AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING
Computer and Systems Engineering

Automatic Pronunciation Error Detection and Correction of the Holy Quran's Learners Using Deep Learning

A Thesis submitted in partial fulfillment of the requirements of
Master of Science in Electrical Engineering
(Computer and Systems Engineering)

by
Abdullah Aml Abdelfattah
Bachelor of Science in Electrical Engineering
(Electronics and Communications)
Faculty of Engineering, Alexandria University, 2019

Supervised By
Prof.Mahmoud I. Khalil
Prof.Hazem M. Abbas

Cairo, 2025

AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING
Computer and Systems Engineering

Automatic Pronunciation Error Detection and Correction of the Holy Quran's Learners Using Deep Learning

by

Abdullah Aml Abdelfattah
Bachelor of Science in Electrical Engineering
(Electronics and Communications)
Faculty of Engineering, Alexandria University, 2019

Examiners' Committee

Name and affiliation	**Signature**
Prof.Mahmoud I. Khalil Computer and Systems Department Faculty of Engineering, Ain Shams University.	. . . . .
Prof.Hazem M. Abbas Computer and Systems Department Faculty of Engineering, Ain Shams University.	. . . . .
Dr. Choose Department Faculty of Engineering, University.	. . . . .

Date: DD Month, YYYY

Statement

This thesis is submitted as a partial fulfillment of Master of Science in Electrical Engineering (Computer and Systems Engineering), Faculty of Engineering, Ain shams University. The author carried out the work included in this thesis, and no part of it has been submitted for a degree or a qualification at any other scientific entity.

Abdullah Aml Abdelfattah

Signature

.....

Date: DD Month, 2025

Researcher Data

Name: Abdullah Aml Abdelfattah
Date of Birth: 28 August, 1996
Place of Birth: Alexandria, Egypt
Last academic degree: Bachelor of Science
Field of specialization: Natural Language Processing
University issued the degree: Alexandria University
Date of issued degree: 2019
Current job: NLP Researcher at Wakeb Data

Abstract

Assessing spoken language is challenging, and quantifying pronunciation metrics for machine learning models is even harder. However, for the Holy Quran, this task is simplified by the rigorous recitation rules (tajweed) established by Muslim scholars, enabling highly effective assessment. Despite this advantage, the scarcity of high-quality annotated data remains a significant barrier.

In this work, we bridge these gaps by introducing: (1) A 98% automated pipeline to produce high-quality Quranic datasets – encompassing: Collection of recitations from expert reciters, Segmentation at pause points (waqf) using our fine-tuned wav2vec2-BERT model, Transcription of segments, Transcript verification via our novel Tasmeea algorithm; (2) 850+ hours of audio (300K annotated utterances); (3) A novel ASR-based approach for pronunciation error detection, utilizing our custom Quran Phonetic Script (QPS) to encode Tajweed rules (unlike the IPA standard for Modern Standard Arabic). QPS uses a two-level script: (Phoneme level): Encodes Arabic letters with short/long vowels. (Sifa level): Encodes articulation characteristics of every phoneme. We further include comprehensive modeling with our novel multi-level CTC Model which achieved 0.16% average Phoneme Error Rate (PER) on the testset. We release all code, data, and models as open-source: https://obadx.github.io/prepare-quran-dataset/

Summary

This thesis presents a comprehensive framework for the automated assessment of Quranic recitation by developing a novel phonetic script and a corresponding deep learning model. To guide the reader, this summary outlines the structure and contributions of each chapter.

In Chapter 1 (Introduction), we introduce the core problem: the need for a precise, computationally tractable system to evaluate Tajweed (Quranic recitation rules). We establish the purpose of this work: to develop a fine-grained Quranic phonetic script capable of capturing all Tajweed rules and articulation attributes, thereby reformulating pronunciation assessment as a sequence-to-sequence problem.

Chapter 2 (Literature Review) provides a review of the relevant literature. We explore existing works in speech recognition, previous attempts at Arabic phonetic notation, and the specific domain of Tajweed rule formalization. This chapter identifies the gaps in current methodologies, particularly the lack of a script that seamlessly integrates phonological and Tajweed-related features, which our work aims to fill.

The primary theoretical contribution of this thesis is presented in Chapter 3 (A Novel Multi-Level Script), where we introduce our novel multi-level Quranic phonetic script. This script is designed to hierarchically represent pronunciation, encompassing everything from basic phonemes to complex Tajweed rules and precise articulation points (*makharij*). This script serves as the foundational representation layer for all subsequent data annotation and modeling.

Building upon this script, Chapter 4 (Data Pipeline and Annotation) details our data creation pipeline. We demonstrate a 98% automated process for generating a large-scale, high-quality annotated dataset. The result is a substantial corpus totaling 890 hours of audio and nearly 300,000 samples, each meticulously aligned with our multi-level script. This dataset is a significant resource for the field.

In Chapter 5 (Multi-Level CTC Model), we address the modeling challenge. We introduce a novel multi-level Connectionist Temporal Classification (CTC) architecture specifically designed to decode the hierarchical nature of our phonetic script. This model is capable of predicting sequences across multiple levels of abstraction simultaneously, directly aligning with our reformulated problem statement.

The efficacy of our entire methodology is validated in Chapter 6 (Results and Discussion). We present extensive experimental results that prove our approach successfully reformulates and addresses the Quranic pronunciation assessment task. The performance of our model on the dataset from Chapter 4 demonstrates high accuracy in capturing the intricacies of Tajweed.

Finally, Chapter 7 (Conclusion and Future Work) concludes the thesis. We summarize our key contributions—the novel script, the large-scale dataset, and the multi-level model—and discuss the limitations of our current work. The chapter concludes by exploring promising directions for future research, building upon the foundation established here.

Keywords: Holy Quran Learning, Mispronunciation Error Detection, Arabic Natural Language Processing, Self-Supervised Learning

Acknowledgment

I express my greatest gratitude to my mother, Azza, for her unwavering support from the beginning—financially, emotionally, and by providing a quiet space in which to work. I can never fully repay her contributions.

We also express our profound gratitude to several individuals and organizations: Sheikh Ahmed Abdelsalam, Sheikh Mustafa Fathy, and Sheikh Mohamed Rabeea for their invaluable guidance in understanding and representing Tajweed rules and common learner mistakes. We extend our thanks to BA-HPC[1] for providing access to high-performance computing resources, which greatly facilitated our data processing. Special appreciation goes to Engineer Khaled Bahaa for his assistance with arranging payment methods for GPU resources.

Abdullah Aml Abdlefattah
Computer and Systems Engineering
Faculty of Engineering
Ain Shams University
Cairo, Egypt
31 August, 2025

[1] https://hpc.bibalex.org/

Contents

List of Figures

List of Tables

List of Algorithms

a distance m
P power W (Js^{-1})

ω angular frequency $rads^{-1}$

Chapter 1

Introduction

The Holy Quran is the word of Allah, the book He chose for humanity until the Day of Judgment. Although this book is entirely in Arabic, people from other nations can learn to recite it even if they do not know the Arabic language. Regarding this, Allah says:

﴿ وَلَقَدْ يَسَّرْنَا ٱلْقُرْءَانَ لِلذِّكْرِ فَهَلْ مِن مُّدَّكِرٍ ﴾

"And We have certainly made the Qur'an easy for remembrance, so is there any who will remember?" (Surah Al-Qamar, 54:17)

Several efforts have been made to utilize Artificial Intelligence (AI) to help Holy Quran learners recite it properly [1], [2], [3]. However, the lack of data and poor representation of Quranic recitation and Tajweed rules have made the task difficult to accomplish.

This project is launched not only as a master's thesis but as a comprehensive initiative to serve three groups of people:

- **Muslims**: Making the Holy Quran accessible in the era of AI.
- **Developers**: By developing a Software Development Kit (SDK) deployable on all devices, including desktops, cloud, and edge devices.
- **Researchers**: By contributing this research as fully open source, including data, code, and models.

This thesis represents the first step: building a foundational model, which, by the grace of Allah, we have achieved.

Assessing pronunciation is not a simple task [4], as it involves not only correct phoneme articulation but also factors such as intonation, prosody, and stress. Furthermore, fluency and completeness are also essential [4]. However, the Holy Quran possesses unique characteristics: it is among the easiest spoken texts to learn, despite containing phonemes absent in other languages.

The pronunciation of the Holy Quran is governed by rigorously defined rules formalized by classical Muslim scholars since the 6th century. Despite their precision and beauty, these rules have not been comprehensively digitized (to our knowledge) for automated Quranic pronunciation assessment.

Although the RDI company pioneered computer-aided Quranic instruction [5], they did not disclose their phoneticization process or release data or models. As a result, new research must start from the basics: defining phoneticization, data, and models. To bridge this gap, we introduce:

- **A Phonetizer**: Encodes *all* Tajweed rules and articulation attributes (*Sifat*) defined by classical scholars, except *Ishmam* .(إشمام)
- **A 98% Automated Pipeline**: Generates highly accurate datasets from expert recitations.
- **A Dataset**: ∼300K annotated utterances (890 hours).
- **Integration**: Our multi-level CTC model demonstrates the learnability of the Quranic phonetic script (achieving a 0.16% average phoneme error rate).

The Thesis is organized as follows:

- **Related Work**: Expands on the strengths and weaknesses of prior research.
- **Quran Phonetic Script**: Introduces our two-level script: **phonemes** and **Sifat** (10 attributes → 11 total levels).
- **Data Pipeline**: Stages include: (1) Digitized Quran script as foundation, (2) *Hafs* methodology criteria, (3) Expert recitation collection, (4) Segmentation at pause points ,(وقف) (5) Segment transcription, (6) Validation via *Tasmee* (تسميع) algorithm.
- **Modeling**: Demonstrates the learnability of the phonetic script.
- **Results**: Analysis of outcomes.
- **Limitations & Future Work**: Proposed research directions.
- **Conclusion**: Summary of contributions.

Chapter 2

Related Word

2.1 Quran Pronunciation Datasets

We discuss the most important datasets here. Everyayah[1] is the largest openly available dataset with 26 complete *Mushafs* segmented and annotated by Ayah by experts like Al Hossary and non-experts such as Fares Abbad. Qdat [6] contains 1509 utterances of single specific Ayahs labeled for three rules: Madd, Ghunna, and Ikhfaa. Although the scale is relatively small, it was widely adopted by the community [7], [7], and [8] due to being open-source. The Tarteel v1 dataset [9] consists of 25K utterances with diacritics and no Tajweed rules. The latter is the Tarteel [10] private dataset, a massive 9K-hour collection annotated and diacritics without Tajweed rules. The most recent benchmark is IqraaEval [11], which presents a test set of 2.2 hours from 18 speakers, but uses Modern Standard Arabic (MSA) without Tajweed rules.

2.2 Quran Pronunciation Models

To our knowledge, the first work addressing automated pronunciation assessment for the Holy Quran is RDI [5], which built a complete system for detecting pronunciation errors. The work does not specify which errors were included or excluded but mentions testing Qalqala, Idgham, and Iqlab rules. It also omits details on Quranic word phoneticization. Subsequent work continued with [1] and [2], using Deep Neural Networks (DNNs) to replace HMMs and improve the system. Many studies rely on modeling phoneme duration for duration-dependent rules like Madd and Ghunna, e.g., [12], [13], but use limited datasets and focus on specific verses rather than the entire Quran. Others concentrate on detecting specific rules like Qalqala [7] or Ghunna

[1] everyayah.com

and Madd [8], [14]. However, most efforts except RDI work train on small-scale datasets from specific Quranic chapters.

At this point, Tarteel [10] emerges; though lacking Tajweed rules, they built a robust ASR system for diacritized character detection. They developed a crowd-sourced dataset [9] of 25K utterances (68 hours), later extended via application users to 9K hours of private annotated data. The work most aligned with our vision of detecting all error types (including Tajweed and *Sifat*/articulation attributes) is [15]. Although it relies on HMMs and minimal data, it introduces a multi-level detection system: *Makhraj* (phoneme level) and Tajweed rules level.

2.3 Pretrained Speech Encoders with Self-Supervised Learning (SSL)

Speech pretraining began early [16] but was constrained by the sequential nature of Recurrent Neural Networks (RNNs) [17]. The rise of Transformers [18] facilitated greater GPU parallelization, enabling large-scale pretraining. BERT [19] using Masked Language Modeling (MLM) introduced large unsupervised pretraining which has better results on downstream tasks. This soon extended to speech with wav2vec [20] and wav2vec2.0, which added product quantization [21]. Conformer later replaced vanilla Transformers for speech by integrating convolution [22]. Google's Wav2Vec2-BERT [23] then applied MLM to speech. Finally, Facebook extended Wav2Vec2-BERT pretraining [24] to 4.5M hours (including 110K Arabic hours), ideal for low-resource language fine-tuning.

Chapter 3

The Quran Phonetic Script

3.1 Introudction

We consider the Quran Phonetic Script to be the most valuable and important contribution of our work. By formalizing the assessment of Holy Quran pronunciation as an ASR problem represented through this script, we provide a comprehensive solution to the task.

3.1.1 Motivation for Developing Quran Phonetic Script

Modern Standard Arabic (MSA) orthography cannot adequately represent Tajweed rules for error detection. For example, MSA cannot measure the precise length of Madd rules. Previous research (e.g., [25]) focused on single rules like Qalqalah. Our phonetic script addresses this limitation by capturing all Tajweed pronunciation errors except Ishmam (إشمام), which involves a visual mouth movement without audible output.

3.1.2 Background

We based our script on classical Muslim scholarship rather than the International Phonetic Alphabet (IPA) for these reasons:

1. **Historical Precedence**: Muslim scholars from the 6th to 14th centuries rigorously defined Quranic errors centuries before modern phonetics emerged in the West.
2. **Scientific Foundation**: Scholars like Al-Khalil ibn Ahmad (6th century AH) systematically described articulations and attributes with remarkable accuracy comparable to modern phonetics [26].

3. **Pedagogical Relevance**: Learners' errors align with classical definitions according to expert Quran teachers.

3.1.3 Defining Mistakes in Quran Recitation

Following [27], Quran recitation errors fall into three categories:

1. **Articulation Errors**: Incorrect pronunciation of phonemes
2. **Attribute Errors**: Mistakes in letter characteristics (Sifat al-Huruf)
3. **Tajweed Rule Errors**: Incorrect application of rules like Ghunnah, Madd, etc.

Our script comprehensively addresses all three aspects through two output levels:

- **Phonemes Level**: Represents letters, vowels, and Tajweed rules
- **Sifat Level**: Represents articulation attributes for each phoneme

3.1.4 Phoneme Set (43 Symbols)

Table 3.1: The table shows our defening for the phonemes level for the Quran Phonetic Script by (43) phonemes.

Phoneme Name	**Symbol**	**الحرف بالعربية**
hamza	ء	همزة
baa	ب	باء
taa	ت	تاء
thaa	ث	ثاء
jeem	ج	جيم
haa_mohmala	ح	حاء
khaa	خ	خاء
daal	د	دال
thaal	ذ	ذال
raa	ر	راء
zay	ز	زاي
seen	س	سين
sheen	ش	شين
saad	ص	صاد

Table 3.1 – continued from previous page

Phoneme Name	Symbol	الحرف بالعربية
daad	ض	ضاد
taa_mofakhama	ط	طاء
zaa_mofakhama	ظ	ظاء
ayn	ع	عين
ghyn	غ	غين
faa	ف	فاء
qaf	ق	قاف
kaf	ك	كاف
lam	ل	لام
meem	م	ميم
noon	ن	نون
haa	ه	هاء
waw	و	واو
yaa	ي	ياء
alif	ا	نصف مد ألف
yaa_madd	ىٓ	نصف مد ياء
waw_madd	وٓ	نصف مد واو
fatha	◌َ	فتحة
dama	◌ُ	ضمة
kasra	◌ِ	كسرة
fatha_momala	◌ٜ	فتحة ممالة
alif_momala	ـٰ	ألف ممالة
hamza_mosahala	أ	همزة مسهلة
qlqla	◌ۜ	قلقة
noon_mokhfah	ں	نون مخفاة
meem_mokhfah	◌ۢ	ميم مخفاة
sakt	◌ۜ	سكت
dama_mokhtalasa	◌ُ	ضمة مختلسة (عند الروم في تأمنا)

3.1.5 Sifat (Attributes) (10)

Table 3.2: The table showes sefining 14 sifa (صفة) as 10 levels.

Sifat (English)	**Sifat (Arabic)**	**Values (English)**	**Values (Arabic)**
hams_or_jahr	الهمس أو الجهر	hams, jahr	همس, جهر
shidda_or_rakhawa	الشدة أو الرخاوة	shadeed, between, rikhw	شديد, بين بين, رخو
tafkheem_or_taqeeq	التفخيم أو الترقيق	mofakham, moraqaq, low_mofakham	مفخم, مرقق, أدنى المفخم
itbaq	الإطباق	monfateh, motbaq	منفتح, مطبق
safeer	الصفير	safeer, no_safeer	صفير, لا صفير
qalqla	القلقلة	moqalqal, not_moqalqal	مقلقل, غير مقلقل
tikraar	التكرار	mokarar, not_mokarar	مكرر, غير مكرر
tafashie	التفشي	motafashie, not_motafashie	متفشي, غير متفشي
istitala	الاستطالة	mostateel, not_mostateel	مستطيل, غير مستطيل
ghonna	الغنة	maghnoon, not_maghnoon	مغنون, غير مغنون

3.1.6 Development Methodology

1. **Imlaey to Uthmani Conversion**: where convert Imlaey Script to Uthmani Script as we rely on Uthmani script to construct Quran Phoneme Script.
2. **Uthmani to Phonetic Script Conversion**: where we convert Uthmani Script to Quran Phonetic Script.

3.2 Converting Imlaey Script to Uthmani Script

As mentioned, we selected the Uthmani script as our foundation because:

- It contains specialized Tajweed diacritics (Madd, Tasheel, etc.).
- It preserves pause rules critical for recitation (e.g., stopping on رحمت).

To facilitate this conversion, we created an annotation UI to manually annotate misaligned words between the two scripts. For example:

Imlaey Script	**Uthmani Script**
يَا ابْنَ أُمَّ	يَبْنَؤُمَّ

Table 3.3: Example of script alignment

Subsequently, we developed an algorithm that relies on these annotations to convert Imlaey to Uthmani.

3.2.1 Annotation Platform

We developed an annotation application to map misaligned words between the Uthmani and Imlaey scripts. The platform consists of two components: a frontend using Streamlit[1] and a backend using FastAPI[2]. The core idea is to align words between the Imlaey and Uthmani scripts. We first loop over every ayah in both scripts; if we find a misalignment (where the number of Imlaey words is not equal to the number of Uthmani words), we prompt the user to align the words, as shown in the figure below.

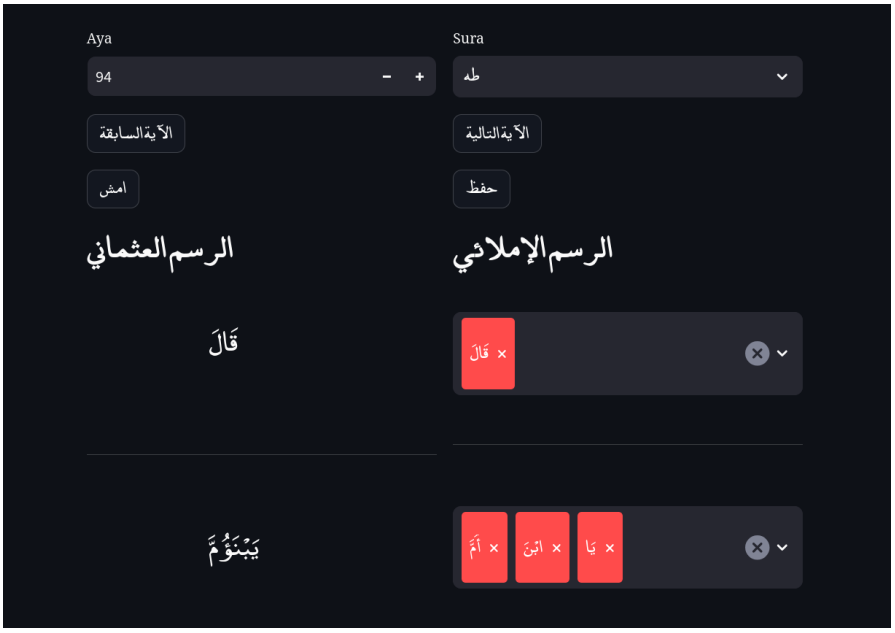


Figure 3.1: A screenshot of the UI where the user aligns words for both scripts.

3.2.2 Observations

After completing the annotation, we attempted to identify patterns of mismatches between the Imlaey and Uthmani scripts. We found the following patterns, as shown below:

Imlaey Script	Uthmani Script	(Surah, Ayah)
يَا ابْنَ أُمَّ	يَبْنَؤُمَّ	(20, 94)
وَأَن لَّوِ	وَأَلَّوِ	(72, 16)

Table 3.4: Examples of script mismatches

Specifically, we found that Imlaey words starting with certain patterns, along with the following word, map to a single Uthmani word.

[1] https://streamlit.io/
[2] https://fastapi.tiangolo.com/

Imlaey Word Start	Count in the Holy Quran
يَا	350
وَيَا	11
هَا	4

Table 3.5: Common Imlaey word patterns

3.2.3 Imlaey to Uthmani Algorithm

Based on these observations, we created an algorithm that performs a lookup for these patterns to align both scripts. This resulted in the `quran-transcript`[3] Python package. With a simple `pip` installation command, the conversion is functional:

```
from quran_transcript import search, Aya

imlaey_text = فأخرج به من الثمرات رزقا لكم
results = search(
    imlaey_text,
    start_aya=Aya(2, 13),
    window=20,
    remove_tashkeel=True
)

uthmani_script = results[0].uthmani_script
print(uthmani_script)
# Output: فَأَخۡرَجَ بِهِۦ مِنَ ٱلثَّمَرَٰتِ رِزۡقٗا لَّكُمۡ
```

Listing 1: Usage example with escaped Arabic

3.3 Quran Phonetic Script Construction

The Quran Phonetic Script is a set of letters and attributes (صفات) that describes what the Holy Quran's reciters **actually** said. It was designed to capture all recitation rules, including all Tajweed rules (except Ishmam إشمام and pausing with rawm روم or إشمام) and Sifat. This script is composed of 11 levels:

- `phonemes` level: Designed to capture pronunciation of letters like baa (ب) and diaractization like (fatha, damma and kasra).

[3] https://github.com/obadx/quran-transcript

- `sifat` level: Consisting of 10 levels to capture the attribute of articulation (صفة) for every phoneme group.

We built this script based on `Hafs` (رواية حفص) and incorporated all the different ways of reciting for `Hafs`. For example, the length of Madd Almunfasil can be (2, 3, 4, or 5 beats). Other variations can be found here **??**.

3.3.1 Phonemes Level

The phoneme level has specific features, which are summarized as:

1. **Madd Representation**:
    - Normal Madd appears as consecutive madd symbols (e.g., 4-beat Madd: اااا).
    - Madd al-Leen is represented with multiple waw/yaa symbols.
2. **Ghunnah**:
    - Stressed Ghunnah for noon (e.g., النون المشددة) is represented as three consecutive noon symbols (ننن).
    - Ikhfa is represented as three consecutive noon_mokhfah (س) or meem_mokhfah (ممم).
3. **Idgham Handling**:
    - Idgham for sakin noon with yaa is represented by consonant doubling (e.g., مَن يَعْمَلْ → مَنييَعمَل).
4. **Special Cases**:
    - Sakin: No following vowel symbol.
    - Imala: Represented by fatha_momala and alif_momala.
    - Rawm: Represented by the dama_mokhtalasa marker.

We only care about pronounced phonemes of letters. If a letter is dropped or not pronounced, we will omit it. For example, we drop the Wasl Hamza (همزة الوصل) when it appears in a context like: (بِسْمِ اللَّهِ).

Table 3.6: Examples of Uthmani to Phonetic Script Conversion with Sifat Attributes

Uthmani	Phonetic	H/J	S/R	T/T	Itb	Saf	Qal	Tik	Taf	Ist	Gho
أَ	ءَ	jahr	shd	mrq	mnf	no	nql	nkr	ntf	nst	nmg
تُ	تُ	hams	shd	mrq	mnf	no	nql	nkr	ntf	nst	nmg
حَ	حَ	hams	rkh	mrq	mnf	no	nql	nkr	ntf	nst	nmg
ـٰـ ـٔـ ـٕـ	ااااا	hams	rkh	mrq	mnf	no	nql	nkr	ntf	nst	nmg
جِ	جُ	jahr	shd	mrq	mnf	no	nql	nkr	ntf	nst	nmg
وٗ	وووو	jahr	rkh	mrq	mnf	no	nql	nkr	ntf	nst	nmg
نٍ	ننن	jahr	btw	mrq	mnf	no	nql	nkr	ntf	nst	mg
ى	ىى	jahr	rkh	mrq	mnf	no	nql	nkr	ntf	nst	nmg

Phonetization of word (أَنُحۡجُوۡنِّى)

Attribute Abbreviations:

H/J: Hams/Jahr S/R: Shidda/Rakhawa T/T: Tafkheem/Taqeeq Itb: Itbaq
Saf: Safeer Qal: Qalqla Tik: Tikraar Taf: Tafashie Ist: Istitala Gho: Ghonna

Value Abbreviations:

shd: shadeed rkh: rikhw btw: between mrq: moraqaq
mof: mofakham mnf: monfateh mtb: motbaq no: no_safeer
nql: not_moqalqal nkr: not_mokarar ntf: not_motafashie
nst: not_mostateel nmg: not_maghnoon mg: maghnoon

3.3.1.1 Disconnected Letters

Disconnected letters (الحروف المقطعة) are letters that are pronounced as individual alphabets one by one. For example: (الٓمٓ) is pronounced (أَلِفْ لَآم مِّيٓمْ). There are 14 forms of these disconnected letters, so we must separate them according to their actual pronunciation.

3.3.1.2 Madd (المد)

There are three types of elongation (مد):

- **Madd Alif** (مد ألف): Fatha followed by alif (ا)
- **Madd Waw** (مد بالواو): Damma followed by waw (و)
- **Madd Yaa** (مد ياء): Kasra followed by Yaa (ي)

These Madd types have different lengths relative to the natural Madd (المد الطبيعي). We created special symbols to denote each Madd type:

- **Madd Alif** is denoted by multiple alif symbols (ا)
- **Madd Waw** is denoted by multiple small_waw symbols, designated as `waw_madd` (ۥ)
- **Madd Yaa** is denoted by multiple small_yaa symbols, designated as `yaa_madd` (ۦ)

Normal Madd (المد الطبيعي) Normal Madd is the type of elongation pronounced at its standard length without excessive prolongation. We denote it by doubling the respective `madd` phonemes. The example below 3.7 shows all three types of Madd in a single word.

Table 3.7: The table demonstrates the three types of normal Madd: Madd Alif (اا), Madd Yaa (ىى), and Madd Waw (وو), each represented with two symbols to indicate a two-beat elongation.

Uthmani Script	Phonetic Script
نُوحِيهَا	نُووحِىىهَاا

Madd Small Silah (مد الصلة الصغرى) Along with Normal Madd, Small Silah Madd (مد الصلة الصغرى) follows the same representation rules. For example 3.8:

Table 3.8: The table shows Small Silah Madd along with noon mushaddad denoted as 3 repeated `noon` (ن) with a special qalqala sign: (چ) for letter jeem (ج).

Uthmani Script	Phonetic Script
إِنَّهُۥ عَلَىٰ رَجْعِهِۦ لَقَادِرٌ	ءِنننَهُۥ عَلَاا رَچِعِهِىى لَقَاادِرٌ

Madd Al-'Iwad (مد العوض) In addition, Madd Al-'Iwad (مد العوض) is represented as shown in 3.9:

Table 3.9: The table shows Madd Al-'Iwad (مد العوض) using the same notation as normal Madd (المد الطبيعي) for Madd alif. This type of Madd occurs when a tanween fatha on a final letter is replaced by an alif madd during pause.

Uthmani Script	Phonetic Script
قَرِيبًا	قَرِىىبَاا

3.3.1.3 Madd Al-Munfasil (مد المنفصل)

For Hafs recitation, Madd Al-Munfasil can be elongated for 2, 3, 4, or 5 harakat, where a haraka here is represented as half of a normal Madd when followed by a hamza (ء) not in the same word, as shown in the example 3.10:

Table 3.10: The example shows elongation for Madd Al-Munfasil with 4 alif madd phonemes, along with a repeated yaa representing yaa mushaddada (ياء مشددة) with both a sakin yaa and a yaa with haraka (damma).

Uthmani Script	Phonetic Script
يَٰٓأَيُّهَا	يَاااءَيُّهَاا

Madd As-Silah Al-Kubra (مد الصلة الكبرى) The same rule is applied to Madd As-Silah Al-Kubra (مد الصلة الكبرى). As shown in the example below: 3.11

Table 3.11: The example shows elongation for Madd As-Silah Al-Kubra with 4 madd waw phonemes (ووو).

Uthmani Script	Phonetic Script
مَالَهُۥٓ أَخۡلَدَهُۥ	مَاالَهُوووو ءَخلَدَه

Madd Al-Muttasil (المد المتصل) For Hafs recitation, Madd Al-Muttasil (مد المتصل) can be elongated for 2, 3, 4, 5, or (6 at pause only) harakat, where a haraka is represented as half of a normal Madd when followed by a hamza (ء) in the same word, as shown in 3.12:

Table 3.12: The example shows elongation for Madd Al-Muttasil (مد المتصل) with 4 madd alif phonemes, along with Madd Al-'Iwad (مد العوض) at the pause point.

Uthmani Script	Phonetic Script
ٱلسَّمَآءِ مَآءً	ءَسسَمَاااءِ مَاااءا

Madd Al-Lazim (المد اللازم) Madd Al-Lazim (المد اللازم) is the type of Madd where a Madd letter is followed by a Sakin letter (حرف ساكن) in the same word and is elongated for 6 harakat (6 حركات), where a haraka is represented as half of a normal Madd.

Table 3.13: The table shows an example of Madd Al-Lazim (المد اللازم) with Madd alif elongated for 6 harakat, along with Madd Al-'Arid Li-S-Sukun (مد العرض للسكون) represented with 4 harakat.

Uthmani Script	Phonetic Script
ٱلضَّآلِّينَ	ءَضضَاااااللِّيءءءءنَ

Madd Al-عارض Li-S-سكون (مد العارض للسكون) Madd Al-عارض Li-S-سكون is the madd that occurs when pausing after a normal madd with a sakin letter. This madd is elongated for 2, 4, or 6 harakat, where the haraka is represented as half of the normal madd length, as shown in 3.13:

Madd Al-لين (مد اللين) Madd Al-لين (مد اللين) occurs when pausing after a ياء (ي) or واو (و) that is preceded by a fatha and followed by a sakin letter. This madd is elongated for 2, 4, or 6 harakat, where a haraka is represented as half of the normal madd length 3.14. We do not create special phonemes for this rule as we did with other madd types because لين represents an elongation of existing واو (و) or ياء (ي) phonemes rather than introducing new phonemes.

For a 4-haraka madd, we denote this with (number_of_harakat - 1) symbols. This approach accounts for cases of Madd Al-لين in the middle of recitation (like وَٱلْمَيْسِرِ) as well as at pause positions, maintaining consistency in the phonetic script. Table 3.14 shows an example of Madd Al-لين:

Table 3.14: The example shows two forms of madd: the first is normal madd followed by Madd Al-لين with 4 harakat (each haraka being half of normal madd), denoted with 3 ياء (ي) symbols.

Uthmani Script	Phonetic Script
لِإِيلَٰفِ قُرَيْشٍ	لِءِءٕلَافِ قُرَيِيش

3.3.1.4 Ghunnah (العنة)

We consider tanween here as a haraka (fatha, damma, or kasra) followed by a sakin noon (نون ساكنة), so we do not need to define separate rules for noon (ن) and tanween.

Noon Mushaddadah (النون المشددة) We first attempted to measure the relative timing of a sakin noon alone (النون الساكنة المظهرة) and compare it to an elongated noon (noon with shaddah - نون مشددة). We found that the elongated noon is approximately 3 to 4 times longer than the sakin noon, so we defined the elongated noon as equivalent to 3 sakin noon repetitions. Example in table: 3.15

Table 3.15: The table shows how Ghunnah disassembly of noon with shaddah (نون مشددة) is represented as 3 repetitive noon (ن) symbols.

Uthmani Script	Phonetic Script
إِنَّ	ءِننن
شَيْءٍ نُّكُرٍ	شَيـِٔننننُكُر

Meem Mushaddadah (الميم المشددة) As we have done with Noon Mushaddadah, we applied the same principle to Meem Mushaddadah (elongated meem). We found the same result: Meem Mushaddadah is approximately 3 to 4 times longer than a regular sakin meem (ميم ساكنة مظهرة). We denote Meem Mushaddadah as 3 repeated meem symbols, as shown in the examples: 3.16

Table 3.16: The table shows how Ghunnah disassembly of meem with shaddah (ميم مشددة) is represented as 3 repeated meem (م) symbols.

Uthmani Script	Phonetic Script
أَمَّا	ءَمممَاا
خَيْرٍ مِّن	خَيرِممِن

Ikhfaa for Noon (إخفاء النون الساكنة) Ikhfaa for sakin noon (إخفاء النون الساكنة) occurs when a sakin noon (نون ساكنة) or tanween is followed by any of the Ikhfaa letters: (ص، ذ، ث، ك، ج، ش، ق، س، د، ط، ز، ت، ض، ظ، ف). We denote this by replacing the noon with three 'noon_mokhfaa' symbols (ں), as shown in the example 3.17:

Table 3.17: The table shows the representation of noon mokhfaa (نون مخفاة) as three dotless noon symbols (ں).

Uthmani Script	Phonetic Script
مِن صَلْصَٰلٍ	مِںصَلصَاااال

Idgham for Noon with Yaa and Waw (إدغام النون الساكنة مع الياء والواو) The Idgham rule is defined as pronouncing two consecutive letters as the second letter with shadda (stress) according to Ibn Al-Jazari [28]. Therefore, we simply delete the noon (ن) and replace it with a yaa (ي) or waw (و).

As with Noon Mushaddadah and Meem Mushaddadah, we represent the resulting stressed yaa or waw with two repetitions rather than three. This maintains consistency with our representation of Madd Al-لين and follows the convention that a stressed letter (مشدد) is represented by both a sakin and mutaharrik (متحرك) form. Table 3.18 shows examples of different forms of yaa:

Table 3.18: This table demonstrates different representations of yaa. The first row shows Idgham of yaa with sakin noon (النون الساكنة) represented by replacing the noon with two yaa symbols. The second row shows yaa with shadda at pause represented with two yaa symbols. The third row shows Madd Al-لين with 4 harakat represented by 3 yaa symbols.

Uthmani Script	Phonetic Script
مَن يَعْمَلْ	مَييَيعمَل
ٱلْحَيِّ	ءَلحَيِي
قُرَيْشٍ	قُرَييش

Ikhfaa for Meem (إخفاء الميم الساكنة) Ikhfaa for sakin meem (إخفاء الميم الساكنة) occurs when a sakin meem (ميم ساكنة) is followed by a baa (ب). Additionally, when a sakin noon or tanween is followed by baa, it is defined in Tajweed literature as Iqlab (إقلاب). We represent both cases with three 'meem_mokhfah' symbols (ۄ). Table 3.19 shows how this rule is applied:

Table 3.19: The first row represents the Iqlab rule (الإقلاب), which is denoted by replacing the noon with 3 ‘meem_mokhfah‘ symbols (ۢمۢ) and (ۚ) donates Qalqala. The second row shows the rule of Ikhfaa for sakin meem with baa (إخفاء الميم الساكنة), represented by 3 ‘meem_mokhfah‘ symbols (ۢمۢ).

Uthmani Script	Phonetic Script
مِنۢ بَعۡدِ	مِۢمۢمۢبَعۡدِۚ
تَرۡمِيهِم بِحِجَارَةٖ	تَرۡمِۦهِۢمۢمۢبِحِجَارَه

3.3.1.5 Idgham (الإدغام)

There are two types of merging (Idgham) in Arabic:

- **Full Merging (إدغام كامل)**: When two letters follow each other and are pronounced as only the second letter, but stressed. Example: (قَد تَّبَيَّنَ) is pronounced as (قَتَبَيَّنَ) where the letter daal is completely not pronounced.
- **Partial Merging (إدغام ناقص)**: When two letters follow each other and the articulation point (makhraj) of the first letter is lost but its attributes (sifat) remain. Example: (بَسَطْتَ) is pronounced the same (بَسَّطَت).

3.3.1.6 Sakin Letter (الحرف الساكن)

A sakin letter is represented in the Uthmani script in three forms:

1. A letter followed by sukun (سُكُون): ’’ْ
2. A letter with shaddah (شَدَّة), which represents a stressed letter composed of two identical letters: the first is sakin and the second has a haraka (fatḥah, ḍammah, or kasrah). Example: (بَّ) → (بَبَ)
3. A letter that is not followed by any haraka (short vowel) or any special symbol, which occurs in Idgham and with madd letters.

We denote a sakin letter by the absence of any following vowel diacritic.

3.3.1.7 Pausing (وَقْف)

At a pause (وَقْف), we make the final letter sakin (سَاكِن) by removing any vowel diacritic. See examples in: 3.18 and other relevant tables.

3.3.1.8 Hamzat Al-Wasl (همزة الوصل)

Hamzat Al-Wasl (همزة الوصل) (ٱ) is defined in Tajweed as a hamza added to avoid beginning with a sakin letter [27]. It is elided during continuous recitation and is only pronounced at the beginning.

The vowel following Hamzat Al-Wasl (fatha, damma, or kasra) depends on the word type:

- For nouns beginning with Alif-Lam at-ta'reef (ال التعريف), the hamza is followed by fatha.
- For proper nouns, the hamza is followed by kasra.
- For verbs: the vowel depends on the third root letter:
  - Damma: hamza is followed by non-transient damma
  - Fatha, kasra, or transient damma: hamza is followed by kasra

Transient damma refers to a damma that is not original but results from a temporary grammatical state. For example, the word (ٱمْشُواْ) has a damma on its third letter, but the verb originates from (ٱمْشِ) where the third letter (ش) has kasra.

Table 3.20: This table shows different forms of Hamzat Al-Wasl (ٱ). The first and second rows demonstrate beginning with hamza followed by fatha due to (ال) at-ta'reef. The third row shows beginning with hamza followed by kasra for a proper noun. The 4th, 5th, and 6th rows show verbs beginning with hamza followed by kasra because the third radical has fatha, kasra, or transient damma. The last row shows beginning with hamza followed by damma because the third radical has a non-transient damma.

Uthmani Script	Phonetic Script	Word Type	Hamzat Wasl Vowel
ٱلْكِتَٰبُ	ءَلكِتَاابُ	Noun beginning with (ال)	fatha
ٱللَّهِ	ءَللَااهِ	Proper Noun beginning with (ال)	fatha
ٱسْتِكْبَارًا	ءِستِكبَارَاا	Proper Noun	kasra
ٱرْكَب	ءِركَبْ	Verb (3rd letter has fatha)	kasra
ٱصْبِرْ	ءِصبِرْ	Verb (3rd letter has kasra)	kasra
ٱمْشُواْ	ءِمشُوو	Verb (3rd letter has transient damma)	kasra
ٱرْكُضْ	ءُركُضْ	Verb (3rd letter has non-transient damma)	damma

Important Note: We rely on Dukes's work [29] for determining word types (nouns, verbs, and particles). Without this foundational research, annotating the Holy Quran's words would require at least a year of dedicated effort, highlighting the critical importance of open-source linguistic resources.

Meeting Two Hamzas (Second One is Sakin) (التقاء همزتان والثانية منهما ساكنة) After converting Hamzat Wasl to a pronounced hamza, certain cases occur where two hamzas meet and the second one is sakin (consonant). In such cases, the second hamza is converted to a madd letter matching the vowel (haraka) of the first hamza [27]. Table 3.21 illustrates this process:

Table 3.21: The table shows the conversion process for verbs that begin with two connected hamzas. The first stage converts Hamzat Wasl to a hamza followed by kasra or damma. The second stage converts the second hamza to either waw_madd (و) or yaa_madd (ے), depending on the vowel of the first hamza. We maintain our established representation where normal madd is represented by two symbols: (اا) for madd_alif, (ےے) for madd_yaa, and (وو) for madd_waw.

Uthmani Script	Converting Hamzat Wasl	Final Conversion
ٱٓۡوۡتُمِنَ	ءُءۡتُمِن	ءُووتُمِن
ٱٓئۡتُونِى	ءِءۡتُووۡنِےے	ءِےےتُووۡنِےے

3.3.1.9 Meeting Two Sakin Letters (التقاء الساكنين)

In Arabic language and the Holy Quran, two sakin letters (الحرفان الساكنان) cannot meet consecutively except at pause (وقف), such as pausing on the word (ٱلۡأَرۡضِ) where the final two letters are sakin. To resolve this meeting, three approaches may be employed:

- Eliminate the first letter
- Elongate the first letter
- Diacritize the second letter with a vowel (fatha, damma, or kasra)

Muslim scholars have simplified this task by comprehensively annotating these rules within the Uthmani script, except for two specific cases:

- When the first letter is (alif, waw, or yaa): we eliminate the first letter
- When the first letter is tanween: we convert the tanween to a noon (ن) followed by kasra

Table 3.22 shows how we apply this rule in our phonetization process:

Table 3.22: The table demonstrates how we resolve the meeting of two sakin letters. The first row shows the meeting of alif (ا) from the word (قَالَا) with the lam (ل) of the word (ٱلْحَمْدُ). In the resulting phonetic script, the alif was deleted. Note that normal madd in (قَالَ) is represented by two alif (اا), and qalqala in the letter daal (د) is represented by (چ). The second example shows the meeting of tanween from (نُوحٌ) with the sakin baa (ب) of the word (ٱبْنَهُۥ), resulting in the conversion of tanween to noon with kasra. Note also that normal madd waw is represented with two (وو) and qalqala for baa (ب) with (چ).

Uthmani Script	Phonetic Script
وَقَالَا ٱلْحَمْدُ	وَقَاالَ لْحَمدچ
نُوحٌ ٱبْنَهُۥ	نُووحِنِ بچنَه

3.3.1.10 Shadda (التشديد)

Shadda (ّ) indicates that a letter is doubled or geminated. We represent this by repeating the letter twice, as shown in 3.24.

3.3.1.11 Pausing (الوقف)

Several rules apply at pause (وقف):

- Vowels (harakat) such as fatha, damma, and kasra are elided, meaning the final letter becomes sakin (ساكن).
- Small Silah Madd is elided.
- Taa marboota (ة) is converted to haa (ه).

3.3.1.12 Qalqala (القلقة)

Qalqala (قلقة) is defined in tajweed as: ”a small sound is followed by on one the letter (ق - ط - ب - ج - د) if one of them is sakin (ساكن) either in between words (وصلا) or at pause (وقفا)” [30]. We denote this small sound as (چ) like in table 3.22.

3.3.1.13 Imala (الإمالة)

Imala (إمالة) is defined in Tajweed as ”pronouncing a fatha somewhere between a fatha and a kasra, and an alif somewhere between an alif and a yaa” [27]. We denote a fatha with imala

as ‘fatha_momala‘ (ۥ) and an alif with imala with two ‘alif_momala‘ symbols (ۦ), similar to the representation of Normal Madd. Table 3.23 provides an example:

Table 3.23: The table shows how we represent fatha with imala as (ۥ) and alif with imala as (ۦ). The letter jeem (ج) also exhibits qalqala, denoted by (ۨچ).

Uthmani Script	Phonetic Script
مَجْرٖىٰهَا	مَجۡرِيۡهَاا

3.3.1.14 Tasheel (التسهيل)

Tasheel is defined in Tajweed as ”pronouncing a hamza (ء) with a quality intermediate between a full hamza and the following madd letter, similar to an intermediate vowel (حركة) between fatha, damma, and kasra” [27]. We denote this facilitated hamza with the symbol ‘hamza_mosahala‘ (ۢأ). Table 3.24 provides an example:

Table 3.24: The table shows a hamza with Tasheel denoted by (ۢأ), along with the disassembly of the letter yaa (ي) with shaddah (ّ) into two yaa symbols.

Uthmani Script	Phonetic Script
ءَاْعۡجَمِىٌّ	ءَأَعۡجَمِيٞ

3.3.1.15 Sakt (السكت)

Sakt is defined in tajweed by ”cutting voice without releasing of breathe for short period learned from expert reciters” [30]. Sakat happens in a specified positions see: **??**. we denote sakt by ‘sakt‘ (ۜ).

3.3.1.16 Implementation

We implemented our phonetic representation by applying 26 operations. Each operation consists of one or more regular expressions:

1. **DisassembleHrofMoqatta** (تفكيك حروف مقطعة): Separates Quranic initials (e.g., الم، الر) into individual letters.
2. **SpecialCases** (حالات خاصة): Handles special words like يبسط that have different pronunciation forms defined in `MoshafAttributes`.

3. **BeginWithHamzatWasl** (البدء بهمزة الوصل): Processes words starting with connecting hamza (ٱ) and converts it to hamza (ء) with appropriate harakah for nouns and verbs.

4. **BeginWithSaken** (البدء بساكن): Manages words beginning with a consonant (sakin) like لْيَقْطَعْ, as Arabic doesn't start utterances with consonants.

5. **ConvertAlifMaksora** (تحويل الألف المقصورة): Converts ى in Uthmani script to either yaa (ي) or alif (ا) based on context.

6. **NormalizeHmazat** (توحيد الهمزات): Standardizes hamza forms (أ إ ؤ ئ) to ء.

7. **IthbatYaaYohie** (إثبات ياء يحيى): Handles words like يُحْيِۦ where two yaa letters occur - resolves conflicts when pausing on words with consecutive consonants (التقاء الساكنين) by adding another yaa at end.

8. **RemoveKasheeda** (إزالة الكشيدة): Deletes elongation marks (ـــ) from text.

9. **RemoveHmzatWaslMiddle** (إزالة همزة الوصل الوسطية): Removes connecting hamza (ٱ) in non-initial positions.

10. **RemoveSkoonMostadeer** (حذف الحرف الذي فوقع سكون مستدير): Eliminates letters with circular sukoon diacritics like alif in جَمَعُواْ.

11. **SkoonMostateel** (سكون مستطيل): Removes alif with elongated sukoon mid-word and adds it at the end during pauses (وقف).

12. **MaddAlewad** (مد العوض): Removes alif after tanween fatha mid-word and adds alif while removing tanween at pause positions (وقف).

13. **WawAlsalah** (واو الصلاة): Replaces letter waw (و) with small alif above combined with alif.

14. **EnlargeSmallLetters** (تكبير الحروف الصغيرة): Resizes miniature Arabic letters to standard proportions.

15. **CleanEnd** (تنظيف النهاية): Removes redundant diacritics and spaces at word endings.

16. **NormalizeTaa** (توحيد التاء): Converts ة (taa marbuta) to ت or ه based on context, and converts final ة to haa (ه).

17. **AddAlifIsmAllah** (إضافة ألف اسم الله): Inserts compensatory alif in derivatives of "الله".

18. **PrepareGhonnaIdghamIqlab** (تهيئة الغنة والإدغام والإقلاب): Preprocesses text for nasalization, assimilation, and conversion rules.

19. **IltiqaaAlsaknan** (التقاء الساكنين): Resolves consecutive consonants by inserting vowels.

20. **DeleteShaddaAtBeginning** (حذف الشدة في البداية): Removes shadda (ّ) from word-initial letters.
21. **Ghonna** (غنة): Applies nasalization during pronunciation of sakin noon and tanween.
22. **Tasheel** (تسهيل): Adds a letter representing alif with tasheel easing.
23. **Imala** (إمالة): Converts fatha with imala to `fatha_momala` phoneme and alif with imala to `alif_momala` phoneme.
24. **Madd** (مد): Adds madd symbols for all madd types, inserting `madd_alif` (ا), `madd_waw` (و), and `madd_yaa` (ي).
25. **Qalqla** (قلقة): Adds echoing effect to ق, ط, ب, ج, د letters with sukoon.
26. **RemoveRasHaaAndShadda** (إزالة رأس الحاء علامة السكون): Deletes sukoon diacritic marks.

3.4 Sifat Level

Sifat (صفة), or in English, attributes of articulation, form a foundational component of our phonetic representation. We based our classification on the classical scholarship of Ibn Al-Jazari. While Ibn Al-Jazari enumerated 17 sifat [31], we have excluded 4 of them for the following reasons:

- **Ismat (إصمات)**: This is a phonological, not a purely phonetic, property.
- **Ithlaq (إذلاق)**: Similarly, this is a phonological characteristic rather than a phonetic one.
- **Leen (اللين)**: This attribute is already accounted for within the rules of Madd Al-Leen (مد اللين).
- **Inhiraf (الانحراف)**: This property explains why the letters lam (ل) and raa (ر) are classified between shidda (شدة) and rakhawa (رخاوة), and is thus subsumed by those attributes.

We have included the sifa of Al-Ghunnah (صفة الغنة), resulting in a system that represents 14 sifat organized into 10 distinct levels, as detailed in table 3.2.

- **Hams/Jahr (الهمس/الجهر)**
  - *Hams*: Whispered letters requiring breath flow (ف ح ث ه ش خ ص س ك ت)
  - *Jahr*: Voiced letters with full the rest of letters
- **Shidda/Rakhawa (الشدة/الرخاوة)**

  - *Shidda*: Complete interruption of sound (ء ج د ق ط ب ك ت)
  - *Between Shiddah and Rqkahwa*: in between of both (ل ن ع م ر)
  - *Rakhawa*: Continuous airflow the rest of letter.

- **Tafkheem/Taqeeq (التفخيم/الترقيق)**
  - *Tafkheem*: Heavy/thickened pronunciation (خ ص ض ط ظ غ ق)
  - *Taqeeq*: Light/thin pronunciation rest of letter with exepsion descripted below

- **Itbaq (الإطباق)**
  - *Motbaq* Letters pronounced with tongue-to-palate contact (ص ض ط ظ)
  - *Monfateh* Rest of the letters

- **Safeer (الصفير)**
  - *Safeer* Whistling sound in س ص ز
  - *No Safeer* the rest of the letters

- **Qalqala (القلقلة)**
  - *Moqlqal* Echo effect on ق ط ب ج د when bearing sukoon
  - *Not Moqlqal* the rest of letters

- **Tikraar (التكرار)**
  - *Mokarrar* the Quranic letter (ر) is trilled just (like Spanish perro)
  - *Not Mokarrar* the rest of letters

- **Tafashie (التفشي)**
  - *Motafashie* Sound dispersion characteristic of ش
  - *Not Motafashie* Rest of letters

- **Istitala (الاستطالة)**
  - *Mostateel* sepecial attribute emphatic and pharyngealized for letter (ض)
  - *Not Mostateel* Rest of letters

- **Ghonna (الغنة)**
  - *Maghnoon* Nasalization in ن and م
  - *Not Maghnoon* The other letters

Our methodology for transcribing Sifat involves first chunking phonemes by grouping similar phoneme categories and then extracting the Sifat for each phoneme group, as shown in the example 3.6. Subsequently, we extract the Sifat for every group.

The extraction of Sifat is straightforward, with the exception of Tafkheem and Tarqeeq.

3.4.0.1 Tafkheem and Tarqeeq (التفخيم والترقيق)

Tafkheem (التفخيم) is defined as "thickening that affects a phoneme, causing it to fill the entire mouth" [27].

In Tajweed, there are 6 levels of Tafkheem and Tarqeeq, ordered from strongest (most mofakham) to weakest (moraqaq):

1. Mofakham followed by fatha and then madd alif.
2. Mofakham followed by fatha without madd alif.
3. Mofakham followed by damma.
4. Mofakham in a sakin (ساكن) state.
5. Mofakham followed by kasra.
6. Moraqaq.

We have formulated these into three labels:

1. 'mofakham' to cover cases 1 to 4.
2. 'moraqaq' to cover case 6.
3. 'low_mofakham' to cover case 5 for the letters (غ, خ, ق), which are monfateh (منفتح) and not motbaq (مطبق). These letters are weakened by a kasra, unlike motbaq letters such as (ص, ض, ط, ظ).

Some phonemes exhibit cases where they can be either moraqaq or mofakham:

- 'madd_alif' (ا): Its Tafkheem or Tarqeeq follows that of the preceding phoneme.
- 'noon_mokhfah' (ں): Its Tafkheem or Tarqeeq follows that of the subsequent phoneme.
- 'raa' (ر) is moraqaq in the following cases:

- When followed by a kasra.
- When preceded by a sakin yaa (ياء).
- When it is sakin (ساكن) and preceded by a mostafel (مستفل) phoneme with a kasra, and not followed by a mosta'lie (مستعلي) phoneme.
- When it appears after a hamzat wasl (ٱ).

Note: The Mosta'lie (حروف الاستعلاء) letters are (خ, ص, ض, غ, ط, ق, ظ), and the Mostafel (مستفل) letters comprise all others.

Chapter 4

Preparing Dataset

4.1 Introduction

The data preparation process began with the definition of clear selection criteria. Our objective was to compile recitations from world-class experts to serve as reference models for evaluating Quranic learners. This study focuses exclusively on the Hafs Way (رواية حفص) due to its status as the most prevalent recitation method globally.

Acknowledging that manual annotation is prohibitively time-consuming, we developed a data collection pipeline that is approximately 98% automated. The procedure consists of the following steps:

1. Selection of a digitized Quranic script as the foundational text.
2. Definition of precise criteria for the حفص methodology.
3. Collection of expert recitations.
4. Segmentation of audio at pause points (وقف).
5. Transcription of the segmented audio.
6. Data validation using a Tasmeea (تسميع) algorithm.
7. Extraction of a phonetic script using our custom Quran Phonetic Script.

For the purposes of this work, a moshaf (مصحف) is defined as a complete recitation of the Quran (chapters 1-114) by a single reciter. The statistics of the collected dataset are summarized in Table 4.1.

Table 4.1: Summary of the proposed dataset. The final collection comprises approximately 848 hours of audio, totaling over 286,000 individual recitation segments.

Moshaf ID	Hours	Recitation Count
0.0	28.48	9,133
0.1	40.31	10,764
0.2	49.47	9,971
0.3	37.19	12,604
1.0	28.41	10,939
2.0	51.05	9,942
2.1	30.03	10,394
3.0	25.19	10,444
4.0	29.12	10,994
5.0	28.02	11,482
6.0	39.39	12,435
7.0	28.26	9,907
8.0	30.86	10,330
9.0	27.95	10,642
11.0	24.01	10,363
12.0	33.42	9,880
13.0	33.99	9,377
19.0	30.11	11,278
22.0	28.11	10,332
24.0	28.51	9,868
25.0	16.93	7,922
26.0	30.44	11,565
26.1	32.71	11,850
27.0	28.05	11,213
28.0	31.05	10,535
29.0	27.79	11,061
30.0	29.14	11,312
Total	**847.99**	**286,537**

4.2 Choose a Digitized Version of the Holy Quran

The Quran has multiple digitized versions including Tanzil[1] and King Fahd Complex[2]. We chose Tanzil because:

- It uses standard Unicode characters
- Contains both إملائي and عثماني versions
- Maintains high accuracy

We excluded KFGQPC due to its evolving/unstable nature compared to Tanzil.

4.3 Defining Variant Criteria for Hafs

The Hafs way(رواية حفص) contains several phonetic and prosodic variants. For instance, the application of مد المنفصل (Madd Al-Munfasil) can vary in duration, extending for 2, 4, 5, or 6 vowel beats depending on the specific recitational rule. These variants were rigorously defined through an analysis of classical Qira'at literature [32]. The criteria for each variant are summarized in the table below.

To capture this variability, each Moshaf in our dataset is accompanied by a 'MoshafAttributes' card that documents its specific recitational features. These cards were manually annotated through a dedicated and meticulous effort, ensuring an accurate representation of each reciter's adherence to the defined rules.

4.3.1 Moshaf Attribute Definitions

- **rewaya** (الرواية)
  - Values: - `hafs` (حفص)
  - Default Value:
  - More Info: The type of the quran Rewaya.
- **recitation_speed** (سرعة التلاوة)
  - Values:
    * `mujawad` (مجود)

[1] https://tanzil.net
[2] https://qurancomplex.gov.sa

    * `above_murattal` (فويق المرتل)
    * `murattal` (مرتل)
    * `hadr` (حدر)
  - Default Value: `murattal` (مرتل)
  - More Info: The recitation speed sorted from slowest to the fastest سرعة التلاوة مرتبة من الأبطأ إلي الأسرع

- **takbeer** (التكبير)
  - Values:
    * `no_takbeer` (لا تكبير)
    * `beginning_of_sharh` (التكبير من أول الشرح لأول الناس)
    * `end_of_doha` (التكبير من آخر الضحى لآخر الناس)
    * `general_takbeer` (التكبير أول كل سورة إلا التوبة)
  - Default Value: `no_takbeer` (لا تكبير)
  - More Info: The ways to add takbeer (الله أكبر) after Istiaatha (استعاذة) and between end of the surah and beginning of the surah. `no_takbeer`: "لا تكبير" — No Takbeer (No proclamation of greatness, i.e., there is no Takbeer recitation) `beginning_of_sharh`: "التكبير من أول الشرح لأول الناس" — Takbeer from the beginning of Surah Ash-Sharh to the beginning of Surah An-Nas `end_of_dohaf`: "التكبير من آخر الضحى لآخر الناس" — Takbeer from the end of Surah Ad-Duha to the end of Surah An-Nas `general_takbeer`: "التكبير أول كل سورة إلا التوبة" — Takbeer at the beginning of every Surah except Surah At-Tawbah

- **madd_monfasel_len** (مد المنفصل)
  - Values:
    * `2`
    * `3`
    * `4`
    * `5`
  - Default Value:
  - More Info: The length of Mad Al Monfasel "مد المنفصل" for Hafs Rewaya.

- **madd_mottasel_len** (مقدار المد المتصل)
  - Values:
    * `4`
    * `5`

    * 6
  - Default Value:
  - More Info: The length of Mad Al Motasel "مد المتصل" for Hafs.

- **madd_mottasel_waqf** (مقدار المد المتصل وقفا)
  - Values:
    * 4
    * 5
    * 6
  - Default Value:
  - More Info: The length of Madd Almotasel at pause for Hafs.. Example "السماء".

- **madd_aared_len** (مقدار المد العارض)
  - Values:
    * 2
    * 4
    * 6
  - Default Value:
  - More Info: The length of Mad Al Aared "مد العارض للسكون".

- **madd_alleen_len** (مقدار مد اللين)
  - Values:
    * 2
    * 4
    * 6
  - Default Value: `None`
  - More Info: The length of the Madd al-Leen when stopping at the end of a word (for a sakin waw or ya preceded by a letter with a fatha) should be less than or equal to the length of Madd al-'Arid (the temporary stretch due to stopping). **Default Value is equal to** `madd_aared_len`. مقدار مد اللين عن القوف (للواو الساكنة والياء الساكنة وقبلها حرف مفتوح) ويجب أن يكون مقدار مد اللين أقل من أو يساوي مع العارض

- **ghonna_lam_and_raa** (غنة اللام و الراء)
  - Values:
    * `ghonna` (غنة)
    * `no_ghonna` (لا غنة)

  - Default Value: `no_ghonna` (لا غنة)
  - More Info: The ghonna for merging (Idghaam) noon with Lam and Raa for Hafs.

- **meem_aal_imran** (ميم آل عمران في قوله تعالى: {الم الله} وصلا)
  - Values:
    * `waqf` (وقف)
    * `wasl_2` (فتح الميم ومدها حركتين)
    * `wasl_6` (فتح الميم ومدها ستة حركات)
  - Default Value: `waqf` (وقف)
  - More Info: The ways to recite the word meem Aal Imran (الم الله) at connected recitation. `waqf`: Pause with a prolonged madd (elongation) of 6 harakat (beats). `wasl_2` Pronounce "meem" with fathah (a short "a" sound) and stretch it for 2 harakat. `wasl_6` Pronounce "meem" with fathah and stretch it for 6 harakat.

- **madd_yaa_alayn_alharfy** (مقدار المد اللازم الحرفي للعين)
  - Values:
    * `2`
    * `4`
    * `6`
  - Default Value: `6`
  - More Info: The length of Lzem Harfy of Yaa in letter Al-Ayen Madd "المد الحرفي اللازم لحرف العين" in surar: Maryam "مريم", AlShura "الشورى".

- **saken_before_hamz** (الساكن قبل الهمز)
  - Values:
    * `tahqeek` (تحقيق)
    * `general_sakt` (سكت عام)
    * `local_sakt` (سكت خاص)
  - Default Value: `tahqeek` (تحقيق)
  - More Info: The ways of Hafs for saken before hamz. "The letter with sukoon before the hamzah (ء)".And it has three forms: full articulation (`tahqeeq`), general pause (`general_sakt`), and specific pause (`local_skat`).

- **sakt_iwaja** (السكت عند عوجا في الكهف)
  - Values:
    * `sakt` (سكت)

    - `waqf` (وقف)
    - `idraj` (إدراج)
  - Default Value: `waqf` (وقف)
  - More Info: The ways to recite the word "عوجا" (Iwaja). `sakt` means slight pause. `idraj` means not `sakt`. `waqf`: means full pause, so we can not determine whether the reciter uses `sakt` or `idraj` (no sakt).

- **sakt_marqdena** (السكت عند مرقدنا في يس)
  - Values:
    - `sakt` (سكت)
    - `waqf` (وقف)
    - `idraj` (إدراج)
  - Default Value: `waqf` (وقف)
  - More Info: The ways to recite the word "مرقدنا" (Marqadena) in Surat Yassen. `sakt` means slight pause. `idraj` means not `sakt`. `waqf`: means full pause, so we can not determine whether the reciter uses `sakt` or `idraj` (no sakt).

- **sakt_man_raq** (السكت عند من راق في القيامة)
  - Values:
    - `sakt` (سكت)
    - `waqf` (وقف)
    - `idraj` (إدراج)
  - Default Value: `sakt` (سكت)
  - More Info: The ways to recite the word "من راق" (Man Raq) in Surat Al Qiyama. `sakt` means slight pause. `idraj` means not `sakt`. `waqf`: means full pause, so we can not determine whether the reciter uses `sakt` or `idraj` (no sakt).

- **sakt_bal_ran** (السكت عند بل ران في المطففين)
  - Values:
    - `sakt` (سكت)
    - `waqf` (وقف)
    - `idraj` (إدراج)
  - Default Value: `sakt` (سكت)
  - More Info: The ways to recite the word "بل ران" (Bal Ran) in Surat Al Motaffin. `sakt` means slight pause. `idraj` means not `sakt`. `waqf`: means full pause, so we can not determine whether the reciter uses `sakt` or `idraj` (no sakt).

- **sakt_maleeyah** (وجه قوله تعالى {ماليه هلك} بالحاقة)
  - Values:
    - `sakt` (سكت)
    - `waqf` (وقف)
    - `idgham` (إدغام)
  - Default Value: `waqf` (وقف)
  - More Info: The ways to recite the word {ماليه هلك} in Surah Al-Ahqaf. `sakt` means slight pause. `idgham` Assimilation of the letter 'Ha' (ه) into the letter 'Ha' (ه) with complete assimilation.`waqf`: means full pause, so we can not determine whether the reciter uses `sakt` or `idgham`.
- **between_anfal_and_tawba** (وجه بين الأنفال والتوبة)
  - Values:
    - `waqf` (وقف)
    - `sakt` (سكت)
    - `wasl` (وصل)
  - Default Value: `waqf` (وقف)
  - More Info: The ways to recite end of Surah Al-Anfal and beginning of Surah At-Tawbah.
- **noon_and_yaseen** (الإدغام والإظهار في النون عند الواو من قوله تعالى: {يس والقرآن}و {ن والقلم})
  - Values:
    - `izhar` (إظهار)
    - `idgham` (إدغام)
  - Default Value: `izhar` (إظهار)
  - More Info: Whether to merge noon of both: {يس} and {ن} with (و) "`idgham`" or not "`izhar`".
- **yaa_ataan** (إثبات الياء وحذفها وقفا في قوله تعالى {آتان} بالنمل)
  - Values:
    - `wasl` (وصل)
    - `hadhf` (حذف)
    - `ithbat` (إثبات)
  - Default Value: `wasl` (وصل)

  - More Info: The affirmation and omission of the letter 'Yaa' in the pause of the verse {آتاني} in Surah An-Naml. `wasl`: means connected recitation without pausing as (آتانيَ). `hadhf`: means deletion of letter (ي) at pause so recited as (آتان). `ithbat`: means confirmation reciting letter (ي) at pause as (آتاني).

- **start_with_ism** (وجه البدأ بكلمة {الاسم} في سورة الحجرات)
  - Values:
    - `wasl` (وصل)
    - `lism` (لسم)
    - `alism` (ألسم)
  - Default Value: `wasl` (وصل)
  - More Info: The ruling on starting with the word {الاسم} in Surah Al-Hujurat. `lism` Recited as (لسم) at the beginning. `alism` Recited as (ألسم). `wasl`: means completing recitation without pausing as normal, So Reciting is as (بئس لسم).

- **yabsut** (السين والصاد في قوله تعالى: {والله يقبض ويبسط} بالبقرة)
  - Values:
    - `seen` (سين)
    - `saad` (صاد)
  - Default Value: `seen` (سين)
  - More Info: The ruling on pronouncing `seen` (س) or `saad` (ص) in the verse {والله يقبض ويبسط} in Surah Al-Baqarah.

- **bastah** (السين والصاد في قوله تعالى: {وزادكم في الخلق بسطة} بالأعراف)
  - Values:
    - `seen` (سين)
    - `saad` (صاد)
  - Default Value: `seen` (سين)
  - More Info: The ruling on pronouncing `seen` (س) or `saad` (ص) in the verse {وزادكم في الخلق بسطة} in Surah Al-A'raf.

- **almusaytirun** (السين والصاد في قوله تعالى {أم هم المصيطرون} بالطور)
  - Values:
    - `seen` (سين)
    - `saad` (صاد)
  - Default Value: `saad` (صاد)

  - More Info: The pronunciation of `seen` (س) or `saad` (ص) in the verse {أم هم المصيطرون} in Surah At-Tur.

- **bimusaytir** (السين والصاد في قوله تعالى: {لست عليهم بمصيطر} بالغاشية)
  - Values:
    * `seen` (سين)
    * `saad` (صاد)
  - Default Value: `saad` (صاد)
  - More Info: The pronunciation of `seen` (س) or `saad` (ص) in the verse {لست عليهم بمصيطر} in Surah Al-Ghashiyah.

- **tasheel_or_madd** (همزة الوصل في قوله تعالى: {آلذكرين} بموضعي الأنعام و{آلآن} موضعي يونس و{آلله} بيونس والنمل)
  - Values:
    * `tasheel` (تسهيل)
    * `madd` (مد)
  - Default Value: `madd` (مد)
  - More Info: Tasheel of Madd "وجع التسهيل أو المد" for 6 words in The Holy Quran: "ءالذكرين", "ءالله", "ءائن".

- **yalhath_dhalik** (الإدغام وعدمه في قوله تعالى: {يلهث ذلك} بالأعراف)
  - Values:
    * `izhar` (إظهار)
    * `idgham` (إدغام)
    * `waqf` (وقف)
  - Default Value: `idgham` (إدغام)
  - More Info: The assimilation (`idgham`) and non-assimilation (`izhar`) in the verse {يلهث ذلك} in Surah Al-A'raf. `waqf`: means the reciter has paused on (يلهث)

- **irkab_maana** (الإدغام والإظهار في قوله تعالى: {اركب معنا} بهود)
  - Values:
    * `izhar` (إظهار)
    * `idgham` (إدغام)
    * `waqf` (وقف)
  - Default Value: `idgham` (إدغام)

  - More Info: The assimilation and clear pronunciation in the verse {اركب معنا} in Surah Hud. This refers to the recitation rules concerning whether the letter "Noon" (ن) is assimilated into the following letter or pronounced clearly when reciting this specific verse. `waqf`: means the reciter has paused on (اركب)

- **noon_tamnna** (الإشمام والروم (الاختلاس) في قوله تعالى {لا تأمنا على يوسف})
  - Values:
    * `ishmam` (إشمام)
    * `rawm` (روم)
  - Default Value: `ishmam` (إشمام)
  - More Info: The nasalization (`ishmam`) or the slight drawing (`rawm`) in the verse {لا تأمنا على يوسف}

- **harakat_daaf** (حركة الضاد (فتح أو ضم) في قوله تعالى {ضعف} بالروم)
  - Values:
    * `fath` (فتح)
    * `dam` (ضم)
  - Default Value: `fath` (فتح)
  - More Info: The vowel movement of the letter 'Dhad' (ض) (whether with `fath` or `dam`) in the word {ضعف} in Surah Ar-Rum.

- **alif_salasila** (إثبات الألف وحذفها وقفا في قوله تعالى: {سلاسلا} بسورة الإنسان)
  - Values:
    * `hadhf` (حذف)
    * `ithbat` (إثبات)
    * `wasl` (وصل)
  - Default Value: `wasl` (وصل)
  - More Info: Affirmation and omission of the 'Alif' when pausing in the verse {سلاسلا} in Surah Al-Insan. This refers to the recitation rule regarding whether the final "Alif" in the word "سلاسلا" is pronounced (affirmed) or omitted when pausing (waqf) at this word during recitation in the specific verse from Surah Al-Insan. `hadhf`: means to remove alif (ا) during pause as (سلاسل) `ithbat`: means to recite alif (ا) during pause as (سلاسلا) `wasl` means completing the recitation as normal without pausing, so recite it as (سلاسلَ وأغلالا)

- **idgham_nakhluqkum** (إدغام القاف في الكاف إدغاما ناقصا أو كاملا {نخلقكم} بالمرسلات)
  - Values:

    - `idgham_kamil` (إدغام كامل)
    - `idgham_naqis` (إدغام ناقص)
  - Default Value: `idgham_kamil` (إدغام كامل)
  - More Info: Assimilation of the letter 'Qaf' into the letter 'Kaf,' whether incomplete (`idgham_naqis`) or complete (`idgham_kamil`), in the verse {نخلقكم} in Surah Al-Mursalat.

- **raa_firq** (التفخيم والترقيق في راء {فرق} في الشعراء وصلا)
  - Values:
    - `waqf` (وقف)
    - `tafkheem` (تفخيم)
    - `tarqeeq` (ترقيق)
  - Default Value: `tafkheem` (تفخيم)
  - More Info: Emphasis and softening of the letter 'Ra' in the word {فرق} in Surah Ash-Shu'ara' when connected (wasl). This refers to the recitation rules concerning whether the letter "Ra" (ر) in the word "فرق" is pronounced with emphasis (`tafkheem`) or softening (`tarqeeq`) when reciting the specific verse from Surah Ash-Shu'ara' in connected speech. `waqf`: means pausing so we only have one way (tafkheem of Raa)

- **raa_alqitr** (التفخيم والترقيق في راء {القطر} في سبأ وقفا)
  - Values:
    - `wasl` (وصل)
    - `tafkheem` (تفخيم)
    - `tarqeeq` (ترقيق)
  - Default Value: `wasl` (وصل)
  - More Info: Emphasis and softening of the letter 'Ra' in the word {القطر} in Surah Saba' when pausing (waqf). This refers to the recitation rules regarding whether the letter "Ra" (ر) in the word "القطر" is pronounced with emphasis (`tafkheem`) or softening (`tarqeeq`) when pausing at this word in Surah Saba'. `wasl`: means not pausing so we only have one way (tarqeeq of Raa)

- **raa_misr** (التفخيم والترقيق في راء {مصر} في يونس وموضعي يوسف والزخرف وقفا)
  - Values:
    - `wasl` (وصل)
    - `tafkheem` (تفخيم)
    - `tarqeeq` (ترقيق)

  - Default Value: `wasl` (وصل)
  - More Info: Emphasis and softening of the letter 'Ra' in the word {مصر} in Surah Yunus, and in the locations of Surah Yusuf and Surah Az-Zukhruf when pausing (waqf). This refers to the recitation rules regarding whether the letter "Ra" (ر) in the word "مصر" is pronounced with emphasis (`tafkheem`) or softening (`tarqeeq`) at the specific pauses in these Surahs. `wasl`: means not pausing so we only have one way (tafkheem of Raa)

- **raa_nudhur** (التفخيم والترقيق في راء {نذر} بالقمر وقفا)
  - Values:
    * `wasl` (وصل)
    * `tafkheem` (تفخيم)
    * `tarqeeq` (ترقيق)
  - Default Value: `tafkheem` (تفخيم)
  - More Info: Emphasis and softening of the letter 'Ra' in the word {نذر} in Surah Al-Qamar when pausing (waqf). This refers to the recitation rules regarding whether the letter "Ra" (ر) in the word "نذر" is pronounced with emphasis (`tafkheem`) or softening (`tarqeeq`) when pausing at this word in Surah Al-Qamar. `wasl`: means not pausing so we only have one way (tarqeeq of Raa)

- **raa_yasr** (التفخيم والترقيق في راء {يسر} بالفجر و{أن أسر} بطه والشعراء و{فأسر} بهود والحجر والدخان وقفا)
  - Values:
    * `wasl` (وصل)
    * `tafkheem` (تفخيم)
    * `tarqeeq` (ترقيق)
  - Default Value: `tarqeeq` (ترقيق)
  - More Info: Emphasis and softening of the letter 'Ra' in the word {يسر} in Surah Al-Fajr when pausing (waqf). This refers to the recitation rules regarding whether the letter "Ra" (ر) in the word "يسر" is pronounced with emphasis (`tafkheem`) or softening (`tarqeeq`) when pausing at this word in Surah Al-Fajr. `wasl`: means not pausing so we only have one way (tarqeeq of Raa)

- **meem_mokhfah** (هل الميم مخفاة أو مدغمة)
  - Values:
    * `meem` (ميم)
    * `ikhfaa` (إخفاء)

– Default Value: `ikhfaa` (إخفاء)

– More Info: This is not a **standard** Hafs way but a disagreement between **scholars** in our century on how to **pronounce Ikhfa** for meem. Some **scholars** do full merging (إدغام) and the others open the **lips** a little bit (إخفاء). We did not want to add this, but some of the best reciters disagree about this.

4.4 Collection of Expert Recitations

Recitations were collected from 22 world-class reciters, amounting to a total of **893 hours** of audio before filtering. The collection process prioritized premium audio quality.

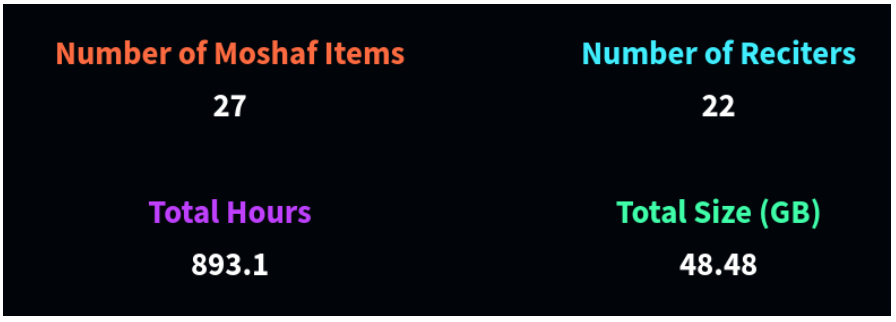


Figure 4.1: Overview statistics of the collected audio database.

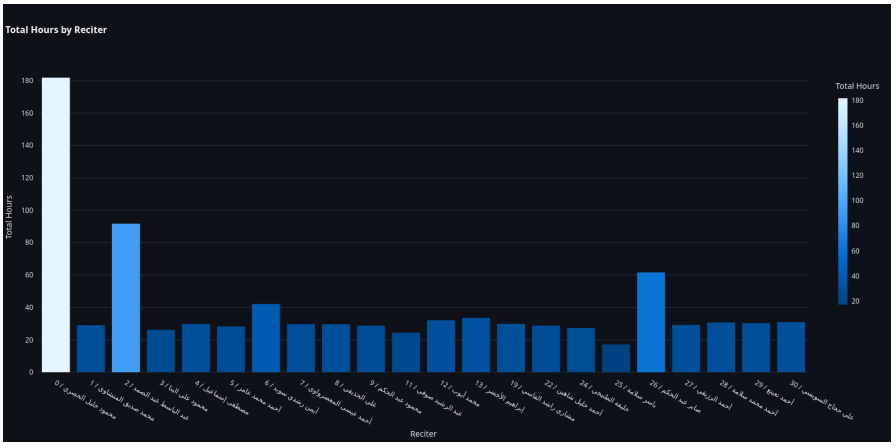


Figure 4.2: Total duration of collected recitations, broken down by individual reciter.

To facilitate this collection, a web GUI was developed using Streamlit[3]. This application performs the following tasks:

- Downloads audio tracks and extracts their metadata.

[3]https://streamlit.io/

- Organizes the data by Moshaf, with each chapter saved as a separate file (e.g., ‘001.mp3‘).
- Provides an interface for annotating Moshaf attribute cards.

4.4.1 Running the Collection Application

4.4.1.1 Cloning the Repository

The application source code can be obtained by cloning the Git repository:

```
git clone https://github.com/obadx/prepare-quran-dataset
```

4.4.1.2 Installing ‘uv‘

The project uses ‘uv‘ for dependency management. It can be installed via ‘pip‘:

```
pip install uv
```

Alternatively, it can be installed directly from the official installer:

```
curl -LsSf https://astral.sh/uv/install.sh | sh
```

4.4.1.3 Installing Project Dependencies

Navigate to the project directory and sync the dependencies, including those for annotation:

```
cd prepare-quran-dataset
uv sync --extra annotate
```

4.4.1.4 Installing Frontend Dependencies

The frontend has additional requirements. Navigate to its directory and install them:

```
cd frontend
uv pip install -r requirements.txt
```

4.4.1.5 Launching the Frontend Application

With dependencies installed, the Streamlit application can be launched from the ‘frontend‘ directory:

```
streamlit run streamlit_app.py
```

4.4.2 UI Snapshots

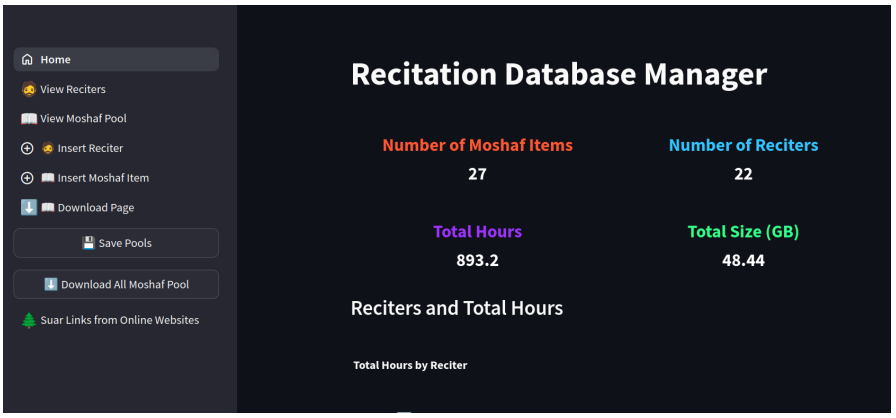


Figure 4.3: The main page of the custom annotation platform.

Figure 4.4: The reciter management view within the application.

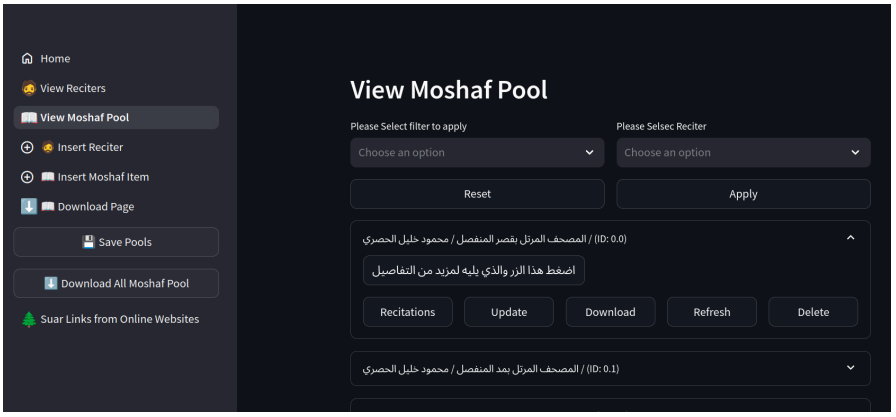


Figure 4.5: View displaying all available Masahif in the database.

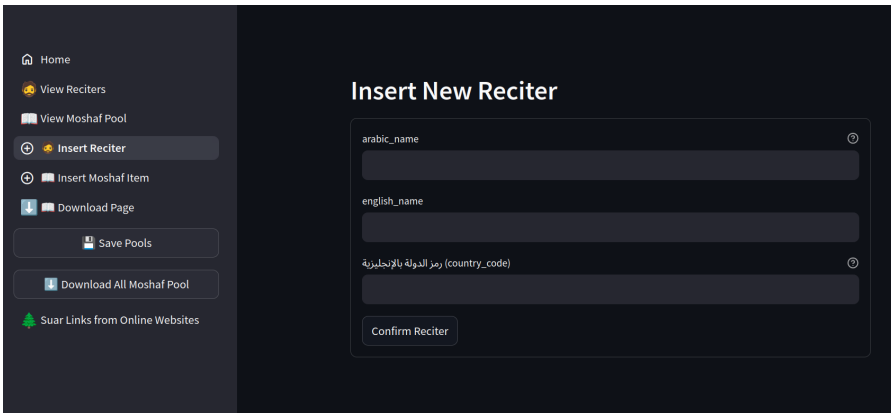


Figure 4.6: Dialog for inserting a new reciter's details.

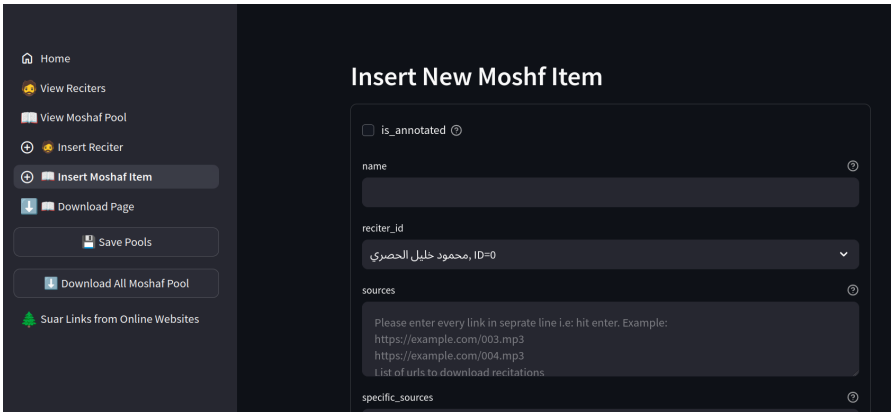


Figure 4.7: Dialog for creating and annotating a new Moshaf attribute card.

Figure 4.8: Interface for viewing a Moshaf's tracks and playing individual recitations.

4.5 Segmentation of Recitations

Accurate segmentation is a critical preprocessing step, as Tajweed rules are directly influenced by pause points (وقف). To address this, we initially evaluated open-source Voice Activity Detection

(VAD) models, including SileroVAD [33] and PyAnnotate [34]. However, their performance on Quranic recitations was unsatisfactory due to the unique acoustic and prosodic characteristics of Tilawah.

Consequently, we developed a custom segmentation model by fine-tuning the Wav2Vec2-BERT architecture [24] for frame-level classification, specifically optimized for Quranic audio.

4.5.1 Preparation of Segmenter Training Data

To create a training dataset, we selected مصاحف from the EveryAyah[4] database that were compatible with SileroVAD v4. This source provided pre-segmented recitations at the verse (ayah) level, which served as our ground truth.

For each Moshaf, we tuned the following segmentation parameters to optimize alignment with the ground truth:

- **Threshold:** Detection confidence level.
- **Minimum Silence Duration:** Durations below this value trigger segment merging.
- **Minimum Speech Duration:** Segments shorter than this value are discarded.
- **Padding:** Duration added to the beginning and end of each detected segment.

The resulting dataset, comprising eight complete مصاحف, is summarized in Table 4.2.

Table 4.2: Dataset used for training the custom segmenter, consisting of eight complete Masahif with tuned parameters.

Reciter Name	ID	Window Size (Samples)	Threshold	Min Silence (ms)	Min Speech (ms)	Pad (ms)
محمود خليل الحصري	0	1536	0.3	500	1000	40
محمد صديق المنشاوي	1	1536	0.3	400	1000	20
عبد الباسط عبد الصمد	2	1536	0.3	400	700	20
محمود علي البنا	3	1536	0.3	400	700	20
علي الحذيفي	5	1536	0.3	350	700	5
أيمن رشدي سويد	6	1536	0.3	500	1000	10
محمد أيوب	7	1536	0.3	400	1000	10
إبراهيم الأخضر	8	1536	0.3	390	700	30

[4] https://everyayah.com

4.5.1.1 Data Augmentation

To improve model robustness and generalize across various recording conditions, we employed data augmentation using the Audiomentations library. The augmentation strategy replicated SileroVAD's noise profile and was applied to 40% of the samples. With additional:

- `TimeStretch` (0.8x-1.5x) to simulate recitation speeds
- Sliding window truncation (1-second windows) for long samples instead of exclusion

4.5.2 Segmenter Training

The Wav2Vec2BERT model was fine-tuned for frame-level classification over a single epoch. The architecture of our VAD model compared to standard streaming models is illustrated in the figure below.

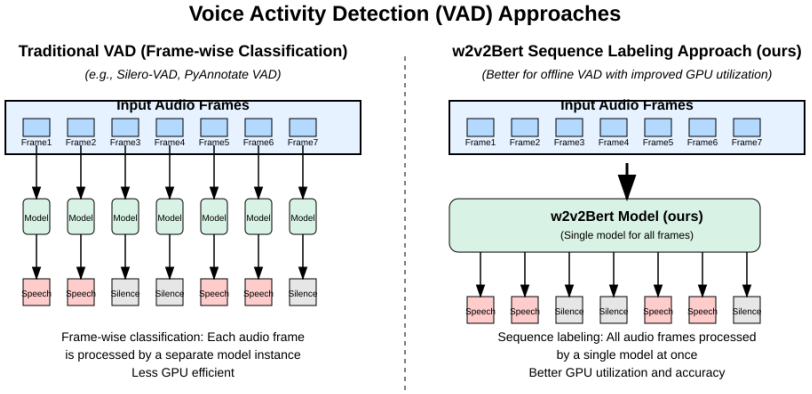


Figure 4.9: Architecture of the fine-tuned Wav2Vec2-BERT model for frame classification, compared to a standard streaming model.

The model's performance on unseen مصاحف demonstrated high accuracy, as shown in Table 4.3.

Table 4.3: Evaluation results of the segmentation model on a held-out test set of Masahif, showing superior performance. The quality of the segmenter was validated by processing our entire dataset, where it maintained this high level of performance. The only exceptions were edge cases involving extremely fast recitation (حدر), which is an expected limitation.

Metric	Value
Test Loss	0.0277
Test Accuracy	0.9935
Test F1 Score	0.99476

4.6 Transcribe Segmented Parts

We employed Tarteel ASR [35] (Whisper fine-tuned on Quranic recitations [36]). To handle its 30-second limit, we used sliding window truncation (10-second windows), with verification in the next step. We used vlLM library because it is really fast thanks to Employing Paged Attention [37].

4.7 Verification of Segmentation and Transcription

4.8 Data Verification

To ensure the highest quality of our dataset, we developed a custom verification interface using Streamlit[5]. We manually inspect 50-75 randomly selected samples per Moshaf, focusing on the following aspects:

- **Segmentation Quality:** Assessing the accuracy of pause detection to determine if adjustments were needed, including:
  - Increasing or decreasing padding durations
  - Merging adjacent segments
  - Splitting undetected segments
- **Qalqala (القلقلة) Duration Inspection:** Verifying that segments containing Qalqala (قلقة) are fully captured without being truncated by brief silences, ensuring the acoustic feature remains intact.
- **Hams (همس) Duration Inspection:** Similarly checking segments for Hams (a whispered or airy phonation) to ensure the subtle release of air was not missed by the segmenter.

[5] https://streamlit.io/

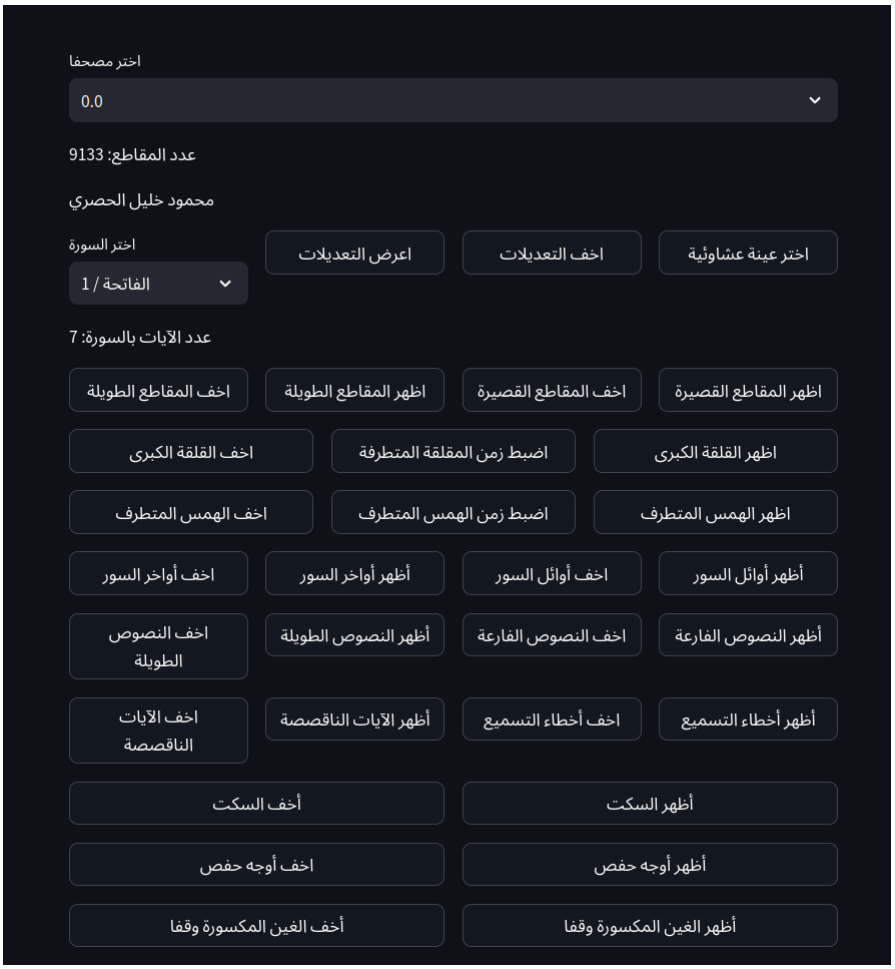


Figure 4.10: The Streamlit-based UI for manually verifying segmentation quality and phonetic feature integrity.

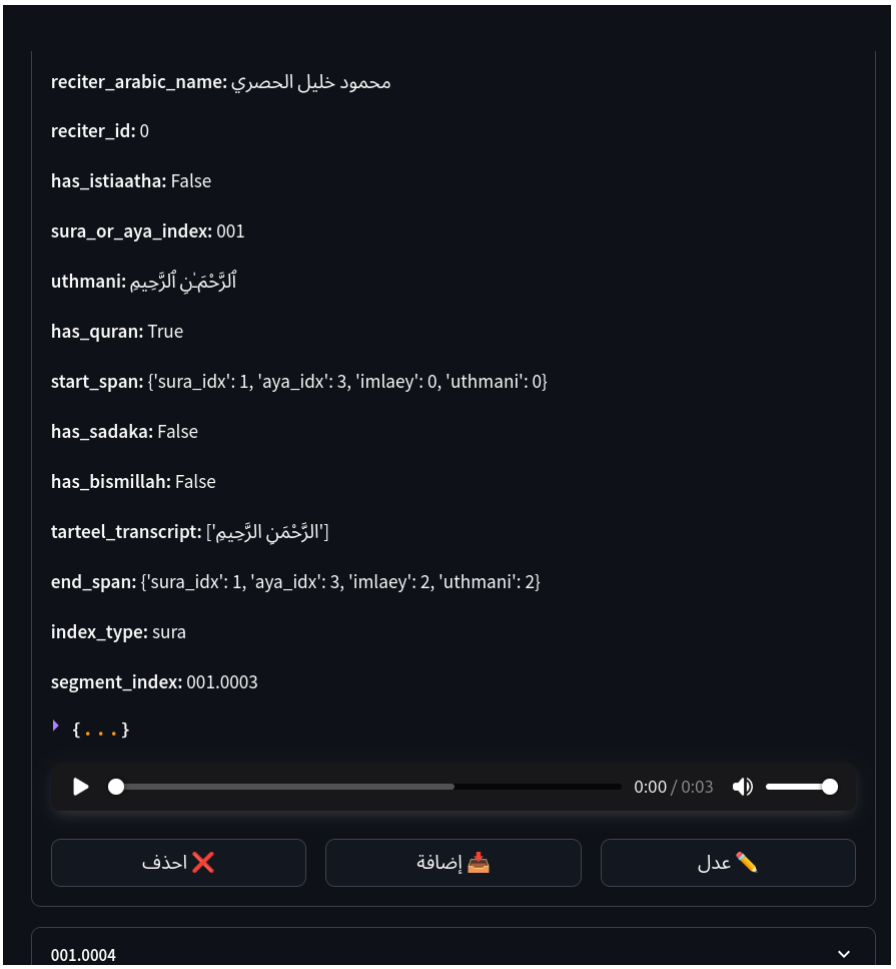


Figure 4.11: The editing view within the verification UI, allowing for manual correction of segment boundaries.

After completing the annotation process for a Moshaf, the defined correction operations were programmatically applied to the entire dataset to ensure consistency.

Note: Moshaf 25.0 was excluded from the final dataset due to irreconcilably poor segmentation quality.

4.8.1 Transcription Verification: A تسميع-Inspired Algorithm

To validate the accuracy of the automated speech recognition (ASR) output, we developed a verification algorithm inspired by Tasmeea (تسميع)—the traditional practice where a student recites for a teacher to correct mistakes. This statistical algorithm operates under the core assumption that the input recitations are 100% correct, and any errors originate from the ASR model (Tarteel model).

The algorithm proceeds through the following steps:

1. **Automatic Matching:** Segments are automatically matched to the canonical Quranic text.
2. **Discrepancy Identification:** The system identifies missing verses, words, or unexpected additions in the transcription.
3. **Manual Correction:** flagged discrepancies are presented for manual review and correction within our annotation UI, completing the تسميع feedback loop.

Algorithm 1: Tasmeea Algorithm

[1] $text\_segments = [s_1, s_2, \ldots, s_n]$, $sura\_idx$, $overlap\_words = 6$, $window\_words = 30$, $acceptance\_ratio = 0.5$, flags for special phrases List of tuples $(match, ratio)$ per segment

$aya \leftarrow 1$ Start at first verse $penalty \leftarrow 0$ each segment s_i in $text\_segments$ $norm\_text \leftarrow$ normalize(s_i) Remove spaces/diacritics $min\_win \leftarrow window\_words - 10$, $max\_win \leftarrow window\_words + 10$ $start\_range \leftarrow [-(overlap + penalty), (overlap + \max(window\_words, max\_win) + penalty]$

first segment $include\_istiaatha$ Check istiaatha special case last segment $include\_sadaka$ Check sadaka special case

$best\_ratio \leftarrow 0$, $best\_match \leftarrow$ null each start position p in $start\_range$ each window size $w \in [min\_win, max\_win]$ $c \leftarrow$ extract candidate at (aya, p, w) $dist \leftarrow$ edit_distance$(norm_text, c)$ $ratio \leftarrow 1 - \min(dist, |norm\_text|)/|norm\_text|$ $ratio > best\_ratio$ ($ratio = best\_ratio$ $|p| < |best\_start|$) update $best\_ratio$, $best\_match$, $best\_start$, $best\_window$

$best\_ratio < acceptance\_ratio$ output (null, $best\_ratio$) $penalty \leftarrow max\_win$ $aya \leftarrow aya + 1$ Default advance output ($best\_match$, $best\_ratio$) $aya \leftarrow aya + best\_start + best\_window$ $penalty \leftarrow 0$ **Complexity:** $O(N \cdot W \cdot L^2)$ N=segments, W=window size, L=segment length

Chapter 5

Modeling Quran Phonetic Script

5.1 Modeling

Our Quran Phonetic Script produces two types of outputs: ‘phonemes‘ and ‘sifat‘ (which comprises 10 distinct attributes). We model this task as follows: Imagine processing an input speech utterance and simultaneously generating transcripts in multiple languages, such as Arabic, English, French, and German. Similarly, we implement a speech encoder with a separate linear output layer for each of our 11 levels (one for ‘phonemes‘ and 10 for the ‘sifat‘ attributes), resulting in 11 parallel transcription heads. We employ the Connectionist Temporal Classification (CTC) loss [38] without a language model, as our objective is to transcribe the actual pronunciation rather than the intended utterance. We refer to this architecture as **Multi-level CTC**.

The total loss is computed as a weighted average of the CTC losses across all 11 levels. The ‘phonemes‘ level is assigned a weight of 0.4 due to its larger vocabulary size (43 symbols), while the remaining levels are weighted proportionally lower:

$$\text{loss} = \sum_i (\text{level\_weight}_i \cdot \text{CTC\_level}_i) \tag{5.1}$$

We used a weight of 0.4 for the ‘phonemes‘ level, 0.0605 for both ‘shidda_or_rakhawa‘ and ‘tafkheem_or_taqeeq‘, and 0.059875 for each of the remaining levels.

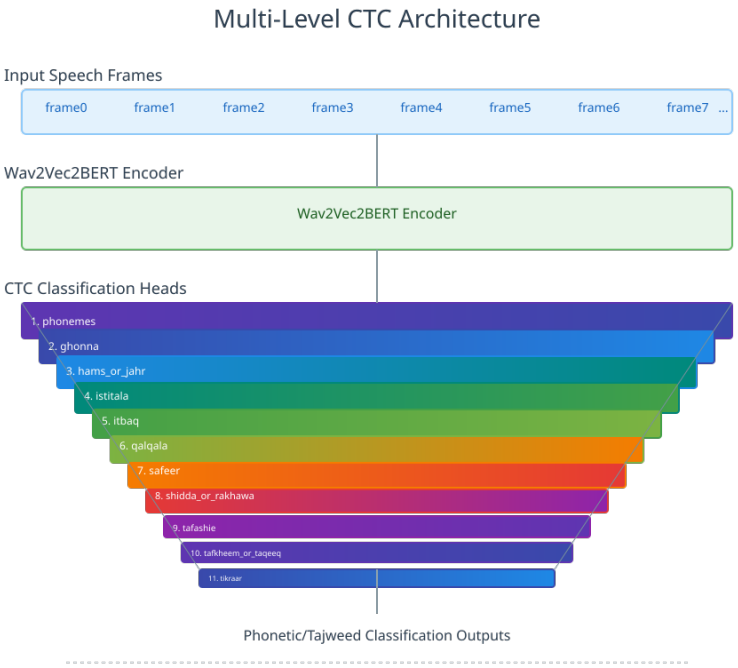


Figure 5.1: Multi-level CTC architecture with 11 output heads, each computing a CTC loss, combined via weighted average.

We fine-tuned Facebook's Wav2Vec2-Bert model [24] for a single epoch using a constant learning rate of `5e-5`. Data augmentations were applied using the `audiomentations` library [39], mirroring the augmentations used in Silero VAD [33], with additional augmentations including `TimeStretch` and `GainTransition`. Samples longer than 30 seconds were filtered out to optimize GPU memory utilization—this resulted in the exclusion of only 3k samples from the 250k training set.

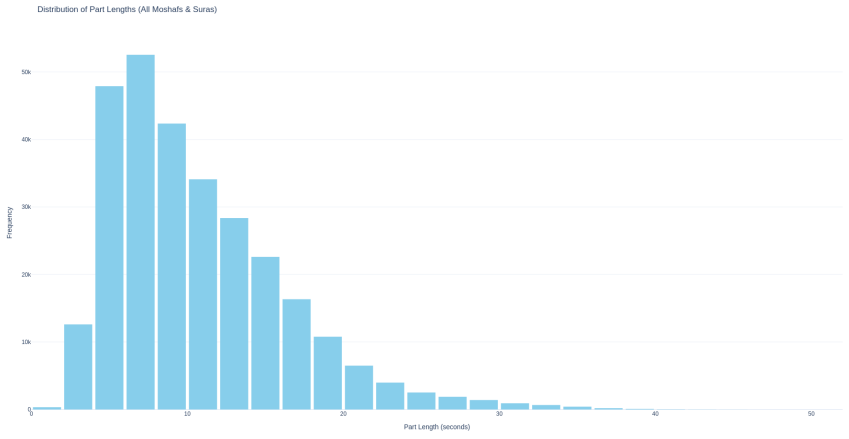


Figure 5.2: Distribution of recitation lengths (in seconds) across the dataset.

Training was conducted on a single H200 GPU with 141 GB of memory and completed in approximately 7 hours.

Chapter 6

Results

6.1 Results

We trained our model on all available Mushaf recitations, reserving Mushaf 26.1 and 19.0 exclusively for testing. The evaluation results are summarized in Table 6.1. The achieved Average Phoneme Error Rate (PER) of 0.16% strongly supports our hypothesis that the Quran Phonetic Script is learnable using modern speech processing techniques.

We further tested the model on actual samples containing errors in مد, غنة, قلقلة, and تفخيم. Despite being trained only on error-free expert recitations, the model successfully detected these common pronunciation mistakes. While these preliminary results are promising, a more comprehensive evaluation on dedicated error-annotated datasets—such as [9]—is planned for future work.

We observe that the PER is well-balanced across nearly all phonetic and attribute levels, with the exception of the phoneme level itself. This is expected, as the phoneme level has a significantly larger vocabulary (44 symbols, including padding), increasing its complexity relative to the attribute levels.

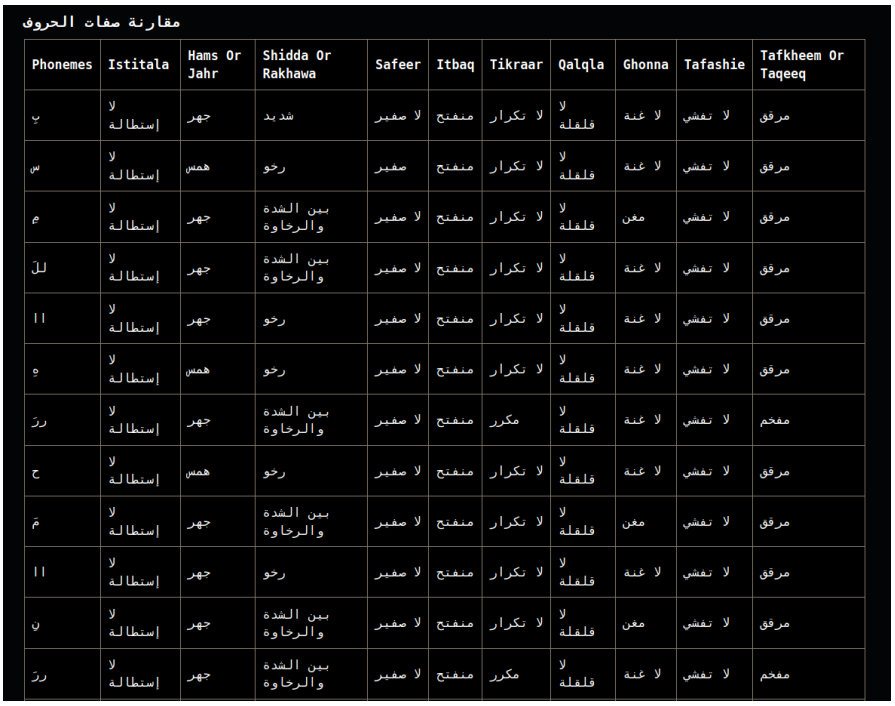
To evaluate real-world performance, we developed a demonstration application using Gradio[1]. The interface allows users to record or upload their recitations and receive immediate phonetic and attribute-level feedback.

[1] https://www.gradio.app/

Table 6.1: Test results on Mushaf 26.1 and 19.0. The Average Phoneme Error Rate (PER) is **0.16%**, confirming the learnability of the Quran Phonetic Script. The phoneme-level PER is higher (0.54%) due to its larger vocabulary.

Metric	Value
loss	0.01162
per_phonemes	0.00543
per_hams_or_jahr	0.00117
per_shidda_or_rakhawa	0.00172
per_tafkheem_or_taqeeq	0.00167
per_itbaq	0.00092
per_safeer	0.00132
per_qalqla	0.00085
per_tikraar	0.0009
per_tafashie	0.0016
per_istitala	0.0008
per_ghonna	0.0013
average_per	**0.0016**

Figure 6.1: Gradio Web App interface allowing users to test our model.


مقارنة صفات الحروف

Phonemes	Istitala	Hams Or Jahr	Shidda Or Rakhawa	Safeer	Itbaq	Tikraar	Qalqla	Ghonna	Tafashie	Tafkheem Or Taqeeq
بِ	لا إستطالة	جهر	شديد	لا صفير	منفتح	لا تكرار	لا قلقلة	لا غنة	لا تفشي	مرقق
سِ	لا إستطالة	همس	رخو	صفير	منفتح	لا تكرار	لا قلقلة	لا غنة	لا تفشي	مرقق
مِ	لا إستطالة	جهر	بين الشدة والرخاوة	لا صفير	منفتح	لا تكرار	لا قلقلة	مغن	لا تفشي	مرقق
للّ	لا إستطالة	جهر	بين الشدة والرخاوة	لا صفير	منفتح	لا تكرار	لا قلقلة	لا غنة	لا تفشي	مرقق
اا	لا إستطالة	جهر	رخو	لا صفير	منفتح	لا تكرار	لا قلقلة	لا غنة	لا تفشي	مرقق
هِ	لا إستطالة	همس	رخو	لا صفير	منفتح	لا تكرار	لا قلقلة	لا غنة	لا تفشي	مرقق
رَ	لا إستطالة	جهر	بين الشدة والرخاوة	لا صفير	منفتح	مكرر	لا قلقلة	لا غنة	لا تفشي	مفخم
حِ	لا إستطالة	همس	رخو	لا صفير	منفتح	لا تكرار	لا قلقلة	لا غنة	لا تفشي	مرقق
مّ	لا إستطالة	جهر	بين الشدة والرخاوة	لا صفير	منفتح	لا تكرار	لا قلقلة	مغن	لا تفشي	مرقق
اا	لا إستطالة	جهر	رخو	لا صفير	منفتح	لا تكرار	لا قلقلة	لا غنة	لا تفشي	مرقق
نِ	لا إستطالة	جهر	بين الشدة والرخاوة	لا صفير	منفتح	لا تكرار	لا قلقلة	مغن	لا تفشي	مرقق
رَ	لا إستطالة	جهر	بين الشدة والرخاوة	لا صفير	منفتح	مكرر	لا قلقلة	لا غنة	لا تفشي	مفخم

Figure 6.2: Gradio Web App interface showing detailed Sifat (attribute) level feedback.

User feedback has been extremely positive. Notably, the model generalized well to female voices despite being trained exclusively on male recitations, successfully detecting common errors such as incorrect مد elongation or weak قلقلة pronunciation. This demonstrates the robustness and practical applicability of our approach.

6.2 Ablation Studies

We conducted more than 13 experiments. To tune the weights for every level, we ran an evaluation set for each experiment on 20% of the training data and logged the Phoneme Error Rate (PER) per level. Our goal was to minimize two metrics:

- Average Phoneme Error Rate (PER).
- Standard deviation across all levels.

We report the best three experiments, labeled ‘EXP1‘, ‘EXP2‘, and ‘EXP3‘. We adjusted the loss weight for every level in each experiment. The weight for the ‘phonemes‘ level was kept constant across all runs at 0.4, while the weights for ‘shidda_or_rakhawa‘ and ‘tafkheem_or_tarqeeq‘ were varied to achieve the best possible results.

Table 6.2: The table above shows different loss weights applied to equation 5.1 for three experiments: ‘EXP1‘, ‘EXP2‘, and ‘EXP3‘. The ‘phonemes‘ level weight was kept constant across all runs. In each run, we tuned the weights for ‘shidda_or_rakhawa‘ and ‘tafkheem_or_tarqeeq‘ to minimize the average Phoneme Error Rate (PER) and the standard deviation across all levels. Note that the sum of all loss weights adds to 1.

Attribute	**EXP1**	**EXP2**	**EXP3**
phonemes	0.4	0.4	0.4
tikraar	0.06	0.058625	0.059625
tafkheem_or_tarqeeq	0.06	0.063	0.0060
tafashie	0.06	0.05825	0.059625
Qalqala	0.06	0.0585	0.059625
Safeer	0.06	0.05825	0.059625
Shidda_or_rakhawa	0.06	0.068	0.059625
Istitala	0.06	0.05825	0.059625
Itbaaq	0.06	0.05825	0.059625
Ghonna	0.060	0.05825	0.059625
hams_or_jahr	0.06	0.05825	0.059625
average_per	0.065	0.05825	0.059625
std_per	0.06	0.05825	0.059625

We observed that ‘EXP3‘ yielded the best results, with an average PER of 0.293% and a standard deviation of 0.0017, as shown in both Table 6.3 and Figure 6.3. ‘EXP3‘ performed the best because it allocates more weight to levels that contain more labels—‘shidda_or_rakhawa‘—and to the more challenging level ‘tafkheem_or_tarqeeq‘, without significantly affecting the rest of the levels.

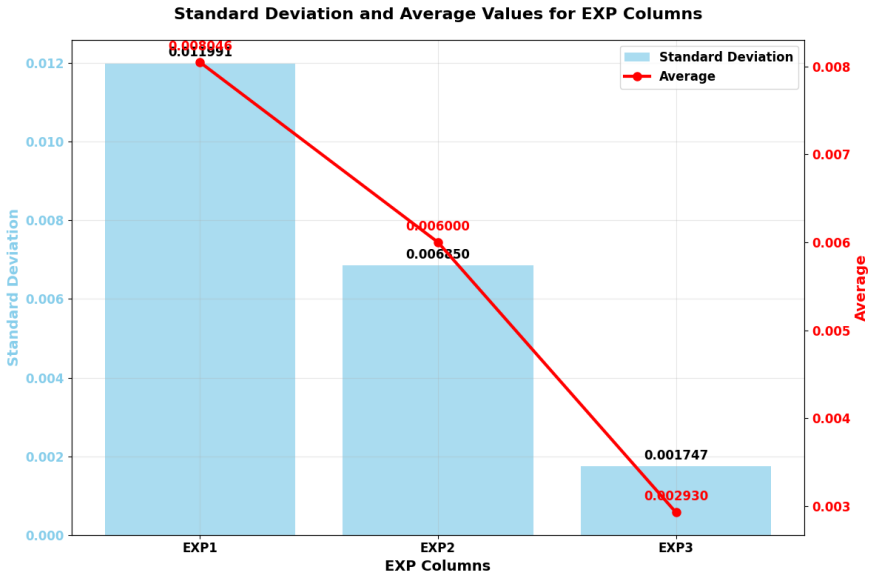


Figure 6.3: Average Phoneme Error Rate and Standard Deviation for all three runs. 'EXP3' performs best by assigning higher weight to levels with more labels ('shidda_or_rakhawa') and the more challenging 'tafkheem_or_tarqeeq' level, without significant differences in the remaining levels.

Table 6.3: Phoneme Error Rate for each level on 20% of the training data. 'EXP3' performs best by assigning higher weight to levels with more labels ('shidda_or_rakhawa') and the more challenging 'tafkheem_or_tarqeeq' level, without significant differences in the remaining levels.

Attribute	**EXP1**	**EXP2**	**EXP3**
phonemes	0.0069	0.0069	0.0063
tikraar	0.006	0.006	0.0017
tafkheem_or_tarqeeq	0.002599	0.00279	0.0065
tafashie	0.001837	0.0025	0.0035
Qalqala	0.001808	0.008	0.00174
Safeer	0.00346	0.00246	0.00174
Shidda_or_rakhawa	0.015276	0.0053	0.0031
Istitala	0.00243	0.00166	0.001525
Itbaaq	0.00176	0.00217	0.00168
Ghonna	0.044	0.02675	0.00199
hams_or_jahr	0.0024	0.00256	0.00234
average_per	0.0080455	0.006	0.00293
std_per	0.0191	0.0062	0.0017

After that we continued the training of 'EXP3' that yields the results in 6.1

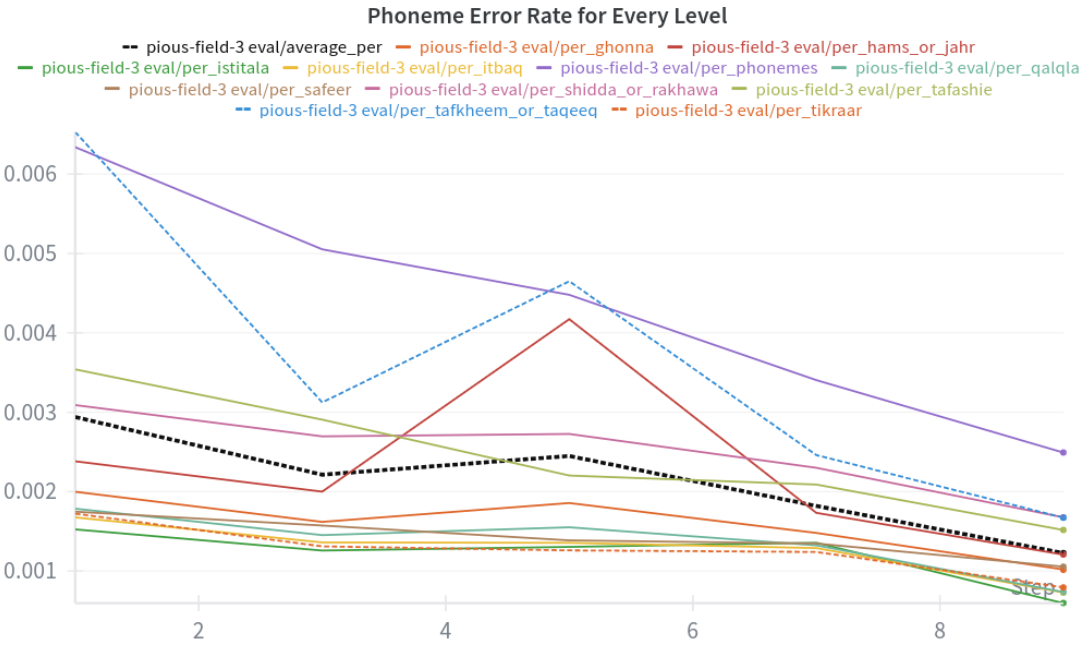


Figure 6.4: Evaluation PER during training steps

6.3 Model Version 3

After completing our initial training and publishing the model, we identified a minor bug and developed a new feature:

- **Bug Fix:** The model was incorrectly classifying a ياء (ي) with a شدة as ‘yaa_madd‘ instead of as two separate ياء characters.
- **New Feature:** A third label, ‘low_mofakham‘, was added to the ‘tafkheem_or_tarqeeq‘ level.

We trained this Version 3 model using the following loss weights: 0.4 for the ‘phonemes‘ level, 0.0605 for both ‘tafkheem_or_tarqeeq‘ and ‘shidda_or_rakhawa‘, and 0.05987 for the remaining levels.

For testing, we removed مصاحف ‘29.0‘ and ‘30.0‘ from the training and validation sets and combined them with the existing test set. This resulted in a final test set containing مصاحف ‘19.0‘, ‘26.1‘, ‘29.0‘, and ‘30.0‘. The results are shown in Table 6.4.

Table 6.4: Testing results on mosahaf ‘19.0‘, ‘26.1‘, ‘29.0‘, and ‘30.0‘ showing a balanced Phoneme Error Rate (PER) across all levels. The ‘phonemes‘ level has a naturally higher PER as it is the largest vocabulary (44 tokens: 43 phonemes + 1 padding token).

Metric	**Value**
per_phonemes	0.00449
per_hams_or_jahr	0.00177
per_shidda_or_rakhawa	0.00315
per_tafkheem_or_taqeeq	0.00299
per_itbaq	0.00130
per_safeer	0.00152
per_qalqla	0.00123
per_tikraar	0.00436
per_tafashie	0.00181
per_istitala	0.00122
per_ghonna	0.00185
average_per	0.00234

Chapter 7

Conclusion

Conclusion

We present a new way of assessing pronunciation errors of Holy Quran learners by developing multi-level Quran Phonetic Script capable of capturing all pronunciation errors for حفص except for إشمام (as it is a sign not pronounced by mouth), along with 890 hours and 300K of annotated data, a 98% pipeline to create similar data, plus modeling and validation.

7.1 Limitations

Our primary limitation is that our dataset consists of golden recitations with no errors, limiting our ability to evaluate performance on real-world data. Although we tested on a few actual samples and successfully detected مد, غنة, and قلقة errors, we need to develop a comprehensive dataset containing error-containing recitations transcribed with our Quran Phonetic Script.

A secondary limitation arises from attribute-specific articulation patterns: Certain attributes apply exclusively to individual letters, such as 'Istitala' for (ض) and 'Tikrar' for (ر). Consequently, we expect our model will be unable to capture instances of (ض) without 'Istitala' or (ر) without 'Tikrar'. This limitation similarly applies to Tajweed rules that occur less frequently in the Holy Quran, such as إمالة, روم, and تسهيل.

7.2 Future Work

To address these limitations, we plan to:

1. **Develop an Error-Included Dataset:** Collect and annotate a large-scale dataset of learner recitations containing common Tajweed errors, transcribed using our phonetic script. This will enable more robust model training and evaluation.

2. **Expand to Other Recitation Styles (روايات):** Extend the phonetic script and modeling framework to support additional recitation styles (e.g., ورش or قالون), facilitating broader applicability across the Muslim world.

3. **Deploy and Evaluate in Real-World Settings:** Integrate the model into user-friendly applications and evaluate its effectiveness in real-world learning environments, incorporating feedback from Quran teachers and students to iteratively improve the system.

By addressing these challenges, we aim to advance the state of Quranic pronunciation assessment and make automated, accurate feedback accessible to learners worldwide.

References

[1] Sherif Mahdy Abdou and Mohsen Rashwan. A computer aided pronunciation learning system for teaching the holy quran recitation rules. In *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*, pages 543–550. IEEE, 2014.

[2] Mubarak Al-Marri, Hazem Raafat, Mustafa Abdallah, Sherif Abdou, and Mohsen Rashwan. Computer aided qur'an pronunciation using dnn. *Journal of Intelligent & Fuzzy Systems*, 34(5):3257–3271, 2018.

[3] Basem HA Ahmed and Ayman S Ghabayen. Arabic automatic speech recognition enhancement. In *2017 Palestinian International Conference on Information and Communication Technology (PICICT)*, pages 98–102. IEEE, 2017.

[4] Yassine El Kheir, Ahmed Ali, and Shammur Absar Chowdhury. Automatic pronunciation assessment–a review. *arXiv preprint arXiv:2310.13974*, 2023.

[5] MA Sherif, A Samir, AH Khalil, and R Mohsen. Enhancing usability of capl system for quran recitation learning. INTERSPEECH, 2007.

[6] Hanaa Mohammed Osman, Ban Sharief Mustafa, and Yusra Faisal. Qdat: a data set for reciting the quran. *International Journal on Islamic Applications in Computer Science And Technology*, 9(1):1–9, 2021.

[7] Dahlia Omran, Sahar Fawzi, and Ahmed Kandil. Automatic detection of some tajweed rules. In *2023 20th Learning and Technology Conference (LT)*, pages 157–160, 2023. doi: 10.1109/LT58159.2023.10092350.

[8] Dim Shaiakhmetov, Gulnaz Gimaletdinova, Kadyrmamat Momunov, and Selcuk Cankurt. Evaluation of the pronunciation of tajweed rules based on dnn as a step towards interactive recitation learning. *arXiv preprint arXiv:2503.23470*, 2025.

[9] Hamzah I Khan, Abubakar Abid, Mohamed Medhat Moussa, and Anas Abou-Allaban. The tarteel dataset: crowd-sourced and labeled quranic recitation. 2021.

[10] Tarteel AI. Tarteel: Ai-powered quran companion, 2023. URL https://www.tarteel.ai/.

[11] Yassine El Kheir, Omnia Ibrahim, Amit Meghanani, Nada Almarwani, Hawau Olamide Toyin, Sadeen Alharbi, Modar Alfadly, Lamya Alkanhal, Ibrahim Selim, Shehab Elbatal, et al. Towards a unified benchmark for arabic pronunciation assessment: Quranic recitation as case study. *arXiv preprint arXiv:2506.07722*, 2025.

[12] Ammar Mohammed, Mohd Shahrizal Bin Sunar, and Md Sah Hj Salam. Recognition of holy quran recitation rules using phoneme duration. In *International Conference of Reliable Information and Communication Technology*, pages 343–352. Springer, 2017.

[13] Ammar Mohammed Ali Alqadasi, Akram M Zeki, Mohd Shahrizal Sunar, Md Sah Bin Hj Salam, Rawad Abdulghafor, and Nashwan Abdo Khaled. Improving automatic forced alignment for phoneme segmentation in quranic recitation. *IEEE Access*, 12:229–244, 2023.

[14] Yousef S Alsahafi and Muhammad Asad. Empirical study on mispronunciation detection for tajweed rules during quran recitation. In *2024 6th International Conference on Computing and Informatics (ICCI)*, pages 39–45, 2024. doi: 10.1109/ICCI61671.2024.10485145.

[15] Budiman Putra, B Tris Atmaja, and D Prananto. Developing speech recognition system for quranic verse recitation learning software. *IJID (International Journal on Informatics for Development)*, 1(2):1–8, 2012.

[16] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[17] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008, 2017.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186, 2019.

[20] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*, 2019.

[21] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

[22] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

[23] Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 244–250. IEEE, 2021.

[24] Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, et al. Seamless: Multilingual expressive and streaming speech translation. *arXiv preprint arXiv:2312.05187*, 2023.

[25] Dahlia Omran, Sahar Fawzi, and Ahmed Kandil. Automatic detection of some tajweed rules. In *2023 20th Learning and Technology Conference (L&T)*, pages 157–160. IEEE, 2023.

[26] مخارج الحروف الصحاح وصفاتها عند الخليل بن أحمد الفراهيدي قراءة في ضوء الدرس اللساني الحديث. .هبيرة, عز مجلة جامعة الأمير عبد القادر للعلوم الإسلامية, 31:165–190, 02 2023. doi: 10.37138/emirj.v31i2.1936.

[27] دار الغوثاني للدراسات القرآنية .التجويد المصور .أمين رشدس سويد, 2021.

[28] تصوير دار الكتاب العلمية .القاهرة, المطبعة التجارية الكبرى .النشر في القراءات العشر .ابن الجزري، محمد بن محمد.

[29] Kais Dukes and Nizar Habash. Morphological annotation of quranic arabic. In *Lrec*, pages 2530–2536, 2010.

[30] معهد .شَرح المقدمة الجزرية: يَجْمَعَ بَيْنَ التُّراثِ الصَّوْتِي العربي القَدِيمِ وَالدَّرْسِ الصوتي الحديث .غانم القدوري حمد الإمام الشاطبي, الطبعة الأولى edition, 2008.

[31] تحقيق وتعليق .المقدمة الجزرية .ابن الجزري, شمس الدين.

[32] مكتبة ومطبعة مصطفى البابي الحلبي وأولاده .صريح النص في الكلمات المختلف فيها عن حفص .علي الضباع, المؤلف: علي الضباع (توفي 1380هـ) .1380هـ"

[33] SileroVAD. Silero vad: pre-trained enterprise-grade voice activity detector (vad), number detector and language classifier. https://github.com/snakers4/silero-vad, 2024.

[34] Alexis Plaquet and Hervé Bredin. Powerset multi-class cross entropy loss for neural speaker diarization. In *Proc. INTERSPEECH 2023*, 2023.

[35] Tarteel AI. Whisper base arabic quran (automatic speech recognition for quranic recitation). https://huggingface.co/tarteel-ai/whisper-base-ar-quran, 2023. Model by Tarteel AI. Company website: https://www.tarteel.ai/.

[36] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.

[37] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[38] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)*, pages 369–376. ACM, 2006. doi: 10.1145/1143844.1143891.

[39] Iver Jordal and Contributors. Audiomentations: A python library for audio data augmentation. https://github.com/iver56/audiomentations, 2025.

الملخص

أنزل الله كتابه وتعهد بحفظه ويسره للذكر ولذلك تعاقب العلماء المسلمون على العناية بالقرآن من جميع الجوانب من اللفظ حتى المعنى وطوعوا الوسائل التقنية المتاحة في زمنهم لخدمة القرآن الكريم. وفي عصر الذكاء الاصطناعي نحاول تقريب القرآن من المسلمين عن طريق تقديم طريقة مبتكرة لكشف وتصحيح أخطاء التلاوة والتجويد وصفات الحروف لدى متعلمي القرآن الكريم عن طريق تدقيم الرسم الصوتي القرآن الكريم متعدد المستويات للحروف وصفاتها ولقلة وفرة التلاوات القرآنية المخصصة لتدريب نماذج الذكاء الاصطناعي قدمنا طريقة شبه مأتمتة بنسبة 98% لبناء قواعد بيانات قرآنية عالية الجودة و قدمنا أيضا قاعدة بيانات مكنونة من 890 ساعة تحتوي على 300 ألف عينة مُعلَّمة وبجانب ذلك قدمنا نموذج مبتكر CTC متعدد المستويات وبفضل الله حققنا متوسط معدل خطأ حرفي قدره 16%. مما يؤكد دقة الرسم الصوتي للقرآن الكريم وسهولة تعلمه ويؤكد تلك الحقيقة ألا وهي: ﴿ وَلَقَدْ يَسَّرْنَا ٱلْقُرْءَانَ لِلذِّكْرِ فَهَلْ مِن مُّدَّكِرٍ ﴾