist temporal classification (CTC)/attention loss function with a Bidirectional Long Short-Term Memory with projections (BLSTMP) architecture and a token-based recurrent neural network language model (RNNLM) using CQDV1 dataset. The results achieved were a token error rate (TER) of 6.4%, a word error rate (WER) of 10.4%, and a sentence error rate (SER) of 55.3% with $\lambda = 0.2$.

INDEX TERMS Acoustic Models, Attention, bidirectional LSTMP, CTC, Language Model, Quran Recitation, Speech Recognition

I. INTRODUCTION

For decades, engineers and scientists have struggled to create a machine that can emulate human behavior, including natural speech and understanding spoken language [1]. This process, known as automatic speech recognition (ASR), involves translating sound waves, or acoustic speech signals, into words or other language units using specific algorithms [2], [3]. The ultimate goal of speech recognition is to enable machines to recognize and respond to sounds [4]. Speech recognition is highly beneficial in the educational sector, as it enables the creation of reliable automatic language correctors. This is illustrated by the English pronunciation learning model developed by [5] for Chinese learners.

The researchers have made significant contributions to various voice processing disciplines for diverse human languages worldwide. It is estimated that 420 million people speak Arabic [6]. The Arabic language can be categorized into three classes. First, the class of Modern Standard Arabic (MSA) is offered in educational institutions and follows the grammatical standards of Arabic. Secondly, Arabic Dialect (AD) class is the language used by native Arabic speakers on a daily basis. It varies throughout nations and even within

the same region. Lastly, the Classical Arabic (CA) class has been well-known worldwide for centuries due to its association with the Holy Quran. This language has very specific recitation rules, but it also has an extensive vocabulary and grammar [7], [8].

The application of voice recognition to Quranic recitation has emerged as a significant area of study in recent years. With over two billion Muslims worldwide [9], it is essential for every Muslim to learn how to recite the Quran accurately and adhere to the regulations of Tajweed. Acquiring knowledge of these guidelines is crucial for proficiency in reciting the Holy Quran Tajweed. Acquiring knowledge of these guidelines is crucial for every Muslim to become proficient in reciting the Holy Quran [7]. Additionally, certain individuals choose to recite from memory to uphold their faith during prayer, especially during night prayer, without relying on reading from the Mushaf. An expert, often referred to as a gari, plays a vital role in teaching Quranic recitation. He/she listens to the learner recite the passage, identifies errors, and provides necessary corrections. While this method of instruction is highly effective, it can be time-consuming as each student's mistakes need to be addressed individually. Indeed, committing the Quran to memory requires a lengthy and continuous process of review. Also, garis encounter challenges in listening and approving lengthy recitations by multiple students. Moreover, certain non-Arabic countries, particularly those with a lower proportion of Muslims, encounter a shortage of qualified teachers to instruct the Holy Quran. Consequently, the development of precise Holy Quran Speech Recognition (HQSR) models has become an important research objective for Muslims globally. Consequently, applications that assist students in memorizing/perfecting their recital of the Holy Quran are both advantageous and indispensable. However, these applications require a robust and accurate speech recognition model. Although contemporary models have been applied to both written and spoken Arabic, as well as non-Arabic recognition, progress in Quran speech recognition remains in its early stages. While there has been limited research on the topic, speech recognition technology for the Holy Quran has not yet produced optimal results. Indeed, these applications require a strong speech recognition model of the Holy Quran with a minimum token error rate (TER), word error rate (WER), and sentence error rate (SER).

Reciting the Quran does not allow for mistakes, as even a single mispronounced letter can change the intended meaning. While it may be straightforward to identify individual words, recognizing continuous recitation and detecting improper recitation and violations of Tajweed rules can pose challenges. Indeed, while reciting the Holy Quran, it is imperative to follow the Tajweed' principles and precisely pronounce the Sifaat (characteristics) and Makhraj (point of articulation) of each alphabet. However, the recognition model may encounter difficulties in accurately recognizing the recitation because of differences between narrations in some Tajweed rules, $e.g.$, the duration of Madd ($i.e.$, prolongation) in the Quranic recitations/narrations. For instance, reciters in the Hafs from Asim narrative have the ability to recite specific types of madd using two, four, five, or six Harakat. In addition, identifying the recitation in the Quran is difficult due to the wide variety of Magam present, including bayat, Ajam, Nahawand, Hijaz, Rost, Sika, and others. Additional factors, including speaker reliance, vocabulary size, and noisy environments, can influence the effectiveness of speech recognition models. When utilizing a large vocabulary and reciter-independent scenarios, the performance of recognition can significantly decline. Conversely, when employing a restricted vocabulary and reciter-dependent settings, the performance tends to improve [3]. However, it is important to note that this can introduce bias into the results.

Furthermore, most research advancements in the field focus on a single or a small number of chapters or Tajweed rules. Another drawback of previous HQSR research is the lack of large datasets. Indeed, there is a lack of a comprehensive dataset encompassing accurate recitations of the Holy Quran by women, children, as well as native and non-native Arabic speakers. Additionally, most contemporary research has overlooked the exploration of end-to-end learning, instead relying on traditional methods. The main goal of this project is to utilize advanced machine learning techniques for Holy Quran recitation, addressing these limitations and advancing the field of HQSR.

To address the existing research gaps in HQSR, this study aims to develop a comprehensive and robust system that encompasses a wide range of chapters and Tajweed rules. Furthermore, we will explore the potential of end-to-end (E2E) learning techniques to enhance the accuracy and efficiency of the recognition system. This work's primary contributions are:

- Creating a new dataset called comprehensive Quranic dataset version 1 (CQDV1) to the HQSR field. The dataset contains recitations of the whole Quran (114 sura, recited by 35 reciters) and is publicly available on Kaggle [1] and IEEE Dataport [2].
- Using an E2E speech recognition architecture with a language model instead of a traditional speech recognition architecture. This research represents one of the initial endeavors to employ an End-to-End (E2E) speech recognition architecture for Quran recitation recognition.
- Identifying the precise error type (substitution, deletion, or insertion) at both the word and token levels, and pinpointing the exact location of the error by comparing the predicted text with the reference text.

The remainder of this paper is structured as follows: Section II provides an overview of the related work in the field. Subsequently, Section III presents the architecture of the proposed solution. Section IV then outlines the experimental setup in detail. Section V presents the experiment results and

[1] https://www.kaggle.com/datasets/quranicdataset/quranic-dataset-v1
[2] http://ieee-dataport.org/12554

discusses the limitations of the current literature and how this solution addresses the existing gaps. Finally, Section VI concludes the paper and suggests directions for future research.

II. RELATED WORK

Several research investigations have been carried out in the realm of speech recognition for both Arabic and non-Arabic languages. This section highlights notable speech recognition technologies that have yielded impressive results in both Arabic and non-Arabic languages, as well as recent advancements in the field of HQSR.

The authors of [10] examined 80 more subjective Arabic language databases in addition to the 27 publicly accessible databases they had found. When it comes to research in Arabic and non-Arabic languages, there are two main categories of solutions: traditional and E2E speech recognition systems. Traditional ASR systems involve training separate components for acoustic, pronunciation, and language models [11]. The Acoustic Model (AM) calculates the likelihood of acoustic elements, such as phonemes or graphemes, based on the audio input. The Language Model (LM) calculates the probability of word sequences by using linguistic data from large text corpora. The Pronunciation Dictionary (PD) matches phonetic transcriptions with unprocessed text. The integration of these components with Finite-State Transducers (FSTs) results in the creation of a search graph. [12]

Unlike traditional speech recognition, E2E speech recognition is a system that directly converts a set of input audio features into a sequence of graphemes or words. This approach significantly reduces the complexity of traditional speech recognition. The neural network has the ability to automatically learn language and pronunciation information without the need for explicit labeling [11]. End-to-end speech recognition commonly employs encoder-decoder technology. This architecture converts an audio file into a condensed vector by processing it through a sequence of convolutional layers, as outlined in [12], [13]. The decoder then takes this encoded vector as input and generates a sequence of characters. The system's performance can be enhanced by incorporating an external LM.

Recently, researchers have proposed E2E-based approaches for speech recognition in both Arabic and non-Arabic languages. In the context of the Arabic language and its numerous dialects, [14] conducted a comprehensive study that compares human speech recognition (HSR), modular Hidden Markov Model-Deep Neural Network (HMM-DNN) ASR, and E2E transformer ASR. To avoid biases in their study, the researchers collected a new assessment set consisting of three hours of conversational speech and news stories, including both MSA and DA. Additionally, they utilized well-known datasets such as MGB2, MGB3, and MGB5 to further enhance the comprehensiveness of their research. They conducted a comprehensive mistake analysis that compared the performance of the ASR system to that of a native speaker and professional linguist. The findings revealed that while the machine ASR system likely outperforms a native speaker, the raw Arabic transcription text still falls short by an average WER gap of 3.5% in achieving the efficiency of an expert linguist. The E2E transformer model demonstrated exceptional performance on the three datasets: MGB2, MGB3, and MGB5, thus outperforming previous state-of-the-art results and achieving new benchmarks of 12.5%, 27.5%, and 33.8%, respectively. In their work, [15] introduced ESPnet, an open-source platform designed for end-to-end speech processing. ESPnet serves as a powerful deep learning engine that seamlessly integrates with Chainer and PyTorch, supporting the entire ASR pipeline, from training to recognition. It follows the style of the Kaldi ASR toolkit [16] [3] for speech recognition and processing tasks, including feature extraction, data processing, and scheme development, thus offering a widely used comprehensive framework. Extensive testing has shown that ESPnet can attain performance levels that are comparable to those of state-of-the-art HMM-DNN systems with standard configurations, resulting in exceptional ASR results. Moreover, ESPnet boasts strong multi-GPU capabilities, allowing for efficient training on multiple GPUs. Impressively, ESPnet completed the training of the Corpus of Spontaneous Japanese (CSJ) assignment, which consisted of 581 hours of data, in just 26 hours. The review by [17] compares standard traditional methods with the end-to-end architecture described in some works that utilized the same datasets. It demonstrates that the majority of the research suggests that the end-to-end design outperforms the traditional architecture.

In the realm of research on speech recognition of the Holy Quran, previous studies can be classified into three groups: Template-Based HQSR, Traditional-Based HQSR, and E2E-Based HQSR.

Template-based models. The articles that used template-based speech recognition, a template method for speech recognition that only used feature extraction (FE), classification, and matching techniques (i.e., without lexical, acoustic, and language models), were summarized in this section. [18] suggested an online method for verifying Quranic phrases in order to protect the Quran's authenticity and integrity toward any corruption. They employ information gathered from 10 qualified Qari (they possess Ejazah in Hafs). The authors recited Surat Al-Nass ten times, with each person reciting it correctly ten times and incorrectly ten times. This exercise should include various challenges, such as Tajweed errors, Makhraj mistakes, and missing words. This study utilized Mel-frequency cepstral coefficients (MFCC) for extracting features and employed Hidden Markov Models (HMMs) for the recognition and matching process. It relies on the traditional approach to speech recognition (i.e., no lexical, linguistic, or auditory models). No testing data was provided for this study. The authors of [19] aimed

[3] Kaldi, an open-source toolkit, offers a speech recognition system based on FSTs and provides comprehensive documentation and scripts for building recognition systems with various features.

to analyze and identify classical Arabic vocal phonemes, particularly vowels, using HMM. Their goal was to tackle the issue of semantic changes resulting from variations in vowel durations in Arabic (short or long). In their research, the authors focused on three specific chapters (Alfateha, Albaqarah, and Alshuraa) from the Holy Quran. The study's outcomes showed a considerable overall accuracy rate of 87.60% without utilizing a particular language model. In a recent study by [20], the authors presented a deep learning model that utilizes a dataset of seven well-known reciters and Convolutional Neural Networks (CNNs). The model employs Mel-frequency cepstral coefficients (MFCCs) to extract and analyze data from audio sources. The proposed model achieved an impressive accuracy rate of 99.66%. They employ MFCCs to extract and assess data from audio sources. Their provided model achieved 99.66% accuracy. [21] developed an algorithm that can identify Ahkam Al-Tajweed in a particular audio clip of a Quranic recital. They recognize eight Ahkam Al-Tajweed: "EdgamMeem" (one rule), "EkhfaaMeem" (one rule), "Ahkam Lam" in the term "Allah" (two rules), and "Edgam Noon" (four rules). Additionally, they take into account both proper and improper application of each regulation. The entire Holy Quran is covered by their categorization issue, with 16 classes that only includes the passages that contain the eight rules considered in this study. The researchers employed a variety of feature extraction techniques, including traditional methods like Linear Predictive Code (LPC) and MFCC, as well as contemporary approaches like Convolutional Deep Belief Network (CDBN) and classifiers like Support Vector Machines (SVM) and Random Forest (RF). Using SVM for classification and MFCC, Wavelet Packet Decomposition (WPD), Markov Model based Spectral Peak Location (HMM-SPL), and CDBN for feature extraction, they achieved the highest accuracy of 96.4%.

The authors of [22] developed a novel system using deep learning to identify the correct recitation of individual alphabets, words from a recited verse, and a complete verse of Al-Quran to assist the reciter. They employ MFCC to extract voice features and LSTM for classification. The proposed approach incorporates the user's voice into the existing dataset upon correct recitation, thereby enhancing its effectiveness. The results demonstrate that the proposed system outperforms the state-of-the-art approaches with an accuracy rate of 97.7%.

Traditional-based models.

[23] use the Carnegie Melon University (CMU) Sphinx [24] trainer of 49 chapters (surah) and 39 different reciters for building the acoustic model. Accurate acoustic models are produced using the reliable continuous speech recognition framework, CMU Sphinx, which operates independently of the speaker. The trained acoustic model's WER was around 15%. Using the Kaldi toolkit, [25] created, developed, and assessed a large-vocabulary speaker-independent continuous speech recognition system. For Chapter 20 (Sūrat Taha), they employ 32 recitations, as per Hafs from the narrative of Asim. With a sub-sampling approach, the optimal trial design utilizes Time Delay Neural Networks (TDNN) to achieve a SER in the range of 0.4% and 17.39% and a WER in the range of 0.27% and 6.31%. The researchers in [26] utilized a deep learning methodology to construct an acoustic model specifically designed for recognizing Quranic speech. The hybrid Hidden Markov Model-Bidirectional Long-Short Term Memory (HMM-BLSTM) technique demonstrated superior speech recognition accuracy in comparison to the Hidden Markov Model-Gaussian Mixture Model (HMM-GMM) method, as determined by the study's examination of 13 distinct reciters. The initial models, specifically the HMM-GMM models, obtained an average WER of 18.39%. Conversely, the acoustic model employing Hybrid HMM-BLSTM had notably superior outcomes, with an average WER of 4.63% under identical testing conditions. The researchers of [27] used MFCC for feature extraction, and then adjusted these features using the minimal phone error (MPE) as a discriminative model. They employed a deep neural network (DNN) model to construct the acoustic model and introduced an n-gram language model. The proposed model was trained and evaluated using a dataset of 10 hours of .wav recitations performed by 60 reciters. The experimental results showed that the proposed DNN model achieved a very low character error rate (CER) of 4.09% and a word error rate (WER) of 8.46%.

E2E-based models. Recently, few studies investigated using End-to-End (E2E) for Quran recitation recognition. In [28], researchers utilized the full transformer model to establish a robust Quran recognition system. They constructed the acoustic model using the PyTorch framework, implementing the encoder and decoder through a multi-head attention mechanism. For feature extraction, they employed a Mel filter bank. Additionally, RNNs and LSTMs were used to train an n-gram word-based LM to construct a language model. To facilitate the training and evaluation of their proposed model, the researchers gathered and processed a new dataset of Qur'anic verses and their corresponding transcripts, comprising 10 hours of .wav recitations by 60 reciters. The suggested end-to-end transformer-based model achieved a significantly low character error rate (CER) of 1.98% and a word error rate (WER) of 6.16% based on the testing data. In another study by [29], the authors proposed an end-to-end deep learning model for recognizing the recitation of the Holy Quran. Their model, a CNN-Bidirectional GRU encoder, employed CTC as an objective function and a character-based decoder employing a beam search decoder. They evaluated the performance of the proposed model using the word error rate (WER) and character error rate (CER). The evaluation was conducted on a public dataset (Ar-DAD) containing approximately 37 chapters recited by 30 individuals, encompassing varying recitation speeds and pronunciation rules. The results demonstrated a WER of 8.34% and a CER of 2.42%. Finally, the authors of

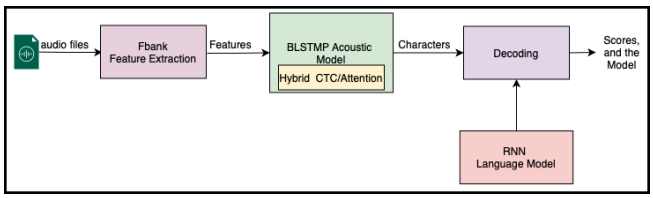


FIGURE 1: Proposed Solution Architecture

[30] developed three models for Arabic speech recognition: transformer, recurrent neural network (RNN)-CTC, and time delay neural network (TDNN)-connectionist temporal classification (CTC). The authors sourced the Quran recordings from a 100-hour collection. According to the data, the RNN-CTC model performed exceptionally well, with a character error rate of 3.51% and the lowest word error rate of 19.43%. The RNN-CTC model's character-by-character recognition is more reliable than transformers' overall sentence recognition performance.

III. PROPOSED SOLUTION ARCHITECTURE

Figure 1 illustrates the basic components of the E2E-based proposed solution for speech recognition. The architecture comprises key elements such as feature extraction, an acoustic model, a language model, and decoding. The fundamental input to the architecture consists of audio files from the dataset, while the primary output is the proposed model. Our suggested architecture uses fbank feature extraction and BLSTMP encoder coupled with a hybrid architecture for the loss that uses CTC and attention, and fed with a token-based language model. In the following sections, we will provide a detailed explanation of each part.

A. FEATURE EXTRACTION

The most crucial step in the ASR process is feature extraction, which involves extracting useful data from speech. In this research, we employ fbank feature extraction techniques, which, despite MFCC feature extraction being more popular, yield better results in ASR [31]. FBank features maximize the impact of the speech recognition system by emulating how the human ear perceives sounds [32]. We employ an 80-dimensional filter bank with a sample frequency of 22050 and further normalize it using Cepstral mean and variance normalization (CMVN), a method that is extensively used in speech recognition for noise reduction.

B. ACOUSTIC MODEL

Once features are extracted they are passed to the encoder which is the main component of our system. The encoder network, as illustrated in Figure 2, consists of a hybrid Connectionist temporal classification (CTC)-Attention architecture for E2E speech recognition and a shared RNN encoder. This framework models the transcription between the input normalized feature sequence $x = (x_1, x_2, \ldots, x_t)$ and the output symbol sequence $y = (y_1, y_2, \ldots, y_z)$. In end-to-end speech recognition systems, the output label sequence is

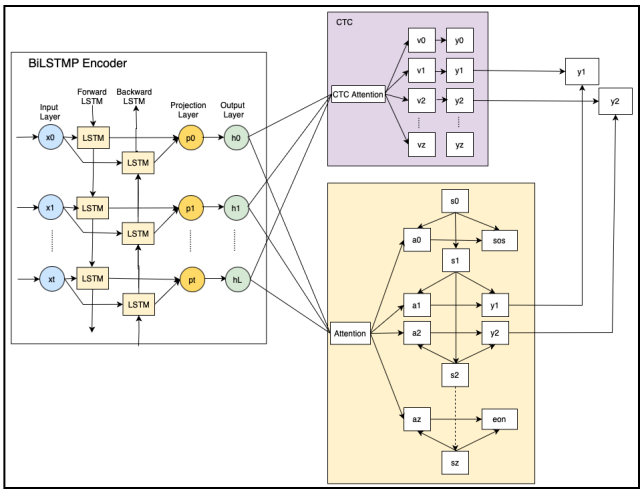


FIGURE 2: Training Architecture

typically shorter than the feature sequence (i.e., $z < t$). The hybrid CTC-Attention architecture incorporates bidirectional long short-term memory (BLSTM) with subsampling, known as pyramid BLSTM [15]. The bidirectional LSTM architecture consists of two unidirectional LSTMs that process the sequence in both forward and backward directions. This design can be viewed as having two independent LSTM networks, with one receiving the token sequence in its original order and the other in reverse. The projection layer then projects the combined probability from the outputs of these LSTM networks, resulting in a high-level representation $h = h_1, h_2, \ldots, h_L$ for the input sequence x. Here, L is the downsampled frame index.

$$h = Encoder(x) \tag{1}$$

Next, leveraging the high-level feature h as a basis, a CTC model and an attention-based decoder simultaneously create targets [33]. A projected layer, a type of deep learning layer, facilitates compression by reducing the number of stored learnable parameters. The output size of the layer and the sizes of the downstream layers are preserved by projecting an LSTM layer instead of reducing the number of hidden units, which may result in enhanced prediction accuracy [34].

In our model, the CTC loss is defined as the negative log probability of the output label sequence. The definition of the CTC loss is as follows:

$$L_{CTC} = -ln(p(y|x)) \tag{2}$$

The attention-based layer is an RNN that uses the attention mechanism to transform the output label sequence from the high-level features h produced by the shared encoder. With the input feature h and the prior labels $y_{1:z-1}$. the decoder uses the chain rule to calculate the likelihood of the label sequence based on the conditional probability of the label y_Z, where the LSTM function is implemented as a bidirectional LSTM layer in this instance.

$$P(y|x) = \prod_{z} p(y_z|h, y_{1:z-1}) \tag{3}$$

A unique start-of-sequence symbol (sos), and an end-of-sequence symbol (eos) have been introduced to the output sequence by the attention-based layer module. It halts the creation of new output labels upon the emission of (eos). Lastly, the negative log likelihood of the target sequence is used to determine the attention loss [33].

The CTC loss and attention loss can be coupled in order to benefit from both models. Figure 2 depicts the hybrid model's general design. Both attention-based and CTC approaches have disadvantages of their own. Because CTC assumes conditional independence between the labels, a robust external language model is necessary to offset the long-term reliance between the labels. The attention system, which may be guided by alignments, generates each output by taking the weighted total of all the input without any restrictions. As a result, training the attention-based decoder is typically challenging. It should be noted that the CTC forward-backward method can discover a monotonic alignment between label sequences and acoustic characteristics, which may speed up the encoder's convergence. Additionally, the target sequence's dependencies can be learned by the attention-based decoder. Consequently, the hybrid model is able to utilize label dependencies and contribute to the convergence of the attention-based decoder by integrating CTC and attention loss [33]. A weighted total of CTC loss and attention-based loss is the definition of the hybrid CTC-Attention objective:

$$L_{hybrid} = \lambda L_{ctc} + (1 - \lambda) L_{Attention} \tag{4}$$

where $\lambda \in (0, 1)$ is a tunable hyper-parameter.

C. LANGUAGE MODEL

The language model P($Token$) is defined to ascertain the probability of various sequences of tokens that a speaker may utter, by training on text data, taking into account the vocabulary and the probability distribution across possible sequences. The language model provides a probability estimate for token sequences [35], [36]. Token-based RNNLM is employed in this study, as Figure 3 illustrates. The token list is illustrated in Figure 4. We employ a compression technique called Byte-Pair Encoding (BPE) [37], commonly used in Natural Language Processing (NLP), to represent a large vocabulary using a limited number of subword units. BPE has found extensive applications in various NLP domains, including speech recognition, text categorization, machine translation, and text synthesis [38].

D. DECODING

The most probable transcription of an audio signal is generated by utilizing the RNNLM and E2E acoustic models obtained in the previous subsections to perform decoding through beam search. Additionally, the scoring metrics, including TER, WER, SER, and accuracy, are calculated, respectively, according to the following equations (see Equations 5-9).

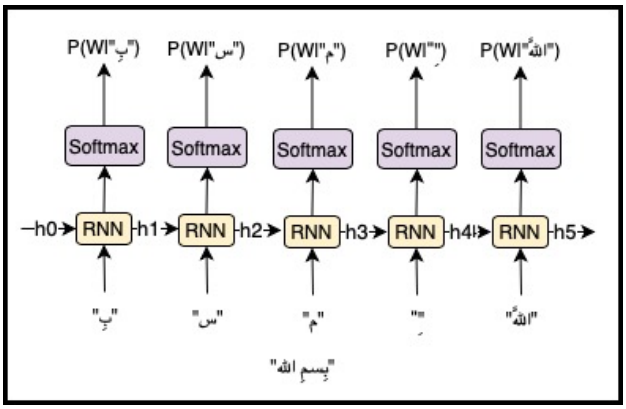


FIGURE 3: Language Model Architecture

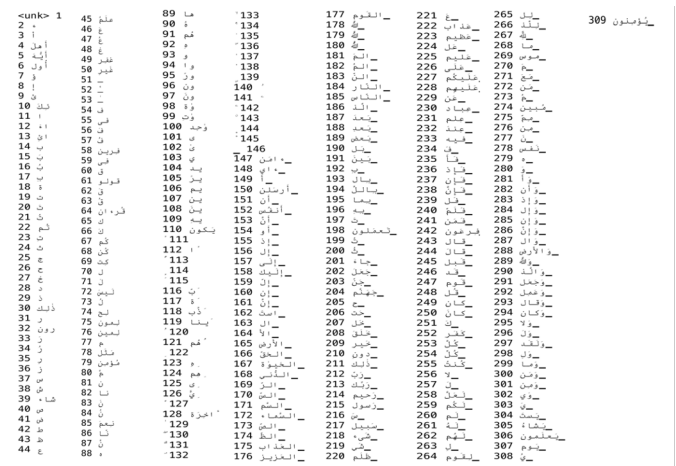


FIGURE 4: Tokens list

$$TER = \frac{(ST + DT + IT)}{NT} \times 100 \tag{5}$$

$$WER = \frac{(SW + DW + IW)}{NW} \times 100 \tag{6}$$

$$SER = \frac{(SS + DS + IS)}{NS} \times 100 \tag{7}$$

$$ACCT = \frac{(NCT)}{NT} \times 100 \tag{8}$$

$$ACCW = \frac{(NCW)}{NW} \times 100 \tag{9}$$

where ST, DT, and IT are the number of substitution, deletion, and insertion errors in terms of tokens; $ACCT$ is the accuracy of tokens; NCT is the number of correct tokens; and NT is the total number of tokens in the test dataset. SW, DW, and IW are the number of substitution, deletion, and insertion errors in terms of words. It is worth noting that the full sentence is considered wrong if at least one

Instance Name	NVIDIA T4 Tensor Core GPUs	vCPUs	RAM	Local Storage	EBS Bandwidth	Network Bandwidth
g4dn.xlarge	1	4	16 GiB	1 x 125 GB	Up to 3.5 Gbps	Up to 25 Gbps
g4dn.2xlarge	1	8	32 GiB	1 x 225 GB	Up to 3.5 Gbps	Up to 25 Gbps
g4dn.4xlarge	1	16	64 GiB	1 x 225 GB	Up to 3.5 Gbps	Up to 25 Gbps
g4dn.8xlarge	1	32	128 GiB	1 x 900 GB	7 Gbps	50 Gbps
g4dn.12xlarge	4	48	192 GiB	1 x 900 GB	7 Gbps	50 Gbps
g4dn.16xlarge	1	64	256 GiB	1 x 900 GB	7 Gbps	50 Gbps

FIGURE 5: g4dn.2xlarge instance

token is wrong. Finally, $ACCW$ is the accuracy of words, with NCW being the number of correct words and NW the total number of words in the test dataset. Besides, SS, DS, and IS are the number of substitution, deletion, and insertion errors in terms of sentences, while NS is the total number of sentences in the test dataset. For example, when considering words, substitution errors occur when a correct word is replaced with another word that contains at least one incorrect character or token, or when the replacement word is in the wrong position. Deletion errors occur when a word is missing from the hypothesis but present in the reference text. Insertion errors occur when a new word is added to the hypothesis that is not present in the reference text.

IV. EXPERIMENTAL SETUP

In what follows, we provide a comprehensive explanation of the experimental details, encompassing the environment, toolkit, dataset, and configuration specifics utilized in the experiment.

A. EXPERIMENTAL ENVIRONMENT

All the experiments were conducted using Amazon Elastic Compute Cloud (Amazon EC2), which offers scalable and on-demand computing capacity in the Amazon Web Services (AWS) Cloud. Specifically, we utilized the g4dn.2xlarge instance type, as depicted in Figure 5, which provides a volume capacity of 1200 GiB.

B. TOOLKIT

Numerous toolkits support E2E speech recognition, but we have a preference for ESPnet [15] due to its ease of comparison with hybrid speech recognition systems. Moreover, ESPnet adopts the Kaldi ASR approach, which is a free open-source toolkit for hybrid speech recognition research [16], for data processing and feature extraction. This integration allows ESPnet to provide a complete framework for speech recognition and other speech processing investigations. ESPnet has exhibited exceptional efficacy in attaining ASR performance that is commensurate with that of state-of-the-art HMM-DNN systems that employ conventional setups. This underscores the significance of leveraging advanced technologies and tools in the development of speech recognition models.

C. DATASET

One of the major limitations in the current studies is that all the previous works used a small dataset. To overcome this limitation, we created a new dataset called CQDV1 to serve HQSR research. The dataset encompasses the entire Quran, with 114 suras (6236 ayah) recited by 35 reciters, 217407 audio files, downloaded from [39]. The audio files were downloaded in mp3 format, and are based on the Hafs from Asim narrative. Figure 6 displays the names of the reciters included in this dataset. It is worth noting that our dataset includes recitations with and without magam, contributing to the model's generalizability.

This dataset contains two types of recitation: mogawwad and morattal. The term mogawwad refers to the act of reading slowly, engaging in the practice of reciting the Qur'an while adhering to the guidelines of Tajweed. Each of these readings preserves the rights and attributes of the letters that should be maintained. Essentially, employing the mogawwad technique while reading enables the reader to understand and contemplate the words of the Qur'an. Morattal is a form of reading characterized by a pace that falls between slow reading and quick reading. Typically, the practice of morattal recitation occurs during congregational (Jama'ah) prayers. Varying recitation styles on the same model could potentially negatively impact accuracy, yet it can also generate a robust model capable of identifying both slow and fast recitations. For example, when any reciter recites slowly using the mogawwad pattern, the model can recognize his recitation; the same thing happens when he recites using the Morattal pattern.

Tajweed utilizes the Arabic term Madd to denote the extension or prolonging of vowel sounds under specified circumstances. Mastery of this exercise is crucial for accurately reciting the Quran. Madd is the elongation of a vowel sound that is induced by specific letters or harakat (vowel markings) [40]. Madd Ja'ez Munfasil is a Tajweed rule that allows for elongation when a Hamzah (ء) is related to the preceding letter in the following word [40]. Observing the recitation of Harakaat, specifically the Mad Munfasil, easily identifies the reference in this context, as it consists of no more than six Harakat. For example, in mogawwad, the Mad Mufasil will be six Harakat, and in Morattal, the Mad Mufasil will be four Harakat.

In the collected dataset, we placed each sura's files in separate folders for all the reciters. Each ayah audio file follows this naming pattern: ReciterKeySuraNoAyahNo.wav, for example, R001001001.wav for the first ayah in the first surah of the Holy Quran (Alfatiha) recited by Abo Baker Alshatery (R001) as illustrated in Figure 7.

After downloading all the data, we converted the audio files from .mp3 to .wav. The audio files were arranged in folders, with each sura in a different folder. The dataset is organized according to the Kaldi dataset preparation guide. It consists of 4 files:

- The file "text" contains the transcriptions of each utterance.
- The file "wav.scp" contains the first token on each line of "wav.scp" file is just the utterance id. The files in

No.	Reciter Name	Reciter Key	Mogawwad	Morattal
1	Abo Baker Alshatery	R001		✓
2	Ahmad Ali Alagamy	R002		✓
3	Ahmad Naena	R003		✓
4	Ebraheem Alakhdar	R004		✓
5	Khalifah Alteneagy	R005		✓
6	Kareem Mansoory	R006		✓
7	Saad Alghamdy	R007		✓
8	Saud Alshoream	R008		✓
9	Sahl Yaseen	R009		✓
10	Shehraiar Berheez Car	R010		✓
11	Slaah Alhashem	R011		✓
12	Slaah Bo Khater	R012		✓
13	Khaled Alqahtany	R013		✓
14	Slah Albedear	R014		✓
15	Abdulrahman Alsodeas	R015		✓
16	Abdullah Almatrood	R016		✓
17	Abd Almuhsin Alqasem	R017		✓
18	Abdullah Basfar	R181		✓
19	**Abdullah Basfar**	R182	✓	
20	Abd Albaset Abd Alsamad	R191		✓
21	**Abd Albaset Abd Alsamad**	R192	✓	
22	Ali Abdulrahman Alhozifi	R020		✓
23	Maher Almaeaqly	R021		✓
24	Mohammad Ayyoob	R022		✓
25	Mohammad Altablawy	R023		✓
26	Mohammad Gebreel	R024		✓
27	Mohammad Abd Alkareem	R025		✓
28	Mohammad Sadeeg Alminshawy	R026		✓
29	Mahmood Kalel Alhosary	R271		✓
30	**Mahmood Kalel Alhosary**	R272	✓	
31	Mahmood Ali Albanna	R028		✓
32	Mshary Al-afasy	R029		✓
33	Naser Algattamy	R030		✓
34	Hany Alrefay	R031		✓
35	Yaser Aldosry	R032		✓

FIGURE 6: Reciters Names

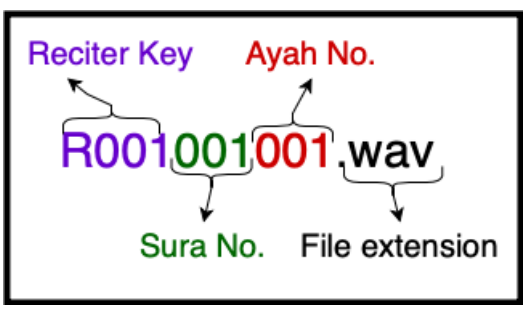


FIGURE 7: File Name Meaning

wav.scp must be single-channel (mono); if the underlying wav files have multiple channels, then a sox command must be used in the wav.scp to extract a particular channel.

- The file "spk2utt" shows all the utterances spoken by each speaker.
- The file "utt2spk" shows the speaker for for each utterance.

The text files include text, wav.scp, spk2utt, utt2spk, and other automatically generated files such as utt2num_frames. The file utt2num_frames contains the utterance ID as the first column on each line, and the second column represents the number of frames. Similarly, in the file utt2dur, the first column on each line denotes the utterance ID, while the second column indicates the utterance duration in seconds.

D. TEST CASES

Seven experiments were conducted in this study as shown in Table 1. For experiments 1-4, the data set is divided into train and test data set as follows: 80% for training and 20% for testing. The training dataset comprises suras (1-56), while the test dataset includes sura (57-114).

In the first experiment, a subset (16.1%) of the entire dataset is used, with 25,000 files for the training dataset and 10,000 files for the test dataset. In the second experiment, a subset (50.7%) of the entire dataset is utilized, with 100,000 files for the training dataset and 10,000 files for the test dataset. For the third and fourth experiments, a subset (91.7%) of the entire dataset is used, consisting of 159,364 files for suras (1-56) for training and 39,840 files for suras (57-114) for testing. Additionally, augmentation is performed on the training dataset by altering the speed of the original audio, generating new audio files with speeds of 0.9 and 1.1, resulting in the tripling of the number of files in the training set to 478,089 files.

For the fifth, sixth, and seventh experiments, all the data for suras (1-114) recited by 32 reciters are used as training data, totaling 517,929 files. Another dataset for the suras (1-114) recited by 3 reciters is added for the testing data. Table 1 provides a detailed overview of the datasets used in each experiment.

E. MODEL CONFIGURATION

In all experiments, we utilized fbank feature extraction techniques to extract features from the dataset, and CMVN for normalization. For the language model, we employed a token-based RNNLM with Stochastic Gradient Descent (SGD) optimizer. Our training approach involved using BLSTMP for the encoder, incorporating three layers: forward LSTM, backward LSTM, and a projection layer. We employed a hybrid CTC/attention as the loss function and used λ = 0.5 for the first, second, third, fourth, and fifth experiments, and λ = 0.2 for the sixth and seventh experiments. Adadelta optimization was employed for training, a stochastic gradient descent method based on adaptive learning rates per dimension. The deep learning engine utilized was PyTorch.

In the last experiment, we introduced an additional text file to the language model. This text file did not contain pause marks, resulting in a significant performance improvement. Indeed, as observed in Figure 12, experiment 6, most of insertion errors are caused by these pause marks. When these marks are removed from the text files fed to the LM, most of these errors are removed as shown in Figure 13.

TABLE 1: Dataset Details of the Experiments

exp #	Dataset Size	Train/ test	Number of Reciters	Number of Files	Duration (Hour)
1	16.1% of CQDV1	Train (1-56) Test (57-114)	32 32	25000 10000	129.3 26.7
2	50.7% of CQDV1	Train (1-56) Test (57-114)	32 32	100000 10000	522 26.7
3,4	91.7% of CQDV1	Train (1-56) Test (57-114)	32 32	478089 39840	2492 106
5,6,7	100% of CQDV1	Train (1-114) Test (1-114)	32 3	517929 17917	2597 101

V. RESULTS AND DISCUSSION

This section presents a comprehensive analysis of the results obtained from the experiments and engages in a detailed discussion of the findings and a comparative study with existing literature.

A. TESTING RESULTS

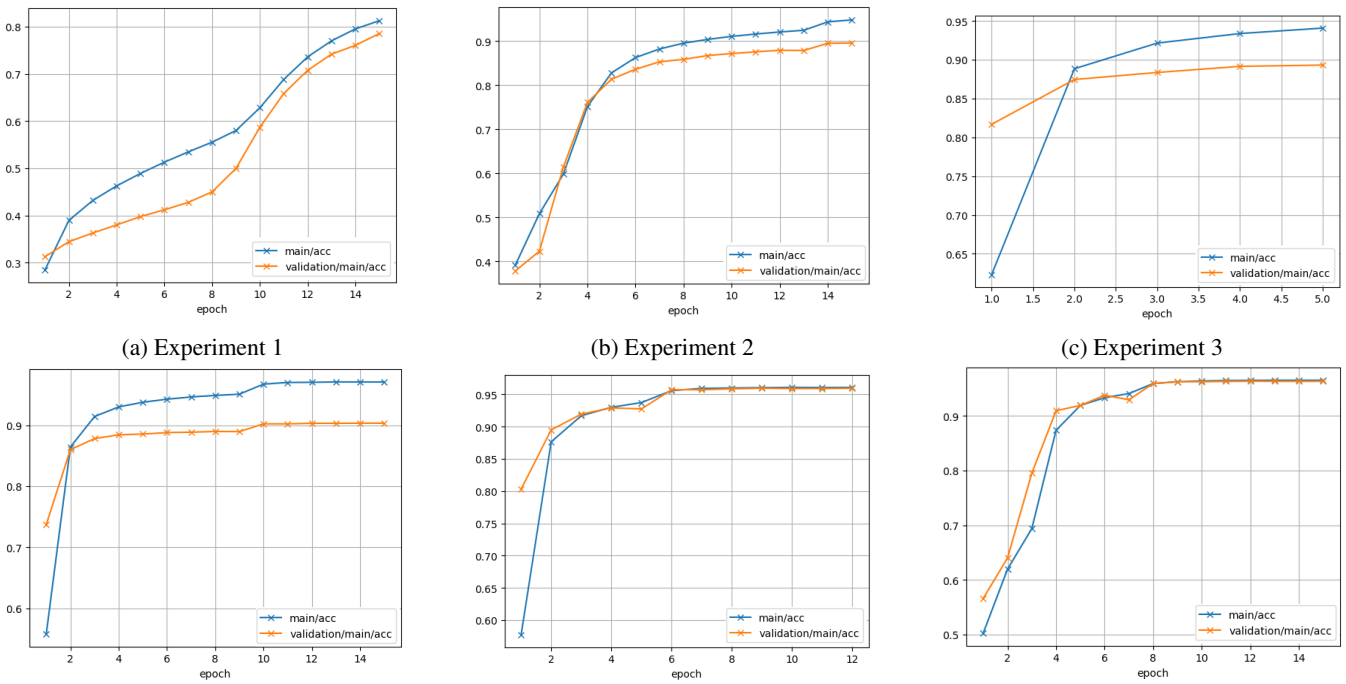
We present the accuracy versus epoch for all the experiments in Figure 8, the CER versus epoch for all the experiments in Figure 9, and the loss versus epoch for all the experiments in Figure 10 during the training of the acoustic model stage. Table 2 presents the detailed final results of all seven experiments, including the number of epochs, TER, token accuracy, WER, word accuracy, and SER.

TABLE 2: Experimental Results

#exp	#epoch	λ	TER	ACCT	WER	ACCW	SER
1	15	0.5	25.3	83.5	39.9	65.8	90.2
2	15	0.5	12.4	90.9	22.9	80.1	73.8
3	5	0.5	12.1	92	21.9	81.7	70.3
4	15	0.5	10.2	92.9	19.2	83.5	66
5	12	0.5	8.1	94	14	87.6	65.6
6	15	0.2	8.1	94.1	13.6	88.2	64.1
7	15	0.2	**6.4**	**95.3**	**10.4**	**90.8**	**55.3**

We observe that Experiment 1, which utilizes a small dataset, accounting for 16.1% of the entire dataset, achieved a TER of 25.3%, a WER of 39.9%, and a SER of 90.2%. Figure 8a illustrates the accuracy; Figure 9a represents the CER; and Figure 10a displays the loss for each epoch in experiment 1. Experiment 2 employs a larger dataset than the first experiment and yields a TER of 12.4%, a WER of 22.9%, and a SER of 73.8%. Figure 8b depicts the accuracy, Figure 9b represents the CER, and Figure 10b displays the loss for each epoch in experiment 2. The third experiment, which utilized 91.7% of the entire dataset, produced superior results compared to the first and second experiments, even with a small number of epochs, achieving a TER of 12.1%, a WER of 21.9%, and a SER of 70.3%. Figure 8c displays the accuracy, Figure 9c shows the CER; and Figure 10c illustrates the loss for each epoch in experiment 3. This underscores the significance of a large dataset in enhancing the performance of end-to-end speech recognition models. Experiment 4 utilized the same dataset as Experiment 3 but with a larger number of epochs, resulting in a TER of 10.2%, a WER of 19.2%, and a SER of 66%. The figures for accuracy, CER, and loss for each epoch in experiment 4 are shown in Figures 8d, 9d, and 10d respectively.

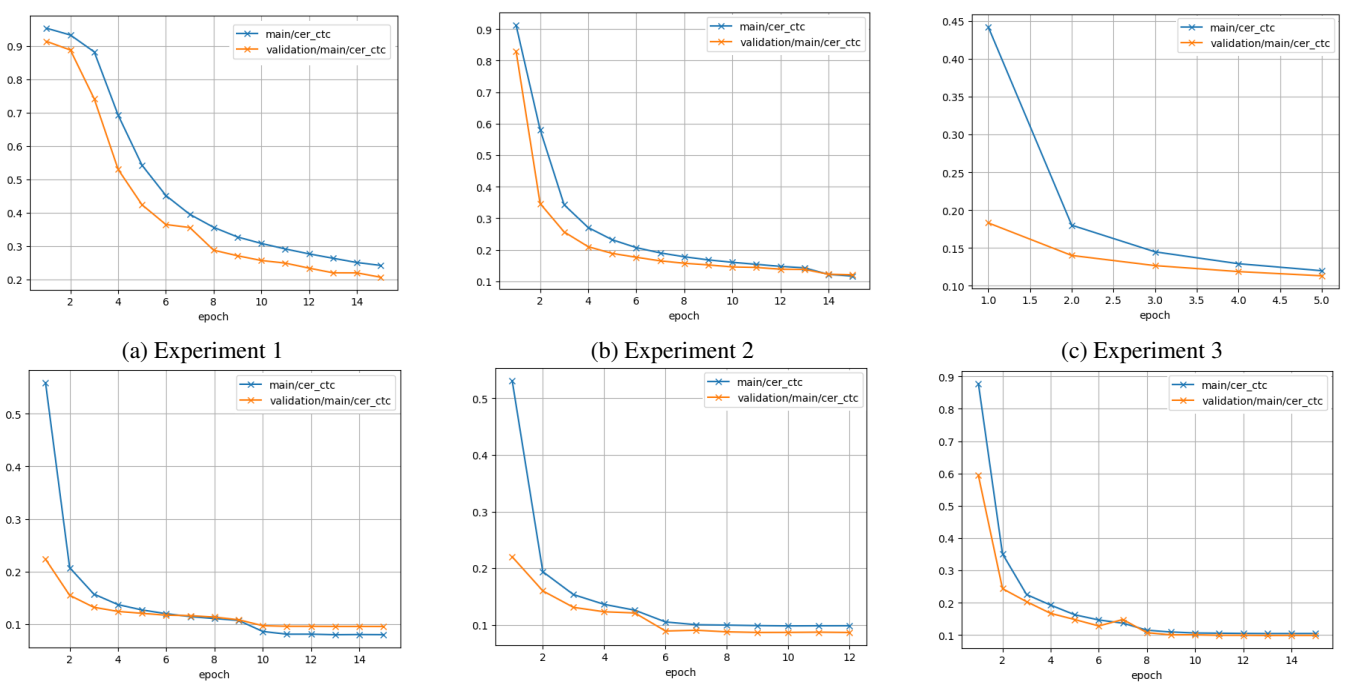
Moreover, for experiments 5, 6, and 7, the entire set of suras (from 1 to 114) was utilized in the training data. The Quran text dataset presents a unique case as it remains fixed and cannot be altered, regardless of any changes to the dataset, such as reciters or magam. Including all of the text in the training dataset is beneficial because the text is immutable and cannot be altered. The only variations between the training and testing datasets will occur in terms of recitation techniques, reciters, magam, narration, and so on, but not in terms of the actual text. Therefore, there is no bias or issue with this approach to dividing the dataset. Notably, experiment 5 yielded promising results with a TER of 8.1%, WER of 14%, and SER of 65.6%. The accuracy, CER, and loss values for each epoch in experiment 5 are displayed in Figures 8e, 9e, and 10e consecutively. It is evident that the performance improved as the number of epochs increased. Furthermore, the analysis of the loss function reveals that the attention loss yields a lower minimum loss compared to the CTC loss. Consequently, in experiment 6, we will adjust the λ value to 0.2 to enhance performance. Subsequently, experiment 6 delivered superior results to experiment 5, achieving a TER of 8.1%, WER of 13.6%, and SER of 64.1%. It is worth noting that in Experiments 6 and 7, the acoustic model training is the same, with the only difference being the LM. Thus, we depict in Figures 8f, 9f, and 10f the accuracy, CER, and loss, respectively, for each epoch in both Experiments 6 and 7. The proposed model has the capability to provide users with feedback regarding error types (substitution, insertion, and deletion) and their respective locations, facilitating easy error correction in their learning process. Figure 11 illustrates this feedback, demonstrating how the model identifies error locations and types at the token and word levels. An example of token-level errors



(a) Experiment 1 (b) Experiment 2 (c) Experiment 3

(d) Experiment 4 (e) Experiment 5 (f) Experiment 6 & 7

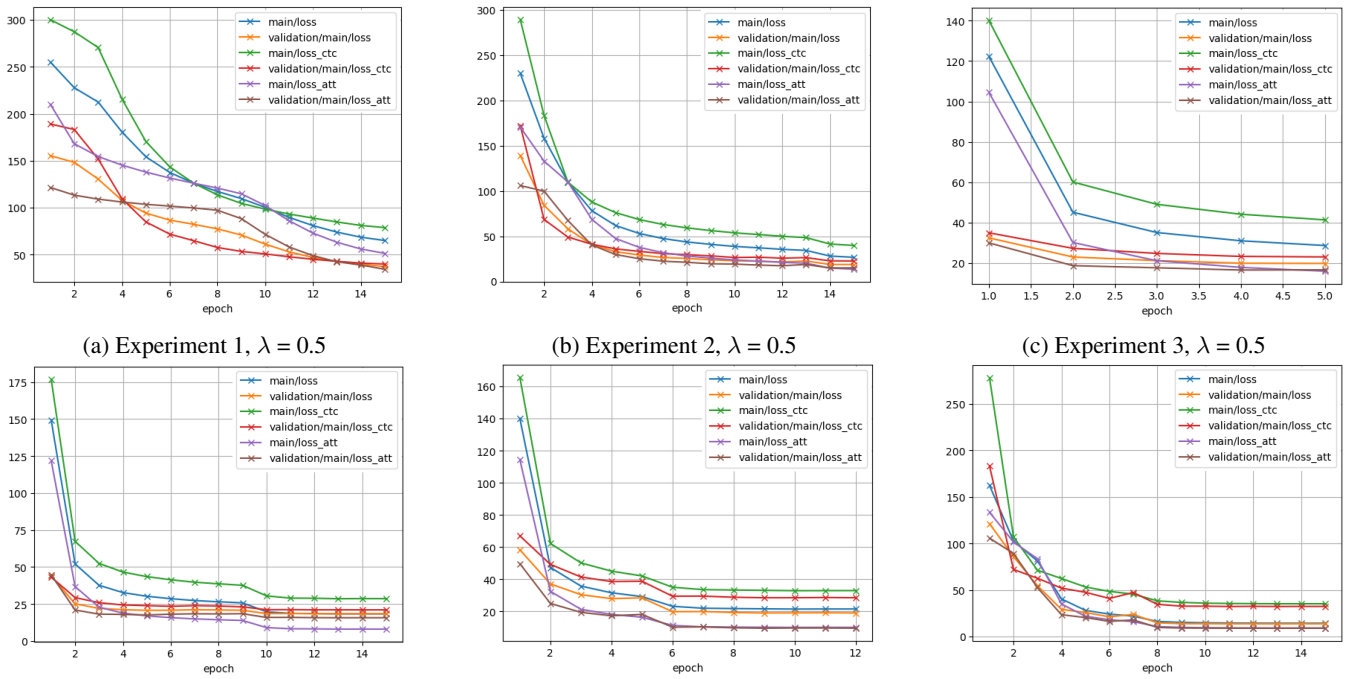
FIGURE 8: Accuracy Vs Epoch Graph For all Experiments



(a) Experiment 1 (b) Experiment 2 (c) Experiment 3

(d) Experiment 4 (e) Experiment 5 (f) Experiment 6 & 7

FIGURE 9: Character Error Rate (CER) Vs Epoch Graph For all Experiments



(a) Experiment 1, $\lambda = 0.5$ (b) Experiment 2, $\lambda = 0.5$ (c) Experiment 3, $\lambda = 0.5$

(d) Experiment 4, $\lambda = 0.5$ (e) Experiment 5, $\lambda = 0.5$ (f) Experiment 6 & 7, $\lambda = 0.2$

FIGURE 10: Loss Vs Epoch Graph For all Experiments

is the sentence with the ID (r016-r016_r016001006) and the score ((#C #S #D #I) 17 0 1 0). This score indicates that the sentence contains 18 tokens, out of which 17 are correct and one has a deletion error. The correct sentence is ا ه د اِ نَ ا الصّ اِ ر أ ط اَ المُ س تَ ق يم, and the predicted sentence is ا ه د اِ نَ ** الصّ اِ ر أ ط اَ المُ س تَ ق يم, ** pointing to the deleted token place. At the word level, we observe the sentence with the ID (r016-r016_r016001004) and the score ((#C #S #D #I) 2 1 0 0), indicating that this sentence comprises three words, with two being correct and one containing a substitution error. The correct sentence is مـلكِ يَومِ الدّين, and the predicted sentence is مـلكِ يَومَ الدّين. The only mistake in the sentence is the wrong word in harakat alkasrah اِ, which was substituted by harakat alfatha اَ. This substitution demonstrates the model's ability to detect precise errors. However, some insertion errors are not identified accurately as shown in Figure 12. For example, in the statement with the ID (r016-r016_r016036020) and score ((#C #S #D #I) 10 0 0 1), this indicates that the statement comprises 11 words, with 10 being correct and one containing an insertion error. The correct sentence is وَجاءَ مِن أَقصَا المَدينَةِ رَجُلٌ يَسعىٰ ** قالَ يـقومِ اتَبعُوا المُرسَلين and the predicted sentence is وَجاءَ مِن أَقصَا المَدينَةِ رَجُلٌ يَسعىٰ قالَ يـقومِ اتَبعُوا المُرسَلين, this mistake is caused by the use of stop marks in the text file.

To address this issue, we attempted to minimize these inaccuracies by improving the text file fed to the language model. The revised text file excludes pause marks, resulting in a significant performance improvement in experiment 7, with a TER of 6.4%, WER of 10.4%, and SER of 55.3%. Upon examining Figure 13, it becomes evident that numerous uncorrected errors have been rectified in the new experiment.

It is worth noting that the our test dataset used in experiments 5, 6, and 7 comprises 3 reciters (R016, R017, and R192). Tables 3 and 4 showcase the TER and the WER percentages, respectively, by reciters for the optimal experiment 7. For each reciter, both tables indicate the number of sentences for in the test dataset. The third column in table 3 denotes the number of tokens, while table 4 indicates the number of words. Then we also present the percentage of correctly recognized tokens or words out of the total tokens or words for each reciter. The table also display the percentage of substitution errors, the percentage of deletion errors, and the percentage of insertion errors. Lastly, the sum of all errors (substitution, deletion, and insertion) is depicted. Finally, the last column illustrates the sentence error rate percentage. For instance, consider the reciter r017 (Abd Almuhsin Alqasem) in Table 3. This reciter has 5924 sentences in the dataset, with 285994 tokens and 273350 correct tokens, resulting in a percentage of correct tokens as shown in the table of 95.6%. There are 5945 tokens with substitution errors, making up 2.1% of the total tokens. Additionally, there are 6699 tokens with deletion errors, representing 2.3% of the total tokens. Furthermore, there are 1610 tokens with insertion errors, which is 0.6% of the total tokens. The sum of all tokens with errors is 14254 tokens, making up 5% of the total tokens. Lastly, there are 3280 sentences with any type of token error,

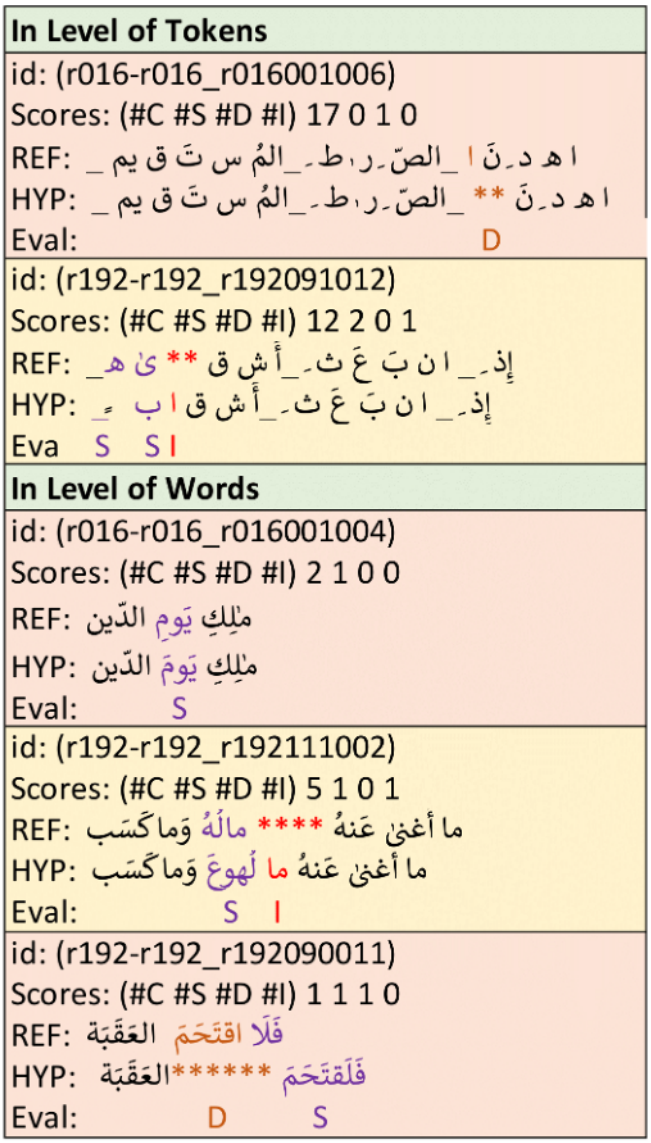


FIGURE 11: Types of errors (substitution, deletion, and insertion)

and the percentage of sentence errors is 55.4%, as shown in the table.

B. DISCUSSION

Though many works have been suggested recently in the field of HQSR, most of the existing solutions have some limitations, either in terms of methodology or the dataset used [17]. In Table 5, we analyze and compare current research studies based on the characteristics of the dataset and the proposed approach using the following criteria:

- Dataset characteristics:
  1) #verses: It represents the total number of verses utilized in the study: The variables L, M, H, and N represent the number of verses falling within different ranges. L, M, and H are used when the number of verses falls in the following ranges: [1..100], [101..200], and [201..6236], respectively, while N is used if the number of verses have not been specified in the research.
  2) #sura: It denotes the number of suras utilized in the study.
  3) #reciters: This indicates the number of reciters who participated in this study.
- Proposed methodology:
  1) DL-based: This criterion indicates if deep learning was applied at any phase throughout the study.
  2) LM: This indicates whether or not the study's solution used a language model..
  3) AM: It shows whether the study uses an acoustic model or no.
  4) Template: This determines if the study follows the template-based speech recognition techniques.
  5) Traditional: This determines if the study follows the traditional-based speech recognition techniques.
  6) E2E: This determines if the study follows the E2E-based speech recognition techniques.

Table 5 reveals that further studies are required to examine deep learning architectures in order to enhance the overall accuracy of HQSR. While some articles have proposed the utilization of deep learning, they typically rely on outdated architectures. For instance, [21] employed an outdated deep learning architecture that only uses deep learning in the feature extraction phase, rather than incorporating it into lexical, acoustic, and language models. Moreover, [22] used a deep learning architecture relyinh on a template-based framework that contains only feature extraction and classification. We also observe that previous studies predominantly relied on template and traditional methods for speech recognition, with only a few research endeavors employing E2E approaches. However, some of these E2E solutions have a limited dataset that does not cover the entire Quran, resulting in biased performance. The best work in terms of number of reciters is [30] but they still need to improve their dataset by increasing the total number of recorded hours (see Table 6). Thus, the development of a dependable and universally applicable voice recognition system is facilitated by a large, comprehensive dataset. This indicates a significant gap in the application of E2E approaches in the field of speech recognition of Quran, highlighting the need for further exploration and advancement in this area. Besides, it is evident that the performance of the traditional ASR system is significantly superior when the data size is less than 400h [14]. As the size of the data increases, E2E performance improves at a significantly faster rate compared to the traditional system. This suggests that the E2E model is likely to exhibit even greater benefits with larger amounts of data [14].

The recent E2E solutions are compared to our solution in Table 6. Our comparison indicates that the current E2E

TABLE 3: TER percentages by reciters for the best Experiment 7

Reciter	# Snt	# Tokens	Corr	Sub	Del	Ins	Err	S.Err
r016	6073	292779	94.8	3.2	2	1.6	6.8	58.2
r017	5924	285994	95.6	2.1	2.3	0.6	5	55.4
r192	5920	282090	95.6	2.8	1.7	3.1	7.5	52.7
Sum/Avg	17917	860863	95.3	2.7	2	1.7	6.4	55.5
Mean	5972.3	286954.3	95.3	2.7	2	1.7	6.4	55.4
S.D.	87.2	5408.8	0.5	0.6	0.3	1.3	1.3	2.7
Median	5924	285994	95.6	2.8	2	1.6	6.8	55.4

TABLE 4: WER percentages by reciters for the best Experiment 7

Reciter	# Snt	# Wrd	Corr	Sub	Del	Ins	Err	S.Err
r016	6073	79582	90	8	2	1.2	11.2	58
r017	5924	77714	91.3	5.6	3.1	0.2	8.9	55.2
r192	5920	76556	91.1	6.9	2.1	2.2	11.1	52.5
Sum/Avg	17917	233852	90.8	6.8	2.4	1.2	10.4	55.3
Mean	5972.3	77950.7	90.8	6.8	2.4	1.2	10.4	55.3
S.D.	87.2	1526.8	0.7	1.2	0.6	1	1.3	2.8
Median	5924	77714	91.1	6.9	2.1	1.2	11.1	55.2

solutions employ a variety of E2E architectures, including CNN-Bidirectional GRU-CTC, TDNN-CTC, RNN-CTC, and Transformer. The authors of [30] devised three E2E models: transformer, RNN-CTC, and TDNN-CTC. They achieve the most favorable outcome with RNN-CTC, achieving a CER of 3.51% and the lowest WER of 19.43% using an approximately 100-hour dataset. They demonstrated that transformer achieved the lowest performance with an WER of 95.03%. This suggests that more intricate and advanced E2E architectures, such as transformers, require additional training data. Besides, we deduce a distinct bias in the results of [28], as the transformer was trained for only 10 hours and the WER was 6.16%. Using a 2698-hour dataset, we implemented an E2E hybrid CTC/attention model with BLSTMP architecture in our investigation. This method yielded an improved outcome, with a weighted error rate (WER) of 10.4%. The objective is to maintain a very low TER, WER, and SER, as even a minor error is considered intolerable when reciting the Holy Quran. To improve the dataset that has been gathered, it is imperative to increase its heterogeneity by incorporating recordings of women and children, as well as native and non-native Arabic speakers. In terms of architecture, it is imperative to implement a more advanced architectural methodology than the one that was previously employed.

VI. CONCLUSION

The field of Holy Quran Speech Recognition (HQSR) is still in its early stage, with no prior research having attained optimal results. The primary challenges in HQSR include the absence of a comprehensive dataset, the presence of numerous diverse narrations, various methods of reading the Quran (Magam), and differing lengths of prolongations (Madd). This paper aimed to address the gap in current literature by comparing existing solutions and highlighting the limitations of previous studies. It introduced a study on HQSR utilizing an end-to-end deep learning architecture. Our suggested architecture uses fbank feature extraction and BLSTMP encoder coupled with a hybrid architecture for the loss that uses CTC and attention, and fed with a token-based language model that combines BLSTMP. We conducted several experiments to explore how performance could be improved by varying the size of the dataset, the division between training and testing datasets, and the weights of CTC and attention loss. The best results achieved a TER of 6.4%, a WER of 10.4%, and a SER of 55.3%. To enhance the current speech recognition models for the Holy Quran, more research is required. The main future work may involve conducting more experiments to improve the performance of the proposed solution. This includes changing hyper-parameters, extracting features using different techniques, and adjusting the acoustic model training architecture. Additionally, expanding the dataset by including recitations from a diverse range of individuals, such as female, children, native, and non-native Arabic speakers, will also be crucial for improving performance. The ultimate goal is to achieve optimal results with very low TER, WER, and SER, as even minor errors in reciting the Qur'an can alter the meaning of an entire ayah. Finally, the extensive training duration reaching 425 hours, was necessitated by the substantial volume of data. This significantly hindered the ability to test multiple algorithms and architectures. Therefore, using big data technologies can be explored.

REFERENCES

[1] Pahini A Trivedi. Introduction to various algorithms of speech recognition: Hidden markov model, dynamic time warping and artificial neural networks. International Journal of Engineering Development and Research, 2(4):3590–3596, 2014.

[2] Xiaodong He and Li Deng. Discriminative learning for speech recognition: theory and practice. Synthesis Lectures on Speech and Audio Processing, 4(1):1–112, 2008.

[3] Ahmed Hamdi Abo Absa. Self-Learning Techniques for Arabic Speech Segmentation and Recognition. Thesis, 2018.

[4] Abdelaziz A Abdelhamid, Hamzah A Alsayadi, Islam Hegazy, and Zaki T Fayed. End-to-end arabic speech recognition: A review. 2020.

[5] Lv Ping. English speech recognition method based on hmm technology. In 2021 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), pages 646–649. IEEE, 2021.

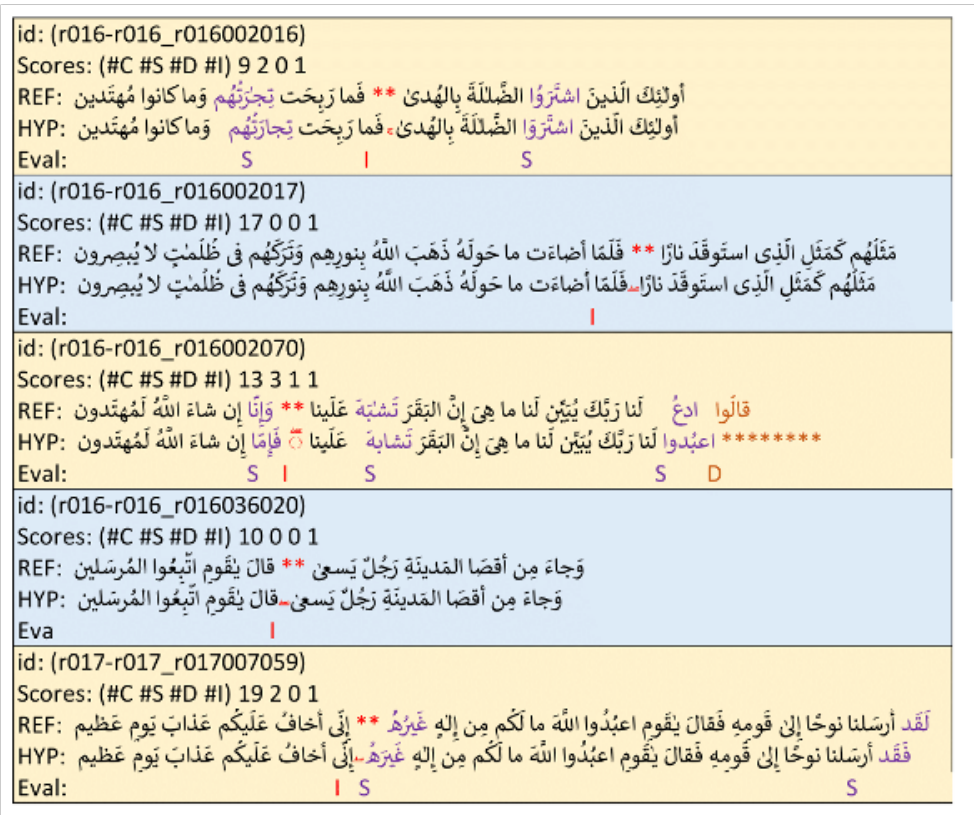


FIGURE 12: Random Samples shows wrong insert errors on experiment 6

[6] Benjamin Elisha Sawe. Arabic speaking countries, Jul 2018.

[7] Fatimah Alqadheeb, Amna Asif, and Hafiz Farooq Ahmad. Correct pronunciation detection for classical arabic phonemes using deep learning. In 2021 International Conference of Women in Data Science at Taif University (WiDSTaif), pages 1–6. IEEE, 2021.

[8] Imane Guellil, Houda Saâdane, Faical Azouaou, Billel Gueni, and Damien Nouvel. Arabic natural language processing: An overview. Journal of King Saud University-Computer and Information Sciences, 33(5):497–507, 2021.

[9] Muslim population by country 2021, 2021.

[10] Ammar Mohammed Ali Alqadasi, Rawad Abdulghafor, Mohd Shahrizal Sunar, and Md Sah Bin H.J. Salam. Modern standard arabic speech corpora: A systematic review. IEEE Access, 11, 2023.

[11] Song Wang and Guanyu Li. Overview of end-to-end speech recognition. In Journal of Physics: Conference Series, volume 1187, page 052068. IOP Publishing, 2019.

[12] Hanan Aldarmaki, Asad Ullah, and Nazar Zaki. Unsupervised automatic speech recognition: A review. arXiv preprint arXiv:2106.04897, 2021.

[13] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, and Guoliang Chen. Deep speech 2: End-to-end speech recognition in english and mandarin. In International conference on machine learning, pages 173–182. PMLR, 2016.

[14] Amir Hussein, Shinji Watanabe, and Ahmed Ali. Arabic speech recognition by end-to-end, modular systems and human. arXiv preprint arXiv:2101.08454, 2021.

[15] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, and Nanxin Chen. Espnet: End-to-end speech processing toolkit. arXiv preprint arXiv:1804.00015, 2018.

[16] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, and Petr Schwarz. The kaldi speech recognition toolkit. In IEEE 2011 workshop on automatic speech recognition and understanding. IEEE Signal Processing Society, 2011.

[17] Alfadhli Sumayya, Alharbi Hajar, and Cherif Asma. Speech recognition models for holy quran recitation based on modern approaches and tajweed rules: A comprehensive overview. International Journal of Advanced Computer Science and Applications(IJACSA), 14(12), 2023. http://dx.doi.org/10.14569/IJACSA.2023.0141297, 2023.

[18] Ammar Mohammed, Mohd Shahrizal Sunar, and Md Sah Hj Salam. Quranic verses verification using speech recognition techniques. Jurnal Teknologi, 73(2), 2015.

[19] Yousef A Alotaibi, Mohammed Sidi Yakoub, Ali Meftah, and Sid-Ahmed Selouani. Duration modeling in automatic recited speech recognition. In 2016 39th International Conference on Telecommunications and Signal Processing (TSP), pages 323–326. IEEE, 2016.

[20] Ghassan Samara, Essam Al-Daoud, Nael Swerki, and Dalia Alzu'bi. The recognition of holy qur'an reciters using the mfccs' technique and deep learning. Advances in Multimedia, 2023, 2023.

[21] Mahmoud Al-Ayyoub, Nour Alhuda Damer, and Ismail Hmeidi. Using deep learning for automatically determining correct application of basic quranic recitation rules. Int. Arab J. Inf. Technol., 15(3A):620–625, 2018.

[22] Natasha Nigar, Amna Wajid, Sunday Adeola Ajagbe, and Matthew O. Adigun. An intelligent framework based on deep learning for online quran learning during pandemic. Applied Computational Intelligence and Soft Computing, 2023, 2023.

[23] Mohamed Yassine El Amrani, MM Hafizur Rahman, Mohamed Ridza Wahiddin, and Asadullah Shah. Towards an accurate speaker-independent holy quran acoustic model. In 2017 4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS), pages 1–4. IEEE, 2017.

[24] Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, and Joe Woelfel. Sphinx-4: A flexible open source framework for speech recognition. 2004.

[25] Imad K Tantawi, Mohammad AM Abushariah, and Bassam H Hammo. A deep learning approach for automatic speech recognition of the holy qur'?n recitations. International Journal of Speech Technology, pages 1–16, 2021.

[26] Faza Thirafi and Dessi Puji Lestari. Hybrid hmm-blstm-based acoustic modeling for automatic speech recognition on quran recitation. In 2018

FIGURE 13: Random Samples shows how experiment 7 improved the performance

International Conference on Asian Language Processing (IALP), pages 203–208. IEEE, 2018.

[27] Alsayadi Hamzah A and Hadwan Mohammed. Automatic speech recognition for qur'an verses using traditional technique. Journal of Artificial Intelligence and Metaheuristics (JAIM), 2022.

[28] Mohammed Hadwan, Hamzah A Alsayadi, and Salah AL-Hagree. An end-to-end transformer-based automatic speech recognition for qur'an reciters. Computers, Materials & Continua, 74(2), 2023.

[29] Ahmad Al Harere and Khloud Al Jallad. Quran recitation recognition using end-to-end deep learning. arXiv preprint arXiv:2305.07034, 2023.

[30] Sarah S. Alrumiah and Amal A. Al-Shargabi. A deep diacritics-based recognition model for arabic speech: Quranic verses as case study. IEEE Access, 11, 2023.

[31] Hilman F. Pardede, Vicky Zilvan, Dikdik Krisnandi, Ana Heryana, and R. Budiarianto S. Kusumo. Generalized filter-bank features for robust speech recognition against reverberation. 2019.

[32] Shaoyun Zhang and Chao Li. Research on feature fusion speech emotion recognition technology for smart teaching. Mobile Information Systems, 2022, 2022.

[33] Zhangyu Xiao, Zhijian Ou, Wei Chu, and Hui Lin. Hybrid ctc-attention based end-to-end speech recognition using subword units. 2018.

[34] Inc The MathWorks. short-term memory (lstm) projected layer for recurrent neural network (rnn) - matlab, url= https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.lstmprojectedlayer.html, 2023.

[35] Maël Fabien. Introduction to automatic speech recognition (asr)maël fabien, May 2020.

[36] Ashifur Rahman, Md Mohsin Kabir, M. F. Mridha, Mohammed Alatiyyah, Haifa F. Alhasson, and Shuaa S. Alharbi. Arabic speech recognition: Advancement and challenges. IEEE Access, 12, 2024.

[37] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. volume 3, 2016.

[38] sanbit876. Byte-pair encoding (bpe) in nlp, Apr 2023.

[39] Quran 70, 6236 mp3, url= https://www.a-quran.com/showthread.php?t=11017, accessed = 24 October 2023, 2023.

[40] quranhouse admin. Madd rules in tajweed: Madd letters, charts, and types, Jan 2024.

PLACE PHOTO HERE

SUMAYYA ALFADHLI was born in Makkah, Saudi Arabia, in 1991. She received the B.S. degree in computer science from Umm Al-Qura University, College of Computing, Computer Science Department, in 2013. She is now studying for an M.S. degree at King Abdulaziz University, Faculty of Computing and Information Technology, Department of Computer Science. From 2017 to now, she has been teaching assistant at Umm Al-Qura University, Adham University College, Department of Computer Science. Her research interests include speech recognition, neural networks, and programming languages.

TABLE 5: Comparative study

Ref#	Dataset			Methodology					
	#verses	#sura	#reciters	DL-based	LM	AM	Template	Traditional	E2E
[25]	M	1	32	✓	✓	✓		✓	
[26]	N		13	✓	✓	✓		✓	
[21]	H		10	✓			✓		
[23]	H	49	39		✓	✓		✓	
[19]	H	3	4				✓		
[18]	L		10				✓		
[20]	H		7	✓			✓		
[27]	L		60	✓	✓	✓		✓	
[28]	N		60	✓	✓	✓			✓
[29]	H	37	42	✓		✓			✓
[30]	H		129	✓		✓			✓
[22]	N			✓			✓		
Our Solution	H	114	35	✓	✓	✓			✓

Note: *L (#verses* $\in [1..100]$*), M (#verses* $\in [101..200]$*), H (#verses* > 200*), and N (#verses not specified).*

TABLE 6: Recent E2E Studies Compared With Our Study

Ref	Dataset			Architecture	WER
	Quran verses	Clips files	Hours		
[28]	1.60%	960	10	Transformer	6.16%
[29]	9%	16207	≈ 50	CNN-Bidirectional GRU - CTC	8.34%
[30]	Not defined	72735	>100	TDNN-CTC RNN-CTC Transformer	45.73% 19.43% 95.03%
Our solution	100%	217121 After Augmentation 517,929	2698	BLSTMP- CTC/Attention	10.40%

PLACE PHOTO HERE

HAJAR ALHARBI received her master's and Ph.D. degrees in computer science from the University of New England, Australia. She is currently an Assistant Professor at the Faculty of Computer Science and Information Technology, King Abdulaziz University. Her research interests include artificial intelligence-related topics, image processing, pattern recognition, and their applications to medical image analysis, computer vision, speech recognition, machine learning, and deep learning.

PLACE PHOTO HERE

ASMA CHERIF received the M.Sc. and Ph.D. degrees in computer science from Lorraine University, France, in 2008 and 2012, respectively. She conducted extensive research with the French Research Laboratory, Inria Nancy Grand Est. Since 2022, she has been leading the IoT Ecosystems Research Team, Center of Excellence in Smart Environment Research (CESER), King Abdulaziz University, Saudi Arabia. She is currently an Associate Professor with the Faculty of Computing and Information Technology, King Abdulaziz University. Her research interests include computational intelligence, distributed systems and communication networks, security, and the Internet of Things.