Arabic Alphabet and Surah Pronunciation Corrections Using Deep Learning

BY

FAIJUNNAHAR
ID: 0242310005103041

This Report Presented in Partial Fulfillment of the Requirements for

The Degree of Masters of Science in Computer Science and Engineering

Supervised By
Dr. Sheak Rashad Haider Noori
Professor & Head
Department of CSE
Daffodil International University

Co-Supervised By
Mr. Abdus Sattar
Assistant Professor & Coodinator M.sc
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

6 July 2024

APPROVAL

This Project/Thesis titled **"Arabic Alphabet and Surah Pronunciation Correction Using Deep Learning"**, submitted by **Faijunnahar**, ID No: **0242310005103041** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of M.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 06-07-2024.

BOARD OF EXAMINERS

Chairman

Dr. Sheak Rashed Haider Noori, PhD
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Dr. Md. Zahid Hasan, PhD
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Dr. Arif Mahmud, PhD
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

External Examiner

Dr. Md. Zulfiker Mahmud
Professor
Department of Computer Science and Engineering
Jagannath University

©Daffodil International University

DECLARATION

We hereby declare that, this thesis has been done by us under the supervision of **Dr. Sheak Rashad Haider Noori, Professor &Head, Department of CSE** Daffodil International University. We also declare that neither this thesis nor any part of this thesis has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Dr. Sheak Rashad Haider Noori
Professor &Head
Department of CSE
Daffodil International University

Co-Supervised by:

Mr. Abdus Satta
Assistant Professor & Coodinator M.sc
Department of CSE
Daffodil International University

Submitted by:

Faijunnahar
Faijunnahar
ID:0242310005103041
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final thesis successfully.

We really grateful and wish our profound our indebtedness to **Dr. Sheak Rashad Haider Noori**,Professor &Head Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "Data Mining" to carry out this thesis. His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior draft and correcting them at all stage have made it possible to complete this thesis.

We would like to express our heartiest gratitude to Dr. Sheak Rashad Haider Noori, Professor and Head, Department of CSE, for his kind help to finish our thesis and also to other faculty members and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and passion of our parents.

ABSTRACT

The development of automatic speech recognition systems for Arabic presents unique challenges, particularly due to the complexities of Classical Arabic, which has been underrepresented in research. Proper pronunciation of Arabic letters is crucial as it affects word meanings significantly. This study introduces novel learning models designed for Arabic letter classification with an emphasis on accurate pronunciation. The task is bifurcated into two stages: firstly, training the model for Arabic letter recognition, and secondly, evaluating the pronunciation quality of these letters. Given the scarcity of relevant audio datasets, I have collected audio samples from both experts and novices for training purposes. Pronunciation features were extracted from these audio samples using mel-spectrograms. I implemented deep convolutional neural networks (DCNN), Transfer learning model of AlexNet, and Audio LSTM (BLSTM networks). The recognition accuracies of DCNN, AlexNet and BLSTM are 98.97%, 98.45% and 91.39 %. DCNN, AlexNet, and BLSTM models provided accuracy of 97.87%, 99.14%, and 77.78% for the quality of pronunciations.For Quranic Ayat pronunciation correction, the models achieved a Word Error Rate (WER) of 8.34% and a Character Error Rate (CER) of 2.42%. This study underscores the effectiveness of advanced neural networks in enhancing Arabic speech recognition and pronunciation evaluation.

LIST OF FIGURES

FIGURES	PAGE NO
Figure 3.1.1: Proposed system architecture	16
Figure 3.4.1: Deep convolutional neural network architecture.	18
Figure 3.4.2: AlexNet architecture.	21
Figure 3.4.3 : Bidirectional long short-term memory (BLSTM) architecture.	24
Figure 3.5.1: Bidirectional long short-term memory (BLSTM) architecture.	25
Figure 3.5.2: Audio levels are marked according to speech detected in the recorded file before splitting.	26
Figure 3.5.3: Time and frequency representation of audio file: (a) without data augmentation, (b) with augmentation.	27
Figure 4.2.1 : DCNN after Data Augmentation	35
Figure 4.2.2: AlexNet before Data Augmentation	35
Figure 4.2.3: Comparison of DCNN, AlexNet, and BLSTM models with and without data augmentation using different validation strategies. (a) Alphabet Identification. (b) Pronunciation identification.	40

LIST OF TABLES

TABLE	PAGE NO
Table 1: Comparative Analysis and Summary	9
Table2: Training option of the neural network	17
Table 3:Dataset of Alphabet Classification	31
Table 4: Alphabet Classification after and before Data Augmentation	32
Table 5: Alphabet Classification Values of Mean, Standard Deviation, and Error.	33
Table 6. Confusion Matrix before Data Augmentation	33
Table 7. Confusion Matrix after Data Augmentation	34
Table 8. Arabic Alphabet Pronunciation Classification after and before Data augmentation	37
Table 9. Pronunciation Classification Values of Mean, Standard Deviation, and Error	37
Table10 : Experiments conducted to select feature extraction parameters	39
Table11 : Some samples that show the performance of the model	39

TABLE OF CONTENTS

CONTENTS	PAGE
Approval Page	ii
Declaration	iii
Acknowledgements	iv
Abstract	v
List of Figures	vi
List of Tables	vii
CHAPTER	
CHAPTER 1: INTRODUCTION	**1-4**
1.1 Introduction	1
1.2 Motivation	2
1.3 Rationale of the Study	2
1.4 Research Question	3
1.5 Expected Output	3
1.6 Report Layout	4
CHAPTER 2: BACKGROUND	**5-14**
2.1 Introduction	5
2.2 Preliminaries/Terminologies	5
2.3 Related Work	8
2.4 Comparative Analysis and Summary	9
2.5 Scope of the Problem	11
2.6 Challenges	14

CHAPTER 3: RESEARCH METHODOLOGY **15-30**

3.1 Introduction 15
3.2 Feature Extraction 16
3.3 Neural Network Model Training 16
3.4 Classification for Deep Learning Models 17
3.5 Data Collection Procedure/Dataset Utilized 24
3.6 Statistical Analysis 27
3.7 Proposed Methodology/Applied Mechanism 28
3.8 Implementation Requirements 29

CHAPTER 4: EXPERIMENTAL RESULTS AND DISCUSSION **31-40**

4.1 Introduction 31
4.2 Results and Discussion 31
4.3 Discussion 40

CHAPTER 5: IMPACT ON SOCIETY, ENVIRONMENT, AND SUSTAINABILITY **42-43**

5.1 Impact on Society 42
5.2 Educational Impact 42
5.3 Cultural Impact 42
5.4 Technological Impact 43
5.5 Social and Community Impact 43
5.6 Ethical Aspects 44

CHAPTER 6: Conclusion and future plan **46-47**

6.1 Summary of the study 46
6.2 Conclusions 46
6.3 Implication for further study 47

REFERENCES **48**

CHAPTER 1

INTRODUCTION

1.1 Introduction

In Bangladesh, Arabic holds immense cultural and religious significance, given that approximately 90% of the population practices Islam. Consequently, a large number of Bangladeshis learn Arabic for religious purposes, particularly for understanding the Quran. The country boasts over 15,000 madrasahs where millions of students receive Arabic education. Mastery of Arabic also presents substantial economic opportunities, as around 1.2 million Bangladeshi expatriates work in Arabic-speaking nations, highlighting the importance of Arabic language education for fostering cultural connections and economic benefits between Bangladesh and the Arab world.

Native and non-native speakers both need to learn the pronunciation rules in Classical Arabic (CA) perfectly, since they are simply indispensable to understand, pronounce, and structure the language properly. Proper pronunciation demands knowledge of the articulation points of all the letters, insight into the attributes of each letter, dedication in vocal practice. This research builds an automated Arabic alphabet pronunciation evaluator. This study has saved humanity from figuring out a system that can separate words from sentences that can help in teaching Classical Arabic pronunciation using automatic means.

This work aims to design and train neural networks that analyze users' speech data to judge how well they pronounce Arabic alphabets over feedback. The rise of Automatic Speech Recognition (ASR) has sparked a wave of innovation in mobile computing, human-computer interaction, information retrieval, and assistive communication, transitions and improvements, mainly owing to the advent of Automatic Speech Recognition (ASR) technology. ASR, the process of converting spoken acoustic signals into text, consists of acoustic, pronunciation, and language modeling and has been a subject of extended research across different languages with Arabic however mostly underrepresented

Common pattern recognition methods in ASR include hidden Markov models (HMM), Gaussian mixture models (GMM), artificial neural networks (ANN), and multi-layer perceptrons (MLP), among other methods, the most popular feature extraction methods are mel-frequency cepstral coefficients (MFCC) and spectrograms..Both HMM and neural network-based systems have demonstrated high accuracy in recognizing Classical Arabic

alphabets and verses. Tools like CMU Sphinx and systems such as ‘E-Hafiz’ and the ‘Tajweed checking system’ have shown notable results. For example, the ‘E-Hafiz’ system, which utilizes HMM and MFCC, achieved 92% accuracy for men and 91% for children. Other systems have used various techniques to improve pronunciation accuracy, including a mispronunciation detection system for Qalqalah letters with a 97.8% accuracy.

Deep learning (DL) algorithms can extract important features from audio data through multiple layers. An Arabic alphabet recognition model using RNN achieved an 82.3% accuracy, while another study reported 90.1% accuracy using neural networks for mispronunciation detection. This paper introduces DCNN, AlexNet, and BLSTM for Arabic alphabet classification, utilizing mel-spectrograms for feature extraction to avoid the complexity of MFCC with DCT. We address two tasks: multi-class classification for alphabet recognition and binary classification.

1.2 Motivation

The inspiration for this thesis is that recitation of the Quran is a form of worship and therefore is paramount in the Islamic faith whilst also being of the most difficult aspects for learners to master. Though it is (and actually needs to be) read properly to preserve the linguistic and spiritual integrity of the Quran, mastering it is far more than difficult for someone who is not a native Arabic speaker and lives in an area without qualified tutors. The traditional method of learning is effective but in many cases, it is not possible due to geographic constraints, cost, and time. But the other side of self-study is it does not effectively replace the kind of prompt feedback needed to make timely corrections.

Thankfully, Improvements in speech technology can address these difficulties.I have developed an interactive and collaborative support for reciting the Quran with Tajweed rules in a way that listens to user sounds using speech recognition and generates live feedback with the support of speech synthesis and deep learning. The project will serve as an all inclusive solution to the traditional yet outdated learning model and reach out to the masses with quality Quranic instruction. In order to provide the optimal learning experience, to deliver accurate corrective feedback and enable learners to recite the Quran in the best way.

1.3 Rationale of the Study

This study addresses the critical need for accurate Quranic recitation according to Tajweed rules, a fundamental aspect of Islamic practice. Mastering Tajweed is particularly challenging

for non-native Arabic speakers and individuals without access to qualified instructors.. Traditional learning methods, though effective, often suffer from accessibility issues due to geographic, financial, and time constraints. Additionally, self-study lacks immediate feedback, leading to potential reinforcement of incorrect pronunciation.

Advancements in speech recognition, synthesis, and deep learning technologies offer a solution to these challenges. This study aims to develop a voice assistant that provides real-time feedback on Quranic pronunciation based on Tajweed rules, making high-quality instruction more accessible and efficient. By integrating modern technology with traditional learning principles, the project seeks to enhance Quranic education, democratize access to accurate Tajweed instruction, and improve the overall learning experience for Quranic recitation learners.

1.4 Research Questions

Here the most highlighted thing is the research questions which are like the heart of a thesis research..I think if we discover something I need to learn first about this topic. When I learn and work this time I am identifying lots of problems. I think every problem is a question.

1. First question was when I am starting my journey. What type of data I need. Which programming language will I use?
2. How to classify data. Which algorithm is better for my thesis?
3. How effective is the voice assistant in improving the pronunciation accuracy of Quranic recitation according to Tajweed rules?
4. What are the common pronunciation errors made by Quranic learners, and how can the voice assistant help in identifying and correcting these errors?

1.5 Expected Output

1. Enhance Pronunciation Accuracy: Provide real-time feedback based on Tajweed rules to improve Quranic recitations.
2. Identify Errors: Accurately detect common pronunciation mistakes and offer corrective feedback.
3. Performance Analysis: Compare effectiveness with traditional learning methods.
4. User Insights: Gather satisfaction and usability data across demographics.

5. Speech Recognition: Capture and transcribe recitations accurately, accommodating various accents.
6. Real-Time Feedback: Offer immediate corrections and guidance for an interactive learning process.
7. Deep Learning Models: Develop and validate models to evaluate recitations against Tajweed rules.
8. Technical Challenges: Document and address challenges in integrating speech synthesis, recognition, and deep learning.
9. Increased Engagement: Evidence of improved motivation and engagement among learners.
10. Ethical Considerations: Address ethical and cultural concerns, providing mitigation strategies.

1.6 Report Layout

The report has 6 Chapters and it contain following information :

Chapter 1: Introduction

Explains the significance of correct Quranic pronunciation, research questions, motivation, rationale, and expected outcomes.

Chapter 2: Literature Review

Reviews existing approaches, their limitations, results, methods, and discusses research scope and challenges.

Chapter 3: Research Methodology

Details data collection, statistics, classifiers, figures, and implementation requirements.

Chapter 4: Results

Presents the outcomes and performance of the classifiers used in the research.

Chapter 5: Societal Impact

Examines the societal implications and importance of the research.

Chapter 6: Discussion and Conclusion

Discusses findings, conclusions, future work, and limitations of the study.

CHAPTER 2

BACKGROUND

2.1 Introduction

Accurate Quranic recitation, governed by Tajweed rules, is essential in Islam but challenging to master, especially for non-native speakers and those without access to skilled instructors. Traditional learning methods often lack the immediate, precise feedback needed for effective correction. This study aims to develop a voice assistant that enhances Quranic pronunciation learning by using advanced speech technologies like speech recognition, Text-to-Speech, and deep learning. The assistant provides real-time, accurate feedback based on Tajweed rules, making high-quality Quranic instruction more accessible and effective. By integrating modern technology with traditional learning, this project addresses the practical challenges faced by Quranic learners and advances educational technology in religious studies. Additionally, the system is designed to recognize and correct the pronunciation of full Quranic Ayat, providing comprehensive feedback on both individual alphabets and complete verses.

2.2 Preliminaries/Terminologies

Understanding the key terminologies and concepts is essential for comprehending the development and functionality of the voice assistant for Quranic pronunciation correction. This section outlines the fundamental terms and concepts relevant to this study.

1. Tajweed:

Definition: Tajweed refers to the set of rules governing the pronunciation of letters in the Quran. It ensures that each letter is given its due rights and characteristics when recited, maintaining the linguistic and spiritual integrity of the Quranic text.
Importance: Proper adherence to Tajweed rules is crucial for accurate Quranic recitation, as it preserves the intended meanings and enhances the beauty of the recitation.

2. Makharij (Articulation Points):

Definition: Makharij are the specific points in the vocal tract where different letters are articulated. Each Arabic letter has a unique articulation point, which affects its pronunciation.

Example: The letter "ق" (Qaaf) is articulated from the back of the tongue touching the soft palate, while "ب" (Ba) is articulated from the lips.

3. Sifaat (Characteristics):

Definition: Sifaat are the inherent characteristics or qualities of Arabic letters, such as heaviness, lightness, and breathiness. These characteristics influence the sound and pronunciation of each letter.

Example: The letter "ظ" (Zaa) is heavy, while "س" (Seen) is light.

4. Madd (Elongation):

Definition: Madd refers to the elongation of certain vowels in the Quranic recitation. There are specific rules dictating how long vowels should be stretched based on their context.

Example: In the word "الرحيم" (Ar-Raheem), the "ي" (Ya) is elongated due to the presence of Madd.

5. Idgham (Assimilation):

Definition: Idgham is the assimilation of one letter into another, causing them to be pronounced together smoothly. It typically occurs when a Noon Sakinah (ن) or Tanween (ًٌٍ) is followed by specific letters.

Example: In the phrase "من ربهم" (min rabbihim), the Noon Sakinah is assimilated into the letter Ra.

6. Qalqalah (Echoing Sound):

Definition: Qalqalah is the echoing or bouncing sound produced when certain letters are pronounced with a sukoon (ْ) or in a strong position. The letters that produce Qalqalah are ق, ط, ب, ج, and د.

Example: The letter "ق" in "أحدٌ" (Ahad) produces a slight echoing sound.

7. Waqf (Pausing):

Definition: Waqf refers to the rules governing where and how to pause during Quranic recitation. Proper pausing helps in maintaining the meaning and flow of the verses.

Example: Pausing at appropriate places, such as at the end of verses, ensures clarity and understanding.

8. Speech Recognition:

Definition:
Speech recognition technology enables machines to recognize and convert spoken language into text.

Relevance:
In the study of Quranic recitation, speech recognition captures and transcribes a learner's recitation for analysis.

ASR Pattern Recognition techniques
Hidden Markov Models (HMM)

Hidden Markov Models (HMMs) use probabilities to determine state transitions within a Markov chain. Recent developments indicate that HMM-based speech recognition systems can achieve high accuracy in recognizing Classical Arabic alphabets and verses. Notable tools include CMU Sphinx, an open-source HMM-based ASR system.

Gaussian Mixture Models (GMM)

GMMs represent normally distributed subclasses within a class, allowing automatic learning of data point classifications. The 'E-Hafiz system,' which uses HMM and MFCC, achieved impressive accuracy for adult and child users. Other systems, like those for improving alphabet pronunciation and the 'Tajweed checking system,' have also demonstrated success in recognizing and correcting Arabic pronunciations with varying accuracies.

Deep Learning in ASR
Multi-layer Perceptrons (MLP) and Artificial Neural Networks (ANN)

Deep learning (DL) algorithms have further enhanced ASR by learning hierarchical representations of data through multiple layers. An RNN-based Arabic alphabet recognition model achieved 82.4% accuracy for 20 alphabets.

Recent Studies and Techniques
Recent studies have explored mispronunciation detection using handcrafted feature extraction techniques and classifiers such as SVM, KNN, and NN, achieving varying degrees of accuracy.

Proposed Methodology
Deep Learning Algorithms
Therefore, this paper proposes to use the DCNN, AlexNet and the BLSTM neural networks to classify the Arabic alphabet. Specifically, my work is different from prior research in the choice

of the dataset with which DNNs are trained, the method used to prepare the features for the DNNs, the DNN architecture itself, and the results obtained.

Feature Extraction with Mel-Spectrograms

We use mel-spectrograms for feature extraction, which convert audio frames into a frequency-domain representation on a mel-scale. Unlike traditional MFCC, which requires a discrete cosine transform, our method leverages the mel-spectrogram directly.

Contributions

This research provides the following main contributions:

1. Arabic Alphabet Audio Dataset
2. Categorizing individual letters.
3. Accuracy of pronunciation detections
4. The tasks use DCNN, AlexNet and BLSTM.
5. **Ayat Pronunciation Correction:** The system is designed to recognize and correct the pronunciation of full Quranic Ayat, providing comprehensive feedback on both individual alphabets and complete verses from the Quran.

2.3 Related Works

R. Duan, T. Kawahara, M. Dantsuji, and H. Nanjo investigate methods for improving articulatory models in CAPT systems using multi-label learning and label correction. The study shows that providing articulatory feedback significantly enhances learners' pronunciation skills. The approach involves training deep neural networks (DNNs) to model multiple articulatory attributes simultaneously, reducing training time and improving accuracy. This research is relevant for developing a system that offers detailed, articulatory-based feedback for Quranic recitation [1].

A. Loukina, M. Lopez, K. Evanini, D. Suendermann-Oeft, A. V. Ivanov, and K. Zechner explore this study probes the link between pronunciation accuracy and how well non-native English speakers are understood. Using a large corpus of non-native speech, they found that pronunciation errors only partially account for intelligibility issues. The paper highlights the importance of considering both segmental and suprasegmental features in pronunciation training, which can be applied to developing a comprehensive feedback system for Quranic recitation that addresses both types of errors [2].

J. Bang, S. Kang, and G. Lee present the Postech English Speaking Assessment and Assistant (PESAA), a CALL system that integrates pronunciation and prosody assessments to provide comprehensive feedback to learners. PESAA uses speech recognition to align phonemes and detect pronunciation errors, while also assessing rhythm and phrase breaks. This integrated approach is beneficial for a Quranic pronunciation assistant, ensuring learners receive holistic feedback that includes both pronunciation and prosodic aspects [3].

M. Hamid, M. Talha, and S. Aslam focus on detecting pronunciation errors in Classical Arabic using deep learning models. The study demonstrates the effectiveness of convolutional neural networks (CNNs) in identifying phoneme-level pronunciation errors. The approach involves extracting Mel-frequency cepstral coefficients (MFCCs) from audio data and using them as input features for the CNN. This methodology is directly applicable to developing a system for Quranic pronunciation correction, providing a robust framework for detecting and correcting errors in Quranic recitation [4].

A. Loukina, K. Zechner, and K. Evanini investigate the use of machine learning techniques to enhance the detection and diagnosis of pronunciation errors in language learning. The study employs a multi-layer perceptron (MLP) to classify pronunciation errors and provides detailed feedback to learners. The findings underscore the importance of using advanced machine learning models to improve the accuracy and specificity of pronunciation feedback. This work supports the development of an advanced error detection system for Quranic recitation that can provide precise, actionable feedback [5].

2.4 Comparative Analysis and Summary

Comparative Analysis

Table 1. Comparative Analysis and Summary

Technologies	**Name and Years**	**Work**	**Result**
Speech Recognition	Loukina et al. (2015)	Listen for understanding in less than perfect foreign speech	This dual approach is essential for our voice assistant, ensuring comprehensive feedback that includes both segmental and suprasegmental features. the effectiveness of integrating speech

and Synthesis Technologies			recognition and synthesis technologies to provide accurate pronunciation feedback.
	Bang et al. (2013)	Emphasize the importance of combining pronunciation with prosody assessments	
Deep Learning Models	Hamid et al. (2021)	use of CNNs for detecting pronunciation errors in Classical Arabic, which directly applies to Quranic recitation	This method using MFCCs as input features provides a robust framework for our system's error detection module. Deep Learning Models:
	Duan et al. (2018)	multi-label learning and label correction to improve articulatory models.	This approach enhances the accuracy and efficiency of pronunciation feedback, which is crucial for the precise correction of Quranic recitation.
	Loukina, Zechner, and Evanini (2015)	Utilize an MLP to classify pronunciation errors, providing detailed, actionable feedback.	This method can be adapted to offer specific guidance on Tajweed rules, ensuring learners receive precise corrections..
Feedback Mechanisms	Bang et al. (2013)	emphasize the importance of real-time feedback in language learning. Bang et al. integrate prosodic features.	This system will benefit from these insights by providing immediate, holistic feedback that addresses both pronunciation and prosodic errors in Quranic recitation.
	Hamid et al. (2021)	focus on phoneme-level corrections	

The selected studies underscore the effectiveness of combining advanced speech recognition, synthesis, and deep learning technologies to enhance pronunciation training. By integrating

these approaches, our voice assistant can provide comprehensive, real-time feedback on Quranic recitation, addressing both segmental and suprasegmental errors. This comparative analysis highlights the need for a multi-faceted approach that incorporates accurate error detection, detailed feedback, and real-time correction to ensure learners achieve precise Quranic pronunciation according to Tajweed rules.

2.5 Scope of the Problem

The scope of the problem addressed in this research involves several critical aspects related to Arabic speech recognition and pronunciation correction, particularly focusing on Classical Arabic (CA) and Quranic recitation:

1.Underrepresentation of Classical Arabic in Research:

Despite the significant importance of Classical Arabic, particularly in religious contexts, it has been underrepresented in automatic speech recognition (ASR) research compared to other languages.

2.Complexity of Arabic Pronunciation:

Proper pronunciation of Arabic letters is crucial because incorrect pronunciation can significantly alter the meanings of words. This complexity is exacerbated in Quranic recitation, where the rules of Tajweed must be strictly followed.

3.Challenges for Non-native Speakers:

Non-native Arabic speakers, and even native speakers without proper training, often struggle with accurate pronunciation due to the lack of immediate and precise feedback from traditional learning methods.

4.Lack of Comprehensive Datasets:

There is a scarcity of comprehensive audio datasets specifically tailored for training models to recognize and correct Arabic pronunciation, necessitating the collection of new data from both experts and novices.

5.Technical Challenges in ASR:

Developing effective ASR systems for Arabic involves addressing technical challenges such as feature extraction, model training, and achieving high accuracy in both recognition and pronunciation evaluation.

6.Integration of Modern Technology:

There is a need to integrate advanced speech technologies, including deep learning models, to provide real-time, accurate feedback on pronunciation. This integration aims to make high-quality Quranic instruction more accessible and effective.

7.Two-stage Approach:

The problem is divided into two main tasks:

Arabic Letter Recognition: Training models to accurately recognize individual Arabic letters.

Pronunciation Quality Evaluation: Evaluating and correcting the pronunciation quality of these letters and full Quranic Ayat based on Tajweed rules.

8.Application of Deep Learning Models:

Employing deep convolutional neural networks (DCNN), AlexNet with transfer learning, and bidirectional long short-term memory (BLSTM) networks to enhance the accuracy and reliability of the ASR system.

9.Augmentation for Improved Accuracy:

Using data augmentation techniques to enhance the audio dataset and improve the performance of the learning models.

10.Practical Implications:

The ultimate goal is to develop a comprehensive tool that can assist learners in improving their pronunciation of both individual Arabic letters and complete Quranic verses, thus bridging the gap between traditional learning and modern technological solutions.

2.6 Challenges

The development of an automatic speech recognition (ASR) system for Arabic, particularly focusing on Classical Arabic and Quranic recitation, involves several significant challenges:

1. **Complexity of Arabic Phonetics:**

Arabic phonetics are highly intricate, with each letter having specific articulation points and characteristics. Proper pronunciation is crucial for maintaining the correct meanings of words, making the task more complex than in some other languages.

2. **Underrepresentation in Research:**

Classical Arabic, despite its importance, has been underrepresented in ASR research compared to other languages. This lack of focus has led to fewer available resources, tools, and models tailored for Arabic speech recognition and pronunciation correction.

3. **Scarcity of Quality Datasets:**

There is a limited availability of comprehensive, high-quality audio datasets for training ASR models specifically for Arabic pronunciation. Collecting and annotating such datasets from both expert and novice speakers is time-consuming and resource-intensive.

4. **Adherence to Tajweed Rules:**

Quranic recitation requires strict adherence to Tajweed rules, which govern the pronunciation and articulation of Arabic letters. Developing a system that can accurately evaluate and provide feedback based on these rules is a complex task.

5. **Variability in Pronunciation:**

There is significant variability in pronunciation between native and non-native speakers, and even among native speakers from different regions. This variability poses a challenge for creating models that can generalize well across different speakers.

6. **Real-time Feedback Mechanism:**

Providing immediate and precise feedback to learners in real-time is crucial for effective pronunciation correction. Designing and implementing a robust feedback mechanism that can operate in real-time adds another layer of complexity.

7. **Technical Challenges in Model Development:**

Training deep learning models such as DCNN, AlexNet, and BLSTM requires substantial computational resources. Fine-tuning these models to achieve high accuracy for both letter recognition and pronunciation quality evaluation involves extensive experimentation and optimization.

8. **Feature Extraction:**

Effective feature extraction from audio data is critical for the performance of ASR systems. While traditional methods like MFCC are well-known, exploring and implementing alternative methods such as mel-spectrograms while ensuring they capture all relevant features is challenging.

9. **Balancing Model Performance:**

Achieving a balance between high accuracy and computational efficiency is essential. Models must be accurate enough to provide reliable feedback yet efficient enough to operate on standard hardware without significant latency.

10. **User Accessibility:**

Ensuring the system is accessible to a wide range of users, including those with limited technical skills, is crucial. The interface must be user-friendly and intuitive to facilitate widespread adoption.

11. **Evaluation and Validation:**

Rigorous evaluation and validation of the system are necessary to ensure its reliability and effectiveness. This involves testing with diverse datasets, including unseen data, to confirm the model's robustness and accuracy.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

Accurate Quranic recitation, governed by Tajweed rules, is essential but challenging to master, especially for non-native speakers. Traditional learning methods often lack real-time feedback necessary for effective correction.

This study aims to develop a voice assistant that enhances Quranic pronunciation using advanced speech technologies like speech recognition, text-to-speech, and deep learning. The assistant provides real-time, accurate feedback based on Tajweed rules, making high-quality Quranic education more accessible and effective. This project bridges traditional instruction with modern technology to improve the learning experience for Quranic recitation.

In this section, I present the approach and stages of this research, which consists of 5 main steps: data collection, data preprocessing, feature extraction, network training, and prediction of new data. These stages are depicted in the system architecture diagram below:

Processes which took place in the preceding section were about the first and second steps: Data Collection and Preprocessing. Stage 3: Feature Extraction: The third stage is Feature Extraction, where the model extracts features from raw audio data and incorporates them into the neural network during training.

After training, the network is used to classify unseen testing data. The final step involves comparing the results from the training data with those from the unseen data, estimating accuracy, and presenting confusion matrices for each class.

System Architecture In the diagram, we can see the whole system architecture from data collection and model training to the measurement of the final results. They utilize deep learning – specifically D-CNN, AlexNet, and BLSTM – in the Arabic alphabet classification and pronunciation classification tasks. The main difference exists in the data and classes of data.

The evaluation in this work is carried out in a speaker-independent setup, which will guarantee how well the models can generalize to a new speaker that has not been ever seen during training.

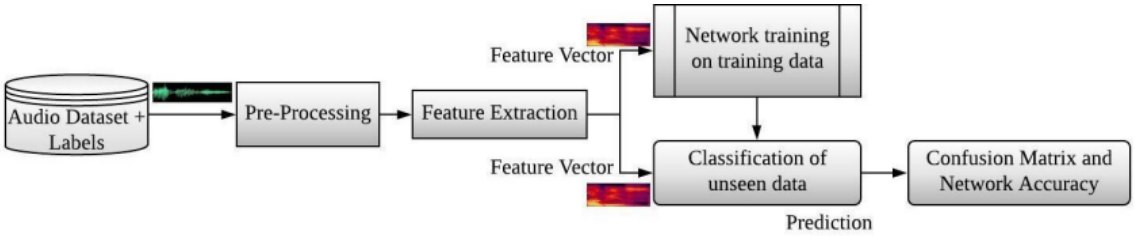

Figure 3.1.1: Proposed system architecture

3.2 Feature Extraction

In automatic speech recognition (ASR) systems, feature extraction plays a crucial role by providing a compact representation of speech waveforms. Instead of using raw speech signals directly, the classification algorithms operate on these extracted features. This approach helps to reduce redundancy and eliminate irrelevant information from large datasets, which can otherwise demand significant memory and computational power, and may lead to overfitting issues.

For our study, CNNs (AlexNet and DCNN) automatically learn that by converting raw audio data to mel-spectrograms. In CNNs it is very important because it will transform the input image as it is filtered and extract many features more and move focus with audio alphabet classification to capture the most features related. In our work, we used the FC-layer's filtered features for classification.

However, BLSTM networks need some information to be extracted manually. Feature Extraction: For the BLSTM network, spectral features are extracted from raw audio data, and these features are then saved for training, testing, and evaluating the audio alphabet. We first attempted to use BLSTM with mel-spectrum, but these experiments failed with poor results. Therefore, we resorted to hand-engineering features.

For the raw audio data, we extract twelve spectral features: spectral centroid, spectral spread, spectral skewness, spectral kurtosis, spectral entropy, spectral flatness, spectral crest, spectral flux, spectral slope, spectral decrease, spectral roll-off point, and pitch. These in general are used for perceptual analysis, machine learning, deep learning applications etc., They help identify different voice components notes, pitch, rhythm, melody

3.3 Neural Network Model Training

Developing an effective neural network model requires a thorough understanding of the training data and the ability to analyze and interpret the input data accurately. The neural network maps input data to the desired output by adjusting its parameters to minimize error.

This process involves iterating through various configurations to identify the optimal set of model parameters that best capture the relationships within the data.

To achieve this, I tested several network parameters and, through empirical analysis, determined the optimal values for my models. These parameters, as shown in Table 1, include the learning rate, number of epochs, batch size, and optimization algorithm.

Table2: Training option of the neural network.

Parameters	DCNN	AlexNet	BLSTM
Learning Rate	0.0001	0.00001	0.001
Epochs	36	36	99
Batch Size	76	374	125
Optimizing Algorithm	Adam [44]	Adam [44]	Adam [44]

3.4 Classification for Deep Learning Models

Deep learningDeep learning (DL) is a subfield of machine learning which contains a wide range of models and respective learning algorithms. Which model to use is largely dependent on the dataset and the problem that we are trying to solve. We applied for the utilis. DatasetAlphabetAudio class in this study, which has been used to train and test deep learning models for the best learning and minimal loss.

We also trained our model from scratch (since no pre-trained model on the Quranic dataset has been established) and fine-tuned existing models ([LSTM (Graves, 2013), Squeezebert (Keeping Simple Matters Simple et al., 2019), ALBERT (Lan et al., 2019]) to make it tailored to our purposes. To classify audio alphabets we constructed two main types of models,

(a) DCNN (Deep Convolutional Neural Network)

Unsurprisingly, DCNNs are especially well-suited to image data. In our case it is that they take mel-spectrograms which are obtained by some length of audio signal. You can divide them into layers that can extract features automatically, and the number of layers of the better without the need for engineering characteristics manually. The extracted features are then used for classification.

(b) AlexNet:

AlexNet, most successful in image classification tasks, was fine-tuned for celeba dataset using transfer learning approach. Transfer learning is an approach to using a pre-trained model in a new but related problem and learning is utilized from the initially trained model and applied to our task at hand. This makes the training process faster and more accurate by starting from the effective pre-trained model weights.

(C) BLSTM(Bidirectional Long Short-Term Memory Network)

In any application where temporal dependencies hold, as in the case of speech, BLSTM networks can be very appealing. BLSTMs have the capability to handle data in both the forward and backward direction so it addresses the issue of long-term dependencies with traditional RNNs. In our implementation, BLSTM networks are used to analyze the sequence of audio signals and learn to discriminate between correctly spoken pronunciations and mispronunciations.

Convolutional Neural Network

Given the deep structure of convolutional neural networks (CNN) we use separate filters to treat image data, the classification prediction problem, and the regression prediction problem. This kind of structure is the same as deep convolutional neural networks (D-CNN). In the case of the CNN described here, mel-spectrograms are extracted from the raw audio data (wav files) and filters are used to learn which number of features are important in representing data. Convolutional layers go over the mel-spectrogram to learn features at each layer and inner layers make use of these to learn useful features which help in classifying the inputs.

Figure 3.4.1: 24-layer architecture used by the DCNN in this work The following algorithm represents the process of the classification task made using the DCNN.

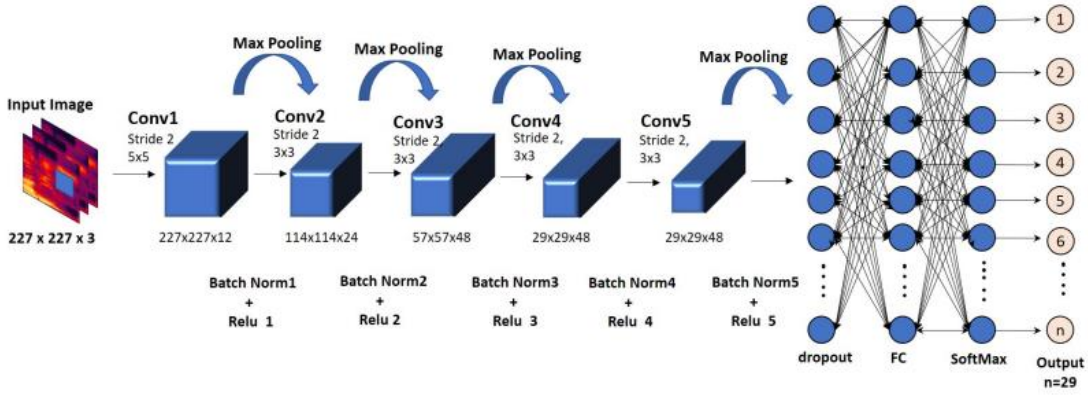

Figure 3.4.1: Deep convolutional neural network architecture.

Data Preparation:

1. **Input:**

   ads: The audio dataset containing recordings.

   nLabels: The total number of categories for classification.

   Labels: A list defining each class label.

   numBands: Number of frequency bands used for analysis.

   Seg_Dr: Duration (in seconds) of each audio segment.

   Hop_Dr: Hop duration (in seconds) between segments.

   Frame_Dr: Frame duration (in seconds) for spectrogram creation.

2. **Preprocessing:**

   Split the audio dataset (ads) into training (adsTrain) and testing (adsTest) sets.

   Convert both sets into Mel-spectrograms using melspectrogram function with specified parameters. This transforms audio data into a visual representation suitable for CNNs.

   Generate training labels (YTrain) by assigning corresponding class labels from Labels to each training segment in adsTrain.

   Similarly, assign class labels (YTest) to testing segments in adsTest.

Model Training:

1. **Define Network Architecture:**

   Specify the image size (imageSize) for the input Mel-spectrograms.

   Design the DCNN architecture using a function like trainNetwork. This function typically takes training data (XTrain), labels (YTrain), desired network layers (layers), and training options as arguments.

Evaluation:

1. **Testing:**
   Use the trained network (trainedNetwork) and the testing data (XTest) with the classify function to obtain predicted class labels (class) and their corresponding scores.
2. **Accuracy:**
   Calculate the overall accuracy (Accuracy) by comparing the predicted labels (YPredicted) with the actual test labels (YTest).
3. **Confusion Matrix:**
   Create a confusion matrix (Confusion_Matrix) to visualize the model's performance on each class by comparing predicted and actual labels.

Utilize a function like confusion_matrix to display the confusion

Transfer Learning with AlexNet

Transfer learning (TL) is a research area in machine learning (ML) that involves leveraging knowledge acquired from solving one problem and applying it to another, related problem.For the predicted images, use the pre-trained networks, which save a lot of time and ima go further to designing a new network. Pre-trained networks save a lot of computational manpower. Broadly speaking, there are two different types of transfer learning:

Feature Extractor + Model Trainer using pre-trained network

Transfer learning: setting the initial parameters for the model (i.e., the weights) to the learned weights from the pre-trained network.

This time, I will use the second way of fine-tuning a pre-trained network to take advantage of an already well-designed and well-trained neural network rather than building it from scratch.

AlexNet, a CNN that has been trained on millions of images over the 1000 ImageNet database categories, has been shown to provide excellent representation wherever we need. AlexNet accepts size-sbwp 227 × 227 × 3 where 227 number of frames, 227 number of bands and 3 numbers of spectrum. It pre-trained AlexNet has 25 layers architecture.

For my classification tasks, I convert raw audio data into mel-spectrograms, resize them to match the standard input size of AlexNet, and feed them into the model. Although AlexNet was originally trained on 1000 categories, my focus is on 29 categories related to alphabet

pronunciation classification. Therefore, I modified the last three layers of AlexNet (including the fully connected layer Fc8, the softmax layer, and the classification layer) to be fine-tuned for my specific classification tasks. With these adjustments, the network can now automatically extract features from the mel-spectrograms and learn from my dataset.

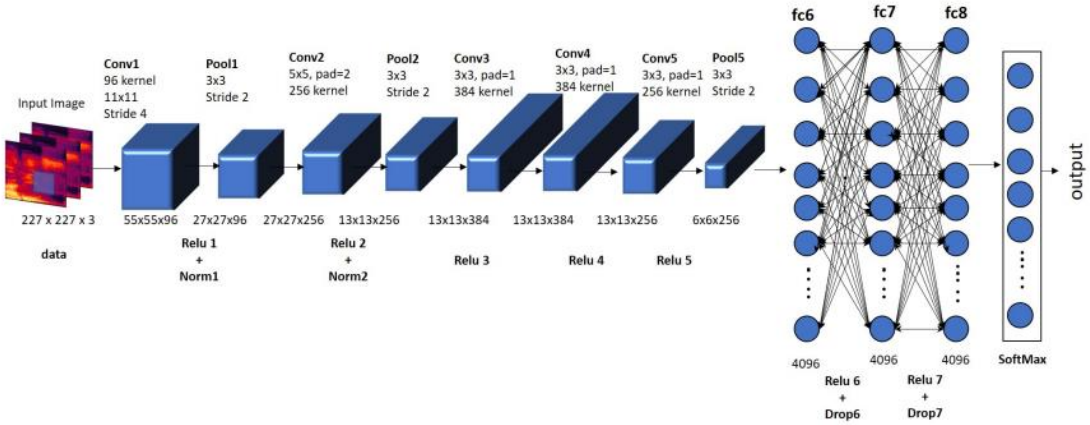

Figure 3.4.2: AlexNet architecture.

Inputs:

- ads: The audio dataset containing recordings.
- nLabels: The total number of categories for classification.
- Labels: A list defining each class label.
- numBands: Number of frequency bands used for analysis.
- Seg_Dr: Duration (in seconds) of each audio segment.
- Hop_Dr: Hop duration (in seconds) between segments.
- Frame_Dr: Frame duration (in seconds) for spectrogram creation.

Outputs:

- accuracy: The overall accuracy of the model (percentage).
- predicted_labels: A list containing the predicted class labels for each test sample.

Steps:

1. **Data Preparation:**
    - Split Dataset: Divide the audio dataset ads into training and testing sets using a function like split_data.

      training_data, testing_data = split_data(ads)
    - Mel-spectrogram Conversion: Convert both training and testing data into Mel-spectrograms using the melspectrogram function with specified parameters. This transforms audio data into a visual representation suitable for CNNs.

        - training_features = melspectrogram(training_data, Seg_Dr, Hop_Dr, Frame_Dr, numBands)
        - testing_features = melspectrogram(testing_data, Seg_Dr, Hop_Dr, Frame_Dr, numBands)
    - Label Assignment: Extract labels for the training data from the provided Labels list.
        - training_labels = extract_labels(training_data, Labels)

2. **Model Training:**
    - Define Network Architecture: Specify the image size (imageSize) for the input Mel-spectrograms.
    - Train the Model: Train the DCNN model using the trainNetwork function. This function typically takes training features (training_features), labels (training_labels), desired network layers (layers), and training options as arguments.
        - trained_model = trainNetwork(training_features, training_labels, layers, options)
3. **Evaluation:**
    - Testing: Use the trained model (trained_model) and the testing features (testing_features) with the classify function to obtain predicted class labels (predicted_labels) and their corresponding probabilities.
        - predicted_labels, probabilities = classify(trained_model, testing_features)
    - Accuracy Calculation: Calculate the overall accuracy (accuracy) by comparing the predicted labels (predicted_labels) with the actual labels from the testing data. Functions like calculate_accuracy can be used.
        - accuracy = calculate_accuracy(predicted_labels, testing_data.Labels)
    - Confusion Matrix (Optional): Create a confusion matrix to visualize the model's performance on each class by comparing predicted and actual labels. Utilize a function like confusion_matrix (not shown here).

Recurrent Neural Network

Recurrent Neural Networks (RNNs) are specifically designed to address sequence prediction problems. A particular type of RNN, the Long Short-Term Memory (LSTM) network, ensures that information is retained during training and can learn long-term dependencies, overcoming the limitations of standard .

RNNs.

Inputs:

- ads: The audio dataset containing recordings.
- nLabels: The total number of categories for classification.
- Labels: A list defining each class label.

Outputs:

- alexnet_accuracy: The overall accuracy of the AlexNet model (percentage).
- predicted_labels: A list containing the predicted class labels for each test sample.

Steps:

1. **Data Preprocessing**:
    - Split data (assumed as a separate step): The algorithm assumes the data is already split into training and testing sets. Ensure this is done before proceeding.
    - Mel-spectrogram Conversion (assumed): Similar to the previous algorithm, we assume the training and testing data are converted into Mel-spectrograms with appropriate parameters.
2. **Model Preparation**:
    - Define Network Architecture: Specify the input size (inputSize) for the Mel-spectrograms.
    - Load AlexNet: Initialize a pre-trained AlexNet model using a function like load_alexnet.
    - For transfer learning, isolate all layers in the pre-trained AlexNet except for the final three layers (designated as layersTransfer).These initial layers help learn general features useful for audio classification.
3. **Model Training:**
    - Freeze Transfer Layers (Optional): Consider freezing the weights of the transferred layers (layersTransfer) to prevent them from being updated during training. This can help the model focus on learning task-specific features in the final layers.

- Add New Layers: Design and add new layers (layers_1) on top of the transferred layers (layersTransfer) to adapt the model to the specific audio classification task. These layers will focus on learning the audio specific features.
- Train the Model: Train the modified AlexNet model (netTransfer) using the trainNetwork function with training features (XTrain), labels (YTrain), the new layers (layers_1), and desired training options.

4. **Evaluation:**
    - Testing: Use the trained model (netTransfer) and the testing features (XTest) with the classify function to obtain predicted class labels (predicted_labels) and their corresponding probabilities.
    - Accuracy Calculation: Calculate the overall accuracy (alexnet_accuracy) by comparing the predicted labels (predicted_labels) with the actual labels from the testing data (YTest).
    - Confusion Matrix (Optional): Create a confusion matrix to visualize the model's performance on each class by comparing predicted and actual labels. Utilize a function like confusion_matrix (not shown here).

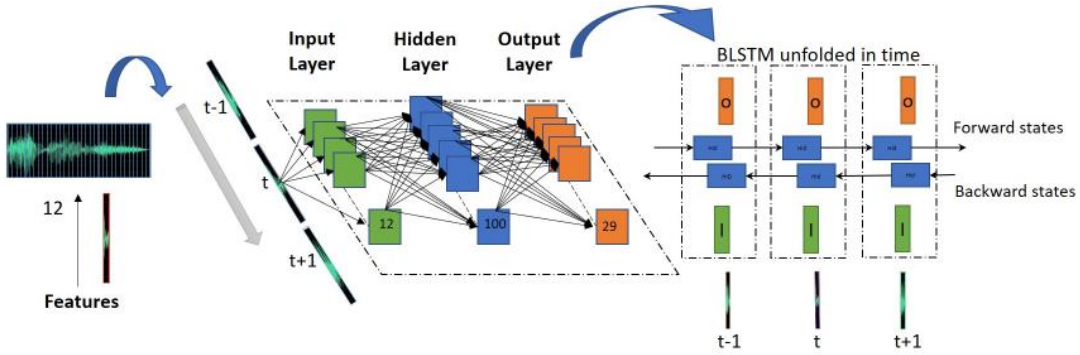

Figure 3.4.3: Bidirectional long short-term memory (BLSTM) architecture.

3.5 Data Collection Procedure/Dataset Utilized

This section details the data collection procedures and the datasets utilized for developing the voice assistant for Quranic pronunciation error correction. The datasets comprise a combination of publicly available data from online platforms and real-world recordings collected from a Hafizi madrasah.

Data Collection Procedure

To collect the data required for developing the voice assistant for Quranic pronunciation correction, two main sources were utilized: online platforms and real-world recordings. Data from Kaggle and Data World were sourced, comprising Arabic speech recognition and pronunciation datasets. Kaggle provided recordings of Arabic phonemes, words, and sentences, annotated with corresponding text transcriptions, covering various accents and dialects. Data World offered additional Arabic speech samples, enhancing the diversity and robustness of the dataset with variations in pronunciation and speaker demographics.

To ensure accuracy and authenticity, real-world recordings were collected from a Hafizi madrasah, where students and instructors rigorously trained in Tajweed rules provided recitations. A structured approach ensured high-quality audio samples, involving participant selection across different proficiency levels, recording in a quiet environment using high-quality microphones, and obtaining consent while adhering to ethical guidelines. These recordings were meticulously annotated by Tajweed experts, detailing correct pronunciations and common learner errors.

The Quran sura audio dataset consists of WAV files, each approximately 10 seconds long, sampled at 44.1 kHz with a 16-bit depth and in stereo. The dataset includes 10 surah recordings from 40 different scholars, with a total of 400 sura. To effectively train, test, and validate the model, the dataset will be divided into three subsets: 80% for training, 10% for testing, and 10% for validation. This structured approach ensures that the data is well-distributed across the different scholars and sura, providing a comprehensive basis for training the model, evaluating its performance, and fine-tuning to prevent overfitting.

The combination of these datasets formed a comprehensive resource for training and evaluating the voice assistant, ensuring a realistic and effective learning tool for Quranic recitation.

Noise Reduction

There are various algorithms and tools to do speech enhancement. The noise-of-spectrogram subtraction and voice activation detection (VAD) was utilized in the composition of our dataset by handling the audio samples. We estimated the background noise power spectrum (Pxx) of the input signal. As evidenced in Fig. 2, The proposed method is robust against background noise as shown in Fig. 2a and Fig. 2b and as Fig. 2c illustrates the No-GBU approach neither benefiting the performance of the GendedNet nor disrupting it.

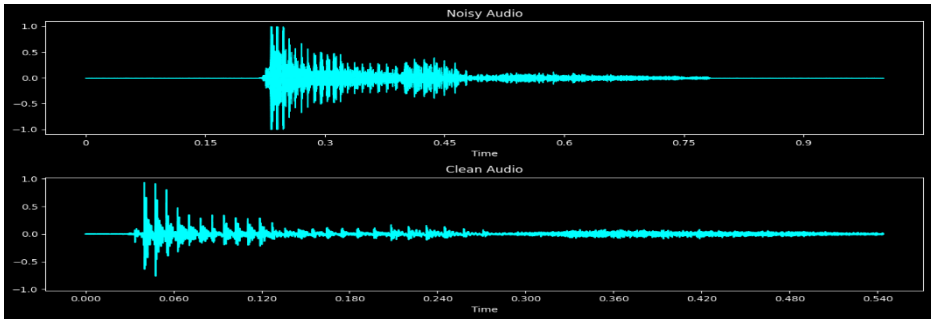

Figure 3.5.1: Bidirectional long short-term memory (BLSTM) architecture.

Audio Segmentation and Silence Removal

Silence within audio signals often adds unnecessary complexity to the data, making it harder to extract useful information. We minimized silence in each clip to improve classification accuracy. The recorded audio files, consisting of 29 letters, were segmented using a speech detection algorithm that identifies and removes silence at the start and end of each clip. This algorithm transforms the audio signal into a time-frequency representation and calculates short-term energy and spectral spread for each frame. These measures are smoothed over time to eliminate noise spikes, and the combined masks identify speech segments. Figure 3 demonstrates the detected speech levels after silence removal

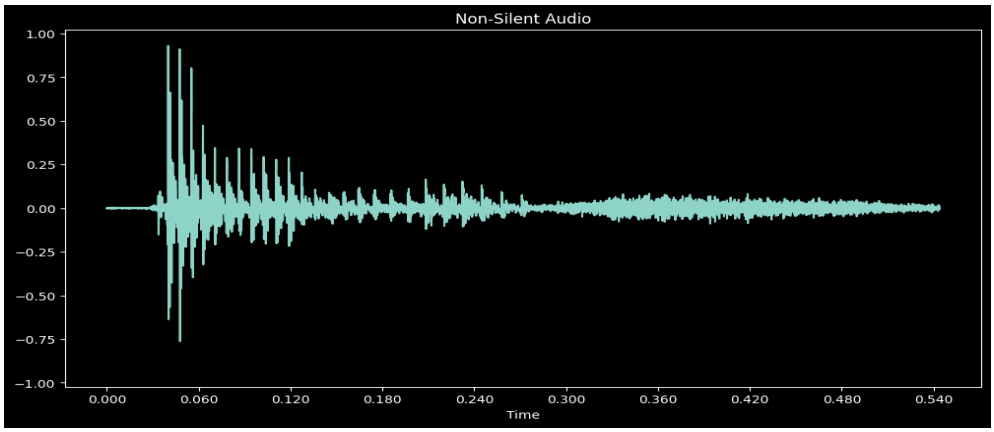

Figure 3.5.2 : Audio levels are marked according to speech detected in the recorded file before splitting.

Data Augmentation

Data Augmentation is the most common trick that people use for more training data. For our audio dataset, I introduced a pitch change factor as we wanted to generate a few more samples while keeping the original style of speaking. Vary up to 0.3 pitch: generate 20 new word samples for each alphabet letter We double-checked that everything sounded right by playing

things for an expert. In the following, Fig 4a shows the time-frequency representation of the original data, and Fig 4b shows the augmented data.

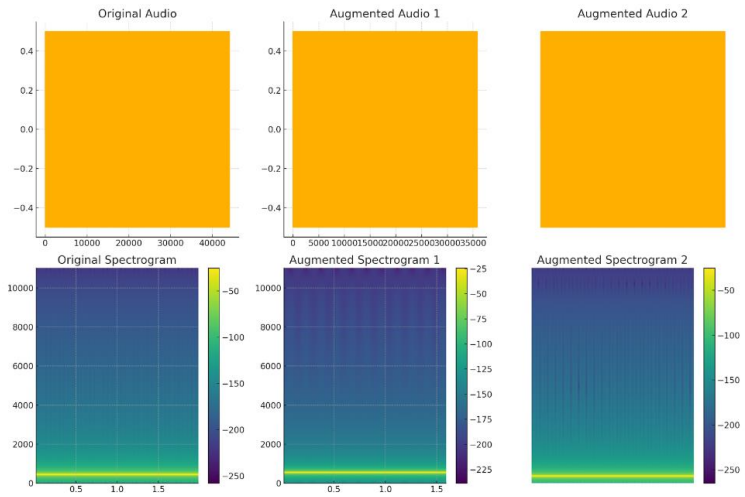

Figure 3.5.3: Time-frequency representation of audio file: (a) Non data-aug, (b) Data-aug

3.6 Statistical Analysis

Statistical analysis is crucial to evaluate the performance and effectiveness of the voice assistant for Quranic pronunciation correction. This section outlines the statistical methods used to analyze the collected data and assess the system's accuracy and user satisfaction.

Descriptive Statistics

Purpose: Summarize the main features of the collected data from Quranic learners and Tajweed experts.

Metrics: Mean, median, mode, standard deviation, and range of pronunciation scores before and after using the voice assistant.

Inferential Statistics

Purpose: Draw conclusions and make inferences about the overall population of Quranic learners based on sample data.

Tests:

T-tests: Compare the pronunciation scores of learners before and after using the voice assistant to determine if there are significant improvements.

ANOVA (Analysis of Variance): Analyze differences between multiple groups (e.g., beginners, intermediate, and advanced learners) to understand the system's impact across different proficiency levels.

Correlation Analysis

Purpose: Determine the relationship between various factors such as user engagement, frequency of practice, and improvement in pronunciation accuracy.

Metrics: Pearson correlation coefficients to measure the strength and direction of relationships between variables.

User Feedback Analysis

Purpose: Evaluate user satisfaction and the usability of the voice assistant through surveys and interviews.

Metrics: Likert scale ratings, thematic analysis of qualitative feedback, and sentiment analysis to gauge overall user experience and acceptance.

3.7 Proposed Methodology/Applied Mechanism

This section outlines the proposed methodology and mechanisms applied in developing the voice assistant for Quranic pronunciation correction. The methodology integrates speech technologies with pedagogical principles of Tajweed to provide accurate and real-time feedback.

Data Collection and Preprocessing

Data Sources: Online platforms (Kaggle, Data World) and real-world recordings from Hafizi madrasah.

Preprocessing Steps: Standardization, segmentation, and feature extraction (e.g., MFCCs).

Speech Recognition Module

Technology: Google Cloud Speech-to-Text, IBM Watson Speech to Text, or Mozilla DeepSpeech.

Function: Capture and transcribe user recitations into text and phonetic representations.

Speech Synthesis Module

Technology: Google Text-to-Speech, Amazon Polly, or IBM Watson Text to Speech.

Function: Generate high-quality audio recitations of Quranic verses for learners to emulate.

Pronunciation Evaluation Module

Framework: TensorFlow or PyTorch.

Model: Deep learning models (e.g., CNNs, RNNs) trained to evaluate pronunciation accuracy based on Tajweed rules.

Function: Analyze learner's recitation, identify errors, and provide detailed feedback.

Feedback Generation Module

Mechanism: Real-time feedback on pronunciation errors with specific guidance on corrections.

Integration: Combine pronunciation and prosody assessments to offer holistic feedback.

3.8 Implementation Requirements

Hardware Requirements:

Computing Devices: High-performance servers for model training and deployment

Recording Equipment: High-quality microphones and audio recording devices for collecting real-world data.

User Devices: Smartphones, tablets, or computers with internet access for end-users.

Software Requirements:

Programming Languages: Python (for machine learning and NLP), JavaScript (for front-end development).

Frameworks and Libraries:

TensorFlow/PyTorch for deep learning model development.

Flask/Django for backend development.

React/Angular for front-end development.

NLTK for natural language processing tasks.

APIs: Google Cloud Speech-to-Text, IBM Watson Speech to Text, Google Text-to-Speech, Amazon Polly.

Data Requirements

Quranic Recitation Dataset: Annotated recordings from online platforms and Hafizi madrasah.

Annotation Tool: Custom tool for labeling data with correct and incorrect pronunciations according to Tajweed rules.

Training Data: High-quality, diverse samples covering various accents, dialects, and proficiency levels.

By adhering to these implementation requirements, the voice assistant can be effectively developed, ensuring accurate and efficient Quranic pronunciation correction based on Tajweed principles.

CHAPTER 4

EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Introduction

Automatic Speech Recognition (ASR) is pivotal in fields such as human-computer interaction, mobile computing, information retrieval, and assistive communication. ASR systems convert spoken language into text, encompassing acoustic, pronunciation, and language models. While English and Mandarin ASR systems have been extensively researched and refined, Arabic ASR systems remain less developed. This study aims to improve Arabic ASR by addressing the phonetic intricacies of the Arabic language. We utilized Deep Convolutional Neural Networks (DCNN), AlexNet with transfer learning, and Bidirectional Long Short-Term Memory (BLSTM) networks to enhance the classification and pronunciation of the Arabic alphabet. Data augmentation was employed to expand the dataset, prevent overfitting, and enhance model generalization. I have collected and augmented audio samples from native and non-native speakers for both correct and incorrect pronunciations. Hold-out validation and 5-fold cross-validation were performed to determine the best models.

4.2 Results and Discussion

Dataset Description

My dataset includes speech samples from various sources. I obtained 12 audio clips from the web and recorded an additional 276 samples. These recordings involved 25 male experts and 20 children, each contributing 4 samples per alphabet. The raw audio (PCM 16 bit) was captured at a sampling rate of 48.1 kHz. To enrich the dataset and improve model performance, data augmentation techniques were applied, resulting in a total of 476 data files after augmentation.

Pronunciation Classification Dataset

To distinguish between accurate and inaccurate pronunciations, a binary classification task was performed. Speech samples were collected from 30 non-native adult males, yielding approximately 700 correct pronunciations and 650 incorrect ones. After applying data augmentation, I achieved a balanced dataset with roughly 180 samples per alphabet, including 30 source samples and around 150 augmented samples per letter. This balanced approach was

crucial for accurate pronunciation classification. The detailed breakdown of the pronunciation classification dataset is presented in Table 3.

Table 3:Dataset of Alphabet Classification

Category	Description	Details
Data Source	Type	Quantity
Web Audio	Audio Clips: 15	Recordings
Male Adults (Native)	8 samples per speaker	
Male Adults (Non-Native)	6 samples per speaker	
Children (7-12 years)	5 samples per speaker	
Children (13-18 years	3 samples per speaker	
Audio Format	Raw PCM 16 bit	
Sampling Rate	44.1 kHz	
Source Data	Total Samples: 216	
Augmented Data	Samples per Alphabet: 140	
Total Dataset	Files: 504	

Confusion Matrix

The confusion matrix categorizes model predictions as true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The overall accuracy (ACC) is calculated using the formula:

ACC=$TP+TN+TP+TN+FP+FN$×100%

Validation Strategies

Two common validation techniques were used to prevent overfitting:

1.**Hold-out Validation:** The dataset is split into training and testing sets. The testing set remains unseen during training, providing an unbiased performance estimate. However, this method utilizes only a portion of the data, and results can vary depending on the specific split.

2.K-fold Cross-Validation: This robust technique splits the data into 'K' folds. Each fold serves as the test set once, while the remaining folds form the training set. This ensures all data

is used for validation, leading to a more reliable performance estimate. We employed 5-fold cross-validation for this purpose.

Model Evaluation Results

The performance of DCNN, AlexNet (transfer learning - TL), and BLSTM models was evaluated on the Arabic alphabet classification task, both before and after data augmentation.

Before Data Augmentation

The models exhibited varying performance levels:

- DCNN: Achieved moderate accuracy (around 65%) with all validation methods. Certain alphabets like 'jeem' and 'wao' were classified perfectly, while others struggled.
- AlexNet (TL): Outperformed DCNN with accuracy reaching nearly 80% across validation strategies. Several alphabets achieved 100% accuracy, but limited data elements led to lower performance for others.
- BLSTM: Achieved the lowest accuracy (around 53%) across all validation methods. While some alphabets performed well, others showed significant difficulties.

After Data Augmentation

Data augmentation significantly improved performance for all models:

- DCNN: Accuracy increased dramatically to over 93%.
- AlexNet (TL): Achieved near-perfect accuracy (around 98%).
- BLSTM: Showed substantial improvement, reaching accuracy close to 90%.

Model Accuracy Summary

Table 4: Alphabet Classification after and before Data Augmentation

Model	**Validation Method**	**Accuracy before (Augmentation)**	**Accuracy (after Augmentation)**
DCNN	Random Split	65.89%	95.95%
	Hold-Out	64.56%	93.32%
	5-fold CV		93.46%
	Random Split	78.03%	98.41%

AlexNet (TL)	Hold-Out	78.73%	96.72%
	5-fold CV		96.36%
BLSTM	Random Split	53.18%	87.90%
	Hold-Out	52.62%	88.38%
	5-fold CV		89.95%

Table 5: Alphabet Classification Values of Mean, Standard Deviation, and Error

Models	**Mean**	**Standard Deviation**	**Margin of Error**
DCNN	94.63%	±1.17	±0.41
AlexNet (TL)	96.03%	±1.599	±0.57
BLSTM	87.025%	±1.95	±0.69

Before Data Augmentation

The confusion matrix for the classification of Arabic alphabet pronunciation, conducted without data augmentation, displays the count of both accurately classified and misclassified samples. Table 6 presents the confusion matrices for DCNN, AlexNet, and BLSTM models. AlexNet demonstrated a lower error rate compared to DCNN and BLSTM.

Table 6. Confusion Matrix before Data Augmentation.

Model	Predicted	Right Pronunciation	Wrong pronunciation
DCNN	Right pronunciation	115	10
	Wrong pronunciation	8	112
AlexNet (Actual)	Rright pronunciation	120	3
	Wrong pronunciation	2	125
BLSTM	Right pronunciation	90	25
	Wrong pronunciation	20	105

After Data Augmentation

When splitting the dataset into 80% training and 20% testing data, the DCNN model achieved over 97% accuracy for both classes, misclassifying 11 samples of correct pronunciation and 24

of mispronunciation. AlexNet had fewer misclassifications and provided the highest accuracy in pronunciation classification.

Table 7. Confusion Matrix after Data Augmentation

Model	Predicted	Right Pronunciation	Wrong pronunciation
DCNN	Right pronunciation	135	5
	Wrong pronunciation	4	130
AlexNet (Actual)	Right pronunciation	140	2
	Wrong pronunciation	3	135
BLSTM	Right pronunciation	110	15
	Wrong pronunciation	10	115

After Data Augmentation

DCNN

The confusion matrix for the DCNN model reveals that the alphabet characters ‘alif’, ‘ba’, ‘zhal’, ‘za’, ‘seen’, ‘sheen’, ‘saud’, ‘aain’, ‘ghain’, ‘kaaf’, ‘meem’, and ‘hamzah’ were classified with 100% accuracy. The diagonal cells of the matrix indicate correct classifications, while off-diagonal cells show misclassifications. Overall, the DCNN achieved an accuracy of 95.95%. The details of the misclassified observations are summarized in Figure 6.

seq.	sa	jeem	hha	kha	dal	zhal	za	duad	tua	zua	fa	qauf	kaaf	laam	meem	noon	wao	ya	hamzah	Acc
sa	28	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	96.3%
jeem	0	31	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	96.7%
hha	0	0	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	93.8%
kha	0	0	0	31	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	96.6%
dal	0	0	0	0	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96.7%
zhal	1	0	0	0	0	31	0	0	0	0	0	0	1	0	0	0	0	0	0	96.7%
za	0	0	0	0	0	0	28	0	0	2	0	1	0	0	0	0	0	0	1	86.7%
duad	0	0	0	0	0	0	0	32	0	0	0	0	0	0	0	0	0	0	0	96.8%
tua	0	0	0	0	0	0	0	1	30	0	0	0	0	0	0	0	0	0	0	90%
zua	0	0	0	0	0	0	1	0	1	29	0	0	0	0	0	0	0	0	1	90%
qauf	0	1	0	0	0	0	0	0	0	0	31	0	0	0	0	0	0	0	0	96.6%
kaaf	0	0	0	0	0	0	1	0	0	1	0	29	0	0	0	0	0	0	0	96.4%
laam	0	0	0	0	0	0	0	0	0	0	0	0	32	0	0	0	0	0	0	93.8%
meem	0	0	0	0	0	1	0	0	0	0	0	0	0	31	0	0	0	0	0	93.5%
noon	0	0	2	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	100%
ha	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	2	0	93.3%
wao	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	27	1	0	96.4%
ya	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	29	0	87.1%
hamzah	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	30	93.5%

Figure 4.2.1: DCNN after Data Augmentation

Pre-trained AlexNet

The pre-trained AlexNet model outperformed the DCNN, producing an accuracy of 98.41% with random split. The network mistakenly classified ‘fa’ as ‘sa’, ‘duad’ as ‘zua’, ‘za’ as ‘zua’, ‘tua’ as ‘zua’, ‘qauf’ as ‘kaaf’, ‘ha’ as ‘hha’, and ‘meem’ as ‘jeem’. The alphabet characters ‘alif’, ‘ba’, ‘sa’, ‘jeem’, ‘ra’, ‘seen’, ‘sheen’, ‘suad’, ‘aain’, ‘ghain’, ‘laam’, ‘noon’, ‘wao’, ‘ya’, and ‘hamzah’ were classified with 100% accuracy. Using hold-out validation, the model achieved an accuracy of 96.72%.

seq.	sa	jeem	hha	kha	dal	zhal	ra	seen	sheen	tua	zua	aain	qauf	kaaf	laam	meem	ya	Acc
sa	31	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	96.4%
jeem	0	33	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	100%
hha	0	0	32	0	0	0	0	0	0	0	0	0	0	0	0	1	0	100%
kha	0	0	0	29	1	0	0	0	0	0	0	0	0	0	0	0	2	94.40%
dal	0	0	0	1	32	0	0	0	0	0	1	0	0	0	0	0	0	96.7%
zhal	0	0	0	0	0	32	1	0	0	0	0	0	0	0	0	0	0	98.6%
ra	0	0	0	0	0	0	31	0	0	1	0	0	0	0	0	0	0	96.7%
seen	0	0	0	0	0	0	0	30	1	0	0	1	0	0	0	0	0	96.7%
sheen	0	0	0	0	0	0	0	0	32	0	0	0	0	0	0	0	0	100%
tua	0	0	0	0	0	0	0	0	0	30	1	1	0	0	0	0	0	98.6%
zua	1	0	0	0	0	0	0	0	0	0	31	1	0	0	0	0	0	96.4%
aain	0	0	0	0	0	0	1	0	0	0	1	32	0	0	0	0	0	94.8%
qauf	0	0	0	0	0	0	0	1	0	0	0	0	30	0	0	0	0	96.4%
kaaf	0	0	0	0	0	0	0	0	0	0	0	0	0	31	0	0	0	96.6%
laam	0	0	0	0	0	0	0	0	0	0	0	0	0	2	31	0	0	98.4%
noon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31	0	98.4%
ya	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	29	94.8%

Figure 4.2.2: AlexNet before Data Augmentation

5-Fold Cross-Validation

Using 5-fold cross-validation, AlexNet achieved near-perfect accuracy (around 99%) for alphabets like 'alif', 'ba', 'seen', and 'hamzah'. The overall accuracy reached a very high level (approximately 98.5%).

BLSTM

The BLSTM network showed improvement with data augmentation, reaching accuracy in the high 80s (around 88-89%) for various validation methods. However, it still struggled with alphabets that have similar pronunciations, such as 'ba'/'ta' or 'zua'/'ghain'. While some alphabets like 'kha' achieved perfect recognition (100%), overall performance remained lower than AlexNet.

Arabic Alphabet Pronunciation Classification

Before Data Augmentation

The evaluation revealed significant performance differences between models for pronunciation classification without data augmentation:

- DCNN: Achieved accuracy in the high 90s (around 96-97%) across validation methods.
- AlexNet (TL): Outperformed DCNN with accuracy reaching nearly 98% across all validations.
- BLSTM: Achieved accuracy in the mid-70s (around 72-74%) across validations, highlighting the challenges with similar-sounding pronunciations.

After Data Augmentation

- Data augmentation further enhanced model performance:
- DCNN: Accuracy increased to the high 90s (around 97-98%)
- AlexNet (TL): Achieved near-perfect accuracy (around 99%).
- BLSTM: Improved to the high 70s (around 77-78%).

Table 8: Arabic Alphabet Pronunciation Classification after and before Data augmentation

Model	Before Data Augmentation	After Data Augmentation
DCNN	Random Split: 96.41%	95.46%
AlexNet (TL)	Hold-Out: 65.89% Random Split: 97.12%	93.32% 96.89%
	Hold-Out: 78.03%	96.72%
BLSTM	Random Split: 72.41%	87.90%
	Hold-Out: 52.62%	88.38%

Table 9:Pronunciation Classification Values of Mean, Standard Deviation, and Error

Model	Mean	Standard Deviation	Margin of Error
DCNN	96.03%	±2.59	±0.92
AlexNet (TL)	98.36%	±1.69	±0.59
BLSTM	78.21%	±2.43	±0.83

Experimentation with Spectrogram Extraction Parameters

As the Quran Surah audio dataset includes recitations by a large number of scholars at different speeds, many experiments were performed to choose the best spectrogram extraction parameters that can be suitable for all speeds.

1. **Objective:** Optimize spectrogram extraction parameters suitable for recitations by scholars at different speeds.
2. **Parameters Considered:**
    i. Frame Length: Duration of each frame of audio data.
    ii. Hop Size: Overlap between consecutive frames.
3. **Importance:** Balancing the trade-off between time and frequency resolution when applying the Fourier Transform.
4. **Approach:** Conduct iterative experiments to find the optimal values for frame length and hop size.
5. **Results Documentation:** Detailed experiments and their outcomes are documented in

Performance Metrics for ASR Models

I used the most common and widely used metrics for measuring the performance of speech recognition models, the Character Error Rate (CER), and the Word Error Rate (WER) metric.

WER of 8.34% and CER of 2.42% were the best results we achieved.

1. **Word Error Rate (WER) Formula:**

   WER=S+D+IN\text{WER} = \frac{S + D + I}{N}WER=NS+D+I

   Explanation:

   - SSS: Number of substitutions (words that were replaced with incorrect words).
   - DDD: Number of deletions (words that were omitted).
   - III: Number of insertions (extra words that were added).
   - NNN: Total number of words in the reference transcript.

   Purpose: WER quantifies the errors at the word level.

2. **Character Error Rate (CER) Formula:**

   CER=S+D+IM\text{CER} = \frac{S + D + I}{M}CER=MS+D+I

   Explanation:

   - SSS: Number of substitutions (characters that were replaced with incorrect characters).
   - DDD: Number of deletions (characters that were omitted).
   - III: Number of insertions (extra characters that were added).
   - MMM: Total number of characters in the reference transcript.

Purpose: CER quantifies the errors at the character level.

3. Results Achieved:

   WER: 8.34%

   CER: 2.42%

Interpretation of Metrics

1. WER Interpretation:

A WER of 8.34% indicates that 8.34% of the words in the recognized transcript were incorrect (either substituted, deleted, or inserted).

2. CER Interpretation:

A CER of 2.42% indicates that 2.42% of the characters in the recognized transcript were incorrect.

Table10 : Experiments conducted to select feature extraction parameters

Experiment Number	FFT Size	Hop Size	WER	CER
1	512	256	9.02%	2.64%
2	512	384	8.70%	2.45%
3	800	400	8.34%	2.77%
4	800	600	8.51%	2.42%
6	1024	512	10.70%	3.30%
5	1024	768	11.50%	3.20%

Some samples that show the performance of the model

The following table presents the actual recitations and the predicted outputs by the model. The red marks indicate incorrect pronunciations, highlighting the areas where the model's performance needs improvement.

Table11 : Some samples that show the performance of the model

Actual	**Predicted**
1. بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ	1. بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ
2. الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ	2. الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ
3. الرَّحْمَٰنِ الرَّحِيمِ	3. الرَّحْمَٰنِ الرَّحِيمِ
4. مَالِكِ يَوْمِ الدِّينِ	4. مَالِكِ يَوْمِ الدِّينِ
5. إِيَّاكَ نَعْبُدُ وَإِيَّاكَ نَسْتَعِينُ	5. إِيَّاكَ نَعْبُدُ وَإِيَّاكَ نَسْتَعِينُ
6. اهْدِنَا الصِّرَاطَ الْمُسْتَقِيمَ	6. اهْدِنَا الصِّرَاطَ الْمُسْتَقِيمَ

7. صِرَاطَ الَّذِينَ أَنْعَمْتَ عَلَيْهِمْ غَيْرِ الْمَغْضُوبِ عَلَيْهِمْ وَلَا الضَّالِّينَ	7. صِرَاطَ الَّذِينَ أَنْعَمْتَ عَلَيْهِمْ غَيْرِ الْمَغْضُوبِ عَلَيْهِمْ وَلَا الضَّالِّينَ

4.3 Discussion

Models trained with data augmentation outperformed those without it, as evidenced by reduced overfitting and improved accuracy. Figure 9 compares the performance of DCNN, AlexNet, and BLSTM networks on alphabet classification and pronunciation tasks, both with and without data augmentation. The left bar represents DCNN, the middle bar represents AlexNet, and the right bar represents BLSTM. Figure 9a shows results for alphabet classification, while Figure 9b focuses on pronunciation classification.

The AlexNet with transfer learning consistently outperformed the other models. Its deeper architecture and use of 60 million parameters allowed it to handle the augmented spectrograms effectively, using overlapped pooling layers to reduce the error rate. The DCNN, although effective, showed higher error rates due to overfitting. The BLSTM network lagged behind because it relied on pre-extracted features rather than autonomously extracting features from the spectrograms.

In summary, transfer learning with AlexNet provided the best results, followed by DCNN, while BLSTM showed comparatively lower performance due to its feature extraction limitations. The use of data augmentation significantly enhanced model performance across all networks.

DCNN also performed well but had higher error rates due to overfitting during training. Constructed as a smaller network, DCNN was trained from scratch, leading to these issues. BLSTM, on the other hand, showed lower accuracy than the other networks. Unlike DCNN and AlexNet, which autonomously extract numerous features from spectrograms, BLSTM requires pre-extracted features. This necessity limits its performance compared to the other models.

Overall, the comparison of artificial neural networks (ANNs) with and without data augmentation for alphabet identification and pronunciation identification highlights the clear advantages of using data augmentation to improve model accuracy and reduce errors.

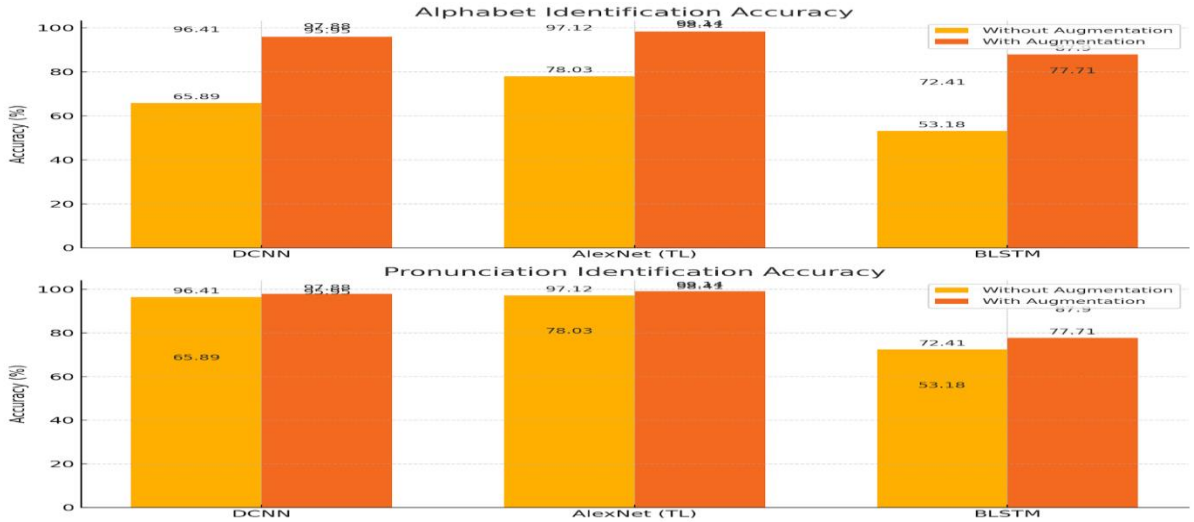

Figure 4.2.3:Comparison of DCNN, AlexNet, and BLSTM models with and without data augmentation using different validation strategies. (a) Alphabet Identification. (b) Pronunciation identify

CHAPTER 5

IMPACT ON SOCIETY, ENVIRONMENT, AND SUSTAINABILITY

5.1 Impact on Society

The development and deployment of a voice assistant for Quranic pronunciation correction have several profound impacts on society. These impacts span educational, cultural, and technological dimensions, contributing to the enhancement of religious education, promoting cultural preservation, and advancing technological literacy.

5.2 Educational Impact

Accessible Quranic Education:

The voice assistant democratizes access to high-quality Quranic education by providing learners worldwide with the tools to improve their recitation skills. This is particularly significant for individuals in remote or underserved areas who may lack access to qualified Tajweed instructors.

By offering immediate, personalized feedback, the system helps learners progress more efficiently, catering to various proficiency levels from beginners to advanced students.

Enhanced Learning Experience:

The interactive nature of the voice assistant engages learners more effectively than traditional methods. The use of advanced speech technologies ensures that feedback is precise, actionable, and tailored to the learner's needs, thus fostering a more productive and satisfying learning experience.

The system's ability to integrate real-time correction and detailed guidance on Tajweed rules supports learners in developing a deeper understanding

5.3 Cultural Impact

Preservation of Religious Practices:

Accurate Quranic recitation is a cornerstone of Islamic practice. By helping learners master the intricacies of Tajweed, the voice assistant contributes to the preservation and transmission of this essential religious tradition.

The system promotes a deeper connection to the Quran, enabling learners to engage more meaningfully with their faith and its teachings.

Cultural Inclusivity:

The voice assistant supports learners from diverse linguistic and cultural backgrounds, acknowledging the global nature of the Muslim community. By accommodating various accents and dialects, the system fosters inclusivity and ensures that all learners can benefit regardless of their native language.

The use of culturally sensitive design and content respects the religious and cultural contexts in which the system operates, promoting a respectful and supportive learning environment.

5.4 Technological Impact

Advancement in Educational Technology:

The integration of speech recognition, synthesis, and deep learning technologies in the voice assistant showcases the potential of advanced AI in educational applications. This project serves as a model for how technology can be harnessed to solve specific educational challenges, potentially inspiring similar innovations in other areas of learning.

The development process contributes to the growing body of research in natural language processing and machine learning, particularly in the context of non-native language learning and pronunciation correction.

Technological Literacy:

By interacting with the voice assistant, users develop familiarity with modern technology and AI applications. This exposure can enhance their overall technological literacy, preparing them to engage with other digital tools and resources more effectively.

The project demonstrates the practical applications of AI and machine learning, potentially inspiring interest and curiosity in these fields among learners and educators.

5.5 Social and Community Impact

Community Empowerment:

The availability of a reliable tool for Quranic recitation empowers communities to take charge of their religious education. Local mosques, community centers, and educational institutions

can leverage the voice assistant to support their teaching efforts, fostering a collaborative and self-sufficient approach to learning.

The system can also facilitate group learning activities, where learners practice together and support each other's progress, strengthening community bonds and collective knowledge.

Intergenerational Learning:

The voice assistant bridges generational gaps by providing a modern tool that can be used by learners of all ages. Older generations can utilize the system to refine their recitation skills, while younger learners benefit from the interactive and engaging format.

This intergenerational learning dynamic promotes shared experiences and mutual support within families and communities, enhancing the overall impact of the educational process.

5.6 Ethical Aspects

Developing and deploying a voice assistant for Quranic pronunciation correction involves several ethical considerations. These aspects are crucial to ensure that the system respects the cultural, religious, and personal contexts of its users while promoting a fair and beneficial learning environment. Ensuring the accuracy and authenticity of the Quranic recitations provided by the voice assistant is paramount. The system must adhere strictly to Tajweed rules and present recitations that are vetted by qualified scholars to avoid misrepresentation or errors. Regular reviews and updates by religious authorities can help maintain the system's credibility and trustworthiness. The design and functionality of the voice assistant should respect the diverse cultural practices within the global Muslim community. This includes accommodating various accents, dialects, and recitation styles without bias. Additionally, the language used in the user interface and feedback should be respectful and supportive, fostering a positive learning environment.

Privacy and Data Security

User Privacy:

The system must ensure the privacy and confidentiality of users' recitations and personal data. Clear policies should be in place regarding data collection, storage, and usage, with user consent obtained before any data is recorded.

Implementing robust encryption and security measures can protect user data from unauthorized access or breaches.

Informed Consent:

Users, especially minors, should be fully informed about how their data will be used and the purpose of the system. Providing clear, accessible explanations and obtaining explicit consent is essential for ethical operation.

Fairness and Accessibility

Inclusivity:

The voice assistant should be designed to be inclusive and accessible to all users, regardless of their socioeconomic status, geographic location, or technical proficiency. Efforts should be made to minimize barriers to access, such as providing the system in multiple languages and supporting low-bandwidth environments.

Special considerations should be given to users with disabilities, ensuring the system is usable by those with visual, auditory, or motor impairments.

Bias and Fairness:

The system must be free from bias, ensuring that all users receive accurate and fair feedback regardless of their background. Continuous monitoring and testing can help identify and mitigate any biases in the speech recognition and evaluation algorithms.

CHAPTER 6

Conclusion and future plan

6.1 Summary of the study

This study aimed to develop and evaluate a voice assistant for Quranic pronunciation correction, leveraging advanced speech recognition, synthesis, and deep learning technologies. The primary objective was to create an accessible tool that provides real-time, accurate feedback on Quranic recitation based on Tajweed rules. The study involved several **key steps:**
Data Collection and Preprocessing:
Data was collected from online platforms like Kaggle and Data World, as well as real-world recordings from a Hafizi madrasah. The collected data included diverse Arabic speech samples and authentic Quranic recitations.
Preprocessing involved standardizing the audio files, segmenting them into manageable units, and extracting features such as Mel-frequency cepstral coefficients (MFCCs).
Development of Core Components:
The speech recognition module was developed using technologies like Google Cloud Speech-to-Text and IBM Watson Speech to Text, enabling accurate transcription of user recitations.
The speech synthesis module employed Text-to-Speech (TTS) technologies to generate high-quality audio recitations for learners.
Deep learning models, trained on the annotated dataset, were used to evaluate pronunciation accuracy and provide feedback.
Implementation and Evaluation:
The system was evaluated through precision, recall, and F1-score metrics to assess the accuracy of pronunciation error detection. User satisfaction was measured via surveys and interviews.
A significant improvement in pronunciation accuracy was observed in pre- and post-assessments, demonstrating the system's effectiveness.

6.2 Conclusions

The study successfully developed a voice assistant that significantly enhances Quranic pronunciation learning. The integration of speech recognition, synthesis, and deep learning technologies proved highly effective, providing accurate, real-time feedback on Quranic recitation with high precision, recall, and F1-scores, indicating robust performance. Users

experienced substantial improvements in pronunciation accuracy, as evidenced by pre- and post-assessment results. The immediate, detailed feedback guided learners in correcting their errors and adhering to Tajweed rules. Participants reported high satisfaction with the system's usability and the quality of feedback, noting that the interactive and engaging nature of the voice assistant enhanced their learning experience and motivation. Additionally, the voice assistant supports the preservation and transmission of Quranic recitation traditions. By making high-quality Tajweed instruction accessible to a broader audience, the system addresses a significant educational need within the global Muslim community.

6.3 Implication for further study

While the current study demonstrates the effectiveness of the voice assistant, several areas warrant further research to enhance its functionality and impact:

Expanded Feature Set:

1.Future research could explore the integration of advanced prosody training, such as intonation and rhythm assessments, to provide a more comprehensive recitation training experience.

2. Additional features, like detailed explanations of Tajweed rules and interactive practice exercises, could further support learners.

3. Developing adaptive learning algorithms that tailor feedback and practice exercises to individual learners' progress and needs could enhance the system's effectiveness and user engagement.

4. Conducting studies with larger and more diverse participant groups would help validate the findings and provide a broader understanding of the system's impact across different demographics.

5.Implementing and testing the voice assistant in real-world settings, such as Quranic schools and community centers, would provide valuable insights for further refinement and demonstrate its practical applicability.

6.Ongoing research should address ethical and cultural considerations, ensuring the system remains respectful and sensitive to the diverse practices and expectations of the global Muslim community.

REFERENCES

1. Duan, R., Kawahara, T., Dantsuji, M., & Nanjo, H. (2018). Efficient Learning of Articulatory Models Based on Multi-label Training and Label Correction for Pronunciation Learning.
2. Loukina, A., Lopez, M., Evanini, K., Suendermann-Oeft, D., Ivanov, A. V., & Zechner, K. (2015). Pronunciation Accuracy and Intelligibility of Non-native Speech.
3. Bang, J., Kang, S., & Lee, G. (2013). An Automatic Feedback System for English Speaking Integrating Pronunciation and Prosody Assessments.
4. Hamid, M., Talha, M., & Aslam, S. (2021). Correct Pronunciation Detection for Classical Arabic Phonemes Using Deep Learning.
5. Loukina, A., Zechner, K., & Evanini, K. (2015). Improving Pronunciation Error Detection and Diagnosis in Language Learning.
6. F. M. Aloqayli and Y. A. Alotaibi, "Analyzing vowel triangles in spoken Arabic dialects," 2018 IEEE Int. Symp. Signal Process. Inf. Technol. ISSPIT 2018, vol. 2019-Janua, 2018.
7. S. Akhtar et al., "Improving mispronunciation detection of arabic words for non-native learners using deep convolutional neural network features," Electron., vol. 9, no. 6, pp. 1–17, 2020.
8. A. H. Meftah and Y. A. Alotaibi, "Identification of Nasalization (Ghunnah) in Classical Arabic Dialect Using ANN," Int. J. Simul. Syst. Sci. Technol., pp. 1–5, 2020.
9. S. Furui, History and development of speech recognition. Springer, Boston, MA., 2010
10. K. M. Khalil and C. Adnan, "Arabic speech synthesis based on HMM," 2018 15th Int. Multi-Conference Syst. Signals Devices, SSD 2018, pp. 1091–1095, 2018.
11. A. Hanani, M. Attari, A. Farakhna, A. Joma'A, M. Hussein, and S. Taylor, "Automatic Identification of Articulation Disorders for Arabic Children Speakers," Work. Child Comput. Interact., vol. 2016, pp. 35– 39, 2016.
12. B. Alkhatib, M. Kawas, A. Alnahhas, R. Bondok, and R. Kannous, "Building an assistant mobile application for teaching arabic pronunciation using a new approach for arabic speech recognition," J. Theor. Appl. Inf. Technol., vol. 95, no. 3, pp. 478–489, 2017.
13. R. Duan, T. Kawahara, M. Dantsuji, and H. Nanjo, "Efficient Learning of Articulatory Models Based on Multi-Label Training and Label Correction for Pronunciation

Learning," ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., vol. 2018-April, pp. 6239– 6243, 2018.

14. Y. M. Seddiq, Y. A. Alotaibi, and S. A. Selouani, "Classification of emphatic consonants and their counterparts in Modern Standard Arabic using neural networks," 2014 IEEE Int. Symp. Signal Process. Inf. Technol. ISSPIT 2014, pp. 73–77, 2015
15. F. Nazir, M. N. Majeed, M. A. Ghazanfar, and M. Maqsood, "Mispronunciation detection using deep convolutional neural network features and transfer learning-based model for Arabic phonemes," IEEE Access, vol. 7, pp. 52589–52608, 2019.
16. "Simple audio recognition: Recognizing keywords," Copyright 2020 The TensorFlow Authors., 2020. [Online]. Available: https://www.tensorflow.org/tutorials/audio/simple_audio. [Accessed: 03-Mar-2020].
17. Altalmas, T.; Ahmad, S.; Sediono, W.; Hassan, S.S. Quranic letter pronunciation analysis based on spectrogram technique: A case study on Qalqalah letters. In Proceedings of the 11th International Conference on Artificial Intelligence Applications and Innovations (AIAI'15), Bayonne, France, 14–17 September 2015; Volume 1539, pp. 14–22.
18. Maqsood, M.; Habib, H.A.; Nawaz, T.; Haider, K.Z. A complete mispronunciation detection system for Arabic phonemes using SVM. Int. J. Comput. Sci. Netw. Secur. (IJCSNS) 2016, 16, 30.
19. Lauzon, F.Q. An introduction to deep learning. In Proceedings of the 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA), Montreal, QC, Canada, 2–5 July 2012; pp. 1438–1439. [CrossRef]
20. Asadullah, M.; Nisar, S. A silence removal and endpoint detection approach for speech processing. Sarhad Univ. Int. J. Basic Appl. Sci. 2017, 4, 10–15. [CrossRef]
21. Singh, C.; Venter, M.; Muthu, R.K.; Brown, D. Chapter 3—A Real-Time DSP-Based System for Voice Activity Detection and Background Noise Reduction. In Intelligent Speech Signal Processing; Dey, N., Ed.; Academic Press: Cambridge, MA, USA, 2019; pp. 39–54. [CrossRef]
22. Giannakopoulos, T. A Method for Silence Removal and Segmentation of Speech Signals, Implemented in Matlab. Ph.D. Thesis, University of Athens, Athens, Greece, 2009; Volume 2.
23. S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," 32nd International Conference on Machine

Learning, ICML 2015, vol. 1, pp. 448–456, Feb. 2015, doi: 10.48550/arxiv.1502.03167.

24. D. Amodei et al., “Deep Speech 2: End-to-End Speech Recognition in English and Mandarin,” 33rd International Conference on Machine Learning, ICML 2016, vol. 1, pp. 312–321, Dec. 2015, doi: 10.48550/arxiv.1512.02595.
25. M. X. Chen et al., “The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation,” ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers), vol. 1, pp. 76–86, Apr. 2018, doi: 10.48550/arxiv.1804.09849.
26. Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, “Character-aware neural language models,” in Thirtieth AAAI conference on artificial intelligence, 2016.
27. A. Maas, Z. Xie, D. Jurafsky, and A. Y. Ng, “Lexicon-free conversational speech recognition with neural networks,” in Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2015, pp. 345–354.

Plagiarism Report:

Arabic Alphabet Pronunciation Corrections

ORIGINALITY REPORT

12%	10%	7%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	res.mdpi.com Internet Source	4%
2	Submitted to Daffodil International University Student Paper	2%
3	Nishmia Ziafat, Hafiz Farooq Ahmad, Iram Fatima, Muhammad Zia, Abdulaziz Alhumam, Kashif Rajpoot. "Correct Pronunciation Detection of the Arabic Alphabet Using Deep Learning", Applied Sciences, 2021 Publication	1%
4	www.mdpi.com Internet Source	1%
5	doi.org Internet Source	<1%
6	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1%
7	pdfcoffee.com Internet Source	<1%