Deep Acoustic Modelling for Quranic Recitation

By

Muhammad Aleem Shakeel

Fall-2021-MS-EE-AI&AS 363493 SEECS

Supervisor

Dr. Kamran Zeb

Department of Electrical Engineering

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Electrical Engineering (MS EE)

In

School of Electrical Engineering & Computer Science (SEECS) ,
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

(October 2023)

Thesis Acceptance Certificate

Certified that final copy of MS/MPhil thesis entitled "**Deep Acoustic Modelling for Quranic Recitation**" written by **Muhammad Aleem Shakeel**, (Registration No **Fall-2021-MS-EE-AI&AS 363493 SEECS**), of School of Electrical Engineering & Computer Science (SEECS) has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: ____________________

Name of Advisor: **Dr. Kamran Zeb**

Date: ____________________

Signature (HoD): ____________________

Date: ____________________

Signature (Dean/Principal): ____________________

Date: ____________________

Approval

It is certified that the contents and form of the thesis entitled "**Deep Acoustic Modelling for Quranic Recitation**" submitted by **Muhammad Aleem Shakeel** have been found satisfactory for the requirement of the degree.

Advisor: **Dr. Kamran Zeb**

Signature: ____________________

Date: ____________________

Committee Member 1: **Dr. Wajahat Hussain**

Signature: ____________________

Date: ____________________

Committee Member 2: **Dr. Numan Khurshid**

Signature: ____________________

Date: ____________________

Committee Member 3: **Dr. Jawad Arif**

Signature: ____________________

Date: ____________________

Dedication

I am humbled and honored to dedicate this thesis book to the most important people in my life: my beloved family. Throughout this challenging yet rewarding journey, your unwavering support, love, and understanding have been my constant pillars of strength. From the early days of my academic pursuits to the late nights of writing and research, you have been there, cheering me on and believing in my abilities even when I doubted myself. Your sacrifices, patience, and encouragement have been the fuel that kept me going, and I am forever grateful for everything you have done.

To my esteemed supervisor, **Dr. Kamran Zeb**, I extend my deepest gratitude for your exceptional guidance and mentorship. Your deep knowledge and lot of learning from **Dr. Hasan Ali Khattak** for passion for research and dedication to their field have shaped the outcome of this thesis in ways I could not have imagined. Your constructive feedback, patience, and unwavering belief in my potential have inspired me to strive for excellence. Working under your guidance has been an honor and a privilege, and I am proud to dedicate this thesis to you as a token of my appreciation.

To my devoted GEC members **Dr. Wajahat Hussain**, **Dr. Numan Khursid**, **Dr. Jawad Arif**, I am immensely grateful for your invaluable contributions to this work. Your expertise and insight have enriched the research and broadened my perspective. Your unwavering support and encouragement have motivated me to explore deeper my studies and explore new avenues of knowledge. I am indebted to each of you for your mentorship and dedication to my academic growth, and I proudly dedicate this thesis to all of my co-supervisors.

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at Department of Electrical Engineering at School of Electrical Engineering & Computer Science (SEECS) or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at School of Electrical Engineering & Computer Science (SEECS) or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: **Muhammad Aleem Shakeel**

Signature: ____________________

Acknowledgments

Glory be to Allah (S.W.A), the Creator, the Sustainer of the Universe. He only has the power to honor whom He pleases and to abase whom He pleases. Verily, no one can do anything without His will. From the day I came to NUST till my departure, He was the only one Who blessed me, opened ways for me, and showed me the path to success. There is nothing that can pay back for His bounties throughout my research period to complete it successfully.

Muhammad Aleem Shakeel

Contents

List of Figures

List of Tables

Abstract

The Holy Quran, the only scripture of the universe preserved in its complete original text since its revelation, is of utmost importance to the Muslim community. Revealed originally in the Arabic language, we need to understand and practice how it should be recited and memorized according to the rules set out by native Arabic speakers. With the advent of AI technology in acoustic modeling, researchers began developing models of various languages; however, due to the variety of accents and dialects of Arabic, it is challenging to develop a robust acoustic model for Quranic recitation. In this research, we developed a deep learning model that is not only robust to the above linguistic properties but is not affected by the recitation styles and intricate Tajweed. When used for classification tasks, deep features from this model produced a maximum accuracy of around 96.30%. To illustrate the importance of our deep learning network as an acoustic model, a content-based verse retrieval system (CBVeRse) was developed by employing the model trained in the previous step with an Average Normalized Modified Retrieval Rank (ANMRR) of 85.39% and mean Average Precision (mAP) of 96.52%.

Chapter 1

Introduction and Motivation

For millions of Muslims around the world, the Quran is of utmost importance as their Holy Book. The poetic recital in Quranic verses, known as "Tilawah," is not only a means of promoting spiritual enlightenment but also a profound art that mesmerizes listeners with its rhythmic flow. The accurate recitation and its preservation have always been of the utmost significance to the Muslim world. Quranic recitation has traditionally been transmitted orally, preserving the art form in its most basic form. However, new options have opened to improve our comprehension of Quranic recitation and bring it to a broader audience using machine learning and deep learning.

Because of its utmost significance in academics and research, people have continually investigated new approaches to improve the Quranic recitation experience and support its accurate preservation and meanings. The use of deep acoustic modeling techniques for Quran recitation is one topic that has recently attracted an immense amount of attention. Speech recognition, natural language processing, and deep learning have succeeded in many areas. In speech recognition and voice analysis, deep acoustic modeling uses complex neural networks for processing and evaluating acoustic data. Researchers aim to better understand the art of "Tilawah" by using these cutting-edge techniques for Quranic recitation. They also seek to increase the accuracy and robustness of current Quranic recitation recognition systems.

Reciting and studying the Quran is of utmost cultural, religious, and educational importance. A solid opportunity to use machine learning, deep learning, and technological advancements to thoroughly research and preserve Quranic recitations exists today. Deep Acoustic Modeling offers a way to delve further into the intricacies of recitation, helping us to comprehend various recitation methods, variances in pronunciation, and emotional expressions. Furthermore, classifying Quranic Surahs can help experts, teachers, and enthusiasts efficiently categorize and retrieve pertinent information, improving the accessibility and comprehension of this treasured scripture. That is why Deep Acoustic Modelling for Quranic Surah Classification aims to develop a comprehensive system model that utilizes advanced deep learning techniques to accurately classify Quranic Surahs based on their acoustic features. The system leverages the power of deep neural networks to extract intricate patterns and representations from audio data, enabling the classification of Surahs into predefined categories. By employing state-of-the-art techniques in speech recognition and natural language processing, this model seeks to provide a novel approach to automate the process of categorizing Quranic Surahs, facilitating efficient navigation, analysis, and study of the Quran for researchers, scholars, and individuals interested in Islamic teachings. The proposed system model combines the richness of acoustic information embedded within Surahs with deep learning capabilities, ultimately contributing to a more accessible and comprehensive understanding of the Quranic text.

1.1 Problem Statement and Research Contribution

The main difficulty is efficiently utilizing Deep Acoustic Modeling for Quranic Recitation and Quranic Surah categorization. Despite their importance, Quranic recitation analysis and classification are challenging because of the wide range of reciters, linguistic variances, and recording settings. The challenge requires creating strong deep learning models that can reliably identify Quranic Surahs, extract valuable features from audio data, and capture the subtleties of various reciter styles. Innovative methods that can handle the complexity of voice modulation, pronunciation, and emotional nuances are required to address this difficulty while preserving the integrity and authenticity of the recitations and surahs.

The scope of this project, titled "Deep Acoustic Modeling for Quranic Recitation," is to create and use cutting-edge deep learning methods to analyze, categorize, and comprehend Quranic recitations. The project's primary goal is to thoroughly examine the numerous reciter styles, acoustic variances, and emotional expressions used in Quranic recitations. The classification of Quranic Surahs is included in the scope, allowing for effective grouping, searching, and categorizing of various chapters. In this research, different audio data will be gathered and preprocessed, deep neural networks will be designed and put into use, and novel methods to capture the intricate features and patterns inherent in Quranic recitations will be investigated.

This project significantly contributes to machine learning, deep learning, and audio processing.

1.1.1 Novel Deep Learning Model

The project contributes by creating cutting-edge deep acoustic models adapted to the complexity of Quranic recitations. These models try to capture the nuanced differences in pronunciation, style, and voice, improving the recitations' understanding and analysis.

1.1.2 Enhanced Quranic Surah Classification

The project's classification framework aids in efficiently and accurately classifying Quranic Surahs.

1.1.3 Dataset Creation

Creating a diversified dataset of Quranic recitations that spans different reciters, styles, and recording circumstances is a significant contribution. The carefully curated dataset will serve as a valuable resource for future research and analysis in the field.

1.1.4 Indexing and Retrival

Accurate and efficient indexing and localization of specific Quranic verses within The Holy Quran is a fundamental challenge that impedes comprehensive Quranic research and access. Existing verse identification and localization methods often lack precision and require significant manual effort, hindering scholars, students, and enthusiasts in their quest to study and reference Quranic verses. Additionally, there is a need for accessible and user-friendly tools that can assist in various aspects of Quranic recitation, education, and research. Therefore, our project aims to address this problem by developing a robust and automated system for accurate verse indexing, localization, and related applications, leveraging deep acoustic modeling techniques.

1.1.5 Insights into Recitation Styles

Deep acoustic modeling allows for in-depth analysis, which offers insights into various recitation styles, emotional nuance, and linguistic differences. These new details deepen our comprehension of Quranic recitations and how they have been interpreted.

1.1.6 Cultural Preservation

This project helps to preserve cultural heritage by using technology to analyze and save Quranic recitations. The effort ensures that the Quranic recitations' integrity, authenticity, and cultural value are safeguarded.

Chapter 2

Literature Review

This review aims for some understanding of the advancements made in the fields of Automatic Speech Recognition (ASR), contextualized classification in Quranic topics, Improvements in the reading of the Quran using Deep learning, feature extraction techniques and comparison between them, reciter classification, and Tajweed, Hijayah, and Makhraj correction and classification based on the context as well.

The objectives of this literature review are as follows:

- To present a comprehensive and current overview of the literature on deep acoustic modeling for Quranic recitation, including ASR, context-based classification in Quranic topics, improvements in reading the Quran using machine learning and deep learning, feature extraction techniques for different acoustic data and comparisons between them to find out the best of it, Reciter classification, Tajweed, Hijayah, and Makhraj checking and correction, and their variety in Holy Quran.
- To describe the approaches, system models, datasets, and evaluating criteria used in the research alongside their merits and shortcomings.
- To evaluate the efficiency and performance of deep learning models for Quranic recitation assessment in contrast to more conventional methods.
- To highlight challenges in existing research areas and propose new directions for the research in this domain.

2.1 Existing Surveys

To this date, as listed in Table 2.1, many research articles are focused on improving user experience in the Quran, whether it is through speech analysis or text analysis. The research investigates their ways of dealing in this domain as Devin et al. [47] offered many methods for identifying various speech types in the Quran. He highlights the kinds of evidence scholars should focus on when examining genres, offers fundamental criteria for interpreting Qur'anic verses, and discusses mistakes and difficulties that should be considered in follow-up research. He advocated closely examining unique words, phrases, and structures. The focus of the discussion is on the incorporation of particular genre texts into suras or longer sections within suras. It illustrates how the Qur'an both refers to and modifies pre-existing categories. Huzaifa et al. [48] discussed the application of natural language processing (NLP) to Quranic commentary and interpretation. He used speech recognition and NLP to enhance Quranic reciting. He demonstrated how NLP techniques help develop tools that facilitate everyday people's acquisition of new knowledge. His research serves as a synthesis compendium of works ranging from automated morphological evaluation to speech recognition-based Qur'anic recitation correction, and it offers an overview of the numerous Qur'anic NLP projects.

Rusli et al. [27] offered a semantic taxonomy for knowledge found in the Quran. She presented an extensive systematic analysis of how existing Quranic ontology models do not account for all concepts in the Quran and are restricted to categories such as nouns, subjects, pronouns, antonyms, and Islamic learning. The research aims to locate pertinent research papers from a variety of electronic data sources to provide a thorough review of this topic. To disseminate a correct understanding of the Quran through semantic technologies, her work carefully assesses the literature relevant to the present ontology models. Wahdan et al. [38] provided a deep learning model text classification for the Arabic language and the Quran. He concentrated on deep learning-based text categorization methods such as CNN, RNN, LSTM, etc. He provides a detailed analysis of the system models, accuracy, and outcomes of twelve relevant research publications before recommending the model that would be most useful. In the end, he offers recommendations on models to employ to enhance text classification.

A comprehensive review of Deep Modeling for Quranic recitation will help the research community to understand these concepts. However, the surveys mentioned in Table 2.1

still lag and face challenges researchers must investigate. However, these surveys mainly focused on NLP, Text Classification, and Semantic Ontology. That's where this paper comes in, to emphasize acoustic modeling and speech analysis in Quranic recitation. The sole focus of this review paper is to find research gaps and challenges the research community faces in speech recognition and Deep Learning modeling.

Year	**Ref.**	**Topic(s) of the survey**	**Primary findings of the survey**
2023	[48]	NLP for Quranic Research	NLP serves as a synthesis compendium of works that span speech recognition-based Qur'anic recitation correction to computerized morphological evaluation
2022	[47]	Speech Genres in Quran	Highlights the kind of evidence that researchers should focus on while investigating genres and explains mistakes and problems that should be considered.
2020	[38]	Text Classification in Arabic Language	Highlights different deep learning models that show the best accuracy for Arabic classification.
2018	[27]	Semantic Ontology for Quranic Knowledge	Analyzes different ontology methods for Quran and highlights the gaps.

Table 2.1: Existing Surveys related to Deep Modeling in Quran

2.2 Background and Existing Literature on Modelling of Quranic Recitation

Recitation is more than just reading in the context of the Quran; it follows standards and rules guiding pronunciation, rhythm, and melody. Deep neural networks, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), are trained to recognize and capture the complex auditory patterns unique to Quranic recitation in deep acoustic modeling for Quranic recitation. The auditory and pronunciation aspects of recitation are two elements the model carefully considers. To do this, it is necessary to record Arabic phonemes precisely and to follow Tajweed guidelines for proper pronunciation. The model also emphasizes intonation and melody to accurately simulate pitch fluctuation and some syllables' lengthening in Quranic recitation. Deep acoustic modeling for Quranic recitation's ultimate goal is to develop a technologically enhanced tool that makes it easier to learn and master the complex art of recitation.

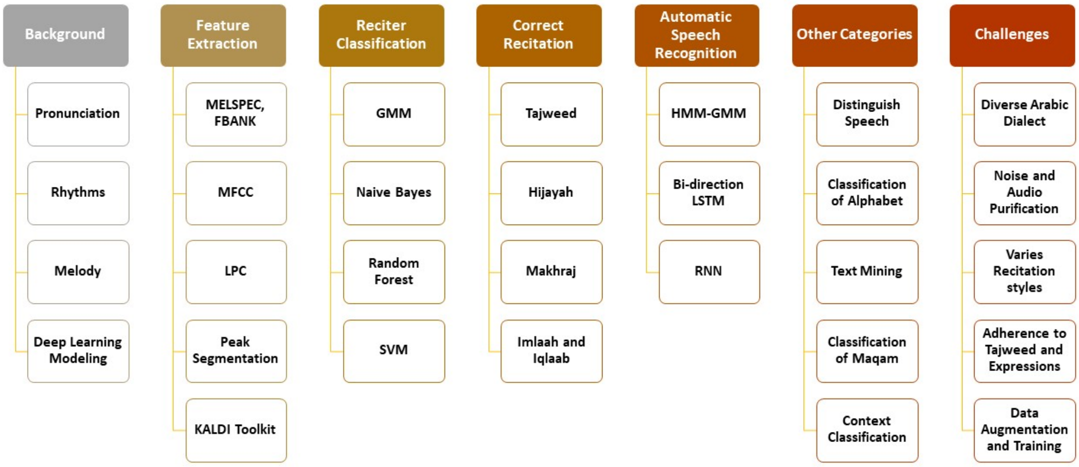


Figure 2.1: Article Organization of Literature Review

The manuscript classifies different research topics for organizing Quranic recitation as shown in Figure 2.1 and conducted a thorough survey on each topic. The article organization is divided into the following categories: Section 2.2 defines Deep Acoustic Modelling for Quranic Recitation. Section 2.2.1 represents the primary feature extraction techniques used for the classification, their results to prove which method works best for the related topic, and their survey and results comparisons. Section 2.2.2 represents the different reciter classification techniques used by the research to classify the reciter's voice and is currently an active research topic. Section 2.2.3 describes a thorough Tajweed, Hijayah, and Makhraj Classification and correction of mistakes techniques using deep learning to improve the reading of the Quran without any errors. Section 2.2.4 represents the Automatic Speech Recognition used for Quranic Recitation, including Hybrid HMM-BLSTM-based modeling and End-to-End Transformer modeling, and Section 2.2.5 describes the classification of different artifacts of Quran recitation, including Feature Identification on both acoustic and textual data and classifying different Maqams of Quranic Recitation.

2.2.1 Feature Extractions

Raw audio signals are frequently multidimensional and packed with information. While maintaining crucial qualities necessary for the application, feature extraction assists in reducing the dimensionality of the data.

Abdo et al. in [6] provided a method to automatically separate Arabic speech from audio signals into emphatic and non-emphatic segments. The principles of Holy Quran recitation are the primary emphasis of the study. The process involves applying the Mel Frequency Cepstral Coefficient (MFCC) to extract key characteristics from Arabic sound signals, locating the target signal's boundary peak position, and assessing the system as a whole for medium-level speech. They developed the database for each signal and tested the system using 80 Arabic words that could be recited. Six distinct speakers of Arabic provided recitations of the dataset, totaling 480 Arabic words for testing. The system achieves a segmentation accuracy of roughly 90%. To improve the effectiveness of the model, future work could add more constraints to the MFCC peaks or combine or use alternative feature extraction techniques, such as spectral envelopes formant frequencies.

Even though The Holy Quran consists of the same verse all over the world, the recited poem probably is different from the other person who repeated the same verse because of the distinct voice of every person. The issue produces variations and contrasts between different reciters. Bezoui et al. in [10] suggested a method for utilizing the KALDI toolkit to train and evaluate the Arabic speech system. The author investigates the feasibility of several feature-extracting techniques, such as the Mel-Frequency Cepstral Coefficient (MFCC), to extract crucial features from Quranic Recitations to construct the system. The author thoroughly explains the MFCC technique, including all of its steps — preprocessing, framing, windowing, etc. The system's highest efficiency is 55% when utilizing the rectangular window technique and 75% when using the Hamming Window technique. Quranic verse audio files are part of the dataset used for this purpose. Nevertheless, the task can be enhanced by applying the fixed-range sliding window method and obtaining the MFCC characteristic for each windowing signal. Meftah et al. in [12] evaluated various feature extraction methods to determine which would produce the best classification accuracy for Arabic phonemes. To achieve this, a dataset corpus containing different Arabic renditions of the Holy Quran has been assembled, and acoustic characteristics are extracted from it. Mel-filter bank coefficient (MELSPEC), Log Mel-filter bank coefficient (FBANK), Perceptual Linear Prediction (PLP), Mel Frequency Cepstral Coefficient (MFCC), Linear Predictive Coding (LPC), and Linear Prediction Reflection Coefficients (LPREFC) are some of these properties. Hidden Markov Model (HMM) classification has been applied following feature extraction. The outcome reveals that the features with the highest accuracy for the system are FBANK and MELSPEC, which yield 85.38% and 83.37%, respectively. Moreover, the results from MFCC and PLP are pretty similar to this accuracy, although LPC is not thought to be appropriate for Arabic voice recognition. Adiwijaya et al. in [14] used various feature extraction approaches to perform comparative analysis to classify pronunciation of the Hijaiyyah letters. The dataset comprises audio samples of Hijayyah letters. Mel frequency cepstral coefficient (MFCC) and linear predictive coding (LPC) are used for analysis, and KNN is used for classification. When comparing the proposed system with Principal Component Analysis (PCA) and without PCA, it was shown that LPC-KNN outperformed MFCC-KNN in terms of accuracy for Hijayah classification, achieving 78.92% compared to 59.87%. For this reason, LPC is a superior feature extraction method.

2.2.2 Reciters Classification

Classification of reciters for Quranic recitation aims to recognize and classify various Qaris (reciters) according to their distinctive recitation styles. This is a particular endeavor in the field of audio analysis. Khan et al. in [30] suggested using machine learning to identify the Holy Quran reciter. 12 different reciters in the dataset recited the last 10 surahs of the Quran, so the model has 12 classes to categorize. Two other methods have been applied to the audio representation. First, features are extracted using the sound pitch and MFCC. Audio spectrogram auto-correlograms are the second. Next, apply J48, Random Forest, and Naive-Bayes for classification. The highest accuracy of 88% was attained with Naive-Bayes and Random Forest.

Munir et al. in [8] suggested an additional feature extraction method for Quranic Surah speaker identification. The author combined the Discrete Wavelet Transform (DWT) and Linear Predictive Coding (LPC) techniques to extract essential characteristics. Random forest (RF) was then utilized for classification. The system's most excellent accuracy was 90.90%; nevertheless, they attempted to combine individual feature extraction approaches and combining them to train the classification model to increase the identification accuracy. However, it does not affect the classifier's accuracy. The Holy Quran recited in Arabic is included in the dataset used for this purpose. More than two feature extraction techniques can be used in the research, and machine learning techniques can be used for classification. But only Arabic recitation is programmed into the system; the task can be extended by introducing recitation in different languages. Elnagar et al. in [21]suggested a method for classifying the reciters in the Quran audio collection using supervised learning. With machine learning techniques, the algorithm may determine the closest or precise reciter. The plan extracted perceptual characteristics from these audio data, such as pitch, tempo, short-term energy, etc.. Next, the support vector machine (SVM) classifier was put into practice. The accuracy of the model was 90%. Seven Saudi Arabian reciters' Quranic audio was included in the dataset used for this purpose, indicating that the model can handle the Arabic dialect of Quranic speech. Other feature extraction methods or their combinations can be used to enhance the task. Qayyum et al. in [26] suggested a deep learning method for speaker recognition that uses Arabic audio sources. Bidirectional Long Short-Term Memory (BLSTM) is used to evaluate and classify the audio signals according to the speaker; this method is more effective and less computationally expensive than previous methods for speaker identification. Gunawan et al. in [23] created an identification mechanism for Quran reciters utilizing GMM and MFCC. Five reciters' randomly chosen Quranic verses were included in the dataset used for this study. Each reciter's fifteen audio samples were gathered, examined using the Mel Frequency Cepstral coefficient, and then categorized using the GMM classifier. The suggested method identified the reciter with 100% accuracy. Moreover, in place of these five reciters, the algorithm may reject unknown reciters. On the other hand, the system can be expanded to incorporate variants of verses spoken on various Quranic Surahs by multiple reciters.

2.2.3 Correct Recitation

Makhraj, Hijayah, and Tajweed deep learning-based Quran correction is a novel and technologically advanced method for improving Quranic text comprehension and recitation. For non-Arabic-speaking Muslims, reading the Quran is always a challenging task. Since many words in the Quran are written differently than they read. Also, to help the parents solve the reading and pronunciation problems of dyslexic children, Basahel et al. in [46] suggested a method for creating an Android application that facilitates self-paced and adaptable learning. To assist with e-learning, the program can translate voice recognition algorithms into text. The program can only process individual words and not parse texts or entire sentences. The program can be enhanced to train on texts and edits in the future.

The right pronunciation and recitation of the Quran mainly depend on these four ideas:

Tajweed

Tajweed is a collection of guidelines for pronouncing and articulating Quranic texts correctly. It ensures every letter is said precisely, melodiously, and with the right rhythm, intonation, and elongation.

Ahsiah et al. in [4] suggested a method for verifying Tajweed and accurate Quran recitation. Tajweed, a system of rules for reciting the Al-Quran, guarantees accurate readings, pronunciations, and interpretations of the text. Experienced religious teachers have historically taught this knowledge. Usually, these teachers listen intently to the students' recitations and correct mistakes. There are limits to the traditional strategy, which requires qualified academics to be present to facilitate a self-learning environment. The author proposes a tajweed rule-checking system that uses speech recognition technology to help students learn and practice proper Al-Quran recitation independently. The suggested approach can identify and draw attention to differences in recitations stored in a database between student and seasoned professors. The system uses the HMM and

MFCC algorithms to extract features for classification. Yosrita et al. in [15] analyzed the various Tajweed techniques utilized throughout the recital of the Qur'an and used MFCCs to extract their attributes. Altalmas et al. in [19] suggested a method for accurately reciting the Quran by Tajweed regulations. Compared to words from various portions of the expression, words at the same point of articulation tended to have more matching sounds or less identical distances. To verify this, the author first used the Mel Frequency Cepstral Coefficient (MFCC) to evaluate the sounds of the words Y and I. Next, the author used the Dynamic Time Warping (DTW) technique to compare the sounds and identify similarities or discrepancies. But this method only applies to the two Quranic terms (Y and I). The work's content can be expanded by expanding the dataset and including all Quranic terms to uncover similarities and contrasts among all words.

Classic Arabic is very hard for non-native Arabic speakers, making it difficult to recite the Holy Quran. Short vowels in Arabic play an essential role in the correct Tajweed. Alqadheeb et al. in [39] suggested the proper Tajweed process employing an audio dataset of Arabic words, including terms with short vowels. There are 84 classes and 2892 Arabic short vowels in the entire dataset. Next, they put the preprocessing methods into practice and use CNN for testing and classification. With the word "ALIF" as a test word, 312 Arabic phonemes were used to test the model, and 100 accuracy was attained. In the future, we can expand the research to include all Arabic phonemes and train the model on them, but for now, the system is only intended to operate on the one phoneme, "ALIF." Rajagede et al. in [42] suggested a method to assist people in learning the Quran by memory without needing a second reciter. He suggested using the current Quranic data to validate the input recitation through an LSTM-based method. The Manhattan LSTM network is employed for recitation verification. If the recitation is comparable, it provides the output in a single numerical data set, and the Siamese classifier produces a binary classifier output. To improve model performance, they also compared various feature preprocessing strategies, such as delta features, Mel Frequency Cepstral Coefficient (MFCC), and Mel Frequency Spectral Coefficient (MFSC). Databases containing data from the Quranic Ayah comprise the dataset utilized for this purpose. With Manhattan LSTM and MFCC, the system achieves the most fantastic accuracy of 77.35%. However, in the future, to achieve better accuracy, it is recommended to use a deeper Siamese LSTM model or an attention-based model and use more data for training. To work on improving the Quranic Recitation system, Alqadasai et al. in [44] suggested a phoneme classification scheme to ensure proper Quranic recitation. The collection includes 30 reciters reciting 21 aayahs from the Quran. An HMM-based ASR model is used for training and analysis of the dataset. The duration added to the Quranic phoneme classification optimizes the method. For phoneme classification, the system's accuracy ranged from 99.87% to 100%. The expanded model should incorporate more datasets to address all Quranic recitation and Tajweed concerns, as the suggested methodology does not address all recitation-related issues. Omran et al. in [50] suggested a method based on deep learning for accurately comprehending the Holy Quran's Tajweed regulations. It is difficult to read the Holy Quran exactly as The Holy Prophet (PBUH) did. The author concentrated on the letters to which Qalqala principles are applied using the dataset of Quranic Audio read aloud by various Arabic reciters. After features are extracted using Mel Frequency Cepstral Coefficients (MFCC), a Convolutional Neural Networks (CNN) based model is applied for classification. The author often attains the greatest validation accuracy of 90.8%.

To encourage the Muslim community to read the Holy Quran with correct Tajweed and without any recitation error, Ahmad et al. in [18] suggested an approach that uses artificial neural networks and digital signal processing techniques to identify two Tajweed methods, namely Musyafahah and Talaqqi. Idghaam's audio files with both accurate and inaccurate recitation are included in the collection. Three distinct ANN classifiers—Levenberg-Marquardt optimization, Resilient Backpropagation, and Gradient Descent with Momentum—have been used to categorize the audio recordings after they have been preprocessed using the Mel Frequency cepstral coefficient technique to extract critical characteristics. The system's maximum accuracy for the Levernberg Marquardt algorithm is 77.7%. The system can be improved by including other classes of Tajweed techniques. The process's dataset is also limited; more audio files for each category can lead to higher accuracy.

Hijayah

Hijayah involves making sure that the written script of the Quran corresponds to the intended pronunciation during recitation.

Marlina et al. in [25] suggested a machine learning method for Makhraj letter recognition in Hijayah. The author extracted features from audio files using the Mel Frequency cepstrum coefficient (MFCC) and then classified Hijayah letters using support vector machines (SVM). Hijayah letter audio files are part of the dataset used for this purpose. Nevertheless, for the Makhraj classification, deep learning methods like ANN or CNN can enhance the system's output. Muslims must correctly pronounce and write the Hijaiyah Letter in the Holy Quran to adhere to the Holy Prophet's (PBUH) injunction regarding reading. Irfan et al. in [11] proposed the use of Dynamic Time Warping to distinguish between letters that are written and those that are pronounced. The collection includes sound recordings and picture images representing the letters of the Qur'an. Mel Frequency Cepstrum Coefficient (MFCC) and Principal Component Analysis (PCA) are used for processing image files when the data is good. Both results are displayed as numerical values, and the difference between them is then calculated using the Euclidian distance. The system's accuracy for image matching is 92.85%, and its accuracy for sound matching is roughly 71.42%. In the future, additional techniques like edge matching and linear discriminant analysis may be employed to process datatoo improve the system's accuracy. Alternatively, the application may only be used with Hijaiyah letters; then, we could expand it to match the differences between individual words or phrases.

Makhraj

Makhraj describes the posture of the tongue, lips, and vocal cords at the point of articulation for each Arabic letter. Proper Makhraj is essential for precise pronunciation.

Correct pronunciation of Hijayah letters in the Quran is essential, and it is always confusing for the user to pronounce similar sound letters. However, the wrong pronunciation changes the meaning of the word. The reciter should have an excellent knowledge of the difference between Hijayah letters. For this purpose, Wahidah Arshad et al. in [45] utilized the Speech analysis technique to tackle the problem of accurately identifying the nine-point articulations for recitation. Experts in a controlled setting have recorded the dataset of Quranic Makhraj letters. Five feature extraction approaches are used to preprocess the audio samples: MFCC, Mel spectrogram, Tonnetz, Spectral contract, and chroma. Three techniques have been used for classification: ANN, KNN, and SVM. The system used the ANN approach to obtain an overall accuracy of 56%. ANN, KNN, and

SVM are less effective in improving the system than Deep learning techniques like CNN and RNN. Farooq et al. in [41] suggested a deep learning-based approach to identify Quranic mispronunciations. The system aims to automate the Quran's manual instruction technique, which calls for a teacher or instructor. The RASTA PLP technique is employed for feature extraction from the dataset, which consists of audio recitations of Quranic words in Arabic. Then, the Hidden Markov Model (HMM) is used to train the system. 85% is the recognition rate attained with RASTA PLP. An extension of the system has been made to create a real-time application. The technique is restricted to single Arabic phoneme words, albeit these can be expanded into Quranic Ayyahs and various Tajweed rules, such as Tanween and Qalqala rules.

Imlaah and Iqlaab

Imlaah is the term used to describe the lengthening of particular Arabic letters inside a word, in contrast to Iqlaab, which involves changing a specific letter, Noon, into a different sound, namely the sound of Meem, when followed by the Arabic letter Ba, when a reciter encounters a letter with an "Izhaar" (clear pronunciation) diacritic, such as Alif, Waw, or Yaa. These letters' specific phonetic properties and how they interact in some word combinations cause this change to happen.

Yousfi et al. in [16] suggested a method for identifying Imaalah guidelines when reciting the Quran. When learning the Quran, accurate recitation is crucial, particularly adhering to Tajweed regulations. The Mel Frequency Cepstral Coefficient (MFCC), a feature extraction approach, was used to preprocess a dataset of accurate Quranic audio recitations. Hidden Markov Models (HMM) were used for the categorization, and their results were compared to the proper recitation and real-time recitation that was gathered and stored in the database. The method achieves an accuracy between 68% and 85%. Yousfi et al. in [17] suggested a Quranic Iqlab verification rule. To do this, the author develops a speech recognition system that can first detect, then identity, and highlight any inconsistencies or incorrect application of Iqlab norms during recitation. Mel Frequency Cepstral Coefficient (MFCC), a feature extraction approach, and Hidden Markov Models (HMM), a feature classification method, are used to preprocess the dataset. The highest accuracy the system could obtain was 70.

2.2.4 Automatic Speech Recognition (ASR)

ASR is necessary to completely comprehend the sensitive nature of the Quran's precious passages. Acoustic modeling is the systematic process of meticulously maintaining the phonetic subtleties and rhythm, especially the Quranic verse recitation style. By bridging the gap between spoken word and digital representation, automatic speech recognition (ASR) for Quranic recitation maintains the recitation's authenticity. It opens the door to innovative applications that enable accurate transcription, analysis, and dissemination of the Islamic message. A drawback of generalizing high-variance and solving non-linear separable datasets is that the Hidden Markov Model (HMM)-Gaussian Mixture Model (GMM) is used in most studies on acoustic signals for Arabic languages. Regarding these issues, Thirafi et al. [28] suggested a novel method for utilizing Deep Learning techniques to train Arabic language acoustic models. The author merged Hidden Markov Models (HMM) with Bidirectional Long-Short Term Memory (BLSTM) to create a hybrid system. Compared to the HMM-GMM model, the system demonstrated satisfactory performance for the Arabic language. The WER development for HMM-GMM was 18.39%, whereas the WER for the suggested method was 4.63%. Additionally, the author examined many Quranic styles as models. To improve system performance, the model might also be expanded to include training on non-professional reciters, as the system is currently taught just on professional reciters reciting the Quran. Additionally, the system's transcription method uses QScript, which isn't typically employed with the Quran. Hadwan et al. [49] suggested a complete framework for Arabic ASR. Using deep learning and attention-based encoder-decoder approaches, the author creates an acoustic model. RNN and LSTM have been employed to construct a sizable Arabic language model for the Quran, while the Mel filter has been utilized for feature extraction. The language model has been trained on textual data, and the dataset includes speech data from 60 distinct reciters and textual Quranic data. Character error rate (1.9%) and word error rate (6.1%) were attained using the model. The model can perform better for huge datasets by looking into the better-proposed model or by employing a transducer model for encoding and decoding instead of a transformer model.

2.2.5 Other Categories

The research of various recitation styles, the categorization of alphabets, the use of text mining tools, the classification of maqams, and context-aware analysis are a few intricate elements that make up the study of the Quran. Investigating many ways to recite the Quran's verses, each infused with unique rhythms and accents that communicate significant meanings, is necessary to distinguish between different styles of recitation.

Ousfi et al. in [13] suggested a method to differentiate between various renditions of the Holy Quran, given the vast range of variations in Qira'at around the globe. The dataset was built for this purpose by utilizing expert teachers' and various students' recitations. Subsequently, the Hidden Markov Model (HMM) classification and the Mel Frequency Cepstral Coefficient (MFCC) feature extraction technique were used. In addition, the technology can identify recitation type inconsistencies.

Khairuddin et al. [29] created an automated system that calculates the various letters in the Holy Quran to allow pupils to practice reciting the verses both once and again. The 'ro' alphabets are the subject of this study. Quranic recitation features have been extracted using formant analysis, Mel frequency cepstral coefficient (MFCC), and Power Spectral Density (PSD). Two methods are used for classification: quadratic discriminant analysis (QDA) and linear discriminant analysis (LDA). With all 19 training features in repetition, the system reached a maximum accuracy of 95.8%, while during the learning phase, it attained accuracy of 82.1%.

Nur et al. [31] created two classes using an automated interpretation classification system: Tafsir Bil Ra'yi and Tafsir Bil Ma'tsur. Using the KNN algorithm, diversity was accomplished with a 98.12% accuracy. However, additional comparisons have been made using Fuzzy KNN and Modified KNN to achieve better performance. The algorithm that yielded the highest precision and the most significant error value was MKNN.

Using deep learning techniques, Shahriar et al. [43] suggested a classification scheme for eight distinct recitation modalities (known as Maqamat), which includes Tajweed, to assist pupils in memorizing the Holy Quran in compliance with the recitation guidelines. Various reciters' audio renditions of the Quran in various genres are included in the dataset used for this purpose. After that, it employs spectral analysis, energy and chroma analysis, Mel Frequency cepstral coefficient, and spectral analysis for feature extraction. It is then trained on deep-learning algorithms, such as ANN, CNN, and

LSTM. Using a 5-layer ANN network trained on 26 input features, the system achieves the best accuracy of 95.7%. If the dataset is expanded utilizing additional reciters rather than the two used in this work, the system can attain better precision; however, the system's performance may be impacted by the implementation of bi-directional LSTMs.

Moulay et al. [22] create an application framework that may be used for voice search and context-based searching of any chapter or verse in the Quran. The voice-search function enables users to do voice-based searches. The dataset includes 36 reciters reciting Quranic chapters on audio, featuring eight well-known interpretations and four translations of the Holy Quran. However, the framework has several drawbacks, such as the total absence of notifications based on GPS location. Adding Hadith information and enabling users to research Islamic teachings extensively could enhance the framework. The table 2.2 reviews research papers that share a dataset system model and compares their model results with other techniques.

Title	References	Dataset Availability	Dataset Used (Speech/Text)	System Model	Comparison of Techniques
MFC peak based segmentation for continuous Arabic audio signal	[6]	×	Speech	✓	×
Feature extraction of some Quranic recitation using Mel-Frequency Cepstral Coefficients (MFCC)	[10]	×	Speech	✓	✓
A Comparative Study of Different Speech Features for Arabic Phonemes Classification	[12]	×	Speech	×	✓
A comparative study of MFCC-KNN and LPC-KNN for Hijayyah letters pronunciation classification system	[14]	×	Speech	✓	✓
Quranic reciter recognition: A machine learning approach	[30]	×	Speech	✓	✓
Arabic speaker identification system using a combination of DWT and LPC features	[8]	×	Speech	✓	×
Automatic Classification of Reciters of Quranic Audio Clips	[21]	×	Speech	✓	✓
Quran Reciter Identification: A Deep Learning Approach	[26]	×	Speech	✓	✓
A Smart Flexible Tool to Improve Reading Skill based on M-Learning	[46]	×	Speech	×	×
Correct Pronunciation Detection for Classical Arabic Phonemes Using Deep Learning	[39]	×	Speech	×	×
Rule-Based Embedded HMMs Phoneme Classification to Improve Qur'anic Recitation Recognition	[44]	×	Speech, Text	✓	×
Automatic Detection of Some Tajweed Rules	[50]	×	Speech	×	×
Tajweed Classification Using Artificial Neural Network	[18]	✓	Speech	✓	×
Makhraj recognition of Hijaiyah letter for children based on Mel-Frequency Cepstrum Coefficients (MFCC) and Support Vector Machines (SVM) method	[25]	×	Speech	✓	×
Signal-based feature extraction for Makhraj emission point classification	[45]	×	Speech	✓	✓
Mispronunciation Detection in Articulation Points of Arabic Letters using Machine Learning	[41]	×	Speech	✓	×
Holy Qur'an speech recognition system Imaalah checking rule for warsh recitation	[16]	×	Speech	✓	×
Isolated Iqlab checking rules based on speech recognition system	[17]	×	Speech	✓	×
An End-to-End Transformer-Based Automatic Speech Recognition for Qur'an Reciters	[49]	✓	Speech	✓	×
Holy Qur'an speech recognition system distinguishing the type of recitation	[13]	×	Speech	✓	×
Implementation of text mining classification as a model in the conclusion of Tafsir Bil Ma'tsur and Bil Ra'yi contents	[31]	×	Speech, Text	✓	×

Table 2.2: Detailed review of research papers that shared dataset, system model and compare their model results with other techniques.

Chapter 3

Design and Methodology

We aim to build a comprehensive system model to build advanced deep learning techniques that could accurately classify Quranic Surahs using their acoustic feature. The system model is divided into a few fundamental phases shown in Figure 3.1, which include:

- Data Acquisition
- Data Augmentation
- Data Pre-processing.
- Voice Activity Detection
- Feature Extraction
- Deep Learning Model
- Extracting Deep Features from Trained Model
- Generating Evaluation Dataset from Training Data
- Testing: Surah Classification and Indexing
- Graphical User Interface (GUI) for Analysis

The general system stages are as follows:

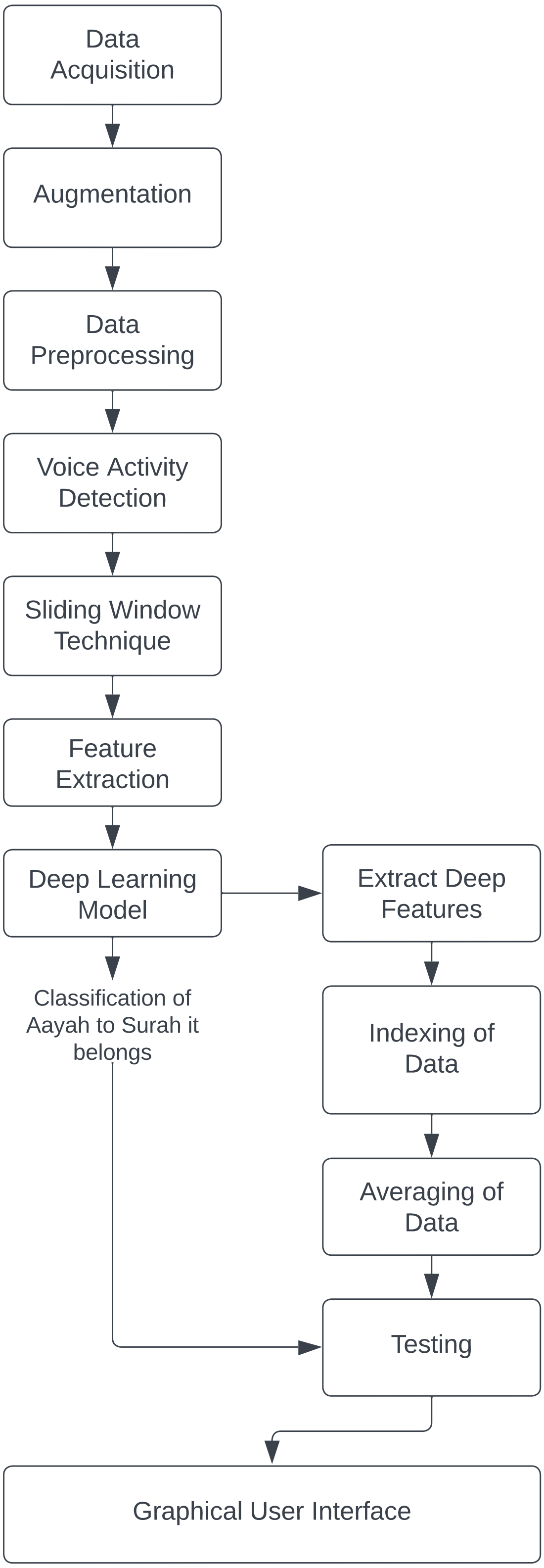


Figure 3.1: General Methodology of the System

3.1 Data Acquistion

Gathering audio recordings of people reading verses from the Quran is known as data acquisition for Quranic recitation. This procedure includes particular considerations to guarantee precise, high-quality, and culturally sensitive data collecting when reciting the last ten surahs (chapters). During the data acquisition process, we took care of a few parameters:

3.1.1 Selection of Reciters

Choose reciters knowledgeable about Quranic recitation and possess the necessary Tajweed (knowledge of Quranic pronunciation) abilities. Reciters with good vocal quality, appropriate rhythm, and clear pronunciation are favored.

3.1.2 Acoustic Environment

For recording, choose a controlled, calm atmosphere. Reduce any echoes, reverberations, and background noise that can affect how the recitation is heard. The recording environment can be made more audiophile by soundproofing.

3.1.3 Recitation Style and Speed

With the reciters, define the recitation speed and style. There may be particular recitation methods for some surahs (such as Hafs or Warsh) that must be followed. Before recording, ensure the reciters are at ease with the pace and style.

The dataset consists of audio files of Quranic Surah recited in Arabic by famous reciters who follow the proper Arabic dialect. These reciters include:

- Sheikh Saad Al Ghamdi
- Sheikh Ali Al Huhaifi
- Sheikh Mishary Rashid Al Afasy
- Sheikh Salah Al Budair

The dataset we constructed consists of the last 10 Quranic Surahs from Surah Al-Fil to Surah An-Nas; each contains a random length of every Surah, depending on their recitation style. The maximum size of a recitation is 27 sec. We have collected these audio files from the internet in MP3 format and converted them to wave format with a sampling frequency of 16kHz. Figure 3.2 shows the number of audio samples of every class.

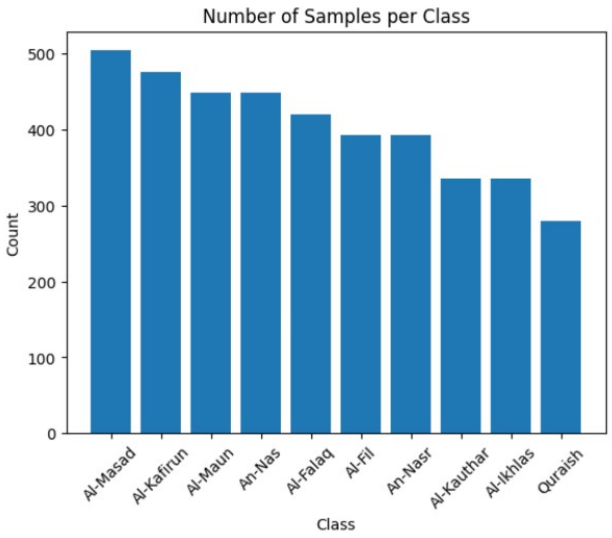


Figure 3.2: Number of Classes Per Sample included in the dataset

3.2 Data Augmentation

The process of making changed versions of the original data while maintaining its fundamental qualities and meaning is known as "data augmentation," it is used to increase the size of datasets artificially. Data augmentation can increase the dataset's diversity, strengthen the models' resilience, and solve problems like overfitting. [32]

Due to the limited number of audio files from every reciter that can be extracted online, we had to use a data augmentation technique to increase the size and produce diversity in training data artificially.

3.2.1 Time Stretching

Time stretching modifies the audio signal's duration while preserving its pitch. Using this, recitation speed variations can be simulated. When producing audio samples with varying pacing while keeping the original recitation style, time stretching can be helpful.

3.2.2 Pitch Scaling

Pitch scaling increases the dataset's diversity by changing the recitation's frequency content while leaving its rhythm and tempo untouched. This vocal augmentation method can imitate diverse vocal traits, giving reciters a more comprehensive representation and allowing models to generalize across varied vocal types more accurately.

3.2.3 Random Gain

This augmentation technique fills the dataset with a range of intensity levels by introducing carefully controlled fluctuations in the signal's amplitude. This version improves the model's capacity to capture changes in recitation styles and emotional nuance by simulating the varied voice dynamics displayed by various reciters.

3.2.4 Adding White Noise

Controlled background noise enables models to adapt and function well even in circumstances with fluctuating levels of ambient sounds by replicating real-world acoustic environments.

3.2.5 Polarity Inversion

This method alters the data by flipping the audio waveform in a minor yet noticeable way. While maintaining the recitations' semantic and cultural integrity, these changes broaden the dataset. The audio data becomes more complex due to inversion scaling.

We used transformation techniques using the Librosa audio library to produce diversity as shown in table 3.1.

3.3 Data Pre-processing

Several methods and procedures are used during the data preparation process to increase the usefulness and efficiency of the audio data.

Transformations	**Rate**
Time Stretch	0.1x to 0.5x
Pitch Scale	1x to 8x
Random Gain	Min Factor: 1x to 3x, Max Factor: 2x to 4x
Adding White Noise	0.1x to 0.8x
Polarity Inversion	-1

Table 3.1: Transformations implemented in data augmentation on Quranic audio data

3.3.1 Sampling

Sampling is digitizing analog audio signals by taking discrete samples of the waveform at predetermined intervals. [37] Selecting the correct sample rate is essential when dealing with audio data for the Quranic surahs.

3.3.2 Conversion to Mono from Stereo

Many audio recordings are made in stereo, which gives the left and right audio sources independent channels. Stereo recordings should be turned into mono since Quranic recitations frequently feature just one voice. This procedure combines The two channels into one channel, simplifying audio processing and requiring less storage. It's crucial to ensure that the mono conversion preserves the clarity and quality of the source audio.

3.3.3 Forced Align for Every Audio

Forced alignment is a method for lining up audio data with the matching textual material. Each recitation of the Quran is aligned with the relevant verses. Researchers and students can more successfully examine, study, and follow the recitations when audio and text are precisely synchronized. [2] The semantic and rhythmic integrity of the spoken verses is maintained by matching audio of the same duration to their associated poems, which makes it simpler to compare and study various recitations.

The data Preprocessing technique starts with sampling all audio data from 44.1 kHz and 22.05 kHz frequency to 16kHz frequency. It converges the audio signal to a mono (single) channel, then converts it from analog to digital, producing an array. But after VAD, when every ayah is extracted from the Quranic Surah in audio, the length of each audio is not the same, so we had to implement the Forced Align technique to map all audio to its maximum height with zero-padding to achieve the full length of ayah (12 seconds). The time-domain signal shown in Figure 3.3 is converted into the frequency domain shown in Figure 3.4 to visualize the spectral characteristics and distribution of frequency components in the signal. The frequency domain helps to extract meaningful features that are not easily visualized in the time domain.

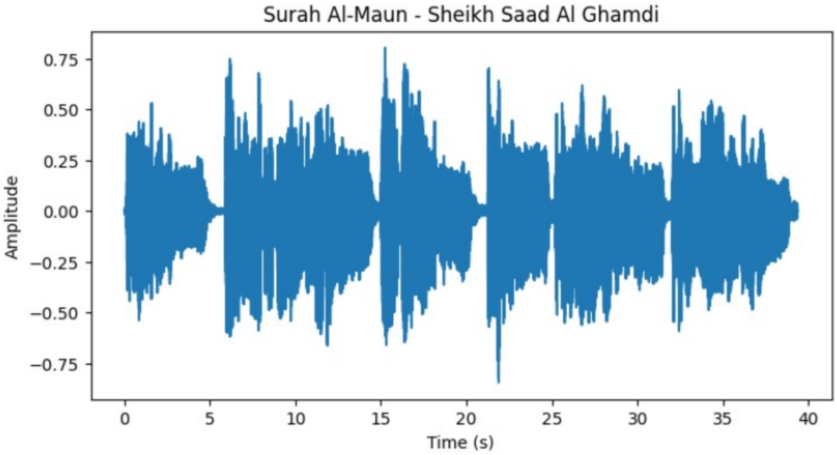


Figure 3.3: Audio Signal of Surah Al-Maun in Time Domain

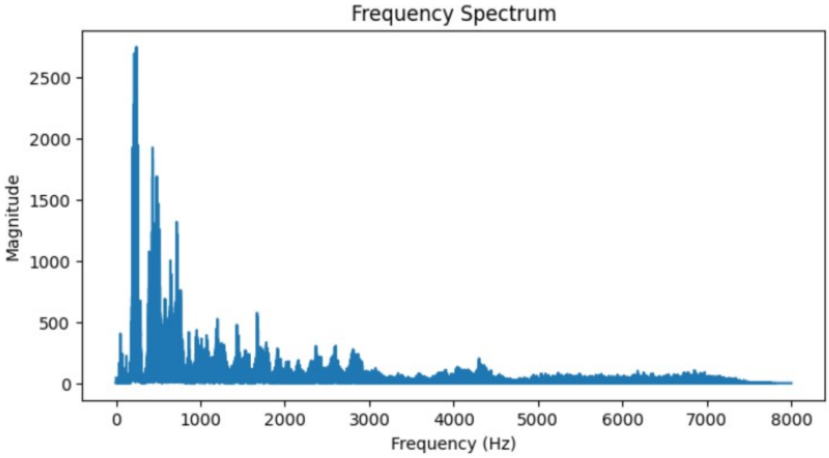


Figure 3.4: Audio Signal of Surah Al-Maun in Frequency Domain

3.4 Voice Activity Detection

A method for separating speech from background noise or silence in audio is voice activity detection (VAD), which locates address in audio segments. To extract individual ”ayahs” (verses) from a more extensive audio recording of a surah (chapter), VAD is essential. We implemented Voice Activity Detection to extract every ayah recited in the complete Quranic Surah. We used Silero-VAD technology, which can distinguish between areas of an audio sample with speech activity and others without. Silero-ASR (Automatic Speech Recognition) systems, on which the Silero-VAD model handles noises and artifacts that might interfere with the identification process.[40] It does its own filtering and noise reduction. The result of Silero-VAD extracts the speech activity along with their start and stop time from an audio signal; we used this start and stop time to remove a single ayah in the time domain from a complete surah recitation.

3.5 Feature Extraction

Feature Extraction technique transforms raw audio data into a more manageable and meaningful representation. Audio signals are typically complex and high-dimensional, consisting of a sequence of samples captured over time. Extracting relevant features from audio signals helps to reduce the dimensionality and capture the essential characteristics of the sound while preserving all necessary information. [35] It aids in accelerating the machine learning system’s learning rate and delivering accurate results with less time, effort, and resource consumption.

3.5.1 Mel Frequency Cepstral Coefficient

The technique used for feature extraction in speech signals is MFCCs. MFCCs are particularly effective in automatic speech recognition (ASR) and speaker recognition applications.

The MFCC computation involves several steps, as shown in Figure 3.5 in detail, including:

- Pre-Emphasis
- Framing

- Windowing
- Discrete Fourier Transform
- Mel Filter Banks
- Discrete Cosine Transform

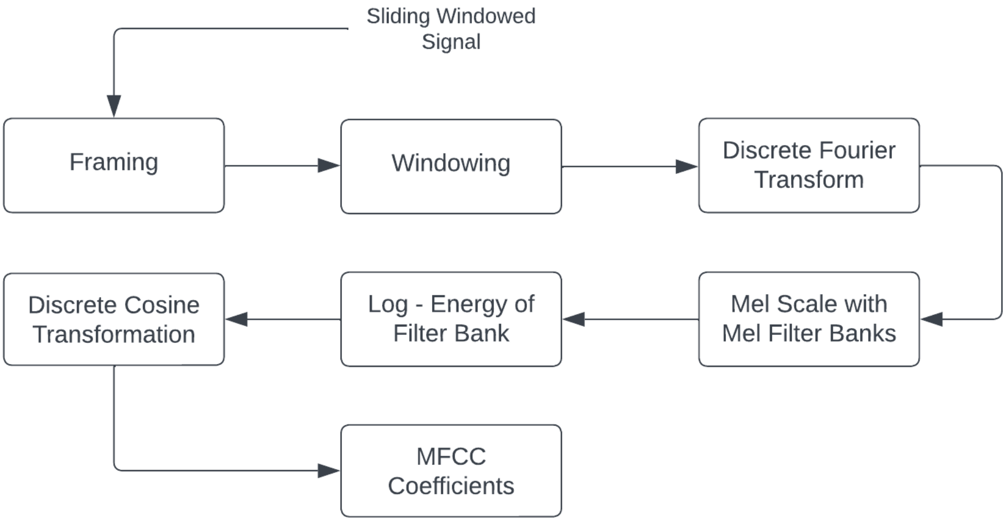

Sliding Windowed Signal
Framing
Windowing
Discrete Fourier Transform
Mel Scale with Mel Filter Banks
Log - Energy of Filter Bank
Discrete Cosine Transformation
MFCC Coefficients

Figure 3.5: Detailed Architecture of MFCC

Pre-Emphasis

Mel-Frequency Cepstral Coefficients (MFCCs), frequently used as features in speech and audio signal processing, must be calculated after pre-emphasis. Pre-emphasis is a technique used to alter a signal's spectral properties before further analysis, which improves the efficiency of later processing processes, including feature extraction, voice recognition, and more.

High-frequency elements frequently carry much energy and information in voice and audio communications. However, the higher frequencies may be diminished or muted because of how speech is produced and transmitted. The signal's high-frequency components may weaken as a result, which may impact the spectral balance of the movement as a whole. Pre-emphasis emphasizes the high-frequency components and amplifies their energy to combat this attenuation.

Pre-emphasis mathematically entails applying a high-pass filter to the signal. By increasing the amplitudes of the higher-frequency components relative to the lower-frequency ones, the filter draws attention to the high-frequency range. The following equation can be used to represent the pre-emphasis operation:

$$y(n) = x(n) - \alpha x(n-1) \tag{3.5.1}$$

where $y(n)$ is the pre-emphasized signal at time index n, $x(n)$ is the original speech signal at time index n, α is the pre-emphasis coefficient. Which is set to 0.97.

The pre-emphasis coefficient regulates how much focus is placed on high-frequency components. A greater value of produces a stronger emphasis on high frequencies, whereas a lower value moderates this emphasis.

The pre-emphasis process mainly focuses on the higher frequencies, compensating for the roll-off during the sound production and recording.

Figure 3.6 displays forty MFCC features. These features are extracted using the Librosa audio signal library.

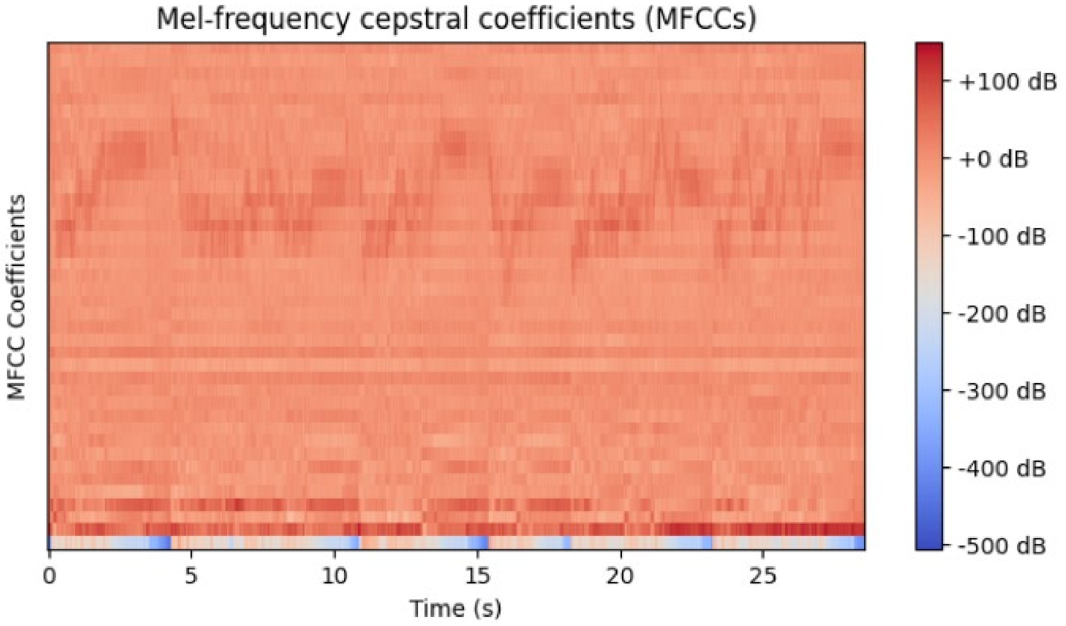


Figure 3.6: 40-MFCC features of an audio signal

Framing

A continuous audio signal is divided into smaller, overlapping pieces called frames during framing. Then, different analyses, including spectral analysis, are carried out over each frame as if it were a stationary signal segment. Framing is important because speech and audio signals are dynamic and change over time due to prosodic variations, phonetic transitions, and other causes.

The signal's properties and the trade-off between temporal and frequency resolution determine the best frame length. While shorter frames capture rapid changes but may not have enough frequency resolution, longer frames have superior frequency resolution but may miss rapid spectrum changes.[35]

Some overlap is introduced to lessen the jarring transition artifacts between neighboring frames. The presence of overlap guarantees adequate representation of spectral characteristics close to frame borders. The beginning of the following structure overlaps with the end of the previous frame by a standard overlap ratio of 50% or 75%. Before implementing MFCCs, we break the audio segments manually into various lengths (1, 2, and 3 seconds). Since the audio signal is converted into 16000 samples/sec, we implemented a sliding window technique of length (1, 2, and 3 seconds) to extract MFCC features so that each second would store 40 features into a mel filter bank.

Short-Time Processing/Windowing

Each frame is given a window function before further processing to reduce these artifacts. The window function is a mathematical function that reduces the abrupt transition from the frame to zero by tapering the signal at the boundaries of the frame. The window function is intended to smoothly taper from values near 1 in the center of the frame to zero at the edges. [24]

There are several types of window functions commonly used in signal processing, and the choice of window function depends on the characteristics of the signal and the desired properties of the analysis. Some common window functions include:

- **Rectangular Window**: Simplest window with constant value within the frame and zero outside.
- **Hamming Window**: Balances the frequency domain's main lobe width and side

lobe attenuation.

- **Hanning Window**: Similar to the Hamming window but with a smoother taper at the edges.
- **Blackman Window**: Provides better side lobe attenuation at the expense of a wider main lobe.

Short-time processing includes windowing the audio signal to reduce the spectral leakage effects caused by the abrupt termination of frames. We used the Hamming Window to obtain the magnitude of the Discrete Fourier Transform (DFT) because it is easy to analyze the signal in the frequency domain. The windowing of the signal is represented as:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) \tag{3.5.2}$$

where $w(n)$ is the windowing function at sample index n, and N is the total number of samples in the frame.

Discrete Fourier Transform (DFT)

The DFT converts a time-domain signal into its frequency-domain equivalent, revealing details about the signal's frequency components. The DFT is used in the context of MFCCs to examine the spectral composition of each signal frame. [9] Once each frame is windowed, the next step is to perform the DFT on each windowed frame. The DFT transforms a finite sequence of time-domain samples into a corresponding sequence of frequency-domain values. The magnitude spectrum is obtained by applying DFT to each windowed frame using the equation:

$$X(k) = \sum_{n=0}^{N-1}\left((w(n)x(n)\exp\left(\frac{-j2\pi kn}{N}\right)\right) \tag{3.5.3}$$

where $X(k)$ is the frequency spectrum at bin k, $w(n)$ is the windowing function at sample index n, $x(n)$ is the pre-emphasized signal at time index n and N is the total number of samples in the frame.

$$M(k) = \log(|X(k)|^2) \tag{3.5.4}$$

Where $M(k)$ is the logarithm of the magnitude spectrum at bin k and $X(k)$ is the frequency spectrum at bin k.

Mel Filter Banks

Mel filter banks extract pertinent spectral information from the signal's power spectrum while simulating the human auditory system's sensitivity to various frequency ranges. The Mel scale is a perceptual frequency scale that roughly represents the sensitivity of the human auditory system to various frequency ranges. It is a nonlinear scale that corresponds more closely to how people perceive pitch. [1] Frequency in Hertz (f) and frequency in Mel (m) are frequently approximated by the following formula:

$$M(f) = 2595 \cdot \log_{10}(1 + \frac{f}{700}) \tag{3.5.5}$$

Where $M(f)$ is the frequency f converted to the Mel scale.

Mel filter banks consist of a set of triangular filters spaced along the Mel scale. Each filter is defined by its center frequency, typically expressed in Mel units and width. The filters overlap and cover the entire frequency range of interest. The filter bank's center frequencies are usually linearly spaced in Mel scale and then converted back to Hertz for practical implementation. The response of each filter is computed using the following equations:

$$H(m,k) = \begin{cases} 0 & \text{if } f(k) < f(m-1) \\ \frac{f(k)-f(m-1)}{f(m)-f(m-1)} & \text{if } f(m-1) \leq f(k) \leq f(m) \\ \frac{f(m+1)-f(k)}{f(m+1)-f(m)} & \text{if } f(m) \leq f(k) \leq f(m+1) \\ 0 & \text{if } f(k) > f(m+1) \end{cases}$$

Where m = { 0, 1, 2 , . . . , $M-1$ }, $H(m,k)$ is the response of the m^{th} filter at frequency bin k and $f(k)$ is the frequency corresponding to bin k.

In the final step of this phase, the log-energy mel spectrum is computed by applying

$$M(k) = \ln\left(\sum_{k=0}^{N-1} |X(k)|^2 H_{m_k}\right) \tag{3.5.6}$$

where $X(k)$ is the DFT of the input audio $x(n)$.

Discrete Cosine Transform (DCT)

The DCT is applied to the logarithmically-scaled filterbank energies to convert the logmel spectrums back to the time domain called MFCCs. The equation for the DCT is:

$$C(i) = \sum_{k=1}^{K} \left(\log(|X(k)|^2) \cos \left(\frac{i(k-0.5)\pi}{K} \right) \right) \tag{3.5.7}$$

$C(i)$ is the $i-th$ MFCC coefficient, and K represents the total frequency bins.

3.6 Deep Learning Model

In the proposed technique, we build a multi-class classifier to classify every Aayah to Surah to which it belongs. We prioritize the classifiers that achieved high accuracy on speech recognition systems. Due to a large number of datasets, saving computation costs, and including reliability and usability of the system, we build a model that can be easily trained using low GPU even with a high number of training parameters. Model details are shown in Table 3.2.

Layers	**No. of Filters**	**Filter Size**	**Padding**	**Activation**
Conv-2D	64	3x3	Same	Tanh
MaxPool-2D	-	2x2	-	-
Conv-2D	64	3x3	Same	Tanh
MaxPool-2D	-	2x2	-	-
Conv-2D	64	3x3	Same	Tanh
MaxPool-2D	-	2x2	-	-
Dropout	Rate = 0.1	-	-	-
Dense	1024	-	-	Tanh
Dense	10 (No. of Classes)	-	-	-

Table 3.2: Detailed Architecture including layers detail for Deep Learning Model

3.6.1 Enhancing Model Generalization

We used the L2 Regularizer on the first layer to protect against overfitting and enhance the model's capacity for generalization. It accomplishes this by training the loss function with a regularization term. The model's weights (parameters) squared magnitudes determine the regularization term.

3.7 Extracting Deep features from Trained model

The process of extracting deep features from a pre-trained convolutional neural network (CNN) model. It is a high-level representation of audio data, which can be valuable for finding the deep parts in our audio file.

3.7.1 Loading the pre-trained Model

Loading a previously trained CNN model that has been trained on a Quranic verses dataset containing learned weights and biases.

3.7.2 Selecting the Intermediate Layer

We need to specify the intermediate layer from which we want to obtain the feature representations to extract deep features. In our case, we are interested in the 'dense' layer having 1024 deep features for every verse of the Quran, which is the model's second-to-last layer. We find the index of this layer in the model's layers list to specify its use.

3.7.3 Creating a Feature Extraction Model

We defined a new model that includes layers up to the selected intermediate layer. This model takes the same input as the original model but outputs the activations of the 'dense' layer. This eventually means that for every audio, the new feature extraction model would give the deep 1024 features of that audio. [20]

3.7.4 Predicting Deep Features

Now that we have our feature extraction model in place, we can use it to predict the deep features of our audio data. We already have preprocessed audio data stored in the variable that includes MFCC features; we used them to predict the deep parts of that audio verse.

3.8 Generating Evaluation Dataset from Training Data: Enhancing Model Performance

We leveraged a pre-trained convolutional neural network (CNN) model trained explicitly on a Quranic verse audio recordings dataset. Our objective was to extract deep features from the second-to-last layer of this model, which we identified as the 'dense' layer. This intermediate layer holds high-level representations of the audio data, capturing complex patterns and features learned during the model's training phase.

To initiate the feature extraction process, we loaded the pre-trained model, preserving its learned weights and biases, ensuring that it retained the knowledge acquired during its training on Quranic verse audio data. Next, we pinpointed the 'dense' layer within the model architecture by identifying its index in the list of model layers. This intermediate layer is strategically positioned before the final output layer, making it an ideal choice for capturing semantically rich features. With the intermediate layer's location established, we constructed a new model, referred to as the feature extraction model, by defining it to include all layers from the input up to and including the 'dense' layer. This model mirrors the architecture of the pre-trained model but is designed to focus solely on extracting deep features. [34] Once the feature extraction model was in place, we were ready to pass our preprocessed audio data. Our input data consisted of MFCC (Mel-frequency cepstral coefficients) representations, which had been prepared to match the format the model was originally trained on. By applying the predict function to the feature extraction model and providing it with our MFCC features, we obtained the activations of the 'dense' layer. These activations represented the deep features of our audio data, encoding intricate information about the Quranic verses' audio characteristics.

With our feature extraction model ready, we began by running our extensive collection of Quranic verse audio recordings through it. This dataset covered a wide range of verses and was carefully prepared to match the format the model originally learned from. Using the predict function with the feature extraction model, we systematically collected the output from the 'dense' layer for each audio sample. These outputs, which we obtained through this thorough process, are essentially the deep features that encapsulate the essence of our Quranic verse audio dataset. Think of them as condensed representations that capture the core audio characteristics of each Quranic verse. This process allowed us to tap into the knowledge acquired by the pre-trained model and capture meaningful, high-level features from our Quranic verse audio dataset. These extracted deep features hold great potential for enhancing the accuracy and interpretability of our audio analysis tasks, enabling us to make more informed insights and classifications.

3.8.1 Indexing of Verses

In our effort to organize and catalog Quranic verses from the last ten surahs, we took a careful approach. Our main aim was to create a structured database of these verses, as shown in Figure 4.9, making it easy to find and study them. To do this, we manually went through each verse's audio data individually and used the CNN model to get their special features. This process began by carefully selecting the verses from the last ten surahs. We made sure each verse was separated and ready for analysis. Then, using our feature extraction model that included a particular layer called 'dense,' we passed each verse's audio data through the model. What made our approach unique was the way we did it in order. We followed the order of verses for each surah as they appear in the Quran. So, the audio from the first ayat went with the first verse, the audio from the second ayat with the second verse, and so on. This way, we got deep features that matched the order of the poems in the Quran. As a result of this careful process, we collected a sequence of deep features, one for each verse. These features are like special codes representing the verses and keeping the Quranic text's original order. We use these features as the basis for our organized database of Quranic verses from the last ten surahs. This structured dataset doesn't just make it easy to find verses; it's also a valuable resource for studying and researching Quranic audio.

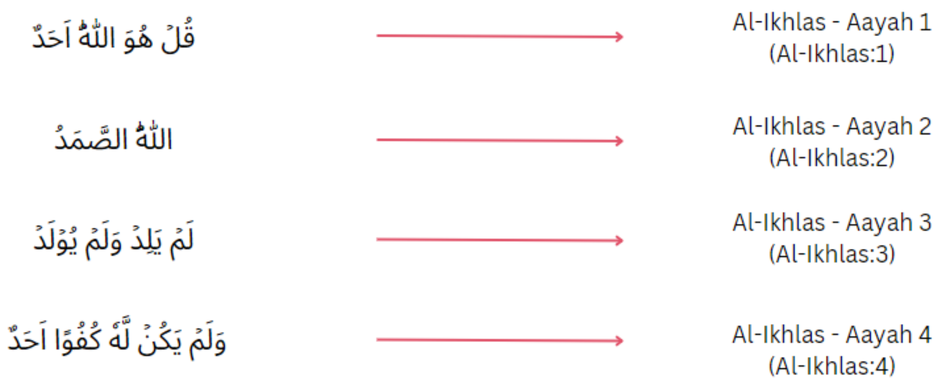


Figure 3.7: Indexing of Surah-Ikhlas

3.8.2 Averaging of Verses

After we've finished organizing and arranging the Quranic verses and created databases for each verse, we face a challenge when dealing with different reciters who may use different dialects. To address this issue, we adopt a clever approach to calculate the average (or mean) of the same verses from multiple reciters. [3] The formula used for the mean is simple Euclidian distance between two vectors as shown in Figure 3.8

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

Figure 3.8: Formula for Euclidian Distance

For each verse, we have recordings from various reciters, and these recordings might sound a bit different due to dialect or pronunciation variations. We create a more consistent and robust representation by taking the mean of these other recordings of the same verse. This means that we smooth out any differences caused by dialects, and we're left with a more unified and reliable version of the verse. In essence, this process helps us eliminate dialect-related problems. It ensures that our database represents the Quranic verses more standardized and consistently, making it easier to work with and study.

After calculating and saving these means in our database, we essentially have a deep feature representation for every verse of the last ten surahs of the Quran. This collection of deep features allows us to make meaningful comparisons and analyses. This standardized representation of each verse allows us to compare and analyze them later quickly. Whether for research, classification, or any other analysis, having these deep features ensures that we work with a consistent and reliable dataset, making our tasks more accurate and efficient. This database of deep parts becomes a valuable resource for our Quranic verse analysis and research.

3.9 Testing: Surah Classification Using the Model and Indexing Based on Mean Representations

For the testing phase, we're presented with an audio ayah recited by a different reciter, one that wasn't included in our training data and may have a distinct dialect. Despite this variation, our trained convolutional neural network (CNN) model demonstrates its capability by accurately predicting the Surah name to which the ayah belongs. This showcases the model's ability to generalize and identify the Surah based on audio features, even when presented with variations in pronunciation and dialect.

Additionally, our database, which contains the mean representations of every ayah in a sequential order, plays a crucial role in this process. When we want to find the index of the test ayah within our database, we leverage the concept of Euclidean distance. We effectively locate the closest match by measuring the distance between the test ayah's deep feature representation and all the stored representations in the database. This distance-based comparison allows us to find the most similar ayah in the database, thereby helping us determine the index or location of the test ayah within the Quranic text. In essence, this combined approach enables accurate Surah prediction despite dialect variations and facilitates the efficient retrieval and indexing of the test ayah within our structured database. It's a testament to the power of deep learning and feature representations in handling diverse Quranic audio data and making it accessible for analysis and study.

3.10 GUI for Quranic Audio Testing and Analysis

Our user-friendly Graphical User Interface (GUI), "Deep Acoustic Modelling for Quranic Recitation," is designed to make Quranic audio analysis easy and accessible. With this tool, users can test and analyze Quranic recitations effortlessly.

3.10.1 Input Section

In the GUI, we offer a straightforward input section. Users can upload their audio files in WAV format, containing the recitation of a specific Quranic ayah. This process is hassle-free and requires no coding skills.

3.10.2 Audio Playback

Listening to the audio is a crucial aspect of analysis. We provide an audio playback feature directly within the GUI to facilitate this. Users upload the WAV file and listen to the uploaded audio file, ensuring they've selected the correct one for analysis.

3.10.3 Output Section

The GUI also provides insightful outputs:

- Predicted Surah Name: After analyzing the uploaded audio, our trained model predicts the Surah name accurately. This is especially helpful for users who want to verify their recitation.
- Index of Surah: To pinpoint the location of the Surah within the Quran corresponding to the tested audio, we display the index or locality. This information is presented clearly, such as "(Surah Name: Indexing)."

Our GUI simplifies testing Quranic audio and obtaining valuable results, making it an accessible tool for anyone interested in Quranic recitation analysis.

Chapter 4

Implementation and Results

The chapter provides a comprehensive overview of the results obtained from each section discussed in Chapter 3. It delves into the details of each experiment and analysis, offering valuable insights and findings. It begins by presenting the outcomes of the methods outlined in Chapter 3, starting with a discussion of the data collection process. It elaborates on the dataset used, its characteristics, and any preprocessing steps applied to prepare the data for analysis. It then delves into the model training and evaluation process, highlighting the various experiments performed to assess model performance. It provides a detailed breakdown of the different configurations and parameters tested and the corresponding results. Key metrics such as accuracy, precision, recall, and F1-score are presented and analyzed for each experiment.

Additionally, it showcases the visual representations of the results, such as graphs, charts, or tables, to provide a clear and concise summary of the findings. These visual aids help readers better understand the patterns and trends observed in the data.

4.1 Data Preprocessing

Visualizing audio data is crucial in understanding and analyzing the acoustic characteristics of Quranic recitations. By visualizing audio in both the time and frequency domains and through power spectrums, we gain valuable insights into the audio signals' properties. In the time domain, we represent the audio as a temporal sequence, enabling us to observe patterns related to speech rhythm, intonation, and pauses. In the frequency domain, techniques such as spectrograms reveal the underlying frequency components, helping us identify variations in pitch, tone, and vocal characteristics as shown in Figure 4.1

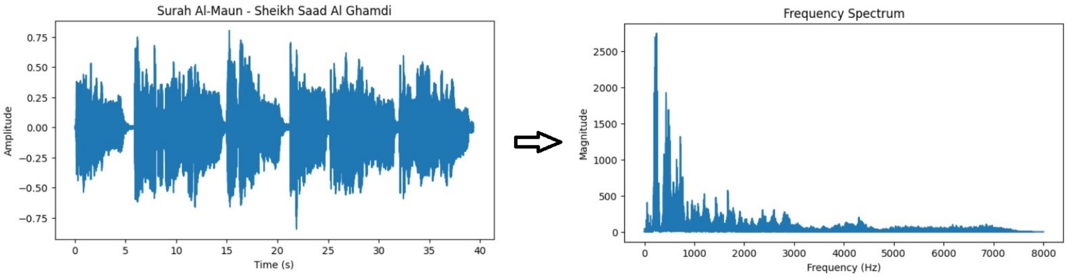


Figure 4.1: Conversion of Time-domain signal into the Frequency domain for visualization

Additionally, power spectrums provide information about energy distribution across frequencies, highlighting moments of emphasis, silence, or vocal intensity variations during recitation, as shown in Figure 4.2. These visualizations enhance our understanding of the Quranic audio and create features for further analysis.

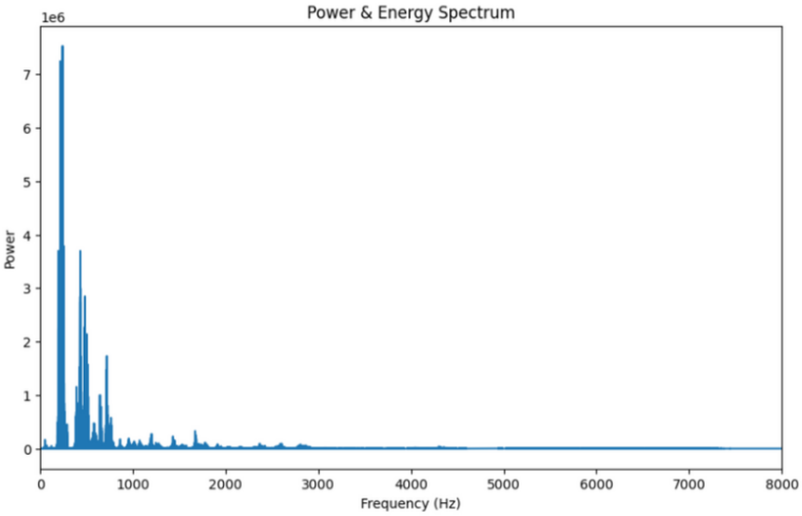


Figure 4.2: Power Spectrum of the audio signal

4.2 Feature Extraction

MFCC coefficients provide a compact yet highly informative representation of audio features, making them an excellent choice for capturing the acoustic nuances present in Quranic recitations. The decision to extract 40 MFCCs for segments of varying

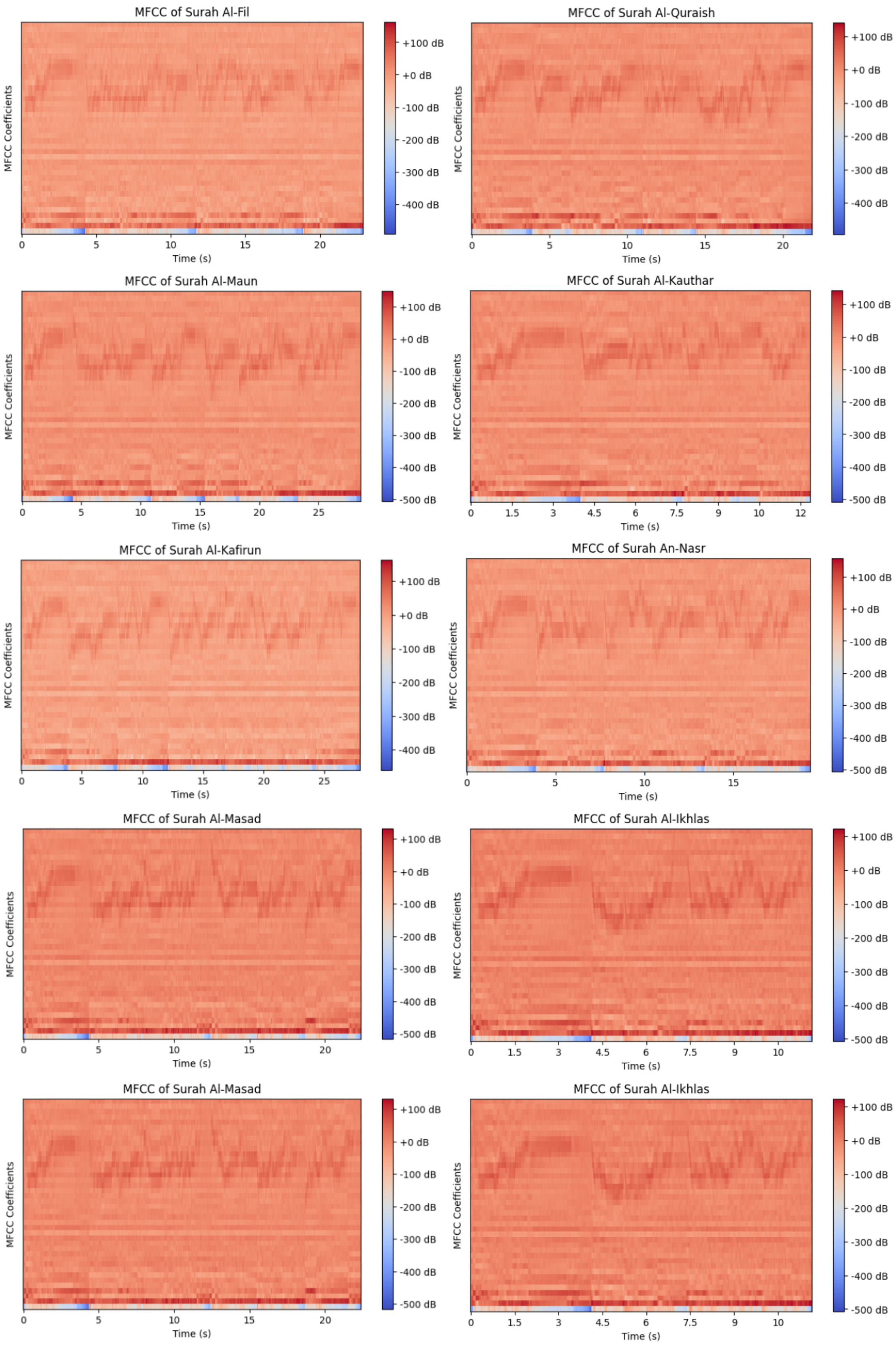


Figure 4.3: MFCCs representation of every Surah

durations, including 1, 2, and 3 seconds, is significant. For each 1-second segment, the extraction of 40 MFCCs allows for a detailed examination of the audio's spectral characteristics within short intervals. This fine-grained representation helps uncover intricate phonetic and tonal patterns within the recitation. Similarly, for two and 3-second segments, maintaining the same number of MFCCs ensures consistency in feature dimensionality across different segment lengths. The MFCCs representation of every Surah is shown in Figure 4.3

4.3 Deep Learning Model

The section shows the experimental results of our proposed model. Performance matrices include Accuracy of the system, Loss per epoch, F1 score, and confusion matrices evaluated in the model.

The total number of 40 MFCC features are extracted from each audio segment of length (1, 2, and 3 seconds), and for each segment, the model is evaluated on the abovementioned parameters. The dataset has been split into different categories of training and testing, including 90-10, 80-20, and 70-30 data. The dataset categories are shared in Table 4.1

Datasets	Sliding Window Length	Total MFCC Features
Dataset-1	1 sec	40
Dataset-2	2 sec	40
Dataset-3	3 sec	40

Table 4.1: Categories Based on Sliding Window Length for Short-Time Processing

4.3.1 Complete Dataset with Various Segment Lengths

Making use of the proposed model (CNN). Different segment lengths and feature counts are used in the suggested model for reciter recognition. The model evaluation aims to identify the ideal segment length and the ideal number of characteristics, resulting in the most significant degree of accuracy.

1-second Segment Length

Performance indicators, such as accuracy, F1-measure, recall, and precision, are used to assess the findings. We used a sliding window size of 1 second to process our data. We divided our dataset into a training set and a testing set using a 90-10 split, which means 90% of the data was used for training, and 10% was used for testing. The accuracy of our model was impressive, reaching 96.3%. This means that our model correctly classified 96.3 of the data points. Additionally, we evaluated our model's performance using precision, recall, and F1-score metrics. The precision was 0.9641, indicating that when our model predicted a positive result, it was correct 96.41% of the time. The recall was 0.9628, signifying that our model successfully captured 96.28% of the actual positive instances. Lastly, the F1-score, a balanced measure of precision and recall, was 0.9627, demonstrating the overall effectiveness of our model in handling this specific task. These results highlight the strong performance of our model in processing data with a 1-second sliding window and a 90-10 train-test split. The results are shown in Table 4.2 It's also noteworthy that the proposed system only requires 40 characteristics to reach this level of accuracy (0.9629), but utilizing only a few features yields the lowest accuracy. It shows that adding features can significantly enhance the model's performance, although the increase is not immediate.

2-second Segment length

For this analysis, we used a sliding window size of 2 seconds to process our data. Similar to the previous experiment, we split our dataset into a training set and a testing set using a 90-10 split, where 90% of the data was used for training, and 10% was reserved for testing. The accuracy of our model in this setup was 96.29%, indicating that it correctly classified 96.29% of the data points as shown in Table 4.2 Let's compare these results with the previous experiment, where we used a 1-second sliding window:

- **Accuracy:** The accuracy remained high in both cases, with a slight drop from 96.3% to 96.29%. This suggests that the change in sliding window size didn't significantly impact overall accuracy.
- **Precision:** The precision increased slightly from 96.41% to 96.43%, indicating that the model was slightly more accurate when predicting positive results in this

experiment.

- **Recall:** The recall remained relatively consistent at 96.28% in both experiments, showing that the model continued to capture most positive instances effectively.
- **F1-Score:** The F1-score, which balances precision and recall, showed a slight decrease from 96.27% to 96.26%, indicating that overall performance was still strong.

The change in sliding window size from 1 second to 2 seconds had a minimal impact on model performance, with only minor fluctuations in accuracy, precision, recall, and F1-score. This suggests that the model was robust to variations in the sliding window size and maintained high-quality results.

3-second Segment Length

In this analysis, we utilized a sliding window size of 3 seconds to process our data. Similar to the previous experiments, we partitioned our dataset into a training set and a testing set using a 90-10 split, where 90% of the data was allocated for training and 10% for testing. The accuracy of our model in this configuration was 95.05%, signifying that it correctly classified 95.05% of the data points as shown in Table 4.2 If we compare these results with the previous experiments that used a 1-second sliding window, we see that:

- **Accuracy:** The accuracy dropped slightly from 96.3% (1-second window) to 95.05% (3-second window). This suggests that the change in sliding window size slightly impacted the model's performance.
- **Precision:** The precision also decreased from 96.41% (1-second window) to 95.30% (3-second window), indicating a slightly lower accuracy in predicting positive results in this experiment.
- **Recall:** The recall followed a similar pattern, decreasing from 96.28% (1-second window) to 95.04% (3-second window), showing that the model captured a slightly lower proportion of actual positive instances.

- **F1-Score:** The F1-Score, a balance between precision and recall, also saw a decrease from 96.27% (1-second window) to 95.04% (3-second window), suggesting an overall reduction in performance.

The change in sliding window size from 1 second to 3 seconds had a noticeable impact on model performance, resulting in a slight decrease in accuracy, precision, recall, and F1-score. This indicates that the more extended temporal context introduced by the 3-second window may not have been as beneficial for this specific task as the 1-second window. The choice of sliding window size should be considered carefully based on the particular requirements and characteristics of the data and job at hand.

The results show that when the number of features rises, accuracy and other performance indicators rise as well, except RMSE, which falls. The suggested system achieves a maximum accuracy of 0.9629. The comparison is shown in Figure 4.4.

Sliding Window Size	Train-Test Split	Max. Accuracy (%)	Precision	Recall	F1-Score
1 sec	90-10	96.30	0.9641	0.9628	0.9627
2 sec	90-10	96.29	0.9643	0.9628	0.9626
3 sec	90-10	95.05	0.9530	0.9504	0.9504

Table 4.2: Maximum accuracy achieved for each Sliding Window Size and Train-Test Splits, with the maximum Recall, Precision, and F1-Score achieved

4.3.2 Complete Dataset with Various Splitting

We manually divided our dataset into train-test segments to evaluate our machine-learning model's performance. We've trained and tested the model separately for each of these segments. After completing this process for all the segments, we can compare the model's performance across them. The goal is identifying which specific train-test split yields the best results according to our chosen evaluation metrics. This approach allows us to assess how well our model generalizes to different data distributions and helps us make informed decisions about its robustness and suitability for real-world applications. Once we've identified the optimal model and train-test split, it's essential to retrain this selected model on our full training dataset and validate its performance on an independent test set, ensuring it performs reliably on unseen data. This manual train-test splitting and evaluation process can provide valuable insights into our model's behavior under various data conditions.

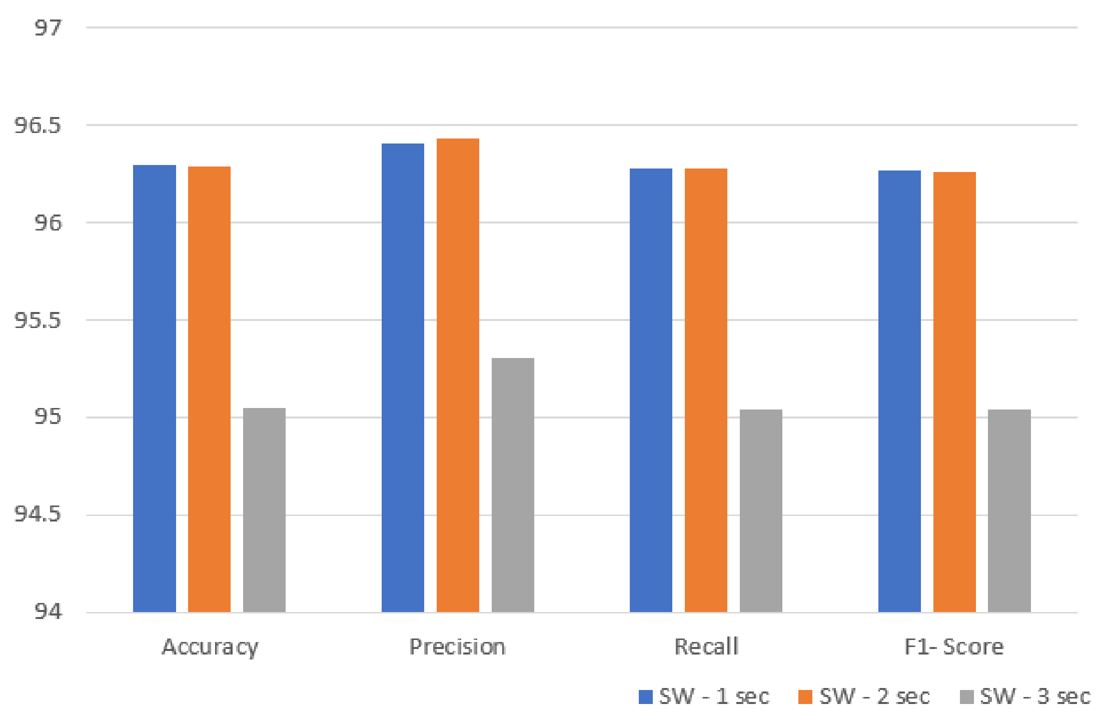


Figure 4.4: Evaluation Matrices representing Accuracy, Precision, Recall and F1-Score of the trained model

We have evaluated a deep-learning model using various sliding window sizes and train-test split ratios. In the first experiment, a sliding window size of 1 was employed, signifying that the dataset was divided into sequential, 50% overlapping segments, each containing a single data point. The 70-30 train-test split was utilized, with 70% of the data allocated for training and the remaining 30% for testing, resulting in an accuracy of 93.31%. The second experiment retained the sliding window size of 1 but increased the training data proportion to 80%, with 20% reserved for testing, yielding a slightly improved accuracy of 93.68%. Lastly, in the third experiment, the same sliding window size of 1 was maintained, but an even more imbalanced 90-10 train-test split was implemented, with 90% used for training and only 10% for testing. This configuration produced the highest accuracy of 96.3%, demonstrating that utilizing a larger portion of the data for training significantly enhanced the model's performance. These results reflect the impact of sliding window size and train-test split ratio on accuracy and provide valuable insights for optimizing model training in similar scenarios. Results are shown in Table 4.3

Sliding Window Size	Train-Test Split	Accuracy (%)
1 sec	70-30	93.31
1 sec	80-20	93.68
1 sec	90-10	96.30

Table 4.3: Accuracies for 1-second Sliding Window Size and different Train-Test Splits

For the 2-second sliding window size, the focus shifted to the impact of different sliding window sizes (measured in seconds) and train-test split ratios on the performance of a machine-learning model. The first experiment employed a sliding window size of 2 seconds, meaning that the dataset was divided into sequential, 50% overlapping segments, each spanning 2 seconds of time. With a 70-30 train-test split, 70% of the data was utilized for training, and the remaining 30% for testing. This configuration yielded an accuracy of 92.23%, indicating that the model performed reasonably well in correctly classifying instances within these time-based segments.

In the second experiment, the sliding window size remained at 2 seconds, but the training data proportion was increased to 80%, with 20% reserved for testing. This adjustment led to an improved accuracy of 93.3%, suggesting that a larger share of the dataset allocated to training contributed to enhanced model performance.

The third experiment maintained the sliding window size at 2 seconds but implemented a more imbalanced 90-10 train-test split, with 90% of the data dedicated to training and only 10% for testing. Remarkably, this setup resulted in the highest accuracy of 96.29%, showcasing the significance of using a more substantial portion of the data for training. These findings underline the influence of sliding window size and train-test split ratio on the model's accuracy, underscoring the importance of carefully optimizing these parameters to achieve the best performance in time-series data analysis or similar applications. Results are shown in Table 4.4

Sliding Window Size	Train-Test Split	Accuracy (%)
2 sec	70-30	92.23
2 sec	80-20	93.3
2 sec	90-10	96.29

Table 4.4: Accuracies for 2-second Sliding Window Size and different Train-Test Splits

In the first experiment, employing a sliding window size of 3 units, the dataset was partitioned into sequential, 50% overlapping segments, each spanning three teams. With a 70-30 train-test split, 70% of the data was dedicated to training, while the remaining 30% was used for testing. This configuration resulted in an accuracy of 91.49%, showcasing the model's ability to classify instances within these more significant temporal segments.

In the second experiment, maintaining the sliding window size at three units, the training dataset was expanded to 80%, with 20% set aside for testing. This adjustment led to an enhanced accuracy of 92.94%, emphasizing the positive impact of increasing the training data proportion on model performance.

The third experiment retained the sliding window size of 3 units but implemented a more imbalanced 90-10 train-test split, with 90% of the data allocated for training and only 10% for testing. This setup produced an even higher accuracy of 95.05%, underscoring the value of a more substantial training dataset in improving model accuracy. Results are shown in Table 4.5

Sliding Window Size	Train-Test Split	Accuracy (%)
3 sec	70-30	91.49
3 sec	80-20	92.94
3 sec	90-10	95.05

Table 4.5: Accuracies for 3-second Sliding Window Size and different Train-Test Splits

Comparing these results to the previous experiments with a sliding window size of 2 seconds, it's evident that a larger window size led to slightly lower accuracies across all three train-test split ratios. However, increasing the training data proportion still resulted in improved performance. The graphical representation and comparison of different splitting on different segment lengths are shown in Figure 4.5

4.3.3 Optimal Model Performance

The proposed model is trained for 20 epochs, with a batch size of 64 for each category with different split sizes, producing nine results. The maximum accuracy achieved is 96.3%, and the F1 Score of 0.9627 for the model trained on segment 1-second with 40 MFCCs feature for every second, making it the highest number of features. The

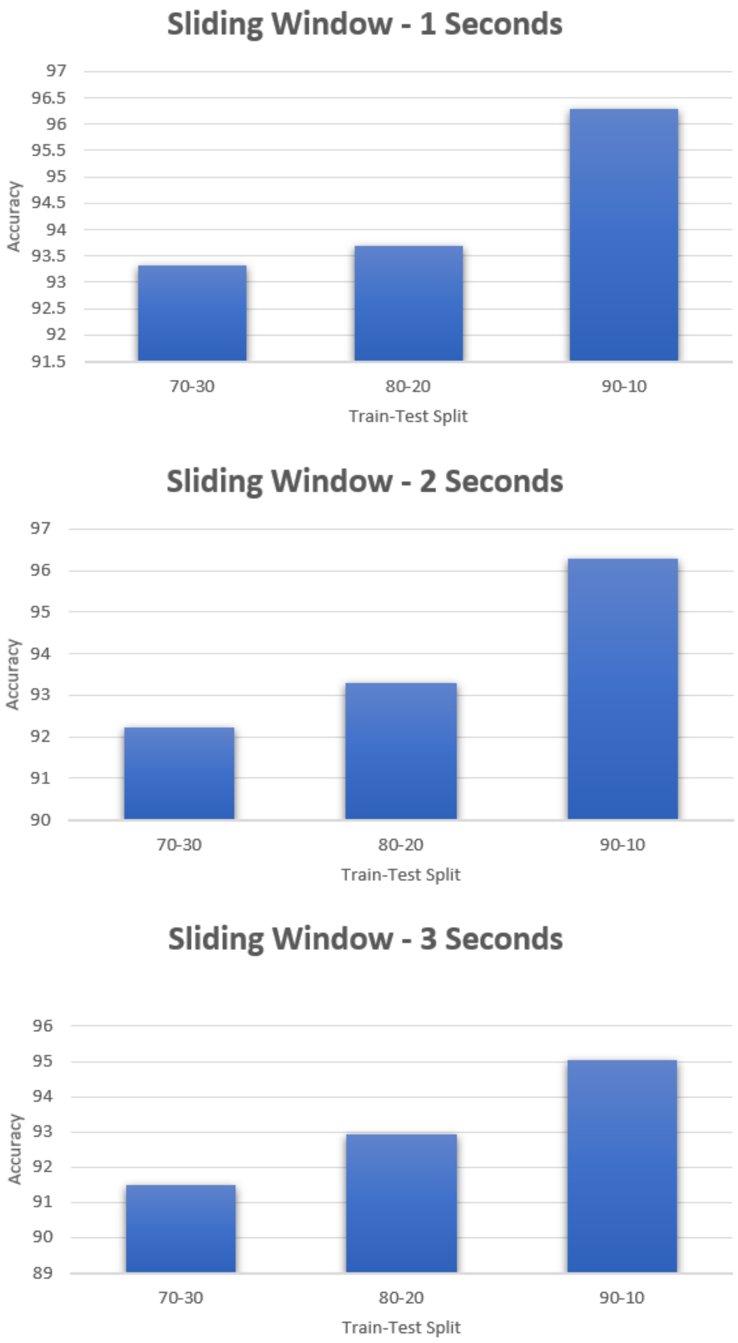


Figure 4.5: Accuracy achieved by trained model on different splitting and segment lengths

optimizer used for the purpose is Adam because of its capability of adaptive learning rate (dynamically adjusts the learning rate) and momentum optimization and loss computed using categorical cross-entropy.

The highest achieved accuracy among the various experiments conducted with different sliding window sizes and train-test split ratios was observed using a 1-second sliding window size and a 90-10 training and test split. The machine learning model achieved an impressive accuracy score of 96.30% in this configuration.

This result indicates the model's exceptional performance when trained on a relatively large proportion of the dataset (90%) and tested on a smaller portion (10%). Such a substantial training dataset enabled the model to capture underlying patterns and relationships within the data, allowing it to make highly accurate predictions on unseen test data. The choice of a 1-second sliding window size suggests that the dataset was divided into sequential segments, each spanning 1 second of time. This granularity allowed the model to analyze and learn from fine-grained temporal patterns within the data, contributing to its robust performance. The optimal training results, including accuracy and loss, are shown in Figure 4.6

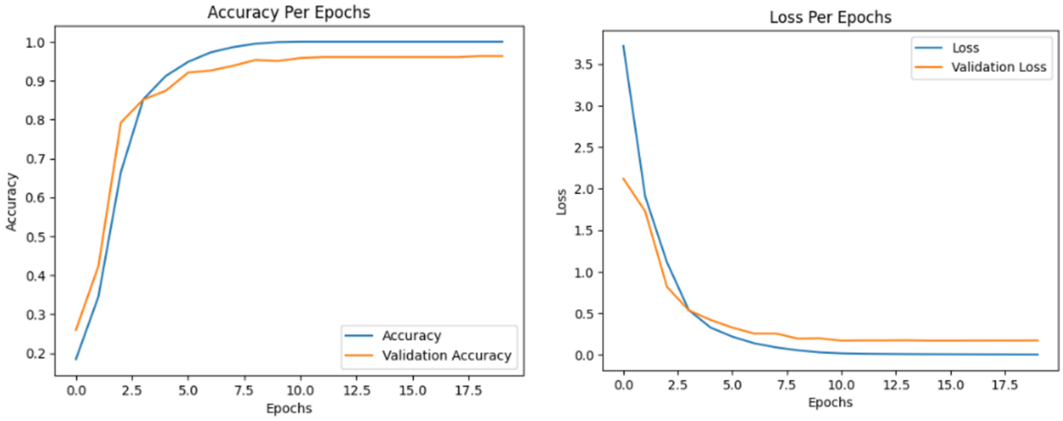
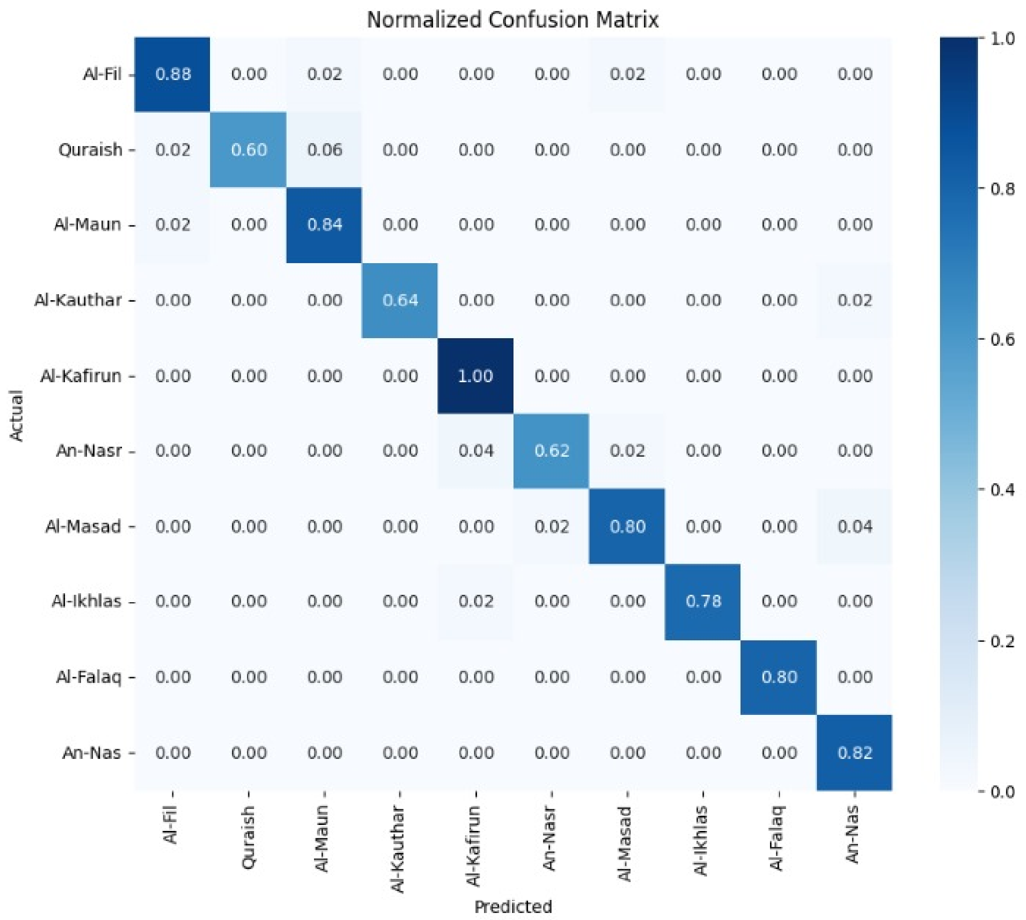


Figure 4.6: Accuracy and Loss per epoch achieved by optimal training settings

Achieving a confusion matrix with all correct predictions highlights the model's high accuracy and reliability. It can be trusted for real-world applications where accurate class assignments are critical. The confusion matrix for our trained model is shown in Figure 4.7

However, the second-highest achieved accuracy in the series of experiments was observed when using a 2-second sliding window size and a 90-10 training and test split. The machine learning model achieved a remarkable accuracy score 96.29 in this configuration. Although it falls just slightly short of the optimal accuracy achieved with 1-second sliding window size and the same 90-10 training-test split (96.30%), it still represents an exceptional level of performance.

The utilization of a 2-second sliding window size suggests that the dataset was divided



	Al-Fil	Quraish	Al-Maun	Al-Kauthar	Al-Kafirun	An-Nasr	Al-Masad	Al-Ikhlas	Al-Falaq	An-Nas
Al-Fil	0.88	0.00	0.02	0.00	0.00	0.00	0.02	0.00	0.00	0.00
Quraish	0.02	0.60	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Al-Maun	0.02	0.00	0.84	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Al-Kauthar	0.00	0.00	0.00	0.64	0.00	0.00	0.00	0.00	0.00	0.02
Al-Kafirun	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
An-Nasr	0.00	0.00	0.00	0.00	0.04	0.62	0.02	0.00	0.00	0.00
Al-Masad	0.00	0.00	0.00	0.00	0.00	0.02	0.80	0.00	0.00	0.04
Al-Ikhlas	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.78	0.00	0.00
Al-Falaq	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.80	0.00
An-Nas	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.82

Figure 4.7: Normalized Confusion Matrices for the optimal trained model

into sequential segments, each spanning 2 seconds of time. This choice allowed the model to analyze broader temporal patterns within the data while maintaining high accuracy.

The 90-10 training and test split ratio ensured that the model was trained on a substantial portion (90%) of the dataset, providing ample opportunities to learn and extract meaningful insights. The slightly lower accuracy than the optimal setting might be attributed to the larger sliding window size, which could result in the model missing out on finer details or rapid changes in the data. The detailed result of accuracy and loss per epoch is also attached in Figure 4.8

4.4 Extraction of Deep Features

The results you see are generated by the second-to-last layer of the trained neural network model, specifically the dense layer with 1024 neurons. In a neural network archi-

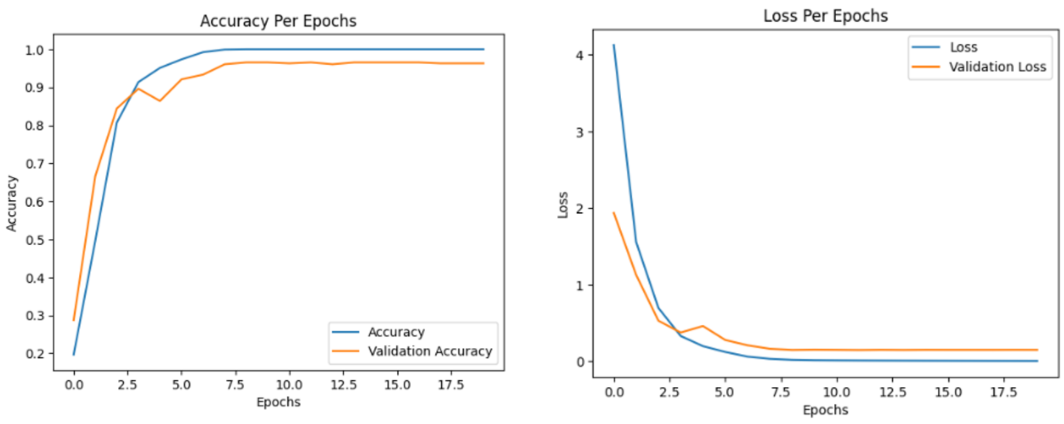


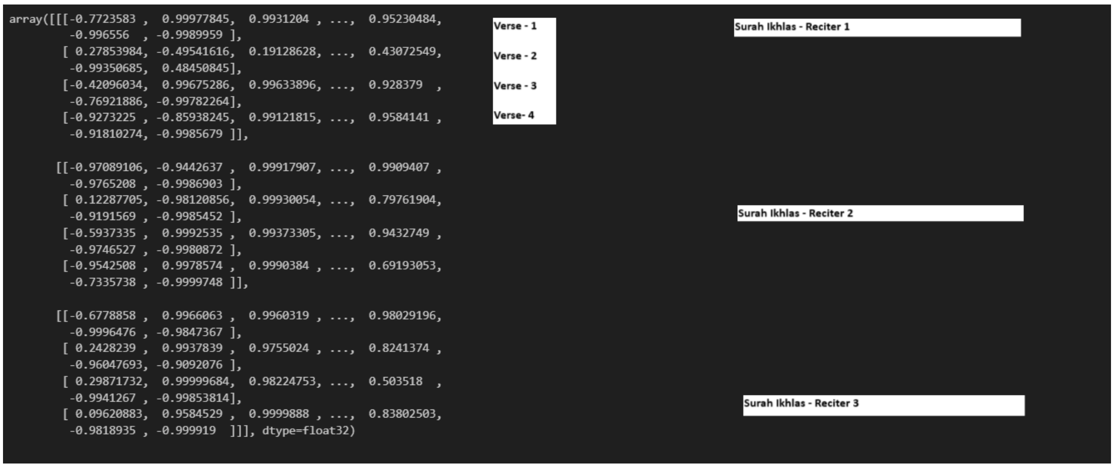
Figure 4.8: Second best Accuracy and Loss per epoch achieved after optimal results

tecture, the layers are organized sequentially, with data flowing from the input layer through various intermediate layers before reaching the output layer. The dense layer with 1024 neurons indicates 1024 individual units, each with weights and biases. These weights and biases are learned during the training process, where the neural network adjusts them to minimize the error between its predictions and the actual target values in the training data. As a result, the neurons in this layer can capture complex patterns and relationships in the data.

In this process, we reserved a portion of the dataset (usually a percentage, such as 80% of the data) for training and the remaining amount (e.g., 20%) for testing. For complete data, we had the recitation of 5 reciters. In this section, we keep the 4 in training and pass them through the second-last layer, i.e., the Dense Layer, and keep the rest of the reciter dataset for testing and evaluation. This division aims to ensure that we can train our model on one subset (training data) and then evaluate it on unseen data (testing data) to gauge its generalization ability.

4.5 Indexing & Retrieval

This indexing process follows the Quranic structure, assigning unique identifiers to each verse, such as "Chapter: Verse." Once this indexing is completed, the verses are prepared for further processing. The data is now organized and indexed through a Dense Layer within a neural network. The purpose of this layer is to transform the indexed verse data, potentially encoding it in a numerical format suitable for computational analysis. This process creates an array or database where each verse is associated with its corresponding Quranic index, allowing for structured and organized access to Quranic text data for various computational tasks or analyses. The representation of the Indexing of the Surah database by different reciters in the sequence is shown in Figure 4.9



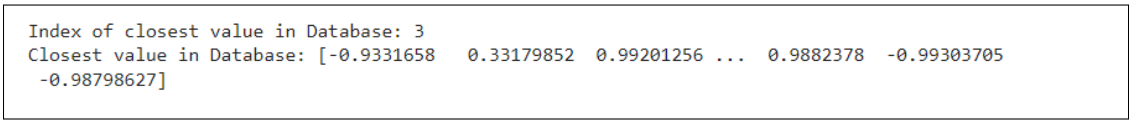
```
array([[[-0.7723583 ,  0.99977845,  0.9931204 , ...,  0.95230484,
         -0.996556  , -0.9989959 ],
        [ 0.27853984, -0.49541616,  0.19128628, ...,  0.43072549,
         -0.99350685,  0.48450845],
        [-0.42096034,  0.99675286,  0.99633896, ...,  0.928379  ,
         -0.76921886, -0.99782264],
        [-0.9273225 , -0.85938245,  0.99121815, ...,  0.9584141 ,
         -0.91810274, -0.9985679 ]],

       [[-0.97089106, -0.9442637 ,  0.99917907, ...,  0.9909407 ,
         -0.9765208 , -0.9986903 ],
        [ 0.12287705, -0.98120856,  0.99930054, ...,  0.79761904,
         -0.9191569 , -0.9985452 ],
        [-0.5937335 ,  0.9992535 ,  0.99373305, ...,  0.9432749 ,
         -0.9746527 , -0.9980872 ],
        [-0.9542508 ,  0.9978574 ,  0.9990384 , ...,  0.69193053,
         -0.7335738 , -0.9999748 ]],

       [[-0.6778858 ,  0.9966063 ,  0.9960319 , ...,  0.98029196,
         -0.9996476 , -0.9847367 ],
        [ 0.2428239 ,  0.9937839 ,  0.9755024 , ...,  0.8241374 ,
         -0.96047693, -0.9092076 ],
        [ 0.29871732,  0.99999684,  0.98224753, ...,  0.503518  ,
         -0.9941267 , -0.99853814],
        [ 0.09620883,  0.9584529 ,  0.9999888 , ...,  0.83802503,
         -0.9818935 , -0.999919  ]]], dtype=float32)
```

Figure 4.9: Visualisation of audio database of Surah Ikhlas by different Reciters as mentioned in Quran

We import a separate set of testing data, which could consist of Quranic audio recordings from different sources or reciters. Just like with the training data, we implement preprocessing steps on the testing data to ensure it's in a suitable format for analysis. Once the testing audio data has been preprocessed, we pass it through the same Dense Layer with 1024 neurons that we used during the training phase. This Dense Layer helps extract relevant features and representations from the audio data.

To determine which verse in the Quran corresponds to a given piece of testing audio, we employ a similarity metric, such as the Euclidean distance. For each piece of testing audio, we calculate the Euclidean distance between its output from the Dense Layer and the representations of verses in our array or database of testing data. The verse representation in our database would have gone through the same preprocessing and neural network encoding as the testing data. By comparing the Euclidean distances, we can identify the verse representation in our database closest to the testing audio regarding the extracted features.

The verse with the closest distance is then considered a potential match, and its index is identified. This process allows us to determine which verse from the Quran corresponds to the provided audio segment, as shown in Figure 4.10



```
Index of closest value in Database: 3
Closest value in Database: [-0.9331658   0.33179852  0.99201256 ...  0.9882378  -0.99303705
 -0.98798627]
```

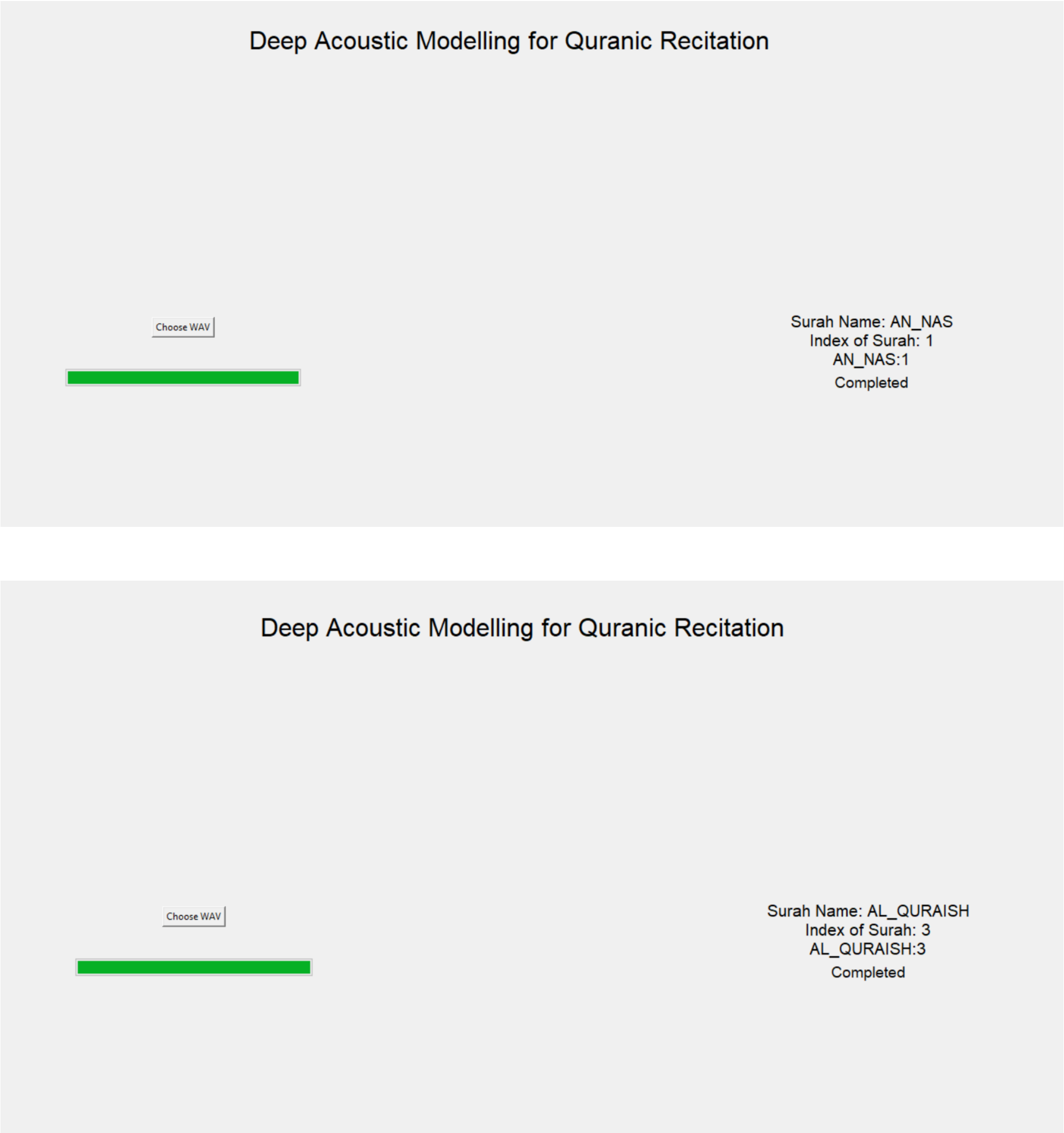
Figure 4.10: Retrival of Testing Audio to get Index of the recited verse in audio

4.6 Graphical User Interface (GUI)

The graphical user interface (GUI) has been designed to enhance user convenience and usability and offer a seamless experience when working with Quranic audio analysis. Users are provided with the capability to import a WAV audio file effortlessly. Once the audio file is imported, the GUI allows users to listen to it in real time, providing an audible preview of the content for verification and validation.

Moreover, the GUI ensures that the output is presented in a standardized and easily interpretable format. When analyzing the imported audio, it processes the data using the previously described techniques, including the Dense Layer with 1024 neurons and the Euclidean distance calculation. After this analysis, the GUI displays the results clearly and intuitively, typically as "Chapter: Verse Index."

The final GUI look is shown in Figure 4.11



Deep Acoustic Modelling for Quranic Recitation

Surah Name: AN_NAS
Index of Surah: 1
AN_NAS:1
Completed

Deep Acoustic Modelling for Quranic Recitation

Surah Name: AL_QURAISH
Index of Surah: 3
AL_QURAISH:3
Completed

Figure 4.11: Graphical User interface (GUI) for the system

Chapter 5

Research Challenges & Applications

5.1 Research Challenges

Building a robust, deep acoustic model specifically for Quranic recitation involves various complex issues requiring careful attention. The challenges in building the model are as follows and as shown in Table 5.1

5.1.1 Diverse Arabic Dialects

Accurate transcription of pronunciation and intonation is complex due to the large variety of Arabic dialects and geographical differences. The intricate tapestry of many Arabic dialects and geographical accents must be considered while building a robust, deep acoustic model for Quranic recitation. This problem requires a thorough strategy incorporating the subtle phonetic variations in different recitation traditions. It takes careful language study and contextual adaptation to accommodate this dialectal diversity while staying true to the traditional recitation methods.

5.1.2 Noise and Audio Purification

Background noise can reduce the output of the model's accuracy in audio recordings. The difficulty of maintaining the original quality of recordings of Quranic recitation arises from the widespread presence of noise inside audio data. Advanced noise reduction techniques must be used to capture profound vocal expressions while successfully overcoming audible interference. It is a complex but crucial endeavor to balance eliminating noise and preserving the evocative details of the recitation.

Almisreb et al. [5] proposed a technique to remove the noise in recitation during recording. The research evaluates the effectiveness of Multiscale Principal Component Analysis (MPCA) with Zero Crossing Rate (ZCR) to remove the noise. Then Mel-Frequency Cepstral Coefficient (MFCC) was used for feature extraction, and Dynamic Time Warping (DTW) was used for recognition. The system proved very effective in removing the noises from the recording.

5.1.3 Varied Recitation Styles

Different reciters have distinctive rhythmic and melodic renditions, necessitating a flexible model. Building a comprehensive deep acoustic model is difficult because of the complex interplay of many recitation styles, each infused with distinctive rhythmic phrases and melodic accents. Creating a model architecture that can skillfully encompass the various recitation patterns while maintaining a flexible and adaptable framework capable of tolerating the different artistic expressions of the reciters is required to navigate this complex terrain.

5.1.4 Adherence to Tajweed and Expressions

The model must accurately distinguish between short and long vowel sounds to represent elongations (Madd) and vowel lengths. It is difficult to capture these little differences accurately while keeping the recitation's rhythmic flow.

5.1.5 Data Augmentation and Training

Creating an extensive and varied training dataset is crucial to building a precise and trustworthy deep acoustic model for Quranic recitation. Data augmentation methods are essential in building a substantial corpus with diverse recitation traditions. Realizing the model's effectiveness will require establishing the right balance between quantity and quality while assuring proper annotation.

Challenges	**References**	**Description**	**Solution**
Diverse Arabic Dialect	[7]	The Arabic language includes a lot of recitation dialects that are difficult for the model to learn.	Utilizing Naive Bayes classifiers and the character n-gram Markov language model achieved 98% accuracy on 18 different Arabic dialects.
Noise and Audio Purification	[5]	Background noise affects the model accuracy and robustness.	Multiscale Principal Component Analysis (MPCA) effectiveness with Zero Crossing Rate (ZCR) to remove the noise.
Varied Recitation Styles	[36]	Different reciters have distinctive rhythmic and melodic renditions, necessitating a flexible model.	SVM-based recognition model for "Qira'ah" achieved success of 96%.
Adherence to Tajweed and Expressions	[19]	The model must be able to distinguish between short and long vowel sounds to represent elongations (Madd) and vowel lengths accurately.	Analyzed signal using Mel Frequency Cepstral Coefficient (MFCC) and compared them using Dynamic Time Warping (DTW) technique to find the similarity differences.
Data Augmentation and Training	[33]	Data augmentation is necessary for including diverse recitation conditions.	Utilize warping the features, masking blocks of frequency channels, and masking blocks of time steps to apply augmentation to the filter bank coefficient directly.

Table 5.1: Research Challenges with their Possible Solutions

5.2 Applications

The real-world applications of our project include:

5.2.1 Verse Indexing and Locality Search

Our project provides a valuable application by enabling users to locate specific verses within the Quran quickly with a simple microphone in real-time. Our system can swiftly identify and give the precise location of the requested verse. This functionality simplifies Quranic research, making it more accessible for scholars, students, and anyone interested in studying the Quran. It streamlines the process of finding and referencing specific verses, ultimately saving time and effort for users.

5.2.2 Namaz e Taraweeh

During Ramadan, Muslims perform additional nightly prayers known as Taraweeh, which involve reciting long portions of the Quran. Our project can seamlessly integrate into prayer apps or systems, aiding in accurate recitation guessing during Taraweeh prayers. This application ensures that Namaz e Taraweeh is conducted with precision and correctness, enhancing the spiritual experience for worshippers.

5.2.3 Quranic Audio Annotations

Our project can facilitate the creation of annotated Quranic audio files. Users can add their comments, explanations, or reflections to specific verses or sections of the Quran. This feature is valuable for educators, scholars, or individuals who want to create personalized audio resources for Quranic study or teaching.

5.2.4 Quranic Accessibility for Visually Impaired Individuals

The project makes the Quran accessible to visually impaired individuals. Users can request recitations or explanations of specific verses, chapters, or topics, enhancing their ability to engage with the Quran independently.

5.2.5 Interactive Quranic Apps for Children

Our project can be incorporated into interactive Quranic apps designed for children. These apps can include audio-guided learning, pronunciation practice, and interactive quizzes, making Quranic education engaging and accessible to young learners.

5.2.6 Content Verification for Quranic Apps

Quranic apps and platforms can use our project to verify the authenticity of Quranic content users upload. This ensures that the recitations and verses shared on these platforms adhere to accurate Quranic recitation standards, promoting reliable and trustworthy Quranic content.

5.2.7 Advanced Quranic Research

Researchers and academics can utilize our project's indexing and navigation capabilities to conduct advanced Quranic research. It can assist in exploring themes, linguistic analysis, and comparative studies across different Quranic chapters and verses.

5.2.8 Integration with Quranic Libraries and Websites

Our project can be integrated into Quranic libraries and websites to enhance search and navigation functionalities. This integration can make Quranic resources more user-friendly and comprehensive for online users seeking knowledge about the Quran.

Chapter 6

Conclusion & Future Directions

In conclusion, our project on deep acoustic modeling for Quranic recitation has significantly enhanced the understanding and analysis of Quranic recitations. We began by conducting an extensive literature review, providing valuable insights into the research landscape. Identifying a research gap, we embarked on the journey to build our deep acoustic model.

Our initial step involved the collection of a diverse dataset encompassing various reciters and dialects, setting the foundation for our research. Subsequently, we employed data preprocessing techniques, including Voice Activity Detection (VAD), to extract single verses from Quranic chapters. Feature extraction was then performed using the Mel-frequency cepstral coefficients (MFCC), preparing the data for training. The heart of our project lay in the training of a Convolutional Neural Network (CNN) based deep learning model, which achieved an impressive accuracy rate of 96.3%. To delve even deeper into the audio features, we utilized the second last layer with 1024 dense neurons to extract deep features from the training data.

We applied the same preprocessing techniques in the testing phase to ensure consistency and accuracy. The matching algorithm we implemented efficiently compared the testing data to the preprocessed data in our database, determining the closest match. This process allowed us to associate the testing data with the appropriate index, providing meaningful insights into Quranic recitation.

To make our tool accessible and user-friendly, we developed a Graphical User Interface (GUI). This GUI enables users to import a WAV file and receive the output in a standardized format, indicating the chapter name and verse number.

In summary, our project has successfully addressed the challenge of deep acoustic modeling for Quranic recitation, culminating in a powerful tool that achieves high accuracy and offers a practical interface for users. This work represents a significant contribution to the Quranic studies and audio analysis field, opening new avenues for research and exploration in this domain.

6.1 Future Work

The successful implementation of our project on deep acoustic modeling for Quranic recitation has opened up several exciting avenues for future work and expansion. Here are some potential directions for further research and development:

6.1.1 Extension to the Entire Quran

While we initially focused on the last 10 Surahs of the Quran, one natural progression would be to extend our model to cover the entire Quran. This expansion would significantly increase the breadth and depth of our dataset, allowing for a more comprehensive analysis of Quranic recitation.

6.1.2 Quranic Language Modeling

The Quranic text can be valuable for training large Arabic language models. Future work could involve creating a specialized Arabic language model trained on the Quranic text. This model could find applications in various fields, including natural language processing, sentiment analysis, and automated translation.

6.1.3 Tafseer and Translation Tools

When combined with a Quranic language model, our deep acoustic model could be used to develop advanced tools for Tafseer (Quranic exegesis) and Quranic translation. These tools could aid scholars and researchers in understanding the Quran's meanings and nuances and facilitate Quranic translation into multiple languages with improved accuracy.

6.1.4 Extension to Hadiths

Building on our expertise in audio analysis, we could extend our project to include the analysis and modeling of Hadiths (sayings and actions of the Prophet Muhammad, peace be upon him). This would involve collecting a diverse dataset of Hadith recitations and implementing similar deep acoustic modeling techniques for Hadith analysis.

6.1.5 User Interface Enhancements

The GUI developed for our project could be further enhanced to provide additional features and functionalities. For example, it could incorporate real-time feedback on recitation quality, pronunciation, or Tajweed rules, making it a valuable tool for learners and enthusiasts.

6.1.6 Multilingual Support

Expanding our project to handle recitations in different languages or dialects would be a valuable addition. This would require dataset collection and preprocessing efforts specific to each language or dialect of interest.

6.1.7 Integration with Educational Platforms

Our technology could be integrated into online educational platforms and mobile apps dedicated to Quranic studies, making it more accessible to a broader audience of learners and scholars.

Bibliography

[1] Martin Vetterli and Cormac Herley. “Wavelets and filter banks: Theory and design”. In: *IEEE transactions on signal processing* 40.ARTICLE (1992), pp. 2207–2232.

[2] Pedro J Moreno et al. “A recursive algorithm for the forced alignment of very long audio segments.” In: *ICSLP*. Vol. 98. 1998, pp. 2711–2714.

[3] Marko Helén and Tuomas Virtanen. “Query by example of audio signals using Euclidean distance between Gaussian mixture models”. In: *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP’07*. Vol. 1. IEEE. 2007, pp. I–225.

[4] I Ahsiah, NM Noor, and MYI Idris. “Tajweed checking system to support recitation”. In: *2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. Ieee. 2013, pp. 189–193.

[5] Ali Abd Almisreb, Ahmad Farid Abidin, and Nooritawati Md Tahir. “Noise effects on the recognition rate of Arabic phonemes based on Malay speakers”. In: *2014 IEEE Symposium on Industrial Electronics & Applications (ISIEA)*. IEEE. 2014, pp. 1–6.

[6] Mohamed S Abdo, Ahmed H Kandil, and Sahar Ali Fawzy. “MFC peak based segmentation for continuous Arabic audio signal”. In: *2nd Middle East Conference on Biomedical Engineering*. IEEE. 2014, pp. 224–227.

[7] Fatiha Sadat, Farnazeh Kazemi, and Atefeh Farzindar. “Automatic identification of arabic dialects in social media”. In: *Proceedings of the first international workshop on Social media retrieval and analysis*. 2014, pp. 35–40.

[8] Shahid Munir Shah and Syed Nadeem Ahsan. “Arabic speaker identification system using combination of DWT and LPC features”. In: *2014 International Conference on Open Source Systems & Technologies*. IEEE. 2014, pp. 176–181.

[9] DS Kumar. “Feature normalisation for robust speech recognition”. In: *arXiv preprint arXiv:1507.04019* (2015).

[10] Mouaz Bezoui, Abdelmajid Elmoutaouakkil, and Abderrahim Beni-hssane. “Feature extraction of some Quranic recitation using mel-frequency cepstral coeficients (MFCC)”. In: *2016 5th international conference on multimedia computing and systems (ICMCS)*. IEEE. 2016, pp. 127–131.

[11] Mohamad Irfan, Imam Zainal Mutaqin, and Rio Guntur Utomo. “Implementation of Dynamic Time Warping algorithm on an Android based application to write and pronounce Hijaiyah letters”. In: *2016 4th International Conference on Cyber and IT Service Management*. IEEE. 2016, pp. 1–6.

[12] Ali Meftah, Yousef A Alotaibi, and Sid-Ahmed Selouani. “A comparative study of different speech features for arabic phonemes classification”. In: *2016 European Modelling Symposium (EMS)*. IEEE. 2016, pp. 47–52.

[13] Bilal Yousfi and Akram M Zeki. “Holy Qur’an speech recognition system distinguishing the type of recitation”. In: *2016 7th International Conference on Computer Science and Information Technology (CSIT)*. IEEE. 2016, pp. 1–6.

[14] Masyithah Nur Aulia et al. “A comparative study of MFCC-KNN and LPC-KNN for hijaiyyah letters pronounciation classification system”. In: *2017 5th International Conference on Information and Communication Technology (ICoIC7)*. IEEE. 2017, pp. 1–5.

[15] Efy Yosrita and Abdul Haris. “Identify the accuracy of the recitation of Al-Quran reading verses with the science of tajwid with Mel-Frequency Ceptral Coefficients method”. In: *2017 International Symposium on Electronics and Smart Devices (ISESD)*. IEEE. 2017, pp. 179–183.

[16] Bilal Yousfi and Akram M Zeki. “Holy Qur’an speech recognition system Imaalah checking rule for warsh recitation”. In: *2017 IEEE 13th international colloquium on signal processing & its applications (CSPA)*. IEEE. 2017, pp. 258–263.

[17] Bilal Yousfi, Akram M Zeki, and Aminah Haji. "Isolated Iqlab checking rules based on speech recognition system". In: *2017 8th International Conference on Information Technology (ICIT)*. IEEE. 2017, pp. 619–624.

[18] Fadzil Ahmad et al. "Tajweed classification using artificial neural network". In: *2018 International Conference on Smart Communications and Networking (SmartNets)*. IEEE. 2018, pp. 1–4.

[19] Tareq AlTalmas et al. "Analysis of two adjacent articulation Quranic letters based on MFCC and DTW". In: *2018 7th International Conference on Computer and Communication Engineering (ICCCE)*. IEEE. 2018, pp. 187–191.

[20] Shahin Amiriparian et al. "Bag-of-deep-features: Noise-robust deep feature representations for audio analysis". In: *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2018, pp. 1–7.

[21] Ashraf Elnagar et al. "Automatic classification of reciters of quranic audio clips". In: *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*. IEEE. 2018, pp. 1–6.

[22] Moulay Ibrahim El-Khalil Ghembaza, Omar Tayan, and Khalid Saleh Aloufi. "Qurani rafiqui: an interactive context-aware quranic application for smartphones". In: *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE. 2018, pp. 1–6.

[23] Teddy Surya Gunawan, Nur Atikah Muhamat Saleh, and Mira Kartiwi. "Development of quranic reciter identification system using MFCC and GMM classifier". In: *International Journal of Electrical and Computer Engineering (IJECE)* 8.1 (2018), pp. 372–378.

[24] Teddy Surya Gunawan, Nur Atikah Muhamat Saleh, and Mira Kartiwi. "Development of quranic reciter identification system using MFCC and GMM classifier". In: *International Journal of Electrical and Computer Engineering (IJECE)* 8.1 (2018), pp. 372–378.

[25] Lina Marlina et al. "Makhraj recognition of Hijaiyah letter for children based on Mel-Frequency Cepstrum Coefficients (MFCC) and Support Vector Machines (SVM) method". In: *2018 International Conference on Information and Communications Technology (ICOIACT)*. IEEE. 2018, pp. 935–940.

[26] Adnan Qayyum, Siddique Latif, and Junaid Qadir. “Quran reciter identification: A deep learning approach”. In: *2018 7th International Conference on Computer and Communication Engineering (ICCCE)*. IEEE. 2018, pp. 492–497.

[27] Asma Salsabila Muhmad Rusli et al. “A systematic review on semantic-based ontology for Quranic knowledge”. In: *International Journal of Engineering and Technology (UAE)* (2018).

[28] Faza Thirafi and Dessi Puji Lestari. “Hybrid HMM-BLSTM-based acoustic modeling for automatic speech recognition on Quran recitation”. In: *2018 International Conference on Asian Language Processing (IALP)*. IEEE. 2018, pp. 203–208.

[29] Safiah Khairuddin et al. “Features Identification and Classification of Alphabet (ro) in Leaning (Al-Inhiraf) and Repetition (Al-Takrir) Characteristics”. In: *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*. IEEE. 2019, pp. 295–299.

[30] Rehan Ullah Khan, Ali Mustafa Qamar, and Mohammed Hadwan. “Quranic reciter recognition: a machine learning approach”. In: *Advances in Science, Technology and Engineering Systems Journal* 4.6 (2019), pp. 173–176.

[31] Afrizal Nur, S Syarifandi, Saidul Amin, et al. “Implementation of text mining classification as a model in the conclusion of Tafsir Bil Ma’tsur and Bil Ra’yi contents”. In: *Int. J. Eng. Adv. Technol* 9.1 (2019), pp. 2789–2795.

[32] Daniel S Park et al. “Specaugment: A simple data augmentation method for automatic speech recognition”. In: *arXiv preprint arXiv:1904.08779* (2019).

[33] Daniel S Park et al. “Specaugment: A simple data augmentation method for automatic speech recognition”. In: *arXiv preprint arXiv:1904.08779* (2019).

[34] Qiuqiang Kong et al. “Panns: Large-scale pretrained audio neural networks for audio pattern recognition”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), pp. 2880–2894.

[35] Qin Li et al. “MSP-MFCC: Energy-efficient MFCC feature extraction method with mixed-signal processing architecture for wearable speech recognition applications”. In: *IEEE Access* 8 (2020), pp. 48720–48730.

[36] KM Nahar et al. "An efficient holy Quran recitation recognizer based on SVM learning model". In: *Jordanian Journal of Computers and Information Technology (JJCIT)* 6.04 (2020), pp. 394–414.

[37] Khoa Nguyen, Konstantinos Drossos, and Tuomas Virtanen. "Temporal sub-sampling of audio feature sequences for automated audio captioning". In: *arXiv preprint arXiv:2007.02676* (2020).

[38] Ahlam Wahdan et al. "A systematic review of text classification research based on deep learning models in Arabic language". In: *Int. J. Electr. Comput. Eng* 10.6 (2020), pp. 6629–6643.

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

[40] Hervé Bredin and Antoine Laurent. "End-to-end speaker segmentation for overlap-aware resegmentation". In: *arXiv preprint arXiv:2104.04045* (2021).

[41] Javeria Farooq and Muhammad Imran. "Mispronunciation Detection in Articulation Points of Arabic Letters using Machine Learning". In: *2021 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*. IEEE. 2021, pp. 1–6.

[42] Rian Adam Rajagede and Rochana Prih Hastuti. "Al-Quran recitation verification for memorization test using Siamese LSTM network". In: *Communications in Science and Technology* 6.1 (2021), pp. 35–40.

[43] Sakib Shahriar and Usman Tariq. "Classifying maqams of Qur'anic recitations using deep learning". In: *IEEE Access* 9 (2021), pp. 117271–117281.

[44] Ammar Mohammed Ali Alqadasi et al. "Rule-Based Embedded HMMs Phoneme Classification to Improve Qur'anic Recitation Recognition". In: *Electronics* 12.1 (2022), p. 176.

[45] Nurul Wahidah Arshad et al. "Signal-based feature extraction for Makhraj emission point classification". In: *Engineering Technology International Conference (ETIC 2022)*. Vol. 2022. IET. 2022, pp. 19–25.

[46] Abdullah M Basahel et al. “A Smart Flexible Tool to Improve Reading Skill based on M-Learning”. In: *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE. 2022, pp. 411–414.

[47] Devin J Stewart. “Approaches to the Investigation of Speech Genres in the Qur’an”. In: *Journal of Qur’anic Studies* 24.1 (2022), pp. 1–45.

[48] Muhammad Huzaifa Bashir et al. “Arabic natural language processing for Qur’anic research: A systematic review”. In: *Artificial Intelligence Review* 56.7 (2023), pp. 6801–6854.

[49] Mohammed Hadwan, Hamzah A Alsayadi, and Salah AL-Hagree. “An End-to-End Transformer-Based Automatic Speech Recognition for Qur’an Reciters.” In: *Computers, Materials & Continua* 74.2 (2023).

[50] Dahlia Omran, Sahar Fawzi, and Ahmed Kandil. “Automatic Detection of Some Tajweed Rules”. In: *2023 20th Learning and Technology Conference (L&T)*. IEEE. 2023, pp. 157–160.