

**Mobile Base Sinhala Book Reader for Visually Impaired  
Individuals**

**Project ID: 23-198**

**Final Project Thesis – Individual**

**IT20146238 - Jayathunga T.M.**

**Supervisor: Prof. Koliya Pulasinghe**

**B.Sc. (Hons) Degree in Information Technology Specialization in  
Information Technology**

**Sri Lanka Institute of Information Technology**

**Sri Lanka**

**September 2023**

**Mobile Base Sinhala Book Reader for Visually Impaired  
Individuals**

**Project ID: 23-198**

**Final Project Thesis – Individual**

**IT20146238 - Jayathunga T.M.**

**Supervisor: Prof. Koliya Pulasinghe**

**B.Sc. (Hons) Degree in Information Technology Specialization in  
Information Technology**

**Sri Lanka Institute of Information Technology**


**Sri Lanka**

**September 2023**

## DECLARATION

I declare now that the thesis I am presenting is entirely my work, and I have not incorporated, without proper acknowledgment, any material previously submitted for a degree or diploma at any other university or institute of higher learning. This proposal does not contain any material previously published or written by another person, except where the appropriate acknowledgment has been made in the text.

I know the potential consequences of academic dishonesty and plagiarism, and I am committed to upholding honesty and intellectual integrity in all my academic endeavors.

Student ID Number	Student Name	Signature
IT20146238	Jayathunga T.M.	

The Supervisor should certify the individual thesis report with the following declaration.

The candidates mentioned above are currently conducting research for their undergraduate dissertation, under my supervision, under my supervision. As their supervisor, I certify this proposal report.

Signature of the Supervisor

Date

-----

-----

## **ABSTRACT**

**In an era of rapid technological progress, education has evolved beyond its conventional boundaries, driven by the integration of Artificial Intelligence (AI) across diverse fields. However, individuals with visual impairments encounter formidable barriers in accessing educational materials, particularly in languages other than English. To tackle this issue head-on, this research project endeavors to develop a holistic Android-based mobile application exclusively catered to the visually impaired community in Sri Lanka. The application's prime focus lies in facilitating access to educational resources in the Sinhala language, thereby empowering visually impaired individuals in their pursuit of knowledge.**

**The accessibility of printed books and other publications in the Sinhala language for blind people in Lanka has not expanded. The main objective of this project is to enable them to have easy access to books and documents. Here, a visually impaired person navigates the app as needed by voice navigation, and the application gives the necessary commands to the visually impaired person. Here the mobile camera performs the necessary commands to accurately capture all the characters on a page of a Sinhala book. After that, optical character recognition separates and identifies the characters from the image with the corresponding characters. The Festival Synthesizing Framework's Text-to-Speech feature then allows the corresponding visually impaired individual to hear clearly. Here he can increase or decrease the speed of reading the book according to the commands. We intended to develop this Mobile Base Sinhala Book Reader to make it easier for visually impaired people by using audio recommendations and Sinhala voice notifications. We anticipate that this would boost the visually challenged community in Sri Lanka's delight in reading Sinhala novels.**

## **ACKNOWLEDGEMENT**

This significant research project received careful evaluation, reinforced by the profound insights offered by our respected research supervisor, Prof. Koliya Pulasinghe, and co-supervisor, Ms. Poorna Panduwawela, whose advice proved priceless. We also thank the Natural Language Processing (NLP) domain evaluation panel for their thorough review. We would like to take this occasion to extend our sincere gratitude to all parties involved for their unflagging perseverance, steadfast commitment, and teamwork, which resulted in the victorious completion of this research project.

# TABLE OF CONTENTS

DECLARATION.....	i
ABSTRACT.....	ii
ACKNOWLEDGEMENT.....	iii
TABLE OF CONTENTS.....	iv
LIST OF TABLES.....	vii
LIST OF FIGURES .....	viii
LIST OF ABBREVIATIONS .....	ix
1. INTRODUCTION.....	1
1.1. Background.....	2
1.2. Literature Review .....	3
1.2.1. Android Related Work .....	4
2. RESEARCH GAP .....	6
3. RESEARCH PROBLEM .....	10
4. RESEARCH OBJECTIVES .....	12
4.1. Main Objective .....	12
4.2. Sub Objectives.....	13
4.2.1. Text-to-Speech (TTS) Synthesizer .....	13
5. METHODOLOGY .....	15
5.1. Requirement Gathering.....	15
5.1.1. Previous Research Analysis.....	15
5.1.2. Identifying Existing System.....	15
5.2. Feasibility Study.....	16
5.2.1. Technical Feasibility Study.....	16
5.2.2. Knowledge on Text-to-Speech .....	17
5.2.3. Knowledge on Machine Learning.....	18
5.2.4. Knowledge on API.....	18
5.2.5. Knowledge on Python .....	18
5.2.6. Knowledge on Flutter .....	19
5.3. Dataset Creation.....	19
5.4. System Overview Diagram.....	20

5.5.	Technologies to be Adopted .....	22
5.6.	Speech Synthesis.....	23
5.7.	Individual System Diagram.....	25
5.8.	Requirement Analysis .....	26
5.9.	System Analysis .....	26
5.10.	Techniques Used for Speech Synthesis .....	27
5.11.	Sinhala Language.....	29
5.11.1.	Sinhala Consonant .....	29
5.11.2.	Sinhala Vowel .....	29
5.11.3.	Sinhala Character Set.....	30
5.12.	Project Requirements .....	33
5.12.1.	Functional Requirements .....	33
5.12.2.	Non-Functional Requirements .....	33
5.12.3.	Hardware Requirements .....	35
5.13.	Commercialization of the Project .....	35
6.	DESIGN & IMPLEMENTATION .....	37
6.1.	Data Preprocessing (Text & Audio).....	37
6.2.	Encoder .....	38
6.3.	Decoder .....	39
6.4.	Converter .....	40
7.	RESULTS.....	42
7.1.	Building Sinhala Front-End.....	42
7.2.	The Training Sequence to Sequence Model .....	42
7.3.	Synthesizing the Training Model .....	44
7.4.	Backend Audiobook Create.....	45
7.5.	Subjective Evaluation .....	46
8.	DISCUSSIONS .....	49
9.	CONCLUSION .....	50
10.	REFERENCES.....	51
11.	APPENDICES .....	53
11.1.	Frontend Design .....	53
11.1.1.	Splash Screen.....	53
11.1.2.	Home Screen .....	54

11.1.3.	Book Reading .....	55
11.1.4.	Image Captioning.....	56
11.1.5.	Voice Navigation & Object Detection.....	57
11.1.6.	Application Working on Local Machine .....	58
11.2.	Backend Design .....	59
11.2.1.	Import Libraries & Modules .....	59
11.2.2.	Transformers TTS Library Models.....	59
11.2.3.	Create Dataset for Sinhala Text-to-Speech (TTS).....	59
11.2.4.	Load Dataset.....	60
11.2.5.	Accessing the Tokenizer from the Processor.....	60
11.2.6.	Text Processing & Building a Vocabulary.....	60
11.2.7.	Generate Speaker Embedding from Audio Waveforms .....	61
11.2.8.	Prepare Dataset .....	61
11.2.9.	Mapping Function to the Dataset .....	61
11.2.10.	Data Collator .....	62
11.2.11.	Transformers Library is Used to Sequence to Sequence Model .....	62
11.2.12.	Text to Speech Model Training (Sequence Model).....	62



## LIST OF TABLES

TABLE 1: COMPARISON BETWEEN EXISTING SYSTEMS .....	6
TABLE 2: PREPARING AUDIOS DATASETS .....	37
TABLE 3: TRAINING MINIMUM DATA SET OF SINHALA DATA.....	47

## LIST OF FIGURES

FIGURE 1: ANDROID AND IOS OS MARKET IN SRI LANKA .....	4
FIGURE 2: ANDROID APPLICATION OVERVIEW DIAGRAM .....	5
FIGURE 3: TEXT-TO-SPEECH AUDIO-BOOK ARCHITECTURE .....	9
FIGURE 4: GENERAL FUNCTIONAL DIAGRAM FOR TTS SYNTHESIZER.....	13
FIGURE 5: TEXT TO SPEECH DATASET COLLECTION.....	19
FIGURE 6: PROJECT OVERALL SYSTEM DIAGRAM.....	20
FIGURE 7: TEXT-TO-SPEECH SYNTHESIS SYSTEM ARCHITECTURE .....	24
FIGURE 8: INDIVIDUAL SYSTEM DIAGRAM.....	25
FIGURE 9: TEXT-TO-SPEECH WAVE GENERATION .....	28
FIGURE 10: SPOKEN SINHALA CONSONANT CLASSIFICATION.....	29
FIGURE 11: SPOKEN SINHALA VOWEL CLASSIFICATION .....	30
FIGURE 12: SINHALA LANGUAGE 20 VOWELS TABLE.....	31
FIGURE 13: SINHALA LANGUAGE CONSONANTS TABLE .....	32
FIGURE 14: ENCODER THE PREPROCESSOR TEXT IN SINHALA TTS .....	38
FIGURE 15: DECODER THE AUDIOS AND TEXT IN SINHALA TTS.....	39
FIGURE 16: ARCHITECTURE OF THE SYSTEM .....	41
FIGURE 17: TRAIN SEQUENCE TO SEQUENCE SINHALA TEXT-TO-SPEECH MODEL .....	43
FIGURE 18: A TYPICAL ARCHITECTURE OF FORMANT SYNTHESIZER.....	44
FIGURE 19: CREATE SINHALA AUDIO BOOK.....	45
FIGURE 20: AUDIO BOOK STORE AS MP3 FILES .....	46
FIGURE 21: TENSOR BOARD RESULT OF SINHALA WAVE OUTPUT .....	48
FIGURE 22: SPLASH SCREEN LOADING .....	53
FIGURE 23: HOME SCREEN.....	54
FIGURE 24: BOOK READING .....	55
FIGURE 25: IMAGE CAPTIONING .....	56
FIGURE 26: VOICE NAVIGATION & OBJECT DETECTION.....	57
FIGURE 27: APPLICATION WORKING ON LOCAL MACHINE .....	58
FIGURE 28: IMPORT LIBRARIES AND MODULES.....	59
FIGURE 29: TEXT-TO-SPEECH TRANSFORMERS LIBRARY MODELS .....	59
FIGURE 30: MAPPING DATASETS .....	59
FIGURE 31: LOAD DATASETS .....	60
FIGURE 32: ACCESSING THE TOKENIZER .....	60
FIGURE 33: SINHALA TEXT PRE-PROCESS.....	60
FIGURE 34: CREATE SPEAKER EMBEDDING FROM THE AUDIO WAVS .....	61
FIGURE 35: PREPARE DATASETS .....	61
FIGURE 36: MAPPING DATASETS .....	61
FIGURE 37: DATA COLLECTOR CLASS .....	62
FIGURE 39: KEYS AND SHAPES OF BATCH SIZE.....	62
FIGURE 40: SEQ2SEQ MODEL TRAINING.....	62

## **LIST OF ABBREVIATIONS**

TTS – Text-to-Speech

OCR – Optical Character Recognition

NLP – Natural Language Processing

WHO - World Health Organization

CART – Classification and Regression Tree

DSP – Digital Signal Processing

AI – Artificial Intelligence

API – Application Programming Interface

CSS – Cascading Style Sheets

GUI – Graphical User Interface

HCI – Human Computer Interaction

ML – Machine Learning

NLU – Natural Language Understanding

OpenCV – Open-Source Computer Vision Library

APK – Android Package

CNN - Convolutional Neural Networks

# 1. INTRODUCTION

The primary objective of this project is to provide alternatives to visually impaired people living in Sri Lanka by creating software that allows them to easily read printed books and stationery in Sinhala. This mobile application mainly uses optical character recognition (OCR) technology and voice navigation incorporating text-to-speech features of the event synthesis framework. Utilizing the capabilities of the mobile camera, the application accurately captures the characters present on a page of a Sinhala book and distinguishes it using OCR technology. This enables visually impaired people to capture text from physical documents and convert it into accessible digital formats. The extracted text is then made clearly audible to the visually impaired user via text-to-speech.

In addition, while navigating the app, the visually impaired person is provided with Voice Navigation support, which is necessary to give him the instructions and other commands he needs and to identify the objects in the surrounding room and thereby guide the visually impaired user. Image recognition and description algorithms are used to clearly describe the pictures in Sinhala to the visually impaired person while reading the story books of visually impaired children through the mobile phone application. This helps visually impaired children to understand the visual content of the picture and further improve their reading skills by bringing them closer to books. For those with visual impairments, our platform offers features that allow users to adjust the reading speed and choose between a male or female voice. Additionally, users can use commands to read each page and move on to the next page when finished. Overall, this software aims to improve the reading experience and skills of visually impaired individuals in Sri Lanka.

The quality of a Text-to-Speech (TTS) system depends on its ability to imitate human speech and ensure clear understanding. The absence of natural expressions in TTS output has a substantial influence on application usability. This emphasizes a key issue in TTS development for creating a synthesized speech that closely matches the human voice from the text. TTS technology's major goal is to recreate the complete range of human speech, including different speech patterns, subtleties, and intonations, while reducing the mechanical or robotic quality of the output voice.

The Sinhala language, the mother tongue of most Sri Lankans, is a crucial area for TTS development due to its complexities and nuances. Despite the large number of Sinhala speakers in Sri Lanka, there is a need for research on Sinhala voice recognition. The complexities of the Sinhala language make it difficult for computers to understand and reproduce it. Currently, there is little progress in developing TTS systems for the Sinhala language. However, this is a key research frontier that must be explored. An efficient TTS system for Sinhala would bridge the gap between human language skills and machine-generated speech, improving user experiences and bridging the gap between human language skills and machine-generated speech. There have been only a few attempts made to develop a Sinhala language TTS. This is

still a major research area that requires investigation, which is one of the key motivations for this research.

### **1.1. Background**

There are many people who have vision problems, and all too frequently they go untreated. Globally, more than two billion people worldwide [1] suffer from some form of vision problems, and of these, a minimum of one billion individuals have a condition that might have been avoided or is still unaddressed.

In the 21st century, knowledge has come to be seen as a necessity for a successful existence. Reading becomes the primary way to learn new things. The difficulty for those who are blind or visually impaired is that there are many sources of knowledge that are not available in braille format, despite their eagerness to learn more. Text-to-speech and computer-assisted braille systems are just two of the technological remedies that have been created in response to this [2]. We anticipate that this would boost the visually challenged community in Sri Lanka's delight in reading Sinhala novels.

Text-to-speech systems and computer-assisted braille systems are just two of the technological remedies that have been created in response to this. Most of these systems are expensive and broadly accessible only in industrialized nations. Additionally, most text-to-speech tools are for the English tongue. However, these amenities are not available to Sri Lankans whose native language is Sinhala and whose level of English literacy is poor. As a result, we require such a mechanism.

Mobile-based technologies are quickly getting acceptance across the globe in the era of modern technological advancements. Even though Sri Lanka is still a developing country, most of its citizens own a smartphone. The international trend also reveals that Android devices predominate the mobile phone market. Therefore, the most effective strategy to handle the problems encountered by visually disabled people in Sri Lanka would be a smartphone solution built on Android.

In conclusion, Android-based mobile book readers offer several benefits to Sri Lankans who are blind. Programmers can create specific apps with compatibility with the Sinhala language and accessibility features because of the open-source platform's versatility. Making mobile book readers for Android is a common option due to the platform's popularity and the accessibility of developer tools. Android-based mobile book readers can reach a larger population of people [3] who are blind because Sri Lanka has a high percentage of Android users.

## 1.2. Literature Review

Vision is crucial for our daily lives, enabling us to learn, walk, read, participate in school, and work. Vision impairment occurs when an eye condition affects the visual system and its functions. At least 2.2 billion individuals worldwide [1] suffer from near- or farsightedness, and in approximately half of these cases, the problem might have been avoided or is still unresolved. Everyone, if they live long enough, will experience at least one eye condition in their lifetime.

Text-to-speech is one of the assertive technologies that help a variety of impaired people improve their interaction with the real world. TTS apps' primary goal is to read digital text into speech by using a voice that is as human-like as possible while reading material that has been taken from any location. TTS may now be viewed as a technology that improves communication for persons with visual and voice impairments by utilizing online accessibility and human-computer interactions.

The simplest way to respond to this is to ask if we really want a TTS to sound like a person at all, or if a "robotic" sounding one would suffice and be far simpler to construct. Most listeners are so sensitive to unnaturalness that they will avoid using non-natural sounding systems no matter what further advantages are offered. Between "listening" and "hearing," there is a big difference. Hearing appears effortless, automatic, and nonselective, claim Rose & Dalton. To hear, we must be awakened [4]. While listening is purposeful, hearing is reactive. According to the explanation above, correct TTS is necessary to produce human performance, which has a more natural voice, and to ensure effective listening and understanding.

Most speech recognition systems that have been created so far use distinct voice synthesis techniques for English speech recognition rather than other languages. There are several TTS apps for the English language that are growing in popularity every day, and most of them are currently undergoing research and development to enhance the functionality of these systems. However, this cutting-edge TTS translation is only capable of converting text into a small number of languages. This subject is still being actively researched to develop technology that can convert text into voice that is as accurate as possible. On the other hand, there is currently no TTS program for the Sinhala language that might make use of the community's level of computer competence.

The same word might be pronounced differently by various people. Two locations can have two distinct sounds for the same letter combination [2]. Additionally, there are regional variations in language known as accents or dialects. In other words, speakers in the same area use the same word in various ways with varying tones and accents. Men also talk in a distinct tonality than women. Emotions may also be heard in sound at times. When it comes to human performance, which might capture all these differences, even the current TTS still struggles. Most text-to-speech technologies read a list of words aloud without any accent or tone since it is difficult to create a model that can capture the dialects and necessary tone to add to the text. Some programs employ different voices as the output, and these voices can talk in a variety of

the most popular accents and pronounce words in accordance with diverse linguistic conventions.

### 1.2.1. Android Related Work

In recent years, mobile devices have become a popular way of accessing information due to their convenience. In Sri Lanka, there is a growing demand for portable text readers that cater to the needs of those who are blind or visually impaired. This community is increasingly relying on mobile technologies to improve their accessibility and independence. The open-source operating system, Android, has gained popularity among developers globally [3] due to its accessibility, flexible nature, and rich development tools. We have decided to explore the development of Android-based mobile applications that cater to the specific needs of visually impaired individuals in Sri Lanka. Our focus is on those who speak Sinhala, and we aim to create customized applications for them.

The diagram below explains our choice to develop a mobile application for the Android operating system, stressing the multiple benefits it provides in solving the particular issues encountered by visually impaired users in Sri Lanka.

#### Mobile Operating System Market Share Sri Lanka

Oct 2011 - Feb 2023

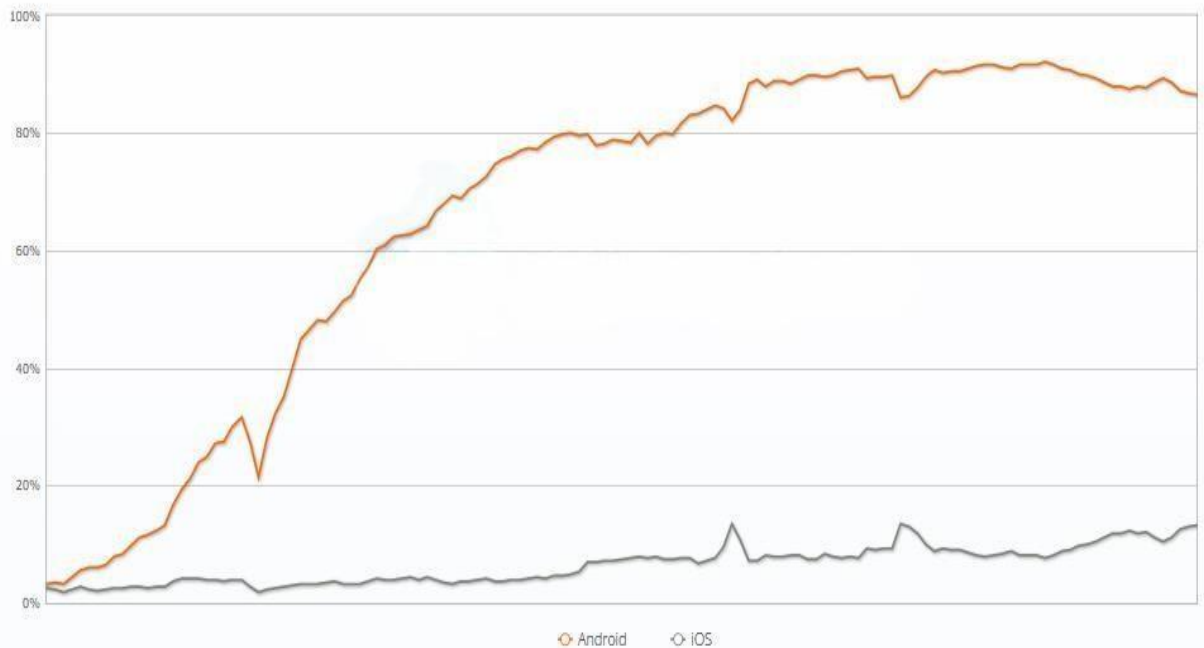


Figure 1: Android and IOS OS Market in Sri Lanka

The goal of this project is to create a technology that allows visually impaired people in Sri Lanka to access and read books and written materials in Sinhala. We produced a mobile

application for the Android operating system to illustrate this technique. This revolutionary program uses the camera on the mobile device to take documents, extract text from them, and then audibly deliver the material to the user in Sinhala [5]. The following figure depicts an overview of the suggested solution.

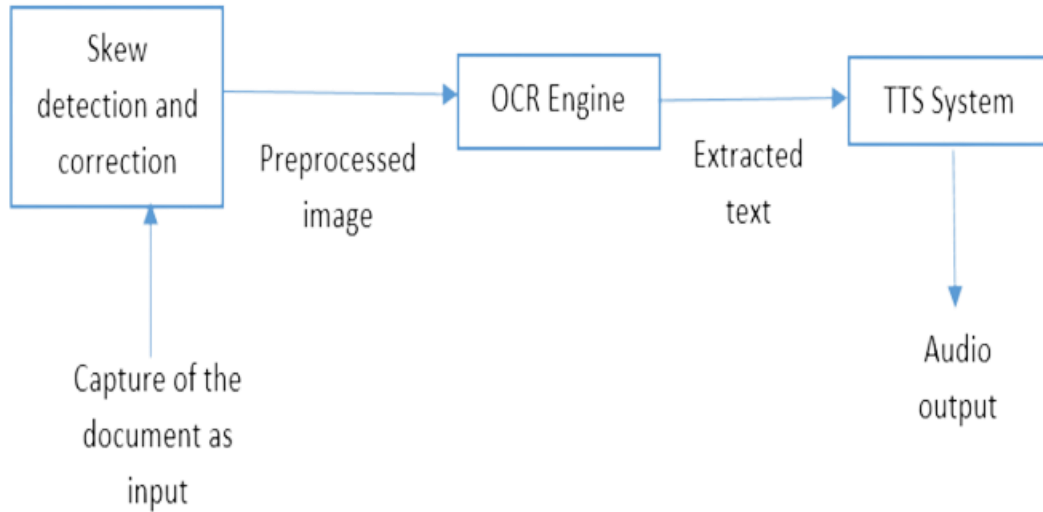


Figure 2: Android Application Overview Diagram

The popularity of Android-based applications among developers has soared, owing to the open-source nature of the platform. This openness allows developers to create specialized programs tailored to specific needs, which is a benefit for improving accessibility. Android-based apps may be made more user-friendly for those with visual impairments by seamlessly adding disability features such as text-to-speech capabilities.

Android-based applications provide a high level of customization, allowing developers to design bespoke solutions that meet the unique needs of visually impaired people in countries like Sri Lanka [3]. As an example, developers may create apps that support the Sinhala language while also incorporating text-to-speech functions, considerably improving the entire user experience.

Furthermore, Android's thriving developer community provides a plethora of tools and resources to help programmers design apps that are not only functional but also highly accessible to persons with vision impairments. These important resources enable developers to create mobile book devices that support the Sinhala language and a variety of accessibility features. Notable among these resources are publicly accessible frameworks and tools, which increase the platform's accessibility and adaptability.



## 2. RESEARCH GAP

There are several mobile applications that provide Text-to-Speech (TTS) functionality, according to the published literature. To turn written text into spoken words, researchers used a variety of approaches, including image processing and machine learning. However, when addressing the unique demands of the Sri Lankan population, a crucial gap appears. There are currently no specific mobile applications created with Sinhala TTS capabilities. Individuals and communities that rely on such services in their everyday lives face significant limitations because of this deficit. Despite the quantity of TTS apps accessible, the possibilities for Sinhala are remarkably limited, with only a few such apps included in the table below. It is critical to fill this research gap to improve accessibility and usefulness for the Sinhala speaking population.

TABLE 1: COMPARISON BETWEEN EXISTING SYSTEMS

Application Reference	Research A	Research B	Research C	Research D	Proposed System
High accurate Sinhala TTS conversion system	✗	✗	✓	✓	✓
OCR Text transferred to a TTS synthesizer in real time	✗	✓	✓	✗	✓
Adjust the Book reading Speed	✓	✓	✗	✗	✓
Support for Sinhala Language	✓	✗	✗	✓	✓
Android Mobile Application	✗	✗	✗	✗	✓
Get Audio Records in Each Pages and Create Sinhala Audio Book	✗	✗	✗	✗	✓

Natural language processing (NLP) is a critical component of the text-to-speech (TTS) generating process known as the "front end." The basic goal of this NLP function is to create a symbolic representation of the text under consideration [6]. This representation is a complex synthesis of various aspects, such as the phonetic transcription of the words, the desired tone, and the desired rhythm or prosody.

The NLP module effectively sets the stage by interpreting and arranging the language input, allowing for a more in-depth comprehension of the textual material. It examines intricacies in the text, such as sentence structure, punctuation, and contextual signals, to generate a rich and relevant picture. After completing the NLP module, we move on to the digital signal processing (DSP) module, which is also referred to as the "backend." Its purpose is to convert the symbolic representation created by NLP into an audible experience, such as a synthetic voice [7]. DSP uses advanced algorithms and methods to manipulate the visual data received from NLP and produce audio output that sounds natural and logical.

In essence, NLP, and DSP work smoothly together in the TTS system to bridge the gap between written text and spoken language, with NLP leading the way by interpreting and encoding the textual essence and DSP bringing it to life as audible speech. They are the dynamic pair driving current text-to-speech technology's astounding powers.

The creation of a highly exact Sinhala Text-to-Speech (TTS) conversion system is a critical scientific activity, particularly in the field of mobile book readers meant for those with visual impairments. Sinhala, Sri Lanka's native language, emphasizes the crucial need for a reliable TTS system capable of correctly transmuting written text into spoken Sinhala. This specialized field of research is dedicated to developing sophisticated models and algorithms capable of accurately converting Sinhala text into voice, including not only correct pronunciation but also proper emphasis and intonation.

The relevance of this research extends far beyond academic endeavor; it has the potential to considerably improve the accessibility of written resources in the Sinhala language for persons who are blind or visually impaired. A high-accuracy TTS conversion system can be a game changer, making books, papers, and other textual materials more accessible. It enables those with visual impairments to participate in Sinhala's rich literary and informational legacy, enabling greater inclusion and knowledge exchange within the Sri Lankan community and beyond. In summary, the quest for an accurate Sinhala TTS system is a driver for improving the quality of life and prospects for a major segment of the people.

The real-time transfer of OCR-processed text to a TTS synthesizer is an important field of research for mobile book users, particularly those with visual impairments. Using the power of Optical Character Recognition (OCR) technology, this method extracts text from photographs and converts it into digital text, which can subsequently be read aloud by TTS systems [8]. This study's main goal is to create an efficient and highly accurate OCR system capable of extracting text from photos of various qualities.

Essentially, the OCR technology acts as a link between visual material and spoken words. Once the OCR system has identified and transcribed the text, the recovered material is quickly and

in real time delivered to a TTS synthesizer. Individuals who are blind or visually impaired can access written content and have it instantaneously turned into spoken words thanks to this seamless integration. This topic of research is extremely important since it allows these people to access and engage with a wide range of textual resources, fostering more inclusion, independence, and accessibility in their daily lives.

The ability to alter the reading rate of books stands out as a key feature of mobile book readers, which are specifically developed to meet the needs of people who are blind or visually impaired. The fundamental goal of this research is to provide a user-friendly interface for the Text-to-Speech (TTS) system that allows users to pick their preferred reading pace. Recognizing that different users may have different reading needs; this function allows them to modify the reading pace based on their own preferences.

Algorithms are rigorously devised and applied in this field of research to allow for the smooth modification of the TTS system's reading speed while retaining the highest accuracy in voice output. This innovation guarantees that users may tailor their reading experience to their specific preferences and needs while maintaining the clarity and precision of the synthesized speech.

The design of an Android mobile application that includes Sinhala language support is a critical component in the development [9] of mobile book readers for people with visual impairments. The fundamental goal of this research is to create an application that interacts perfectly with Android devices, allowing for a simple download and installation process. The dedication to building an application with a user-friendly interface, guaranteeing that visually impaired users may browse it with ease, is central to our research. This user-centric approach emphasizes accessibility, with the goal of enabling users to engage with the app's capabilities with ease.

Furthermore, the application's primary operations are designed to provide extensive Sinhala language support, including Text-to-Speech (TTS), Optical Character Recognition (OCR), and book-reading capabilities. This comprehensive approach attempts to establish a solid foundation that allows those with visual impairments simple and rapid access to books and other written materials in Sinhala.

Creating a Sinhala audiobook is a critical stage in the process. A customized pattern is used at this step to painstakingly record the audio information of each page of the book. This entails extracting text from pages and turning it into spoken Sinhala using TTS technology or, in some circumstances, human narrators to assure accuracy and naturalness. The goal is to create high-quality audio recordings of the complete book, ensuring that the text of each page is correctly translated into clear and comprehensible Sinhala audio. This phase is critical to offer consumers, particularly those with visual impairments, a smooth and immersive hearing experience while accessing the book's information.

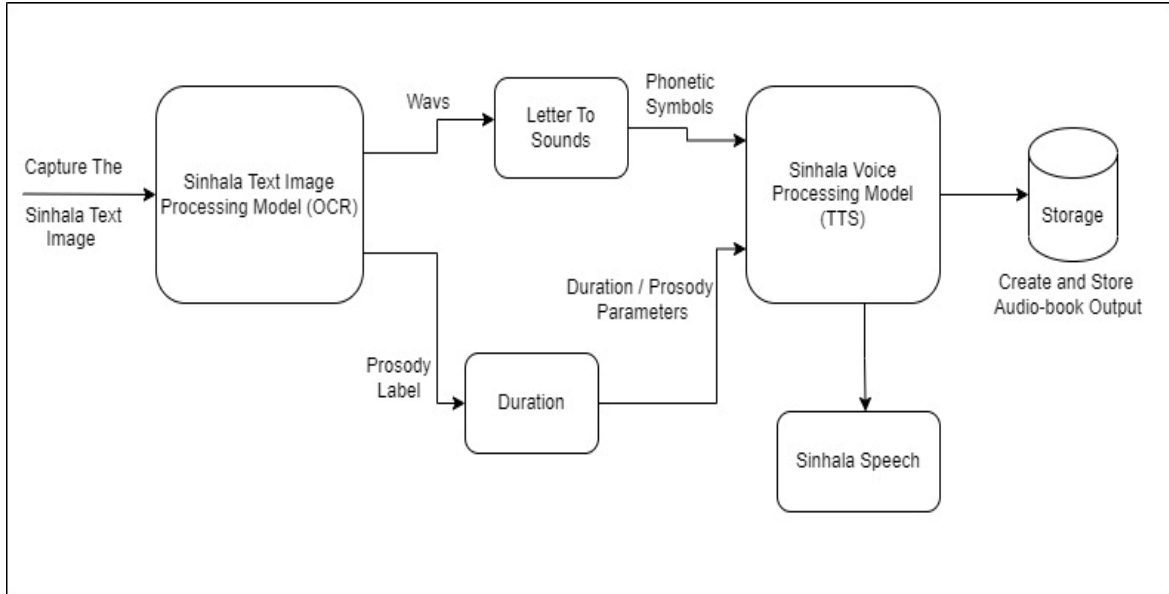


Figure 3: Text-to-Speech Audio-book Architecture

After obtaining the audio files for each page, the following step is to combine and synchronize them to generate a full Sinhala audiobook. The separate audio segments corresponding to each page are meticulously sorted and blended throughout this procedure to provide a seamless and cohesive flow of the book's content. To improve the user experience, any relevant metadata, such as chapter divisions or book information, is also supplied. The resulting Sinhala audiobook is a valuable resource, providing an accessible and engaging way for people to enjoy literature, educational materials, or any written content in audio format, promoting inclusivity and accessibility for a diverse range of readers, including those with visual impairments.

In summary, the goal of this study is to close the accessibility gap by creating a sophisticated Android application adapted to the unique needs of the visually impaired population, encouraging inclusion, and enhancing the lives of people with visual impairments. Furthermore, the incorporation of real-time OCR-to-TTS technology represents a significant advancement in assistive technology [8], allowing people with visual impairments to access printed content in spoken form, improving information gathering capabilities, and overall quality of life. By allowing users to set their own reading pace, this technology gives them more power and autonomy while engaging with textual information, thereby improving their reading experience.

### 3. RESEARCH PROBLEM

The World Health Organization (WHO) estimates that there are presently over 290 million visually impaired people worldwide [1], with about 40 million of them completely blind. The mere act of reading books might be extremely difficult for some people, owing to their inability to obtain written materials in a comfortable manner. Despite significant advances in assistive technology, such as text-to-speech software and Braille displays, many books continue to be out of reach for the visually impaired people. This limited access not only limits their capacity to broaden their knowledge, but also creates considerable impediments to participating in literary encounters, upgrading their education, and seeking professional prospects.

The situation necessitates further efforts to make literature more inclusive. We can considerably improve the accessibility of information for visually impaired people by making printed materials available in accessible formats such as audiobooks or digital texts compatible with screen readers. Furthermore, encouraging more works to be transcribed into Braille and other tactile forms might offer up new paths for inquiry and study. We must continue to invent and invest in technology that overcomes the accessibility divide, allowing blind people to fully connect with the literary world, further their education, and realize their professional potential. By doing so, we may contribute to a more inclusive and equal society for all people, regardless of their visual skills.

The situation necessitates further efforts to make literature more inclusive. We can considerably improve the accessibility of information for visually impaired people by making printed materials available in accessible formats such as audiobooks or digital texts compatible with screen readers. Furthermore, encouraging more works to be transcribed into Braille and other tactile forms might offer up new paths for inquiry and study. We must continue to invent and invest in technology that overcomes the accessibility divide, allowing blind people to fully connect with the literary world, further their education, and realize their professional potential. By doing so, we may contribute to a more inclusive and equal society for all people, regardless of their visual skills.

The visually impaired face a shortage of audiobooks and Braille materials. Despite the increasing availability of audiobooks, their numbers are still small compared to the large library of written literature. Furthermore, the process of turning written literature into Braille is not only time-consuming but also costly, making access to Braille resources much more difficult. As a result, blind people may be unable to access the most recent bestsellers or frequently utilized instructional materials, worsening their reading impairments.

Addressing this issue would need coordinated efforts to enhance audiobook output and extend Braille libraries. Collaborations between publishers, technological firms, and organizations that serve the visually impaired can assist ensure that a wider choice of literary and educational resources are made available. By eliminating these impediments, we can provide visually impaired persons with equitable access to modern literature and critical educational materials.

This not only enhances their lives, but it also helps to foster a more inclusive and fair reading culture.

Finally, the major barrier that blind people face in their quest to read books is a significant lack of access to written materials. Despite significant advances in assistive technology, numerous challenges remain, such as the prohibitive costs of specialized hardware and software, the scarcity of audiobooks and Braille resources, and the difficulty of replicating a reading experience comparable to that of traditional printed books.

To overcome these obstacles, a concerted and determined effort is necessary to improve book accessibility for blind readers. It is critical that we work to ensure that they have equal access to reading materials and educational opportunities as sighted people. We may strive toward a more inclusive society where blindness does not limit one's access to the great world of literature and knowledge by promoting affordability, extending the availability of audiobooks and Braille materials, and continually developing assistive technology. By doing so, we may enable visually impaired people to reach their full potential and completely engage in the realms of learning and reading.

## 4. RESEARCH OBJECTIVES

### 4.1. Main Objective

A Sinhala book reader for the visually impaired is a unique software tool that has been painstakingly designed to empower those with visual impairments by opening the world of reading and information to them. This complete solution cleverly combines a plethora of cutting-edge technology, resulting in a seamless and user-friendly reading experience that surpasses the constraints of visual impairment.

In Sri Lanka, where most of the population speaks Sinhala as their first language, the development of mobile book readers capable of reading Sinhala texts for the blind or visually impaired is critical. This technology is a revolutionary force, allowing these people to access textual material with unprecedented ease and freedom, thereby improving their overall quality of life.

The rising popularity of mobile book readers that use Text-to-Speech (TTS) technology emphasizes its importance. TTS technology enables persons who are blind or visually handicapped to receive information more easily through their sense of hearing by translating written text into spoken speech. Our invention is a critical step toward closing the accessibility gap and promoting diversity in our community.

The application we hope to create has two key components: a Text-to-Speech (TTS) synthesizer and powerful Optical Character Recognition (OCR) technology created particularly for detecting common Sinhala characters [10]. This revolutionary system uses OCR to smoothly convert printed text from physical books to digital format. The TTS synthesizer then takes control, converting the text into clear and genuine Sinhala voice. This dynamic mix means that visually impaired people may easily follow and grasp the text, boosting their access to literature and information greatly.

In addition to the text-to-speech synthesizer, our gadget includes a very useful feature: auditory guidance. This feature has two functions: it supports users in properly navigating the app and it offers real-time feedback on the distance to the book being read. By providing this multidimensional assistance, our gadget enables visually impaired users to not only easily find their place within the book, but also to keep a clear feeling of their reading progress. This deliberate integration improves the overall user experience while also encouraging greater freedom in the quest of literary inquiry.

## 4.2. Sub Objectives

### 4.2.1. Text-to-Speech (TTS) Synthesizer

Speech synthesis is the practice of reproducing human-like speech via mechanical means. The voice synthesizer, a complex machine that uses the computational power of computers to make sounds that closely mimic actual human speech, is at the center of this project. material-to-Speech (TTS) is a popular speech synthesis technique that translates written language material into audible speech, providing a key link between written information and spoken communication. Meanwhile, other methods investigate other ways of encoding symbolic linguistics, increasing the range of strategies used to accomplish the astonishing feat of mechanical speech production. These improvements together contribute to the removal of communication and accessibility obstacles, therefore improving the lives of people all over the world.

Concatenation, a commonly used approach in speech synthesis, is the expert combination of pre-recorded speech fragments painstakingly preserved in a database. This approach is the foundation for creating artificial speech that mimics human speech patterns. Text-to-Speech (TTS) is another strong mechanism for converting written text into spoken language, providing a seamless connection between textual information and audible communication [11]. Aside from these ways, many systems use a wide range of methodologies to translate symbolic representations of language into understandable speech.

Two basic speech synthesis methods, formant synthesis and concatenative synthesis, take center stage. To make synthetic speech, formant synthesis relies on mathematical modeling of vocal tract resonance frequencies. Concatenative synthesis, on the other hand, uses the fusing of carefully selected, pre-recorded speech fragments to produce synthetic speech that closely resembles the subtleties of actual human speech. These two technologies jointly propel speech synthesis advancement, defining the landscape of accessible and lifelike artificial speech creation.

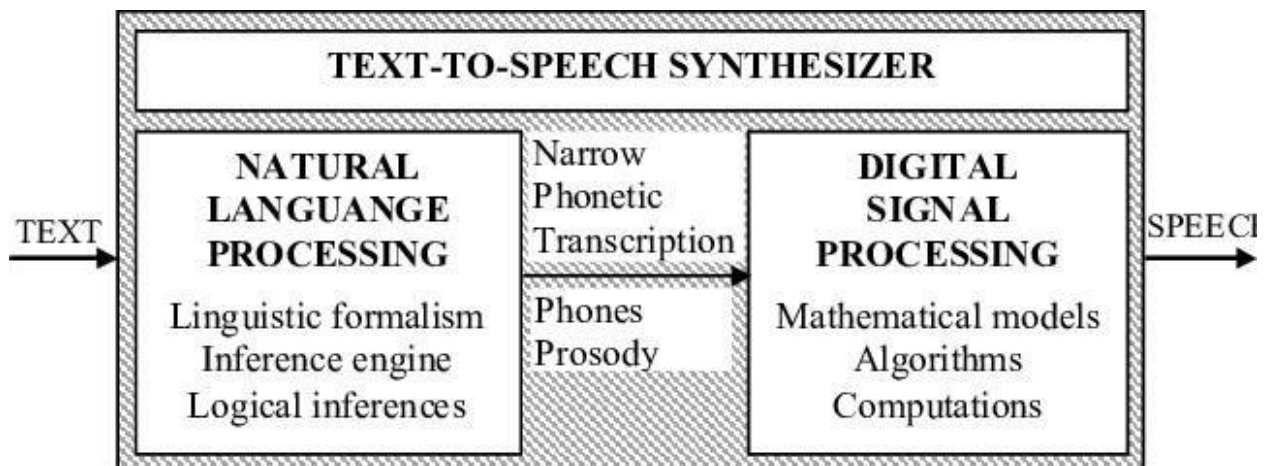


Figure 4: General Functional Diagram for TTS Synthesizer



TTS technology is built on two key components: the front end and the back end, both of which play critical roles in the synthesis process [11]. The front end is largely in charge of two critical functions: text normalization and pre-processing. The front end's responsibility during the pre-processing step is to recognize and segment individual words within the input text and then transform these words into their corresponding phonetic transcriptions [6]. This stage is critical in dividing the text into phonetic units, which serve as the foundation for producing cohesive and natural-sounding speech.

Concurrently, the front end's text normalization process guarantees that the input text follows a specified format. This procedure harmonizes spelling, punctuation, and formatting differences, assuring uniformity and clarity throughout the synthesis process. The front end of TTS technology establishes the groundwork for accurate and fluent speech synthesis, eventually improving the quality of the synthesized speech output by solving three important tasks.

The front end of the TTS process conducts numerous critical steps to convert the incoming text into a format appropriate for speech synthesis. One important part is dividing the text into prosodic units, which might comprise phrases, sentences, and clauses. These prosodic units are then examined to obtain the relevant prosody information, which includes elements like pitch, rhythm, and intonation.

The front end generates a symbolic language representation of the input text after identifying and tagging the prosodic units with relevant prosody information [12]. This representation was created by combining prosody data with phonetic transcriptions. The front end guarantees that the synthesized speech not only delivers the words but also captures the natural rhythm, stress, and intonation patterns seen in spoken language by combining these features. This careful approach eventually leads to the creation of high-quality, expressive synthetic speech.

The front ends precisely designed symbolic language representation is critical to the TTS process, acting as the foundation for the succeeding phases, notably at the back end. The transition of this symbolic representation into audible sound, known as synthesis, occurs at the back end. Advanced algorithms and models analyze the symbolic language representation during synthesis, considering factors like phonetic transcriptions, prosody information, and other linguistic signals. These components are analyzed and turned into voice waveforms that closely resemble actual human speech patterns. This sophisticated synthesis process guarantees that the final product closely matches spoken English, giving the listener with a vivid and cohesive aural experience. In essence, it is this synthesis stage that brings the textual representation to life, providing a comprehensible and expressive synthesized sound.

## 5. METHODOLOGY

### 5.1. Requirement Gathering

The requirement collecting procedure for our Sinhala mobile blind book reader application included a thorough evaluation of existing research spanning several years. This included finding and extensively assessing existing systems in the sector, as well as doing considerable web research. The identification of existing or comparable systems on the market was an important part of this study. Furthermore, we tried to acquire insights into the development of comparable systems by investigating the processes and procedures used in their development. This methodical approach to demand collecting provided a solid basis for improving our text-to-speech (TTS) capability, ensuring that our application fits the special needs of our visually impaired Sinhala users.

#### 5.1.1. Previous Research Analysis

The preceding study analysis was a careful procedure that largely focused on an exhaustive assessment of research paper publications concentrating on key areas critical to our Sinhala mobile blind book reader application. Text-to-Speech models, and Text-to-Speech approaches were among the topics covered. We came up a wide range of books that probed into the junction of TTS and dialogue management among this vast body of research.

The key goal of our prior study analysis was to identify the tactics and instruments used in the creation of current systems. This investigation sought to offer insight on the strategies and approaches used in the field. Furthermore, it provided a tool for identifying the difficulties and drawbacks that prior research had experienced. We acquired great insights by completing this extensive investigation, which will be useful in improving the capabilities of our Sinhala TTS system within our blind book reader application.

#### 5.1.2. Identifying Existing System

We did a thorough examination of existing systems and technologies in the sector in our effort to build Sinhala text-to-speech (TTS) capabilities for our Sinhala mobile blind book reader application. While there aren't as many Sinhala TTS systems as there are in English or other commonly spoken languages, we did find a few notable existing systems and tools that paved the path for our development work.

- **Google Text-to-Speech (Sinhala):** Google's TTS engine now supports Sinhala, making it one of the significant current systems for our consideration. Although it is

aimed for a broad audience, its capabilities give useful insights into natural language processing and Sinhala text pronunciation.

- **eSpeak (Sinhala):** Sinhala has been derived from eSpeak, an open-source voice synthesis program. While it is not as sophisticated as other commercial solutions, it provides insights into the technical components of Sinhala TTS and can be a useful resource.
- **Local Research efforts:** We also investigated local research efforts and academic programs that have focused on Sinhala TTS. These initiatives frequently provide innovative techniques to addressing language subtleties and cultural context, both of which are critical for our Sinhala mobile blind book reader application.
- **Commercial TTS Systems:** We looked at commercial TTS systems with multilingual support, some of which offer Sinhala. Although these systems do not explicitly cater to the blind or visually handicapped, they do provide insights into the development of high-quality, natural-sounding Sinhala TTS voices.

We hoped to obtain a full grasp of the problems and potential in designing a successful Sinhala TTS solution for our mobile blind book reader application by researching these current systems and technologies. This study served as the cornerstone for our attempts to develop a customized and accessible TTS experience for our visually impaired Sinhala-speaking users.

## **5.2. Feasibility Study**

### **5.2.1. Technical Feasibility Study**

It is critical that we do a technological feasibility analysis for our Sinhala text-to-speech (TTS) implementation within the Sinhala mobile blind book reader application. This research focuses on setting the necessary tools to develop a Voice-Enabled Intelligent Programming Assistant, as well as smoothly integrating and TTS features. During the requirement analysis portion of our research project, we focused heavily on determining the technological viability of this attempt.

Our team's goal in this critical component was to guarantee that members have the experience and resources needed to set up both conversation management and text-to-speech approaches efficiently. The effective integration of these technologies is critical to satisfying our Sinhala TTS solution's needs and aims.

Several major issues are included in the technical feasibility study:

1. **Text-to-Speech Integration:** Setting up the Sinhala TTS engine and confirming that it can transform written material into natural-sounding audio. This stage entailed assessing the capabilities of existing TTS systems and identifying any potential issues with Sinhala language support.
2. **Availability of Hardware, Software, and Language Resources:** We analyzed the availability of hardware, software, and language resources necessary for the development and implementation of our Sinhala TTS system. Access to computational resources, language data, and development tools are all part of this.
3. **Scalability and Performance:** Evaluating our system's capacity to handle a wide range of programming-related tasks and user interactions. We tested its functionality in a variety of settings to guarantee responsiveness and dependability.
4. **Testing and Validation:** Plan rigorous testing and validation methods to assure the TTS system's correctness, usefulness, and accessibility for our visually impaired users.
5. **User Feedback and Iteration:** Using user feedback to enhance our settings and guarantee that the system meets the unique demands of our target audience.

We hoped to lay the basis for a solid and efficient Sinhala TTS solution within our Sinhala mobile blind book reader application by completing a detailed technological feasibility analysis. This research directed our development efforts and ensured that we were well-prepared to face the technological hurdles of developing a voice-enabled programming aid for the visually impaired.

### 5.2.2. Knowledge on Text-to-Speech

To enable the effective implementation of Sinhala text-to-speech (TTS) inside our Sinhala mobile blind book reader application, the team member in charge of this component must be well-versed in TTS technology. This should be capable of setting up the Sinhala TTS system, including hardware and software configurations. Furthermore, to achieve the needed functionality and quality of the Sinhala TTS output, a good technical foundation in TTS implementation is required.

It is critical to be able to fine-tune and adjust the TTS system to the intricacies of the Sinhala language. This comprises understanding of Sinhala linguistic traits, phonetics, and pronunciation. With this underlying understanding and technical aptitude, the team member can play a critical role in providing accurate and natural-sounding speech feedback to improve the accessibility and usability of our Sinhala-speaking mobile blind book reader application.

### **5.2.3. Knowledge on Machine Learning**

A thorough grasp of machine learning is required in the context of our Sinhala mobile blind book reader application and the development of our Sinhala text-to-speech (TTS) solution. Machine learning is critical in training and fine-tuning our TTS models to transform Sinhala text into genuine, understandable voice. This understanding enables our team to use cutting-edge machine learning techniques to generate TTS voices that are not only linguistically exact but also sensitive to the distinctive nuances of the Sinhala language. Furthermore, our expertise in machine learning allows us to continuously improve our Sinhala TTS skills. We can adjust and refine our models over time to ensure that our visually impaired customers have an ever improving and personalized TTS experience when reading their favorite Sinhala books on our mobile blind book reader app. This skill in machine learning demonstrates our dedication to providing the Sinhala-speaking blind population with an inclusive and accessible reading experience.

### **5.2.4. Knowledge on API**

To smoothly connect our Sinhala text-to-speech (TTS) system and dialogue management units with the key components of our mobile blind book reader application, relevant team members must have a basic grasp of Python API development. Understanding the basic concepts of API functionality, connecting with cloud-deployed machine learning models, and skillfully building Python modules to promote seamless communication across system components are all part of this knowledge.

In this context, the team members in charge of this integration must be skilled in creating, managing, and leveraging APIs. This involves developing APIs that allow our conversation management and TTS units to work in tandem with other system components, guaranteeing a consistent and user-friendly experience for our Sinhala-speaking visually impaired users. Their ability to handle Python API development will be critical in ensuring smooth integration and optimal performance of our Sinhala TTS system within our mobile blind book reader application.

### **5.2.5. Knowledge on Python**

Proficiency in Python is required for our Sinhala text-to-speech (TTS) model training for the Sinhala mobile blind book reader application. Python is the major programming language used to develop and fine-tune our TTS models. We may construct and tweak Sinhala-specific TTS models by leveraging Python's rich libraries and frameworks, such as TensorFlow and PyTorch.

### 5.2.6. Knowledge on Flutter

We are committed to creating an intuitive and accessible user experience for our Sinhala mobile blind book reader application by using our expertise in Flutter for front-end application development. Our expertise in Flutter allows us to develop a smooth and user-friendly experience designed exclusively for our visually impaired Sinhala-speaking consumers, hence improving the accessibility and usefulness of our Sinhala TTS capabilities.

### 5.3. Dataset Creation

We've collected a large dataset of roughly 3,300 voice WAV files to help us improve the development of our Sinhala text-to-speech (TTS) technology within our mobile-based Sinhala book reader for visually impaired people. This dataset, obtained from a reliable GitHub repository, is a critical asset in our model training efforts. These voice recordings include a wide range of Sinhala language variances and subtleties, which are critical for developing a TTS system that is truly appealing to our target audience.

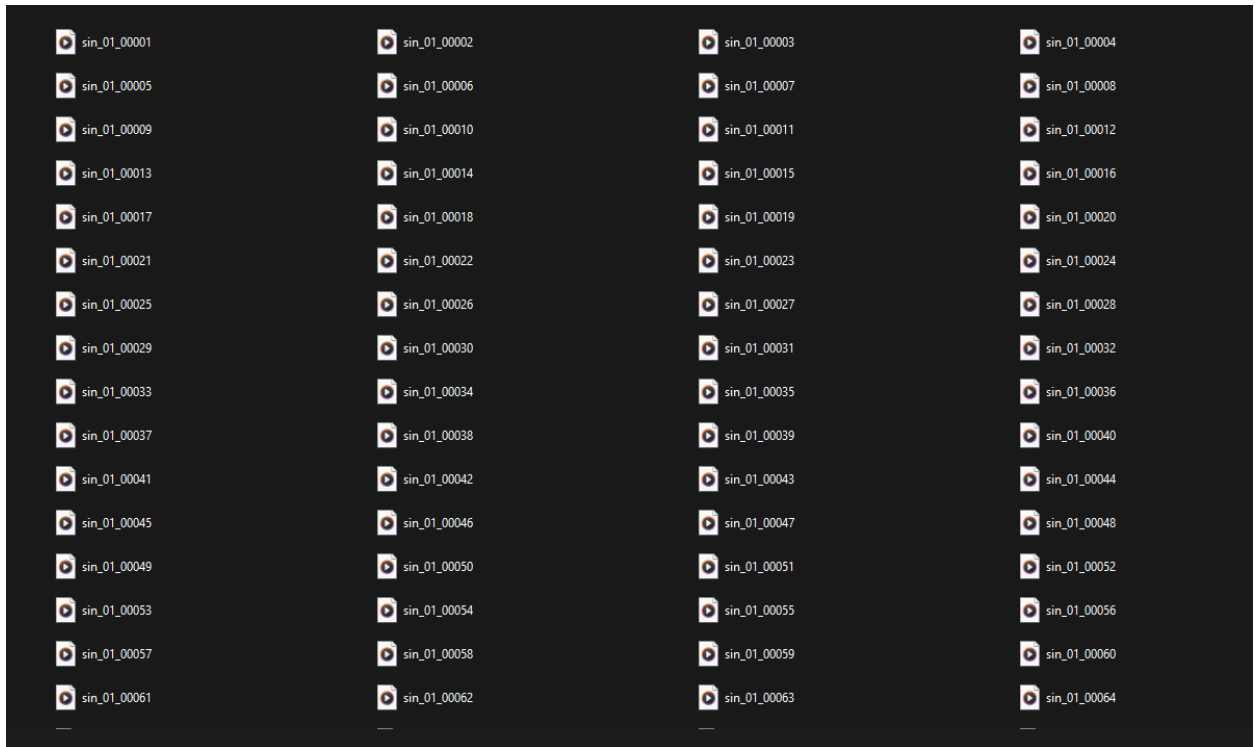


Figure 5: Text to Speech Dataset Collection

The use of this large dataset allows us to begin complete model training, allowing our TTS system to understand the nuances of Sinhala pronunciation, intonation, and rhythm. With this wealth of data at our disposal, we are well-positioned to create a sophisticated TTS model that not only improves the accessibility of our mobile-based book reader but also caters especially

to the linguistic demands of visually impaired Sinhala-speaking persons. We think that including these voice WAV files will considerably improve the quality and naturalness of the synthetic speech, thereby improving the user experience and fostering more inclusion for our visually impaired users.

#### 5.4. System Overview Diagram

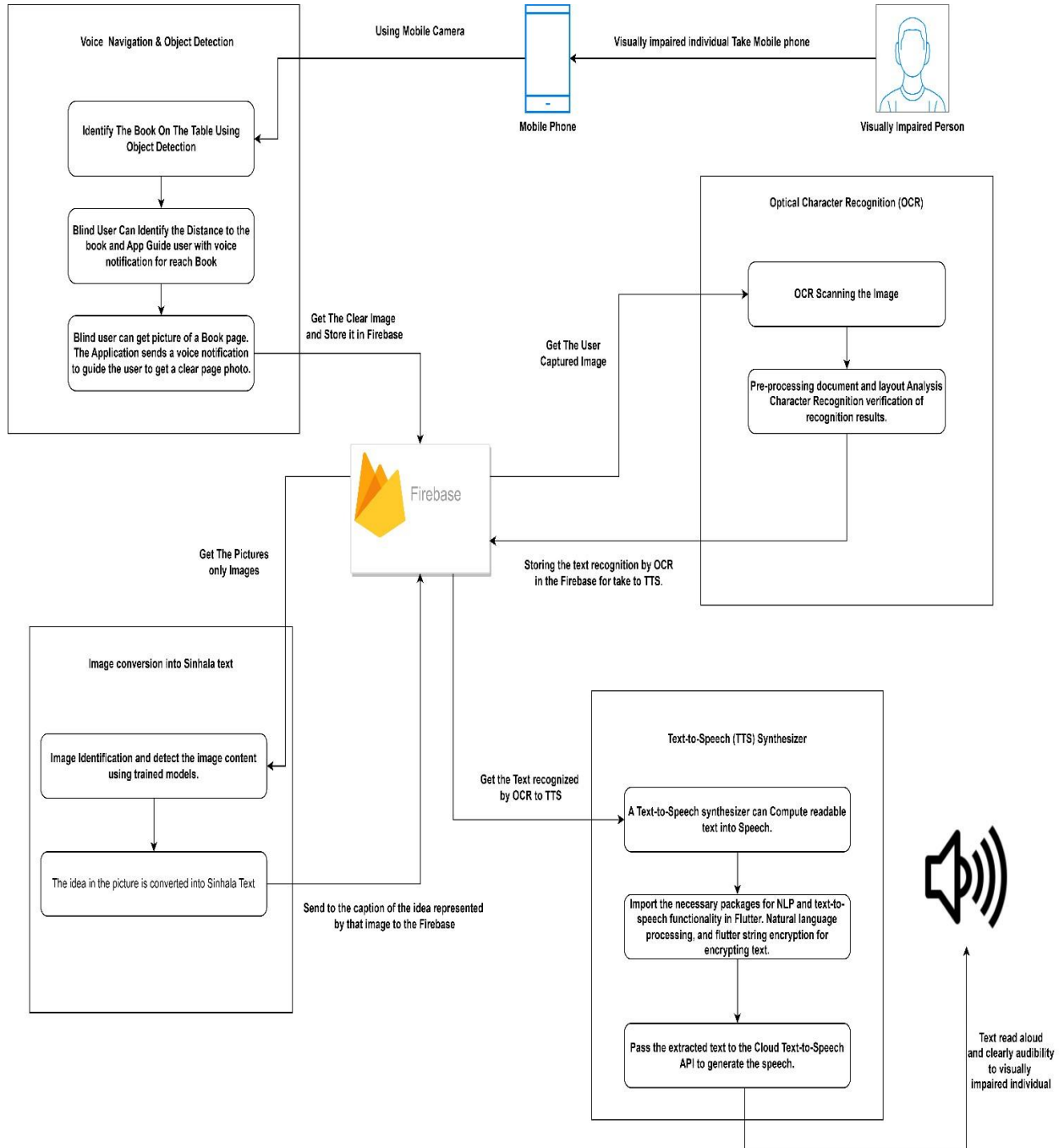


Figure 6: Project Overall System Diagram

Within the program, Audible advice is an essential companion, leading users effortlessly through its multifarious capabilities and providing clear and succinct aid whenever they find issues executing tasks. For example, when a user wants to find a book on a table, the app uses real-time image processing technologies. The software can recognize the user's hand and effectively navigate to the selected book by opening the camera through the app and directing it toward the table or desk.

This smart app's powers go beyond book discovery. It makes use of cutting-edge image processing technology to detect possible risks and harmful things in the immediate surroundings of blind people. The software audibly warns the user of the existence and distance of such items in real time, adding an important degree of protection. Furthermore, it computes the likelihood of an accident occurring, guaranteeing that the user is well-informed and may take proactive actions to avoid injury. Users may comfortably explore their surroundings with this fantastic program, enjoying a beautiful atmosphere while avoiding risks and dangerous components. It enables people with vision impairments to navigate around the world with greater independence and security, making their everyday life easier and safer.

The engine's ability to recognize Sinhala characters and create words is extremely impressive. It effortlessly analyzes and turns text into speech, allowing users to communicate more effectively. What distinguishes this engine is its ability to operate in the background, allowing users to effortlessly check the time even when the app is not actively in use. Voice commands are an easy method to begin the software, and the application is responsive. Upon activation, it immediately activates the camera, allowing for quick document scanning. It can detect and focus on the paper placed in front of the camera, providing users with real-time instruction until the document is properly aligned within the capture frame.

Audible notifications play a critical part in improving user experience by keeping people always informed. When taking a page from a book using the phone's camera, the software gives users explicit directions to help them achieve a successful outcome. Furthermore, it keeps the taken image in the device's storage for subsequent use. The program's devotion to excellence is one of its most notable features. It performs automated skew detection and correction [11] prior to delivering information to the OCR (Optical Character Recognition) system. This attention to detail guarantees that the OCR process produces accurate and dependable results.

The image detection program for blind pupils uses cutting-edge computer vision technologies to perform real-time item and scene recognition and description. This sophisticated application takes advantage of the capabilities of smartphones and tablets by using their cameras to gather photographs of the user's surroundings. To recognize and describe the objects therein and their associated qualities, these photos are subjected to several advanced image-processing techniques such as edge recognition, color analysis, and feature extraction.

What distinguishes this program is its use of machine learning models, which are critical in recognizing and categorizing objects inside photos. Object detection methods are used to identify the existence of objects, and once recognized, these models, which are frequently



based on resilient architectures such as convolutional neural networks (CNNs), go into action. These models run in real time, allowing for quick and precise object detection. They have been rigorously trained on large datasets of annotated photographs, allowing them to execute with outstanding accuracy and dependability.

The fundamental goal of utilizing Text-to-Speech (TTS) technology is to provide blind folks with the ability to easily access the textual content of Sinhala literature. This priceless technology allows individuals to interact vocally with the rich substance of Sinhala literature, considerably improving their reading experience. TTS technology does this by effortlessly transforming written Sinhala text into spoken words, employing a natural-sounding voice that allows visually impaired users [5] to have a deeper grasp of the information.

TTS technology uses complex computer algorithms to thoroughly evaluate the Sinhala text behind the scenes. These algorithms can produce accurate pronunciations, intonations, and rhythms for each word and phrase, resulting in a fluent and cohesive audio representation of the textual material. This transformational technology not only enables accessibility but also encourages inclusion, allowing those with visual disabilities to experience and explore the world of Sinhala literature.

This cutting-edge program combines computer vision techniques and machine learning skills to create a formidable solution for blind pupils. It allows individuals to engage with their environment in real time using object detection, navigate using Sinhala voice commands, and read Sinhala text using TTS technology. This complete tool not only promotes independence and accessibility, but it also provides a user-friendly experience, making it a significant asset for those with visual impairments looking for rapid document identification, scanning, and Sinhala text processing options.

## **5.5. Technologies to be Adopted**

Speech is the principal mode of human communication, employing a complex system that mixes words and names from large vocabularies in a syntactic framework. Each spoken word is made up of a small number of phonetically related vowel and consonant speech sound components [13]. Tens of thousands of different and mutually unintelligible linguistic systems originate from the global variety of human languages.

The choice of interface is critical to user success in the field of human-machine interaction. Voice user interfaces, powered by voice recognition and synthesizing technologies, have grown in popularity. Voice communication is the most basic means of human contact, involving spoken verbal expression. It is the most natural and effective way for people to exchange their expertise.

Voice processing is a discipline that studies the analysis and manipulation of spoken signals. It is widely used as a front-end component in a variety of language processing systems. The scope of speech processing includes a wide range of issues, such as:

**Speaker Identification:** This involves recognizing and distinguishing individual speakers based on their unique vocal characteristics. It has applications in security and authentication systems.

**Speech Identification:** Speech recognition systems are designed to recognize and transcribe spoken words into written form. This technology lies at the heart of voice assistants and transcribing services.

**Speech Coding:** Speech coding is concerned with efficiently compressing and encoding voice signals for storage and transmission, which is useful in telecommunications and multimedia applications.

**Speech Enrichment:** Speech enrichment refers to techniques for improving the quality and intelligibility of speech signals, which benefit applications such as hearing aids and audio restoration.

**Speech Compression:** This pertains to the reduction of data size in speech signals while maintaining perceptual quality. It's crucial for efficient data transmission and storage.

**Speech Synthesis:** Speech synthesis involves generating artificial speech from text or symbolic representations, enabling applications like text-to-speech (TTS) systems and voice assistants.

In essence, speech processing is a multidisciplinary discipline that bridges the gap between human communication and technology, improving interactions and enabling a diverse range of applications in a variety of domains.

## 5.6. Speech Synthesis

Text-to-speech systems must first convert input text into lexical or phonological interpretations before generating the matching sounds associated with these representations. Given that the input is often in plain text format, linguistic models must be enhanced with insights into elements such as pitch and tempo to guarantee that the synthesized speech reflects natural patterns. A Natural Language Processing (NLP) module, which is a fundamental component of most speech processors, manages this fine-tuning of prosody and linguistic subtleties.

The NLP module functions as a text analysis behemoth, comprehending the complexities of the supplied text [6]. It considers the semantic and syntactic context, as well as understanding of tone, rhythm, and language structures. The NLP module, armed with this extensive information, enables the synthesis process to sound more realistic and human-like.

Following that, an Artificial Sound Generation phase takes center stage, which is powered by a Digital Signal Processing (DSP) module. This module makes use of the NLP modules precisely obtained prosodic and linguistic data. It converts this data into artificial sound by painstakingly constructing speech waveforms that closely resemble the intricacies of human

speech. Text-to-speech systems bring to life the translation of plain text into expressive and understandable synthetic speech by integrating these crucial components, a feat accomplished through the harmonic interplay of language analysis and powerful signal processing.

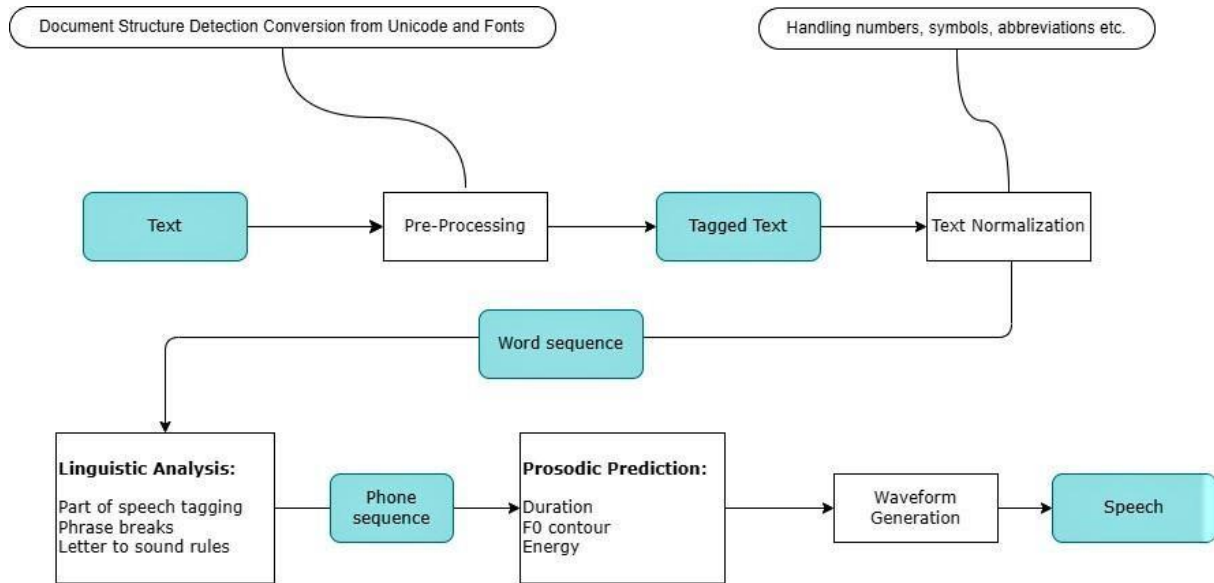


Figure 7: Text-to-Speech Synthesis System Architecture

There are two types of text-to-speech (TTS) systems: restricted domain TTS and generic TTS.

- **Restricted Domain TTS:** The first type, Restricted Domain TTS, is intended for a specific, predetermined purpose. As a result, the value of this sort of TTS is restricted to its intended purpose. It works by utilizing a preset collection of words and phrases designed specifically for voice synthesis. Restricted Domain TTS can be found in applications such as talking dictionaries, travel information systems, talking clocks, and other similar situations. In many cases, the TTS system is fine-tuned to meet the unique demands and vocabulary of the targeted application.
- **Generic TTS:** The second category, Generic TTS, is designed to accommodate a wider range of information. This adaptable TTS system can read a broad range of text formats, including internet news, emails, articles, and more. It can produce synthetic voices for any words or phrases, making it a good choice for general-purpose applications. Generic TTS systems enable the conversion of a wide range of written information into speech, meeting a wide range of user demands and preferences.

Restricted Domain TTS excels in specialized areas, but Generic TTS offers adaptability for a wide variety of applications and content kinds.

### 5.7. Individual System Diagram

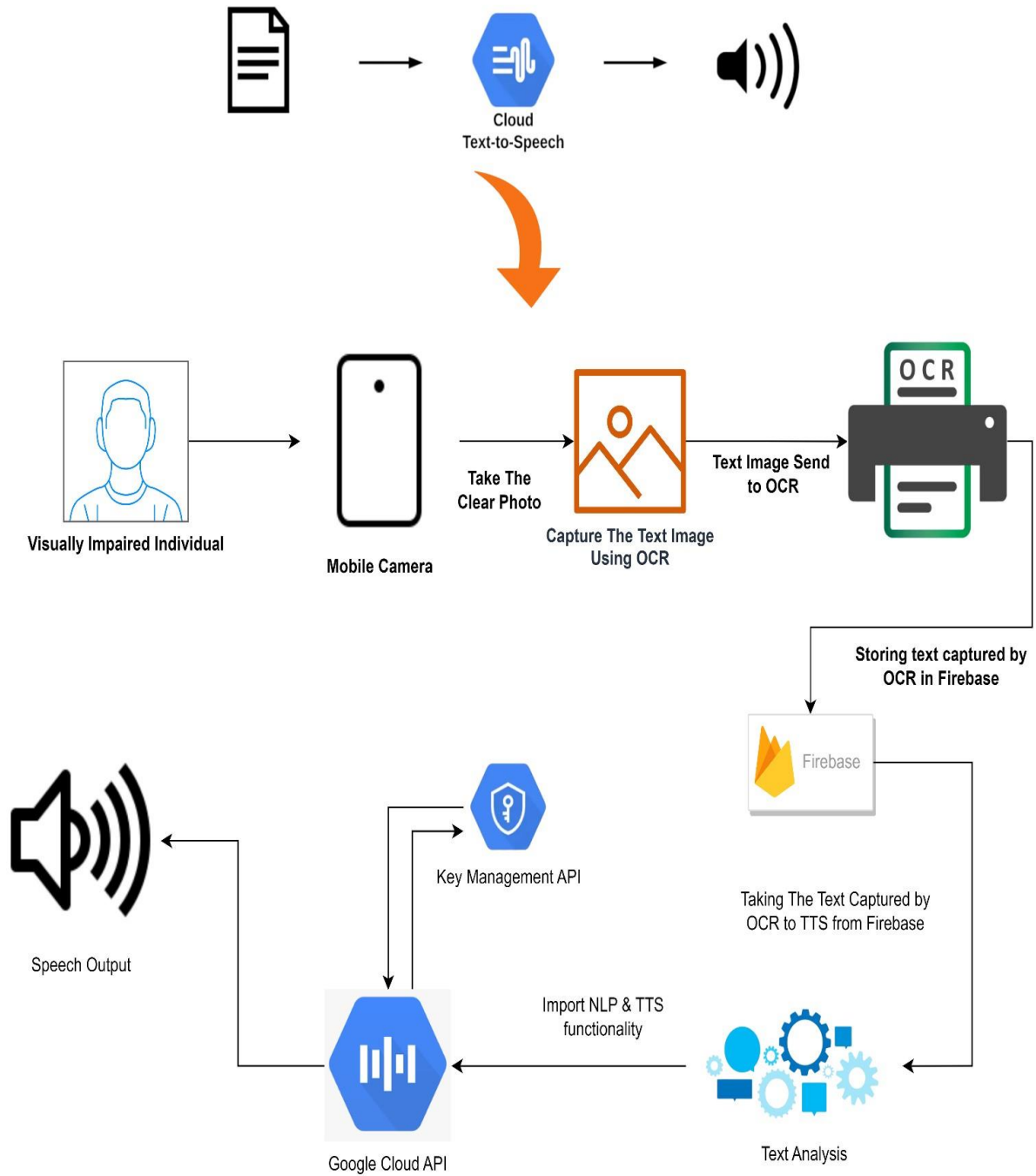


Figure 8: Individual System Diagram

A computer-based system known as a text-to-speech synthesizer is critical in converting computer-readable text into audible speech. This procedure includes multiple critical stages that may be divided into three basic phases: text analysis, linguistic analysis, and wave-form generation.

The procedure is as follows:

1. A visually impaired individual is instantly told vocally upon completion of OCR, ensuring they are aware of the task's completion.
2. At this point, the system passes the extracted and recognized Sinhala characters from the scanned text onto the TTS system for processing.
3. The TTS system then takes center stage, utilizing its ability to vocally recreate the Sinhala text from the camera picture, making it accessible to blind folks.

In summary, the goal of incorporating TTS into the Sinhala language book reader for the visually impaired is to offer them with a way to access written information and fully understand the content of books while overcoming the hurdles imposed by their visual disability. This technology enables people to successfully engage with books and information, fostering diversity and accessibility.

## **5.8. Requirement Analysis**

A critical stage in our development process is requirement analysis for our mobile-based Sinhala text-to-speech (TTS) system within the Sinhala book reader for visually impaired users. The first component of our requirement analysis is a thorough study of the subtleties and nuances of the Sinhala language to achieve proper pronunciation and natural-sounding speech synthesis.

In addition, to improve the user experience for those with visual impairments, we prioritize accessibility elements including intuitive user interfaces and interoperability with multiple mobile devices. We also realize the significance of incorporating excellent conversation management technologies to enable user interactions, allowing our application to serve as both a reader and an intelligent helper for programming-related activities. Our in-depth requirement research seeks to link our development efforts with the specific demands of our target audience, thereby creating inclusion and accessibility in the field of Sinhala literature for the visually impaired.

## **5.9. System Analysis**

System analysis is critical in the development of our mobile-based Sinhala book reader for visually impaired people, which includes Sinhala text-to-speech (TTS) capabilities. During this phase, we thoroughly examine our application's requirements, functionality, and user demands. Understanding the specific issues that visually impaired users confront is critical because it influences the design and execution of user-friendly features. To provide a seamless and

inclusive reading experience, we investigate user interfaces, navigation techniques, and accessibility standards.

Furthermore, system analysis comprises a thorough examination of Sinhala TTS technologies, with a focus on their accuracy, naturalness, and adaptation to the linguistic subtleties of our target audience. We establish the groundwork for a robust and user-centric Sinhala TTS book reader by doing extensive system analysis, enhancing the lives of visually impaired folks via accessible literature and technology.

### **5.10. Techniques Used for Speech Synthesis**

Speech synthesis, the fascinating discipline of reproducing human-like speech using technology, has evolved significantly with substantial technical advances. Voice synthesizers, which are among the most outstanding technologies in this sector, use computers' computational capacity to meticulously recreate the intricacies of spoken language [14]. Text-to-Speech (TTS) is the most common and effective solution in this arena, beautifully converting written text into expressive speech that nearly mimics the richness of human vocalization. Furthermore, various methodologies within this discipline are investigating novel ways to describe symbolic linguistics, pushing the limits of what is possible in the world of artificial speech creation. These improvements are not only improving accessibility for people with speech difficulties, but they are also finding useful uses in areas such as entertainment, education, and customer service. The future of speech synthesis promises great prospects for more human-like and emotionally expressive voices as technology advances.

Among these strategies, concatenation stands out as a trailblazing strategy for producing artificial speech. This is accomplished by skillfully combining pre-recorded speech fragments drawn from large databases, resulting in cohesive and lifelike synthetic speech. This approach works in tandem with another effective strategy: the conversion of textual input into voice using Text-to-Speech (TTS) technology. The panorama of voice synthesis, on the other hand, stretches even farther, incorporating systems that cleverly transform symbolic language representations into captivating vocal renditions, pushing the frontiers of what is feasible in artificial speech production.

Concatenation, a popular approach for producing artificial speech, entails combining pre-recorded speech fragments from a database. Text-to-Speech (TTS) is another famous technique that turns written text into speech. Aside from this, several systems investigate novel approaches for converting symbolic language representations into spoken speech [15]. Two fundamental technologies dominate the realm of speech synthesis: formant synthesis and concatenative synthesis. They both attempt to develop synthetic speech that closely mimics genuine human speech, demonstrating the many ways in which the area is evolving and improving.

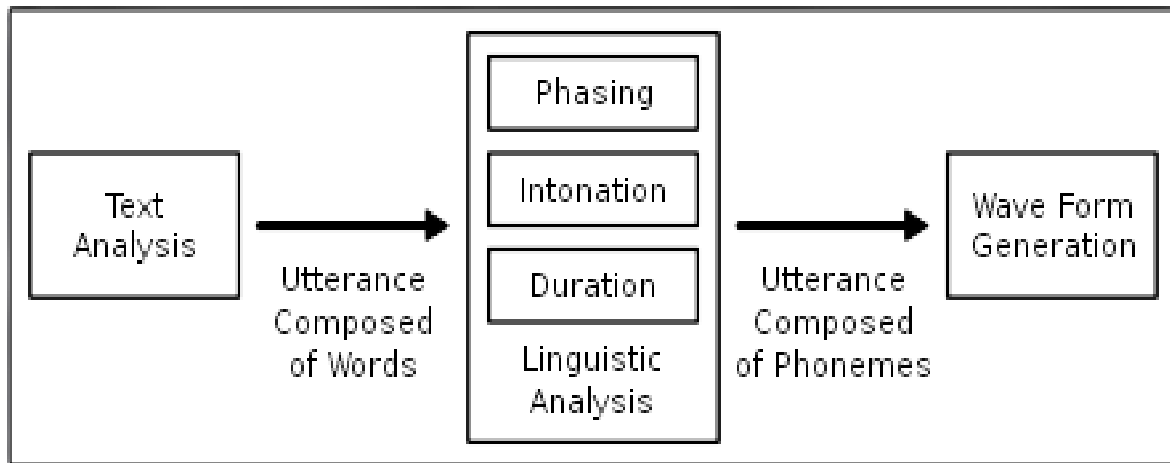


Figure 9: Text-to-Speech Wave Generation

TTS technology is a multidimensional system made up of two important components: the front end and the back end. The front end is critical because it performs two key tasks: text normalization and pre-processing. During the pre-processing step, the system recognizes individual words in the incoming text and turns them into phonetic transcriptions. Meanwhile, the text normalization step ensures that the incoming text follows a specified structure, allowing for more accurate and smoother voice synthesis. These two critical front-end procedures jointly establish the groundwork for TTS technology's effective functioning, allowing it to convert written text into natural-sounding spoken language.

Text-to-Speech (TTS) technology refines the input text further by segmenting it into prosodic units, which include phrases, sentences, and clauses [16], before transforming them into phonemes. This technique also entails identifying these prosodic units with pertinent prosody information, which captures details such as pitch, rhythm, and stress. The front-end creates a symbolic language representation of the input text using this augmented data. This representation combines prosody information with phonetic transcriptions, which improves the TTS system's capacity to create expressive and lifelike synthetic speech that mimics the complexities of actual human conversation.

The front ends precisely created symbolic language representation serve as the foundation for the back end, where actual sound synthesis occurs. This translation of the text's symbolic representation into audible sound is accomplished using a process suitably dubbed "synthesis." Synthesis is the critical process that transforms the symbolic representation into a cohesive and natural-sounding output, bringing artificial speech to life. The confluence of complicated algorithms and acoustic models joins here, at the back end, to generate synthetic speech that closely matches the rhythm, tone, and articulation of real voice.

## 5.11. Sinhala Language

### 5.11.1. Sinhala Consonant

Sinhala, Sri Lanka's native language, has a distinct aural identity due to its 26 consonants, which are an important component of the language's phonetic inventory [16]. Figure 10 in the chart presents a thorough list of these sounds, emphasizing their importance in the linguistic composition of Sinhala. The retroflex sounds stand out among these consonants because they are spoken by bending the tongue backward, providing a distinct phonetic feature. The language also has hissing alveolar fricatives, which distinguishes it from others. Notably, Sinhala has a sequence of pre-nasalized voiced stops, which contributes to its phonetic profile and makes it a linguistically intriguing language.

		Labial	Dental	Alveolar	Retroflex	Palatal	Velar	glottal
Stops	Voiceless	p	t		t̪		k	
	Voiced	b	d		d̪		g	
Affricates	Voiceless					c		
	Voiced					j		
Pre-nasalized voiced stops		ḃ	ḍ		ḍ̪		ḡ	
Nasals		m		n		ɲ	ŋ	
Trill				r				
Lateral				l				
Spirants		f	s			ʃ		h
Semivowels		v				y		

Figure 10: Spoken Sinhala Consonant Classification

The sound patterns inherent in Sinhala consonants are not only important to the language's identity, but also to its cultural past. These different phonetic traits have played an important part in the language's growth, allowing it to adapt to the changing terrain affected by the increased usage of English loanwords. As Sinhala incorporates these loanwords, its consonants continue to play an important role in molding the language's growth and integrating it with modern society. This dynamic combination between tradition and modernity demonstrates Sinhala's endurance and flexibility, demonstrating how it has developed to stay relevant in today's linguistic and cultural setting.

### 5.11.2. Sinhala Vowel

Sinhala's 14 vowels play an important role in defining the language's distinct aural character [13]. These vowels, as seen in Figure 11, are an important part of Sinhala's phonetic inventory. What distinguishes Sinhala vowels from vowels in other languages is their distinguishing qualities. Sinhala has a great degree of phonemic length difference between its vowels, which



gives depth and subtlety to its spoken sound. The language also utilizes a sophisticated system of vowel harmony, which further enriches its phonetic landscape. These outstanding characteristics of Sinhala vowels contribute greatly to the language's rich phonological heritage and undeniable aural beauty.

	Front		Central		Back	
	Short	long	Short	long	short	long
<b>High</b>	i	i:			u	u:
<b>Mid</b>	e	e:	ə	ə:	o	o:
<b>Low</b>	æ	æ:	a	a:		

Figure 11: Spoken Sinhala Vowel Classification

The vowels and consonants that make up the phonetic tapestry of the Sinhala language are representative of the language's rich cultural background. These linguistic characteristics not only show the language's historical roots, but also its adaptation to modern conditions. They are important pillars in creating Sinhala's identity and heritage, demonstrating the language's ability to adapt while keeping its own characteristics. The rising assimilation of English loanwords is a noteworthy example of this linguistic development, demonstrating how Sinhala adopts new sounds and vocabulary while retaining its core identity.

Consonants and vowels work together to form the foundation of Sinhala's acoustic and linguistic uniqueness. They continue to play an important role in the language's evolution, acting as conduits for its cultural heritage. These phonetic components continue to play an important role in Sinhala, strengthening its linguistic legacy. As Sinhala adopts new sounds and vocabulary, such as the increasing use of English loanwords, it demonstrates its adaptability and resilience, balancing tradition, and modernity in an ever-changing linguistic context.

### 5.11.3. Sinhala Character Set

Sinhala, Sri Lanka's official language, predominantly employs the Sinhala character set, which has a diverse linguistic palette. There are 40 consonants and 20 vowels in this set, each with unique features that contribute to the language's peculiar aural quality [10]. The usage of diacritics, which are symbols used to change the sounds generated by both consonants and vowels, further demonstrates the adaptability of these letters. It is possible to control and refine the composition of consonants and vowels by expertly adding or deleting diacritics, allowing for a subtle and expressive portrayal of the phonetic nuances of the Sinhala language.

A total of 18 diacritics are used in the Sinhala letter set to shape the language's phonetic subtleties. 17 of these diacritics are specially created as vowel modifiers, making them vital tools for differentiating Sinhala's varied array of vowels. They allow for the difference of nasalized vowels as well as long and short vowels, both of which contribute to the language's phonetic richness. The last diacritical mark has a specific function: it indicates the unmodified

consonant form, completing the set and guaranteeing clarity in the depiction of Sinhala consonants and vowels.

### Vowels

අ	ආ	ඇ	ඈ	ඉ	ඊ	උ	ඌ
a	ā	æ	æ̃	i	ī	u	ū
[a/ə]	[a:/a]	[æ]	[æ:]	[i]	[i:]	[u]	[u:]
ඊ	උ	එ	ඒ	ඹ	ඹ	ඹ	ඹ
ri	ri	e	ē	ai	o	ō	au
[ri/ru]	[ri:/ru:]	[e]	[e:]	[aj]	[o]	[o:]	[aʷ]

Figure 12: Sinhala Language 20 Vowels Table

Vowels are extremely important in the Sinhala language, playing a critical role in defining its distinct aural character. Each of the 20 vowels in the Sinhala letter set has unique features that add to the richness and complexity of the language. Sinhala vowels are classified into three types: short, long, and nasalized [16]. These many vowel kinds not only add to the distinctive sound of the language, but also give a nuanced and expressive framework for communication, reflecting the complexities of Sinhala's linguistic legacy.

To express distinct sorts of vowels in Sinhala, a systematic technique is used. Short vowels are represented by plain vowel symbols, but long vowels are represented with diacritical markings known as "pure." In addition, nasalized vowels are denoted by the niggahita tilde diacritic. The appropriate use of these diacritics is required to correctly discern and portray the subtleties of distinct vowels within the Sinhala language. This thorough use of diacritics improves the precision and clarity of written Sinhala, allowing readers and speakers to properly navigate its vast phonetic terrain.

The Sinhala letter set consists of 40 consonants, each with its own distinct qualities and sounds, making them essential components of the Sinhala language. These consonants are divided into two categories: pure consonants and compound consonants. These consonants' specific characteristics and phonetic properties add greatly to the depth and complexity of the Sinhala language, highlighting their critical significance in both written and spoken communication.

### Consonants

ක	ඛ	ග	ඝ	ඞ	ඟ	ච	ඡ	ජ	ඣ	ඤ	
ka	kha	ga	gha	ṇa	ṅga	ca	cha	ja	jha	ṇa	
[ka]	[ka]	[ga]	[ga]	[ṇa]	[ṅga]	[tʃa]	[tʃa]	[dʒa]	[dʒa]	[na]	
ට	ඨ	ඩ	ඪ	ණ	ඬ	ත	ථ	ද	ධ	න	ඳ
ṭa	ṭha	ḍa	ḍha	ṇa	ṅḍa	ta	tha	da	dha	na	ṇda
[ṭa]	[ṭa]	[ḍa]	[ḍa]	[na]	[ṅḍa]	[ta]	[ta]	[da]	[da]	[na]	[ṇda]
ප	ආ	බ	භ	ම	ඹ	ය	ර	ල	ව	ළ	
pa	pha	ba	bha	ma	m̐ba	ya	ra	la	va	ḷa	
[pa]	[pa]	[ba]	[ba]	[ma]	[m̐ba]	[ja]	[ra]	[la]	[va]	[la]	
ශ	ෂ	ස	෪	හ	ආ						
śa	ṣa	sa	ṣa	ha	fa						
[ʃa]	[ʃa]	sa	[za]	[ɦa]	[fa]						

Figure 13: Sinhala Language Consonants Table

Pure consonants, which lack intrinsic vowels, constitute simple and distinct phonetic sounds in Sinhala. Compound consonants, which are generated by mixing consonants and vowels, serve as linguistic carriers for more complex sounds. The complicated interaction of pure and compound consonants, which is frequently adjusted by the addition or removal of diacritics, enables exact identification of consonant and vowel compositions [13]. This dynamic use of consonants and diacritics not only allows for the articulation of a broad range of phonetic subtleties, but it also emphasizes the intricate character of the Sinhala script, where consonants and vowels merge to produce a harmonious language system.

Vowels and consonants work together to produce the aural character of the Sinhala language. The Sinhala letter's set of 40 consonants spans a broad range of both fundamental and nuanced sounds, while the 20 vowels are essential for differentiating between the language's various phonetic variants. The correct use of diacritics is critical in identifying the composition of consonants and vowels, bringing depth and complexity to the Sinhala script [16].

Furthermore, the availability of text-to-speech alternatives has substantially improved the accessibility and inclusivity of learning and communication in the Sinhala language, particularly for those with visual impairments. These technological advances have created new opportunities for making Sinhala more accessible and user-friendly, guaranteeing that the beauty and depth of the language may be enjoyed and embraced by a larger audience.

## 5.12. Project Requirements

### 5.12.1. Functional Requirements

At the heart of our goal is a dedication to tearing down barriers and guaranteeing equitable access to education for all students, regardless of ability. We are excited to release the Sinhala TTS Synthesizer as we work to improve the capabilities of our mobile Audio Sight Sinhala Book Reader. This novel element changes how visually challenged pupils interact with instructional information. We recognize the importance of readily available learning resources, and this cutting-edge Sinhala TTS capability is ready to transform the educational environment for those who rely on it. By offering a tool for visually challenged kids to read print storybooks, we hope to foster independence and diversity in education. Here are the primary functional requirements for text-to-speech.

- **Sinhala Text-to-Speech (TTS) Synthesizer:** The mobile book reader must include a high-quality Sinhala TTS synthesizer capable of transforming Sinhala text into spoken words accurately and organically.
- **Adjustable Reading Speed:** To accommodate the different preferences of visually impaired people, the TTS function should have adjustable reading speed options. Users should be able to modify the speed to their liking, delivering a pleasurable and accessible reading experience.
- **Pause and restart:** Recognizing that users may need to pause and restart their reading at any time, the TTS function should accommodate this functionality flawlessly. This feature allows users to take pauses or go through the text at their own leisure, improving the mobile book reader's overall usefulness.

We acknowledge the tremendous impact of this Sinhala TTS option for our mobile Audio Sight Sinhala Book Reader on the lives of visually challenged students as we improve it. Access to educational materials is a vital right, and this technology gives these kids a new sense of independence and the capacity to engage in their studies on par with their sighted counterparts. We hope to bridge the educational opportunity gap by creating a flexible and user-friendly application that translates text into natural, audible Sinhala speech.

### 5.12.2. Non-Functional Requirements

Our mission is to use technology to empower visually challenged pupils in Sri Lanka. We are currently working on developing a Sinhala TTS option for our mobile Audio Sight Sinhala Book Reader. We recognize that this is about more than simply software development; it has a substantial influence on the lives of individuals we hope to help. We are devoted to providing

these students with a platform that promotes accessibility, usability, quality, and security while also simplifying their educational experience. Our commitment to ensuring that our Sinhala TTS function passes high non-functional standards illustrates our commitment to making education more inclusive and accessible to all. The key non-functional needs for text-to-speech are listed below.

- **Accessibility:** Accessibility should be prioritized for visually challenged persons, particularly Sinhala-speaking pupils. This involves making sure that screen readers, voice commands, and other assistive technology are compatible with the audience.
- **Usability:** The user interface and controls must be designed to be simple to accommodate users who may be unfamiliar with complicated mobile interfaces. To guarantee simplicity of use, clear and concise instructions in both Sinhala and English are required.
- **TTS Voice Quality:** The Sinhala TTS synthesizer should provide natural-sounding speech of excellent quality. To improve the reading experience for visually challenged pupils, it must precisely recreate Sinhala text while keeping pronunciation, intonation, and rhythm.
- **Performance:** The application should run well on mid-range mobile devices that are widely available to the target audience. This features quick and fast TTS conversion, which ensures that text-to-speech capabilities run smoothly and without pauses.
- **Security:** Because user information and reading history are sensitive, rigorous security measures should be in place to secure this data. To protect personal information and guarantee user privacy, encryption and user authentication procedures must be used.
- **Reliability:** The consistency of TTS conversion is critical. The program should translate Sinhala text to speech consistently, so that visually challenged students may rely on it as a consistent and reliable resource for reading Sinhala literature.
- **Sinhala TTS Functionality:** For the Sinhala-speaking audience, the TTS function should be designed to correctly pronounce Sinhala letters, words, and phrases. It should be able to handle complicated Sinhala script peculiarities properly, ensuring that the audio output is clear and understandable.

With a strong feeling of duty and purpose, we are creating the Sinhala TTS function for our Audio Sight Sinhala Book Reader for mobile devices. We appreciate the confidence you have placed in us to deliver a product that enhances the lives of visually impaired children by providing them with access to information, reading, and learning. As the cornerstone of this transformational tool, we focus non-functional needs such as accessibility, usability, voice quality, performance, security, reliability, and Sinhala TTS capabilities. We are devoted to not just meeting but exceeding our users' requirements and expectations by concentrating on these elements. Our path highlights our view that, when used with care and forethought, technology

can help break down barriers and create a more inclusive society in which everyone has an equal opportunity to learn, grow, and thrive.

### **5.12.3. Hardware Requirements**

The most essential hardware needed throughout the creation of the Text-to-voice (TTS) system in this research is the necessity for a speaker to create voice. The technology requires the integration of a speaker to turn text into an audible voice. The speaker is the channel via which the synthesized speech is supplied to the user, ensuring that textual information is successfully communicated in an audio manner. Meeting this need is critical to the basic operation of the TTS system, boosting its usability and accessibility for diverse applications and users.

### **5.13. Commercialization of the Project**

The commercialization plan for a mobile app catering to visually impaired individuals include effective communication, monetization strategies, and partnerships.

#### **1. Identifying the Target Audience:**

- It is critical to define and comprehend your target audience. Consider conducting surveys or connecting with visually impaired groups to learn more about their unique needs and preferences. Create user personas to help you better personalize your application to their needs.

#### **2. Revenue Creation:**

- **Freemium Model:** Provide a free version of your mobile application with minimal functions, with the opportunity to pay a charge to upgrade to a premium version with additional functionality. This method allows customers to test the software before making a purchase.
- **Subscription Plans:** Implement subscription plans with varying pricing levels to meet the demands of varied users. For example, offer monthly, yearly, or lifelong memberships.
- **In-App Purchases:** Think about including in-app purchases for extra features or content, such as specialist tools and store audiobook features.
- **Donations:** Include an option for users and supporters to make voluntary donations to support your application's development and accessibility initiatives.

### 3. Promotions

- **Social Media Campaigns:** Develop aesthetically appealing and useful postings for networks such as Facebook, Twitter, Instagram, and LinkedIn. Use relevant hashtags and share success stories, user testimonials, and updates with visually impaired groups.
- **Educational Content:** Create blog entries, articles, or videos that highlight the advantages and distinguishing aspects of your product. Share these materials with potential users and caregivers via our website and social media.
- **Partnerships and Collaborations:** Collaborate with organizations, schools, libraries, and advocacy groups that serve visually impaired people. Collaborative activities can help spread the word about our software and give critical feedback for future enhancements.
- **Availability Conferences and Events:** Attend or sponsor accessibility and assistive technology-related events to promote our application and engage with potential users and partners.
- **User Communities:** Create online forums or communities for users to discuss their experiences, seek help, and provide feedback. Engage actively in these communities to establish a loyal user base.

## 6. DESIGN & IMPLEMENTATION

### 6.1. Data Preprocessing (Text & Audio)

To improve the text-to-speech experience, raw text with correct space and punctuation is required. During the preprocessing step, our major aim was to smoothly integrate the raw text input into the function of our system. By correcting mispronunciations, omissions, and redundancies, we were able to solve the issue of confusing and non-fluid spoken information. Our approach was developed by creating a preprocessing model that normalizes incoming text and removes impediments.

The developed preprocessing method included numerous critical transformations. To begin, we transformed all the input text to uppercase to ensure uniformity. Following that, we deleted intermediate punctuation marks one by one, simplifying the text for more effective linguistic analysis. To guarantee genuine prosody, each statement was then suitably punctuated with a question mark or a period.

We developed specific separator characters to distinguish between words, words with relaxed pronunciation, words with brief pauses, and spaces to address the issue of spaces introduced during speaking [17]. This flawless alignment of text and audio significantly minimized long silences and permitted exact synchronization, considerably improving auditory performance.

Our research has significantly improved text preprocessing, leading to more enriched text-to-speech outcomes. The proper format sliced audios are taken from public domains listed in TABLE 2.

TABLE 2: PREPARING AUDIOS DATASETS

Functionalities	Sinhala
Name	“pathnirvana”
Audio Resource from	Kaggle & GitHub
File Format	.wav file
Audio Count	3300
Speaker Type	Single Speaker
Single audio Duration	1seconds to 10seconds
Text Resource from	Converted Sinhala font



## 6.2. Encoder

The system primarily handles text input and uses an encoder to translate it into an internal representation. This representation records learning results through fully convolutional layers. The encoder is built to handle two types [18] of input: phoneme embeddings and phoneme stress embeddings. It converts phonemes or letters into vector representations, which can be further enhanced through training. The embedding percentage gradually becomes a fully linked structure after passing through a few convolutional blocks. Attention key vectors are then generated by referencing the embedding proportions. These key vectors are used within each attention block to determine attention weights. Finally, the context vector is calculated by merging the value vectors in a weighted fashion using a standard procedure. According to Figure 14, the following diagram shows how to encode texts in Sinhala TTS.

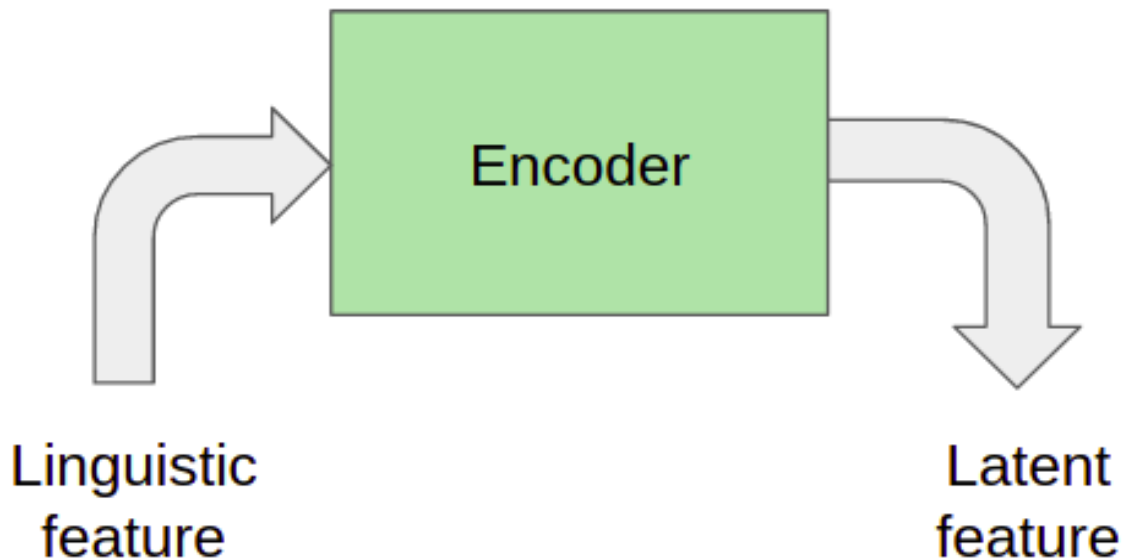


Figure 14: Encoder the Preprocessor Text in Sinhala TTS

The encoding procedure begins with embedding the input data and progresses through a chain of convolutional blocks from embeddings to a fully connected representation. Following that, the attention key vectors are constructed by looking back to the initial embedding proportions. These key vectors are used to calculate attention weights inside each attention block. The final context vector is calculated using a typical manner as a weighted aggregate of the value vectors.

The Encoder design is distinguished by a convolutional block with excellent connection to the input data. It also includes sequential gated components, which, curiously, do not rely on rigid sequence-based dependencies but rather feature certain scaling characteristics. To accommodate various sequence lengths, the input is supplemented with timestamps, allowing the Encoder to handle inputs with diverse temporal scopes successfully. Because of its adaptability, the Encoder is a reliable component for a broad range of applications, particularly

those needing the extraction of significant characteristics and representations from textual or sequential input.

### 6.3. Decoder

The decoder's primary goal is to make predictions in an auto-regressive manner, successfully decoding the learned outcomes. The decoder is specifically developed to estimate future audio sequences based on previous audio files. The decoder is built as a fully convolutional neural network, with a strong emphasis on totally causal convolutions [18]. This design decision is critical in ensuring that predictions are created in a sequential and causal fashion, which means that each prediction is exclusively based on previous information, matching the natural flow of audio data.

This decoding procedure is based on transferring the attention mechanism of multi-hop convolutional networks into an audio representation of low-dimensional Mel-band spectrograms [11]. This modification enables the model to focus on essential audio elements when making predictions, improving its ability to grasp complex patterns and structures in audio data. Furthermore, the decoder operates on audio sequences that have been grouped together, which has a substantial influence on performance. Decoding numerous audio collections at the same time has been found to be more efficient than processing individual audio files separately. This method makes use of the contextual information contained in linked audio sequences, resulting in more accurate and coherent predictions. According to Figure 14, the following diagram shows how to decoder Audio and Texts in Sinhala TTS.

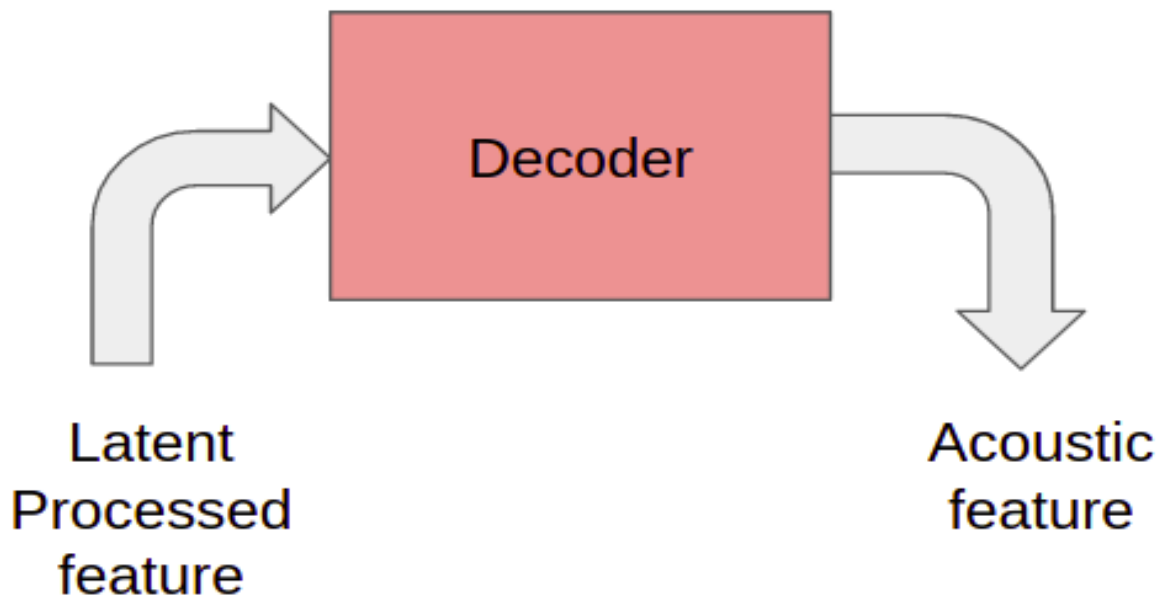


Figure 15: Decoder the Audios and Text in Sinhala TTS

Rectified Linear Units (ReLU), attention blocks, and output layers are all important components of the decoder network. The ReLU activation function covers nonlinearities in several fully linked layers, allowing the model to capture complicated data connections [11]. The attention blocks are arranged in a certain order to assist the decoder in focusing on relevant information from the encoder's output. The output layers are completely linked layers that forecast the next batch of audio frames. A binary wrapping procedure is used to the last frame to assure continuity. Dropout is selectively applied to all fully connected layers before the attention blocks, save the first layer, to reduce overfitting and increase model generalization. Using the output spectrograms, the model creates predictions for the missing data functionality, and its performance is assessed using the cross-entropy loss function, with a focus on the "done" prediction.

The decoder's hidden state is critical for its operation, especially when utilizing the dot-product attention technique. This technique computes the output vector, which is a weighted average, using the encoder's vectors at each time step. It's worth noting that the model only receives audio data from a single speaker during training. The encoding position rate is set to one to maintain alignment with the decoder and remains constant about the encoder, allowing the model to successfully train and output coherent audio sequences.

The architecture of the decoder network, which includes ReLU activations, attention blocks, and output layers, is intended to reliably forecast future audio frames. Dropout techniques are used to improve model resilience, and the dot-product attention mechanism makes precise context-based predictions using the encoder's per-timestep vectors. This method works especially well when trained with data from a single speaker, resulting in coherent and high-quality audio creation.

#### **6.4. Converter**

The converter's principal job is to forecast the final output characteristics required by the post-processing network. This prediction is strongly reliant on the mechanism used to produce the waveform from the decoder's hidden states. The converter network is built as a fully convolutional architecture, with a strong emphasis on gathering context information relevant to the job at hand.

The converter network's input activations are obtained from the decoder's final hidden layer outputs [18]. The converter uses its non-causal and non-autoregressive convolution blocks to anticipate important vocoder parameters, based on the decoder's upcoming context. This contextual data is critical for making accurate predictions and producing high-quality audio outputs that accurately replicate the intended audio waveforms.

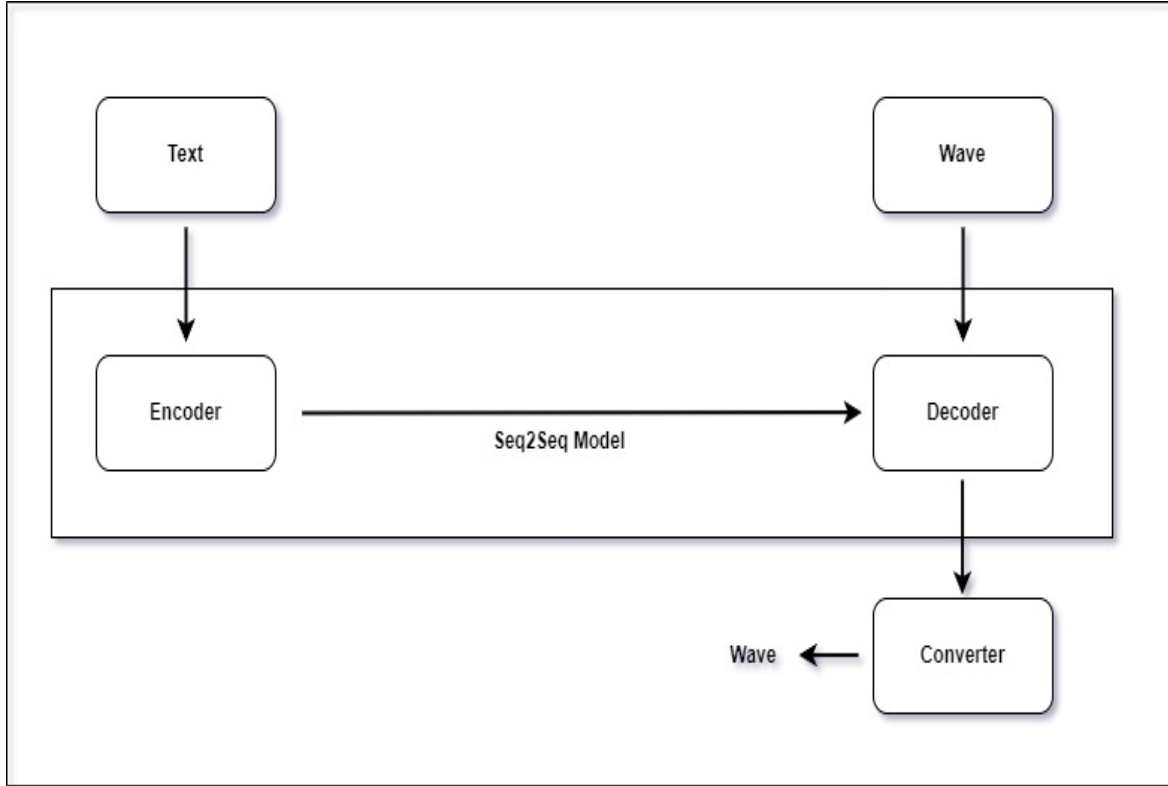


Figure 16: Architecture of The System

The design of the proposed text-to-speech application is based on a thorough review of current literature, as seen in Figure 16. This research serves as the foundation for further assessments, which must be closely aligned with the results of the system design guide to guarantee consistent and stable performance [4]. The architectural framework is made up of four important components, each of which is critical to the system's seamless operation.

The author used the Griffin-Lim method for converting spectrograms during the procedure, which was crucial in creating the final audio synthesis [18]. The loss function was chosen with care, as it is determined dependent on the performance of the vocoder. The authors specifically used a loss function on linear-scale (log-magnitude) spectrograms. This choice is consistent with the Griffin-Lim method and guarantees that the model's training process efficiently optimizes for the desired audio output quality.

## 7. RESULTS

### 7.1. Building Sinhala Front-End

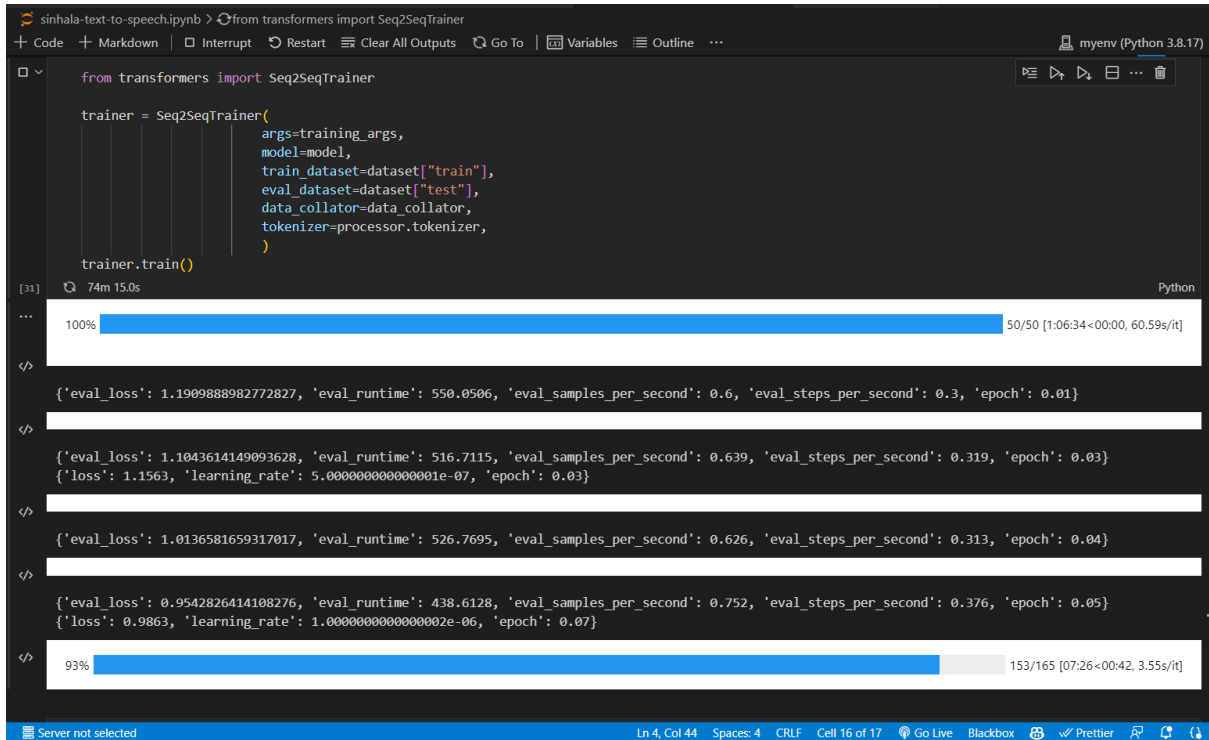
The major purpose of text data preparation is to provide optimal performance in the creation of a Sinhala text synthesizer front-end. The Sinhala front end meticulously prepares input text, not only for cleaning but also to improve training accuracy. Using a mixed pronunciation method is a critical component of this procedure. This approach breaks down words in the dataset into individual letters and employs a Sinhala alphabet system that has been painstakingly constructed by compiling a complete Sinhala phoneme database [19]. This database contains a large collection of Sinhala dialects in both spoken and written form. The input dataset is then processed using a sequence-to-sequence text conversion technique, which converts the textual data into a numerical sequence.

The Sinhala text synthesizer front-end not only cleanses and prepares input text, but it also applies a complex mixed pronunciation approach, which is backed by a Sinhala alphabet system based on a large phonetic lexicon that contains considerable Sinhala dialect variants, both spoken and written. This preprocessed data is then smoothly integrated into a sequence-to-sequence conversion process, allowing language to be transformed into numerical sequences for enhanced training accuracy.

### 7.2. The Training Sequence to Sequence Model

The training phase is critical in the process of sequence-to-sequence speech synthesis. It is based on a predefined configuration file, which plays an important role in designing the training process. This preset includes settings for optimizing training, such as the critical checkpoint interval, which controls the number of repetitions before a checkpoint is stored. The eval interval parameter oversees analyzing the TTS waveform and creating voice output for the input text at regular intervals [10]. Furthermore, the batch size is chosen dynamically by dividing the overall dataset size by the number of repetitions, resulting in efficient data processing.

Input data is divided into three unique objects throughout the training process: Text Data Source, Linear Spec Data Source, and Mel Spec Data Source. The main goal is to load the training data for the specified number of iterations. Following that, a model is built with the provided hyperparameter values in mind. If a GPU is available, training is performed on this high-performance hardware to accelerate and improve efficiency. This methodical technique guarantees that the training model is carefully calibrated and capable of delivering high-quality synthesized speech.



```
from transformers import Seq2SeqTrainer

trainer = Seq2SeqTrainer(
    args=training_args,
    model=model,
    train_dataset=dataset["train"],
    eval_dataset=dataset["test"],
    data_collator=data_collator,
    tokenizer=processor.tokenizer,
)

trainer.train()
```

[31] 74m 15.0s Python

100% 50/50 [1:06:34<00:00, 60.59s/it]

```
{'eval_loss': 1.1909888982772827, 'eval_runtime': 550.0506, 'eval_samples_per_second': 0.6, 'eval_steps_per_second': 0.3, 'epoch': 0.01}
```

```
{'eval_loss': 1.1043614149093628, 'eval_runtime': 516.7115, 'eval_samples_per_second': 0.639, 'eval_steps_per_second': 0.319, 'epoch': 0.03}
```

```
{'loss': 1.1563, 'learning_rate': 5.000000000000001e-07, 'epoch': 0.03}
```

```
{'eval_loss': 1.0136581659317017, 'eval_runtime': 526.7695, 'eval_samples_per_second': 0.626, 'eval_steps_per_second': 0.313, 'epoch': 0.04}
```

```
{'eval_loss': 0.9542826414108276, 'eval_runtime': 438.6128, 'eval_samples_per_second': 0.752, 'eval_steps_per_second': 0.376, 'epoch': 0.05}
```

```
{'loss': 0.9863, 'learning_rate': 1.0000000000000002e-06, 'epoch': 0.07}
```

93% 153/165 [07:26<00:42, 3.55s/it]

Figure 17: Train Sequence to Sequence Sinhala Text-to-Speech Model

Training a Seq2Seq model for Sinhala Text-to-Speech (TTS) is a difficult task that requires both devotion and experience. The model's error loss has dropped to an astounding 0.9863, which is far below the critical threshold of 1. This achievement highlights the model's ability to create Sinhala speech that closely mimics the intricacies of human pronunciation and prosody.

This astonishing achievement was achieved with a rigorous training schedule that included 165 steps and lasted an impressive 16 hours. This highlights not just the computational resources necessary for such an attempt, but also the depth and complexities of the Sinhala language. The positive conclusion is a huge step forward in providing inclusive and accessible TTS solutions for the Sinhala-speaking population. This advancement has the potential to improve accessibility, education, and communication for many people, so contributing to a more inclusive and connected world.

In conclusion, training the Sequence-to-Sequence model is an important step in achieving cutting-edge voice synthesis. We develop a model that is ready to produce great outcomes by carefully configuring parameters, managing data diligently, and utilizing available GPU resources. As time goes on, this well-trained model has the potential to empower a wide range of applications, from voice assistants to accessibility aids, therefore making human-machine interactions more natural and inclusive. The strong foundation established during the training phase assures that the future of speech synthesis is not only promising, but also accessible and adaptable to a wide range of linguistic and contextual needs.

### 7.3. Synthesizing the Training Model

The synthesis phase represents the pinnacle of the sequence-to-sequence model's capabilities, as it attempts to convert text input into a coherent waveform. This approach, crucially, makes use of the useful checkpoints created during training [20], which serve as critical milestones in the model's learning path. These checkpoints capture the model's comprehension of language subtleties, auditory patterns, and contextual information, allowing it to create astonishingly accurate and natural-sounding speech.

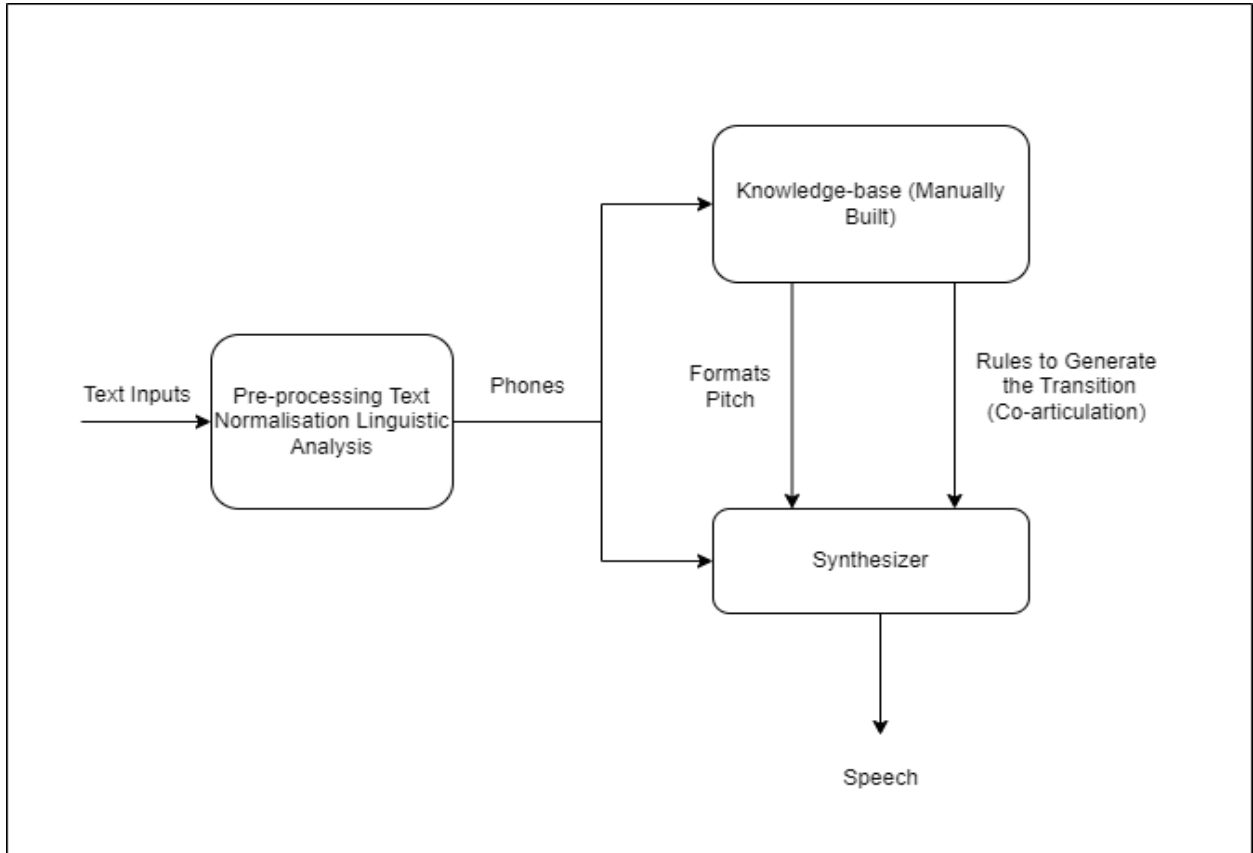


Figure 18: A Typical Architecture of Formant Synthesizer

During synthesis, an external vocoder works with the trained model to bring the produced speech to life by leveraging the checkpoints [12]. A vital component of the synthesis pipeline, the vocoder converts the model's linguistic output into a precise waveform that properly depicts the desired voice. The collaboration between the trained model and the vocoder guarantees that the synthesized speech has high quality, smooth prosody, and a genuine voice, making it appropriate for a variety of applications such as virtual assistants, audiobooks, and more.

The synthesizing phase draws on the model's information and skills gained during its training journey. This, along with the seamless integration of an external vocoder, allows the model to translate text inputs into astonishingly natural-sounding voice, ushering in a new age of human machine connection and enhancing the landscape of audio content creation.

## 7.4. Backend Audiobook Create

Audio Sight is a breakthrough smartphone application that aims to empower visually impaired people by providing them with easy access to Sinhala audiobooks. This unique program is specifically designed to meet the requirements of the Sinhala-speaking blind population, ensuring that reading becomes an inclusive experience for all, regardless of visual disability.

We can easily construct your own personal library of Sinhala audiobooks on your mobile device with Audio Sight. The program makes use of Sinhala Optical Character Recognition (OCR) technology to convert text from our favorite Sinhala books into clear and natural Text-to-Speech (TTS) audio. In practice, this means that by just photographing a book page, Audio Sight can turn the text into an audiobook that blind users can listen to on their smartphone. This program provides access to a world of Sinhala literature, ranging from timeless masterpieces to instructive resources, all in a handy and user-friendly audio format.

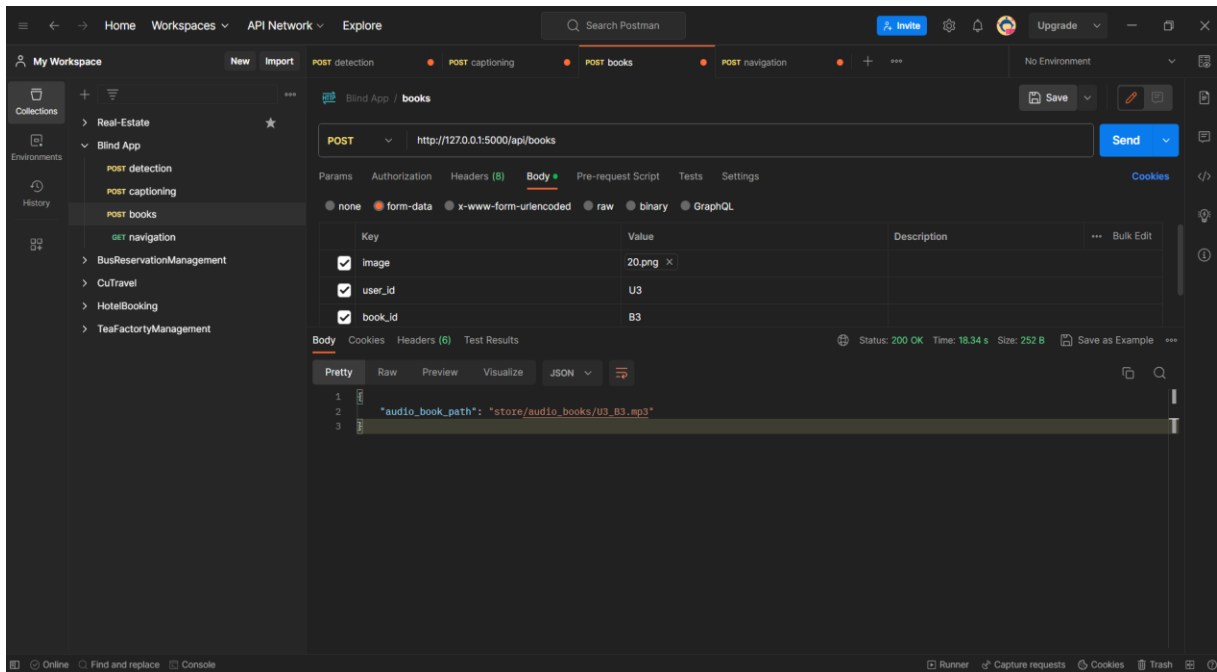


Figure 19: Create Sinhala Audio Book

Audio Sight stands as a beacon of accessibility and inclusivity in the realm of literature, revolutionizing the way visually challenged individuals experience the world of Sinhala books. This app not only breaks down barriers but also fosters a sense of independence, enabling the blind community to immerse themselves in the rich literary heritage of the Sinhala language effortlessly.



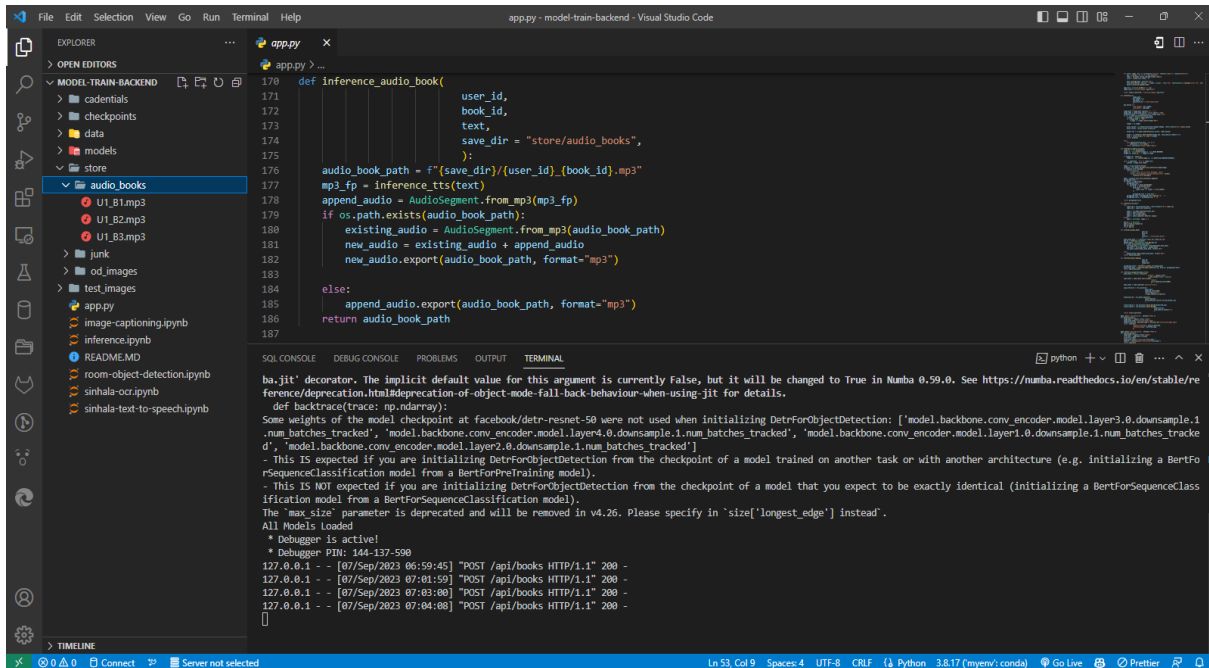


Figure 20: Audio Book Store as Mp3 Files

Audio Sight also focuses user-friendliness, making it simple for visually impaired people to engage with the program. The Sinhala voice command function allows you to use voice commands in the Sinhala language to control the playback of your audiobooks, move across chapters, alter playback speed, and much more. This hands-free technique allows you to fully immerse yourself in the world of Sinhala literature without having to perform intricate movements or engage with touch-screen interfaces.

## 7.5. Subjective Evaluation

The most important achievement of a reliable software system is substantial accuracy. A software system's accuracy is inextricably linked to its correctness and dependability. Accuracy testing, in essence, serves as a litmus test, determining the system's dependability and precision in delivering the desired outcomes.

Quality testing, which is frequently considered an essential parameter, acts as an important milestone in evaluating the system's performance [3]. It serves as a complete assessment of the system's ability to continuously provide the proper outputs while remaining true to its intended purpose. The significance of accuracy testing cannot be emphasized, since it not only guarantees that the system satisfies its design parameters but also that it operates reliably under a variety of scenarios.

To achieve an accurate software system, a well-defined formula is used to monitor accuracy testing, as shown in (1). This formula acts as a quantitative standard, allowing the system's correctness to be measured and monitored over time. Accuracy testing is the cornerstone that

binds it all together, guaranteeing that the finished result is not only stable but also trustworthy and dependable in providing the required outputs. Finally, the objective is to offer users a software system that they can rely on to constantly execute with the highest accuracy and correctness.

$$r = \frac{x}{y} \times 100 \quad (1)$$

r: The accuracy percentage, which measures how well the software system translates input text (x) into proper spoken output (y).

x: This variable represents the total number of input texts or test cases handled by the program.

y: The number of times the program provides the right voice output out of the total number of test instances (x).

In their current condition, the tested techniques have achieved an 87% accuracy rate, a solid basis for further development. The Author is devoted to improving accuracy by training the model with a larger dataset to improve system performance. This proactive strategy emphasizes the commitment to obtaining accuracy and excellence prior to project completion.

One interesting technique is to find the minimal data set, which is made easier by referring to TABLE 3 from the English speech dataset. This procedure entails several training procedures that are customized and suited to the complexities of the Sinhala language. The goal is to refine the model's understanding and proficiency by drawing insights from established practices in the field of speech synthesis and applying them judiciously to the Sinhala context, ultimately pushing the boundaries of accuracy and paving the way for an even more refined and precise speech synthesis system.

TABLE 3: TRAINING MINIMUM DATA SET OF SINHALA DATA

Data Sets	Hours of Training	Steps
3300	16	165
2200	11	50

Training a minimal Sinhala data set entails fine-tuning an existing machine-learning model using a small amount of Sinhala speech data. In this scenario, 3,300 speech datasets totaling around 16 hours of audio recordings provide the basis for this training effort. The goal of this exercise is to improve the understanding and generation of Sinhala text or voice using a pre-trained model.

The training process is often comprised of numerous iterations or phases, which in this case totals 165. The model improves its comprehension of the Sinhala language with each step, eventually enhancing its capacity to transcribe or create correct Sinhala text or voice depending on the datasets supplied. The training process's efficiency and efficacy are determined by a variety of criteria, including data quality, model design, and the precise purpose for which the

model is being fine-tuned. As a result, even with a small dataset, important advances in Sinhala language processing may be made, making the language more accessible and adaptable for a variety of applications.

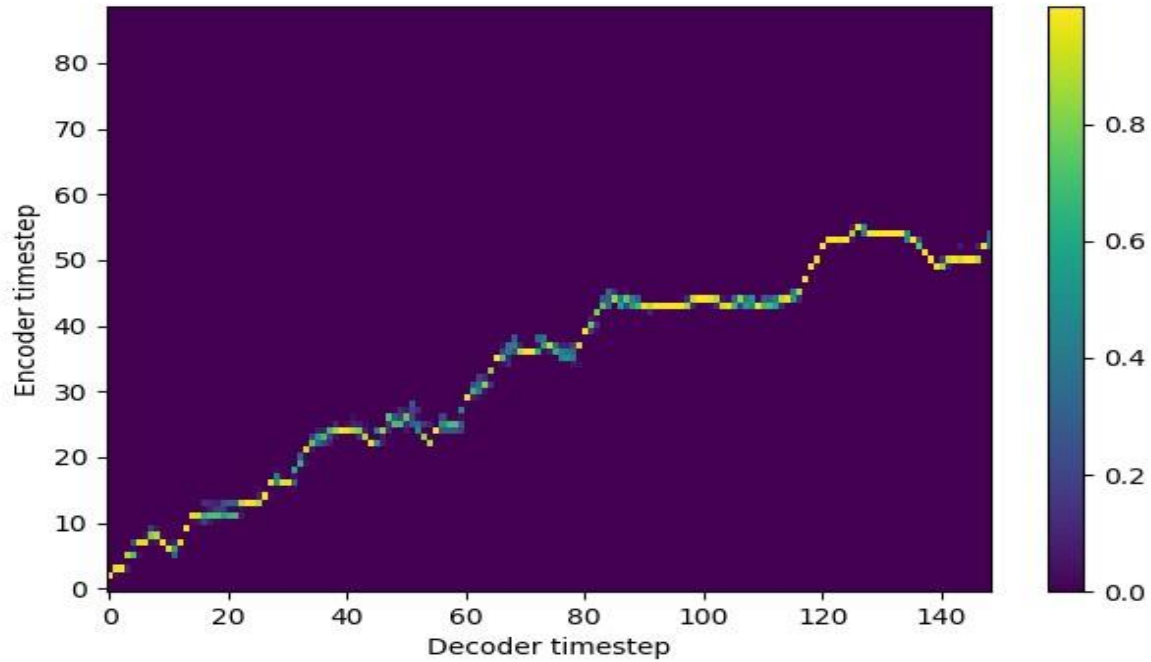


Figure 21: Tensor Board Result of Sinhala Wave Output

Improving the quality of the Sinhala output represented in the graph, as shown in Figure 21, demands more tuning. The graph now has an obvious curve, which may be corrected by enlarging the training dataset. While our current components, such as the voice synthesizer and text preprocessors, are well-suited to English language processing, we may solve the lack of multilingual capabilities by including Sinhala support for Text-to-voice (TTS) capability.

Furthermore, in line with the findings of the Literature Review, our dedication to intelligent TTS systems requires us to constantly evolve and improve the present algorithms and approaches. This proactive technique not only resolves the existing curve in the Sinhala result graph, but it also significantly improves the user experience. We are committed to ensuring that our TTS system excels in producing high-quality Sinhala audio output, giving a seamless and enriching experience for our users by concentrating on these enhancements.

## 8. DISCUSSIONS

According to our results, there is currently a noteworthy lack of a well-established and successful strategy for learning and mastering Sinhala-specific open-source projects. This disparity in accessible resources is a critical issue that must be addressed. Furthermore, when we analyze the graph depicting Sinhala language results, as shown in Figure 21, we can see that the curve in the line shows great space for development. To address this, we must increase the size of our training dataset, which is one step we can take to improve the quality and accuracy of our Sinhala language processing skills. While our current components, such as the voice synthesizer and text preprocessors, can perform English language jobs, we recognize the necessity to overcome multilingual support limitations. To address this issue, we are currently working on Sinhala language support for our Text-to-Speech (TTS) technology. This strategic change intends to make our system more adaptable and inclusive, allowing it to adapt to the complexities of the Sinhala language with ease.

Furthermore, in keeping with the findings of our extensive literature research, the idea of intelligent TTS emphasizes the significance of constantly improving our algorithms and methodology. This proactive strategy not only addresses the curve shown in the Sinhala language result graph, but also emphasizes increasing the value we deliver to our consumers. We are completely dedicated to developing a TTS system that excels at generating high-quality Sinhala audio output by focusing our efforts on these developments. In doing so, we want to provide a smooth and enhanced experience for our users while also enhancing accessibility and language processing skills in the Sinhala language area.

The fundamental goal of this project is to create an intelligent Sinhala-based multilingual Text-to-Speech (TTS) system. This huge project was dependent on the use of cutting-edge machine learning techniques, notably neural networks. This project's trajectory has been distinguished by both big accomplishments and difficult disappointments, inspiring exploration of unique concepts, rigorous investigations, and inventive developments. The road to accomplishing the project's final aim was not easy, requiring unwavering devotion and work.

The author has gathered a plethora of information encompassing many aspects of project management and machine learning during this research-based endeavor. This experience is quite like the complications found in a real-world business context. As a result, while the initiative began as an academic endeavor solely targeted at benefiting the blind population, it has grown into a potentially profitable company. The result has both technical capacity and commercial promise, demonstrating the effort and experience put into its creation.

## 9. CONCLUSION

In this thesis, our research emphasizes the crucial need of establishing a Text-to-voice (TTS) system capable of producing more human-like voice output, particularly in languages such as Sinhala, where little work has been done in the field of speech synthesis. Our results highlight the importance of neural network-based techniques in producing natural-sounding speech, linguistic flexibility, and effective storage consumption. We present "Audio Sight," a game-changing smartphone application aimed at enabling visually challenged people in the Sinhala-speaking community. This program was particularly created to bridge the reading accessibility gap for visually challenged users. Users of Audio Sight may easily build their personal collection of Sinhala audiobooks on their mobile devices.

Sinhala Optical Character Recognition (OCR) technology is used by Audio Sight to convert text from their favorite Sinhala books into clear and natural Text-to-Speech (TTS) audio. Audio Sight turns text into an audiobook that visually challenged individuals may listen to on their cellphones by just taking a snapshot of a book page. This breakthrough provides access to a large universe of Sinhala literature, ranging from timeless masterpieces to instructional materials, all in an easy-to-use audio format.

The program not only removes barriers to reading literature, but it also fosters independence within the blind people by allowing them to easily immerse themselves in Sinhala's rich literary legacy. Furthermore, Audio Sight lays a great emphasis on usability. It supports Sinhala voice commands, allowing users to control audiobook playing, explore chapters, alter playback speed, and more using simple voice commands in their own language. This hands-free technique improves the user experience, making Sinhala literature more accessible than ever before.

Aside from user-facing features, our study entails the creation of a Seq2Seq Sinhala TTS model. This architecture contains data preprocessing, an encoder, a recorder, and a converter, all of which work together to provide a high-quality Sinhala TTS experience for our mobile-based Sinhala book reader application. We are committed to offering a holistic solution that improves accessibility, inclusiveness, and the overall reading experience for visually impaired persons in the Sinhala-speaking world by combining cutting-edge technologies and a strong grasp of the grammatical subtleties of Sinhala. Audio Sight is more than simply a program; it is a transforming instrument that opens a world of literary possibilities for individuals in need.

## 10. REFERENCES

- [1] WHO, *World report on vision*, vol. 214, no. 14. 2019. [Online]. Available: <https://www.who.int/publications-detail/world-report-on-vision>
- [2] D. S. S. De Zoysa, J. M. Sampath, E. M. P. De Seram, D. M. I. D. Dissanayake, L. Wijerathna, and S. Thelijagoda, "Project Bhashitha - Mobile based optical character recognition and text-to-speech system," *13th Int. Conf. Comput. Sci. Educ. ICCSE 2018*, no. Iccse, pp. 623–628, 2018, doi: 10.1109/ICCSE.2018.8468858.
- [3] M. Awad, J. El Haddad, E. Khneisser, T. Mahmoud, E. Yaacoub, and M. Malli, "Intelligent eye: A mobile application for assisting blind people," *2018 IEEE Middle East North Africa Commun. Conf. MENACOMM 2018*, no. September, pp. 1–6, 2018, doi: 10.1109/MENACOMM.2018.8371005.
- [4] P. Jayawardhana, A. Aponso, N. Krishnarajah, and A. Rathnayake, "An Intelligent Approach of Text-To-Speech Synthesizers for English and Sinhala Languages," *2019 IEEE 2nd Int. Conf. Inf. Comput. Technol. ICICT 2019*, no. May, pp. 229–234, 2019, doi: 10.1109/INFOCT.2019.8711051.
- [5] A. Mishangi, "Android based sinhala document reader for visually impaired people," 2021.
- [6] S. Gallege, "Analysis of Sinhala Using Natural Language Processing Techniques," 2010, [Online]. Available: [www.defence.lk/](http://www.defence.lk/)
- [7] S. Y. Senanayake, K. T. P. M. Kariyawasam, and P. S. Haddela, "Enhanced Tokenizer for Sinhala Language," *2019 Natl. Inf. Technol. Conf. NITC 2019*, pp. 8–10, 2019, doi: 10.1109/NITC48475.2019.9114420.
- [8] R. Weerasinghe, A. Wasala, D. Herath, and V. Welgama, "NLP applications of Sinhala: TTS & OCR," *IJCNLP 2008 - 3rd Int. Jt. Conf. Nat. Lang. Process. Proc. Conf.*, vol. 2, pp. 963–966, 2008.
- [9] A. Joy\* and D. R. Saranya, "A Pilot Research on Android Based Voice Recognition Application," *Int. J. Recent Technol. Eng.*, vol. 8, no. 4, pp. 7272–7277, 2019, doi: 10.35940/ijrte.d5284.118419.
- [10] G. Na, G. N. Surname, G. N. Surname, G. N. Surname, G. N. Surname, and G. N. Surname, "Mobile Base Sinhala Book Reader For The Visually Impaired Individuals," 2007.
- [11] A. A. Kumar, B. Senthilvasudevan, and H. U. Farhan, "Translation of Multilingual Text into Speech for Visually Impaired Person," *7th Int. Conf. Commun. Electron. Syst. ICCES 2022 - Proc.*, no. Icces, pp. 60–64, 2022, doi: 10.1109/ICCES54183.2022.9835819.
- [12] D. H. Klatt, "Review of text-to-speech conversion for English," *J. Acoust. Soc. Am.*, vol. 82, no. 3, pp. 737–793, 1987, doi: 10.1121/1.395275.
- [13] K. Kumara, N. Dias, and H. Sirisena, "Automatic segmentation of given set of Sinhala

text into syllables for Speech Synthesis,” *J. Sci. Univ. Kelaniya*, vol. 3, no. January 2007, pp. 53–62, 2011, doi: 10.4038/josuk.v3i0.2738.

- [14] R. Weerasinghe, A. Wasala, V. Welgama, and K. Gamage, “Festival-si : A Sinhala Text-to-Speech System”.
- [15] L. Nanayakkara, C. Liyanage, P. T. Viswakula, T. Nadungodage, R. Pushpananda, and R. Weerasinghe, “A Human Quality Text to Speech System for Sinhala,” *6th Work. Spok. Lang. Technol. Under-Resourced Lang. SLTU 2018*, no. February 2019, pp. 157–161, 2018, doi: 10.21437/SLTU.2018-33.
- [16] A. Wasala, R. Weerasinghe, and K. Gamage, “Sinhala grapheme-to-phoneme conversion and rules for Schwa epenthesis,” *COLING/ACL 2006 - 21st Int. Conf. Comput. Linguist. 44th Annu. Meet. Assoc. Comput. Linguist. Proc. Main Conf. Poster Sess.*, no. July, pp. 890–897, 2006, doi: 10.3115/1273073.1273187.
- [17] Y. Karunanayake, U. Thayasivam, and S. Ranathunga, “Sinhala and Tamil Speech Intent Identification from English Phoneme Based ASR,” *Proc. 2019 Int. Conf. Asian Lang. Process. IALP 2019*, pp. 234–239, 2019, doi: 10.1109/IALP48816.2019.9037702.
- [18] S. Ueno, M. Mimura, S. Sakai, and T. Kawahara, “Synthesizing waveform sequence-to-sequence to augment training data for sequence-to-sequence speech recognition,” *Acoust. Sci. Technol.*, vol. 42, no. 6, pp. 333–343, 2021, doi: 10.1250/ast.42.333.
- [19] L. Nanayakkara, C. Liyanage, P. T. Viswakula, T. Nadungodage, R. Pushpananda, and R. Weerasinghe, “A Human Quality Text to Speech System for Sinhala,” *6th Work. Spok. Lang. Technol. Under-Resourced Lang. SLTU 2018*, no. August, pp. 157–161, 2018, doi: 10.21437/SLTU.2018-33.
- [20] H. Miyauchi, “A systematic review on inclusive education of students with visual impairment,” *Educ. Sci.*, vol. 10, no. 11, pp. 1–15, 2020, doi: 10.3390/educsci10110346.

## 11. APPENDICES

### 11.1. Frontend Design

#### 11.1.1. Splash Screen

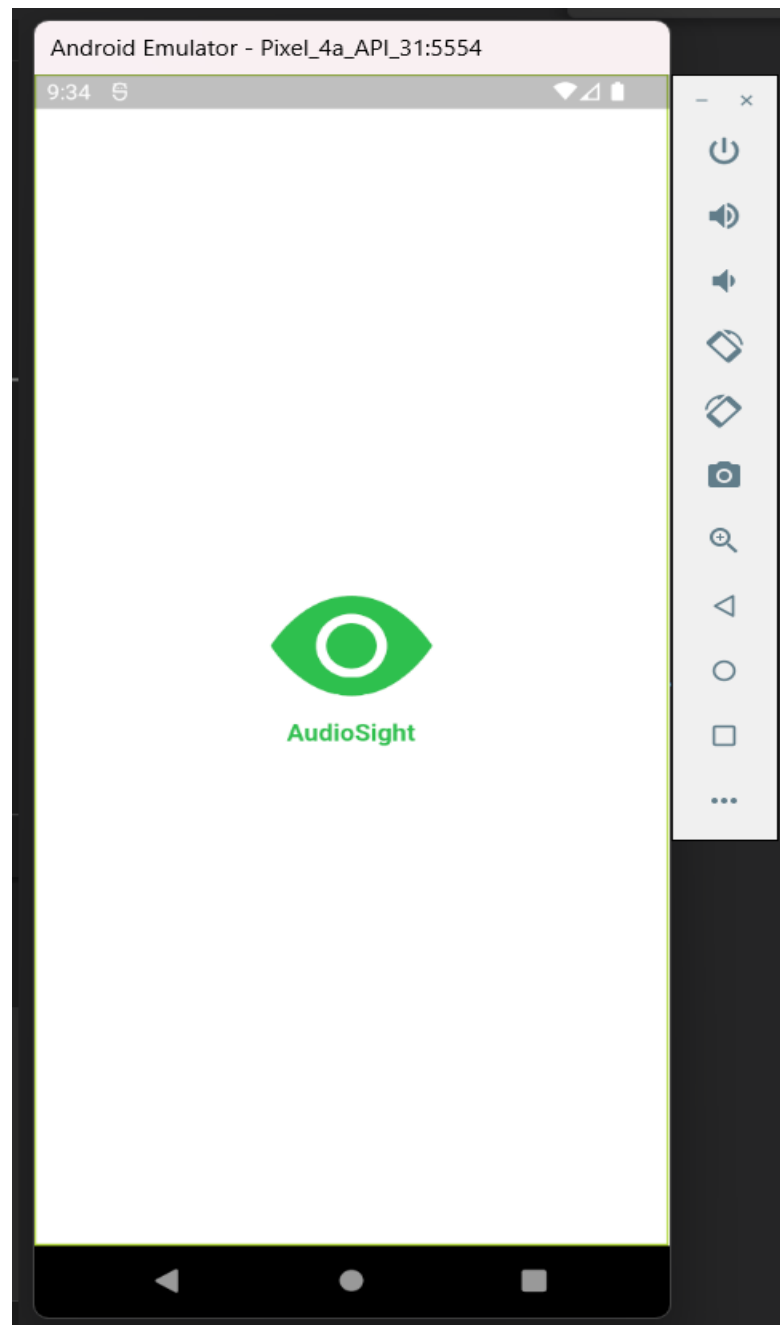


Figure 22: Splash Screen Loading



### 11.1.2. Home Screen

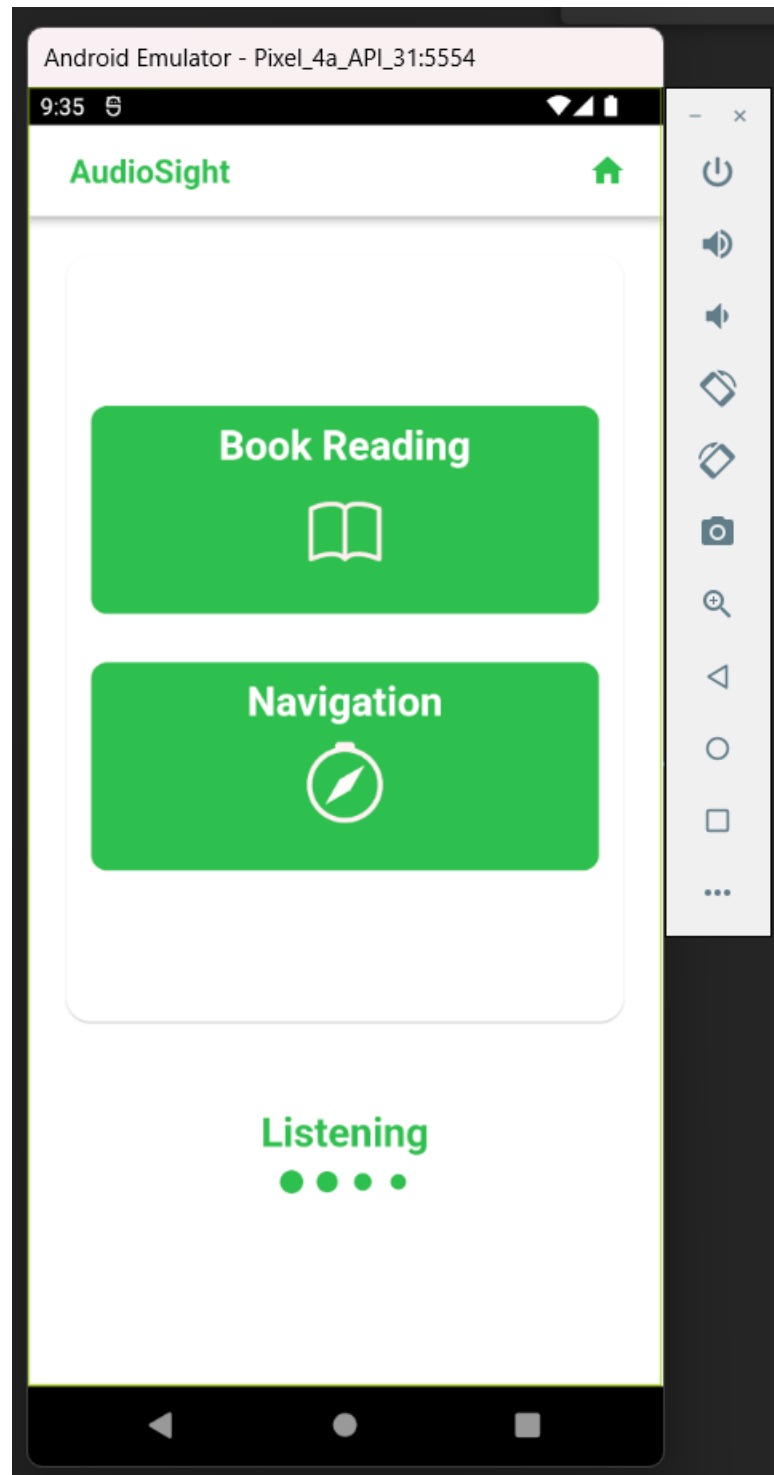


Figure 23: Home Screen

### 11.1.3. Book Reading



Figure 24: Book Reading

#### 11.1.4. Image Captioning

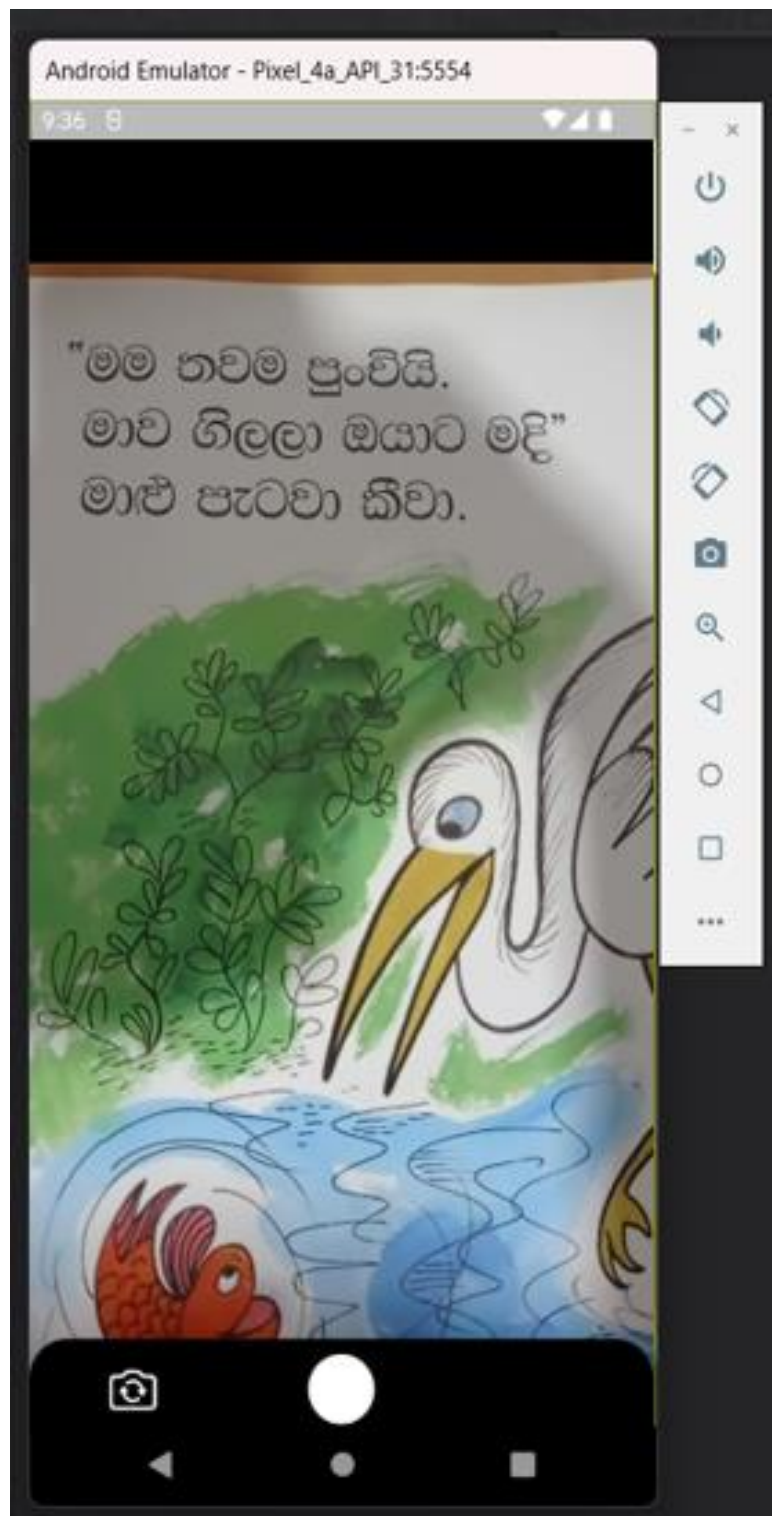


Figure 25: Image Captioning

### 11.1.5. Voice Navigation & Object Detection

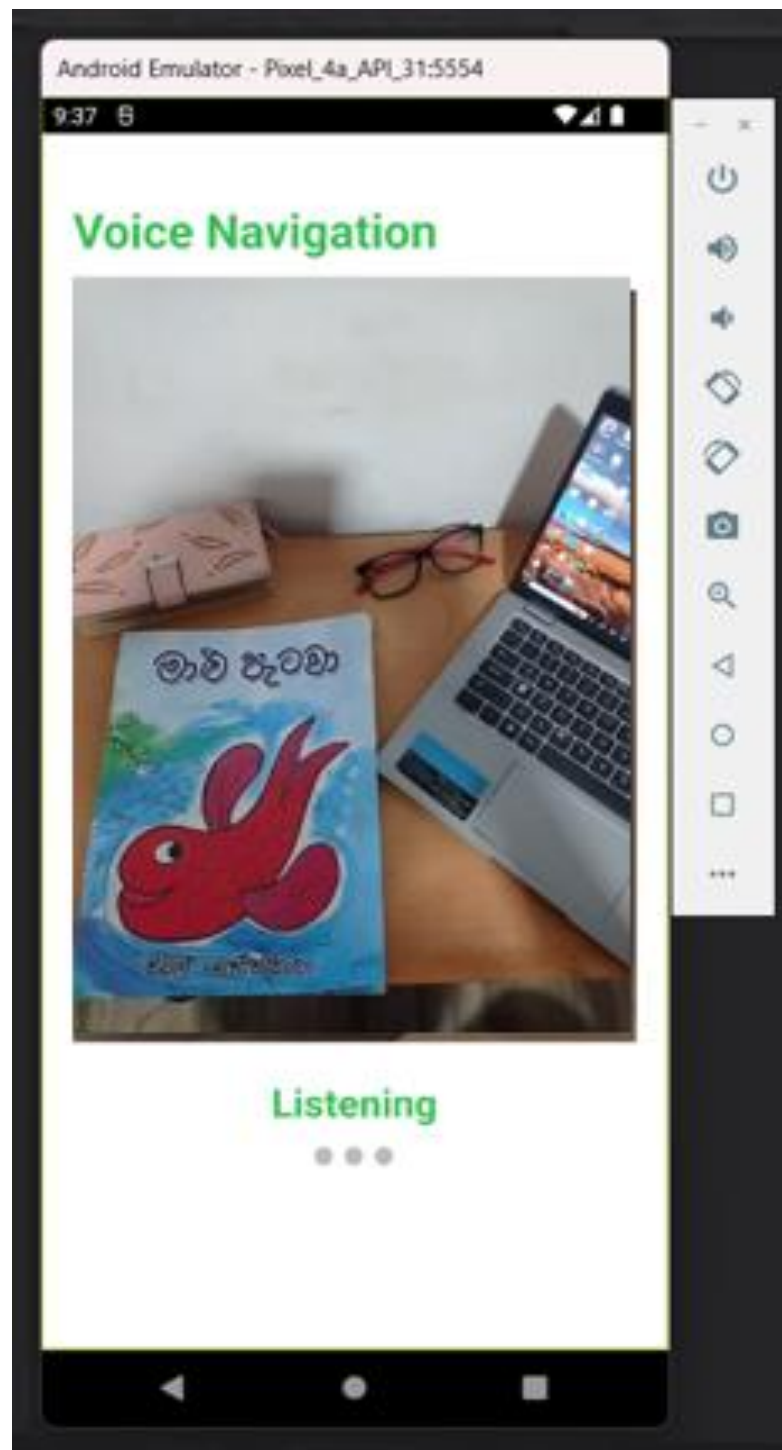


Figure 26: Voice Navigation & Object Detection

### 11.1.6. Application Working on Local Machine

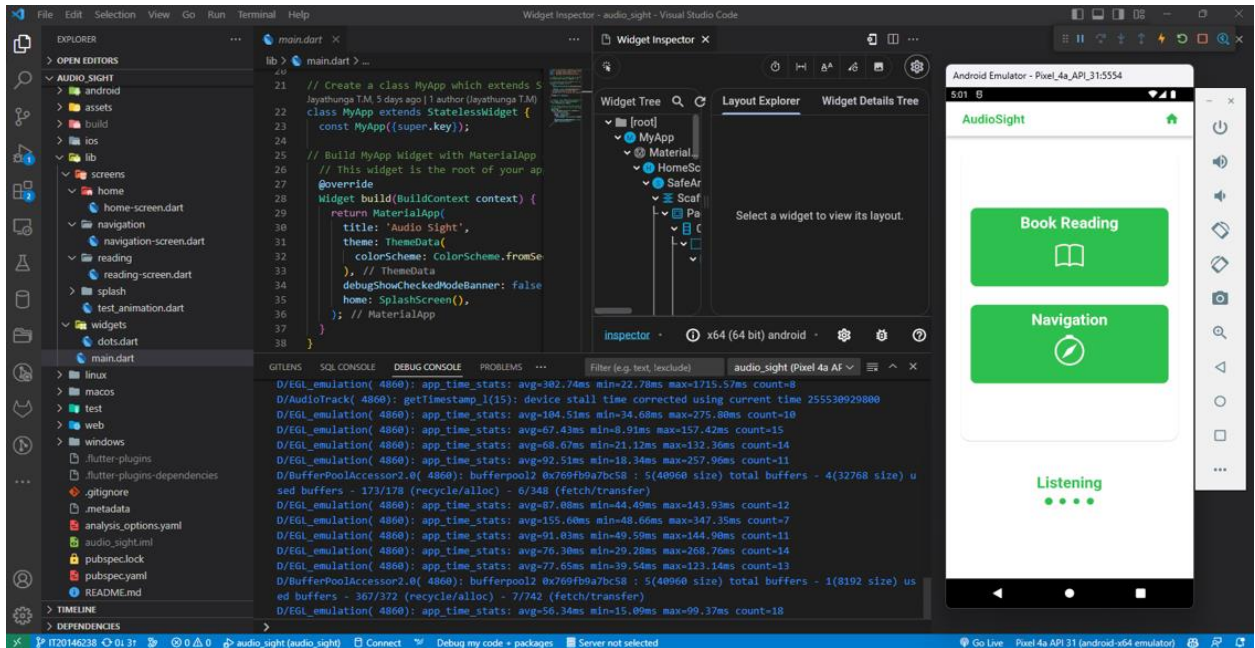


Figure 27: Application Working on Local Machine

## 11.2. Backend Design

### 11.2.1. Import Libraries & Modules

```
import os
import torch
import librosa
import numpy as np
from gtts import gTTS
import torch, json, os
import soundfile as sf
from tokenizers import Tokenizer
from dataclasses import dataclass
from collections import defaultdict
from datasets import load_dataset, Audio
from typing import Any, Dict, List, Union
from datasets import Audio, Dataset, Value, Features, load_dataset
from transformers import SpeechT5Processor, SpeechT5ForTextToSpeech, SpeechT5HiFiGAN
```

Figure 28: Import Libraries and Modules

### 11.2.2. Transformers TTS Library Models

```
processor = SpeechT5Processor.from_pretrained("microsoft/speecht5_tts")
model = SpeechT5ForTextToSpeech.from_pretrained("microsoft/speecht5_tts")
vocoder = SpeechT5HiFiGAN.from_pretrained("microsoft/speecht5_hifigan")
```

Figure 29: Text-to-Speech Transformers Library Models

### 11.2.3. Create Dataset for Sinhala Text-to-Speech (TTS)

```
def SinhalaTTSDataset(mapping_json = 'data/tts/file-mapping.json'):
    with open(mapping_json) as f:
        mapping = json.load(f)

    data = {}
    data["audio"] = []
    data["normalized_text"] = []
    for val_dict in mapping.values():
        audio_path = f"data/tts/wavs/{val_dict['newfn']}"
        sinhala_text = val_dict['sinhala']
        if os.path.exists(audio_path):
            audio, sampling_rate = librosa.load(audio_path, sr=16000)
            audio_data = {
                "path": audio_path,
                "array": audio,
                "sampling_rate": sampling_rate
            }
            data["audio"].append(audio_data)
            data["normalized_text"].append(sinhala_text)

    return Dataset.from_dict(data)
```

Figure 30: Mapping Datasets

### 11.2.4. Load Dataset

```
[5]: try:
      dataset = SinhalaTTSdataset()
    except:
      dataset = load_dataset(
          "facebook/voixpopuli",
          "nl", split="train"
      )
```

Python

Figure 31: Load Datasets

### 11.2.5. Accessing the Tokenizer from the Processor

```
[6]: dataset = dataset.cast_column("audio", Audio(sampling_rate=16000))
      tokenizer = processor.tokenizer
```

Python

Figure 32: Accessing the Tokenizer

### 11.2.6. Text Processing & Building a Vocabulary

```
[7]: def extract_all_chars(batch):
      all_text = " ".join(batch["normalized_text"])
      vocab = list(set(all_text))
      return {"vocab": [vocab], "all_text": [all_text]}

      vocabs = dataset.map(
          extract_all_chars,
          batched=True,
          batch_size=-1,
          keep_in_memory=True,
          remove_columns=dataset.column_names,
      )

      replacements = [
          ('ā', 'a'),
          ('c', 'c'),
          ('ē', 'e'),
          ('ē', 'e'),
          ('ī', 'i'),
          ('ī', 'i'),
          ('ō', 'o'),
          ('ū', 'u'),
      ]

      def cleanup_text(inputs):
          for src, dst in replacements:
              inputs["normalized_text"] = inputs["normalized_text"].replace(src, dst)
          return inputs

      dataset = dataset.map(cleanup_text)

      dataset_vocab = set(vocabs["vocab"][0])
      tokenizer_vocab = {k for k, _ in tokenizer.get_vocab().items()}
```

Python

Figure 33: Sinhala Text Pre-Process

### 11.2.7. Generate Speaker Embedding from Audio Waveforms

```
def create_speaker_embedding(waveform):  
    with torch.no_grad():  
        try:  
            speaker_embeddings = speaker_model.encode_batch(torch.tensor(waveform))  
            speaker_embeddings = torch.nn.functional.normalize(speaker_embeddings, dim=2)  
            speaker_embeddings = speaker_embeddings.squeeze().cpu().numpy()  
        except:  
            speaker_embeddings = np.random.rand(512, )  
    return speaker_embeddings
```

Figure 34: Create Speaker Embedding from the Audio Waves

### 11.2.8. Prepare Dataset

```
def prepare_dataset(example):  
    # load the audio data; if necessary, this resamples the audio to 16kHz  
    audio = example["audio"]  
  
    # feature extraction and tokenization  
    example = processor(  
        text=example["normalized_text"],  
        audio_target=audio["array"],  
        sampling_rate=audio["sampling_rate"],  
        return_attention_mask=False,  
    )  
  
    # strip off the batch dimension  
    example["labels"] = example["labels"][0]  
  
    # use SpeechBrain to obtain x-vector  
    example["speaker_embeddings"] = create_speaker_embedding(audio["array"])  
  
    return example
```

Figure 35: Prepare Datasets

### 11.2.9. Mapping Function to the Dataset

```
dataset = dataset.map(  
    prepare_dataset, remove_columns=dataset.column_names,  
)  
  
def is_not_too_long(input_ids):  
    input_length = len(input_ids)  
    return input_length < 200  
  
dataset = dataset.filter(is_not_too_long, input_columns=["input_ids"])
```

Figure 36: Mapping Datasets



### 11.2.10. Data Collator

```
@dataclass
class TTSDDataCollatorWithPadding:
    processor: Any
    def __call__(self, features: List[Dict[str, Union[List[int], torch.Tensor]]]) -> Dict[str, torch.Tensor]:
        input_ids = [{"input_ids": feature["input_ids"]} for feature in features]
        label_features = [{"input_values": feature["labels"]} for feature in features]
        speaker_features = [feature["speaker_embeddings"] for feature in features]
        # collate the inputs and targets into a batch
        batch = processor.pad(
            input_ids=input_ids,
            labels=label_features,
            return_tensors="pt",
        )
        # replace padding with -100 to ignore loss correctly
        batch["labels"] = batch["labels"].masked_fill(
            batch.decoder_attention_mask.unsqueeze(-1).ne(1), -100
        )
        # not used during fine-tuning
        del batch["decoder_attention_mask"]
        # round down target lengths to multiple of reduction factor
        if model.config.reduction_factor > 1:
            target_lengths = torch.tensor(
                [len(feature["input_values"]) for feature in label_features]
            )
            target_lengths = target_lengths.new(
                [length - length % model.config.reduction_factor for length in target_lengths]
            )
            max_length = max(target_lengths)
            batch["labels"] = batch["labels"][:, :max_length]
        # also add in the speaker embeddings
        batch["speaker_embeddings"] = torch.tensor(speaker_features)
        return batch
```

Figure 37: Data Collector Class

### 11.2.11. Transformers Library is Used to Sequence to Sequence Model

```
data_collator = TTSDDataCollatorWithPadding(processor=processor)

features = [
    dataset["train"][0],
    dataset["train"][1],
    dataset["train"][20],
]

batch = data_collator(features)
```

Figure 38: Keys and Shapes of batch Size

### 11.2.12. Text to Speech Model Training (Sequence Model)

```
from transformers import Seq2SeqTrainingArguments

training_args = Seq2SeqTrainingArguments(
    output_dir="models/sinhala-text-to-speech",
    per_device_train_batch_size=2,
    gradient_accumulation_steps=2,
    learning_rate=1e-5,
    warmup_steps=500,
    max_steps=50,
    gradient_checkpointing=True,
    fp16=False,
    evaluation_strategy="steps",
    per_device_eval_batch_size=2,
    save_steps=10,
    eval_steps=10,
    logging_steps=25,
    report_to=["tensorboard"],
    load_best_model_at_end=True,
    greater_is_better=False,
    label_names=["labels"],
    push_to_hub=True,
)
```

Figure 39: Seq2Seq Model Training