Audio Phonic

An Audiovisual Auto corrector

By

Capt Muhammed Bilal Sadiq

Capt Danish Zarif

Capt Arslan Waqar

Capt Shehryar Ali Ahmad

Supervised by:

AP Mobeena Shahzad

Submitted to the faculty of Department of Software Engineering,

Military College of Signals, National University of Sciences and Technology, Islamabad,

in partial fulfillment for the requirements of B.E.S.E Degree in Software Engineering.

June 2023

DEDICATION

To Allah Almighty

&

Our Respected Faculty and Parents

CERTIFICATE OF CORRECTNESS AND APPROVAL

This is to officially state that the thesis work

contained in this report

"Audio Phonic An Audiovisual Auto corrector"

is carried out by

Capt Muhammad Bilal Sadiq

Capt Danish Zarif

Capt Arslan Waqar

Capt Shehryar Ali Ahmad

under my supervision and that in my judgement, it is fully ample, in scope and excellence, for the degree of Bachelor of Software Engineering in Military College of Signals, National University of Sciences and Technology (NUST), Islamabad.

Approved by

Supervisor

AP Mobeena Shahzad

Department of SE, MCS

Date: ___________

DECLARATION OF ORIGINALITY

We hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification in either this institute or anywhere else.

ACKNOWLEDGEMENTS

We'd want to convey our heartfelt gratitude to our parents, advisor, and co-advisor, Mobeena Shahzad and Naima Altaf, as well as all the other faculty members who inspired us and assisted us throughout the project. They are always available to help us, whether it is from our advisors or other faculty members, and they have given us the moral support to work on this wonderful idea of the audiovisual autocorrector. They also assisted us in conducting extensive research prior to implementing this project, and we learned a lot of new things, such as speech recognition, voice generation, text-to-speech and vice versa. We are thankful to them (faculty and advisors). Secondly, we would also like to thank our parent and friends who helped us a lot in testing the final product and finalizing this project within limited time frame.

Plagiarism Certificate (Turnitin Report)

This thesis has 9% similarity index. Turnitin report endorsed by Supervisor is attached.

Capt Muhammad Bilal Sadiq

Capt Danish Zarif

Capt Arslan Waqar

Capt Shehryar Ali Ahmed

Signature of Supervisor

Table of Contents

Contents

List of Figures

List of Tables

ABSTRACT

Media industry, especially after the rise of the internet as it is readily availableto all households, there is a lot of increase in the internet users both as well as consumer and content creators. Now the ultimate problem of proof-editing the videos and audio created by the creators and voice-over artists respectively. According to the statistics, more than 40% of the pupils living in Ireland, find it cumbersome in preparing videos and sounds for the internet. This leads us to the utmost problem editors and voice-over artists face, which is proofreading, changing the audio/video again and again to matchperfection which takes a lot of time, Secondly, the problem is with the audiobook creators, one small mistake in the voice generation and you'll have to start all over again from the Last sentence and often from the last paragraph to avoid any feelings of unmatched voices. This system i.e., "AudioPhonic – an audiovisual auto corrector" aims to automate the process of making changing in the audio as well as video files . System will provide many features such as, upload your audio as well video files, get editable transcript (change the audio transcript) and changings will be added as the original voiceon the audio. Finally, you can download the final version of the audio/video files from the system.

The system will be easy to use and provide a great user experience throughout the process. It will help the content creators as well as the voice-over artists to minimizetheir time of editing the voice/video due to the process of automation using deep neuralnetworks.

There will be a lot of improvements and future work that can be done as well ascater in this project, such as shifting all the learning practices to online has left us with a need for a solution to provide quality content despite the internet connectivity issues.This project also has the branch to solve this internet connectivity issue by using the technology of deepfake and TTS (Text-to-speech) solutions. We can work for online transfer of text and generation of audio on the other end. According to a report, more than 1.2 million Californian students are reported to have internet or computer issues for learning online. And the others are also getting either a very low bandwidth or a constant connectivity issue for their network. Hence, we can solve this problem too by providing them a solution like free Facebook for their online lectures and meetings. Moreover, we make the edits in the audio file on the face (in case of video) more realistic and accurate.

Chapter 1

INTRODUCTION

The programme is what offers its services so that you can take back control of your audio and video gestures. This tool has you covered for anything from paying attention to your pronunciation to adding dialogues you forgot. In order to make quick changes to the audio or video, AudioPhonic is helpful while creating audiobooks as well as in the YouTube community. Our speech recognition and text-to-speech (AudioPhonic) technology-based audiovisual auto corrector solution was primarily built for content producers to streamline their process of editing audio/visual files utilising the most recent deep learning technology. The system is built and deployed as a Web Application, and it provides very easy user-experience for the users to configure and see outputs. Users canuse almost any other web application, every process is self-explanatory, and uses a veryeasy approach so that any non-technical person can easily use and enjoy the product.

It is a simple-to-go website. Right now, there is no account creation system on the website. Any user can go on the platform and benefit from the services. The first step provides an option to upload your audio or video files. AudioPhonic will then be able to generate the transcript from those audio/video files. Then, you will see the transcript generated from the audio/video files. You can edit the transcripts text i.e., mostly where you want to change the audio from your audio/video files. After necessary modifications, submit it to the system to apply the changed text to the audio (with the help of voice-generation). There will be graphics and visualization to show the changings the Mel-coefficients (the characteristic of the voice plus text).

The system's objective is to make it simple for users to change the audio, support voice actors, and produce audiobooks utilising cutting-edge neural network techniques rather than outdated ones. The word level editing stage is offered by AudioPhonic, a pioneer in this field of audio/video editing, to content producers, voice actors, and many others.

The main task of our Project i.e., “AudioPhonic – An audiovisual auto corrector” is to extract the transcript from the audio. This extracted transcript will be editable, and users can make edits in this text and submit the new text. The new audio will be generated according to it. For example, if in some audio, the user made some mistake in pronunciation of some words, or if he/she wanted to add words to the video. So, rather than going through the whole process again, he/she can simply upload his video on our application. Then, make changes in the transcript (add, change, or remove the text), and simply submit it, then all the changings will be applied to the audio.

Our project i.e., “AudioPhonic – an audiovisual auto corrector” is an audio auto corrector that uses multiple techniques at one place such as autonomous voice generation, speech recognition, text-to-speech etc. There are multiple operational sides that combined as well as work together to make the AudioPhonic perform useful operations such as:

- **Speech Recognition:** using Kaldi Acoustic and language model
- **Text To Speech:** using SV2TTS model
- **Programming Language:** python 3.7
- **Frontend:** Flask – python module to develop web application
- **SV2TTS:** multiple small dependencies used for model that is used for audio generation

The Final product will enable the users to upload any kind of audio files on the application. Users will be able to see the transcription of the audio. Users can make edits in the transcript and submit it. Then, a **new audio** will be generated based on the new transcription and, users can download the new version of audio.

The main functions of the product listed below are in order (read it to understand, incase of any problem)

1. User will input the audio files
2. Editable transcription will be generated from audio
3. User made edits in the transcription and submit it
4. User can finally download the new version of the audio from our application

In this **chapter 1** (i.e., introduction) of our project, we briefly discussed the reasonsfor doing this project and its target audience as well. We lay out the current problem, the generic approach as well as our brief proposed solution. In **chapter 2** (i.e., Literature review) we will discuss the background of the problem. In **chapter 3** we will briefly discuss the problem statement. In **chapter 4** (i.e., solution system) and **chapter 5** (i.e., detailed design and Architecture), we will discuss the various methodologies and more detailed solution and finally we will finish the report by identifying the implementation details in **chapter 6** (Implementation and Testing), and results in the last **chapter 7** (Results and conclusion). In **chapter 8**, there will be a section for the conclusion and future work for our project.

Chapter 2

LITERATURE REVIEW

With the advancement in the field of machine learning and deep learning-basedalgorithms, there is a huge rise in the field of solving complex problems using deep neural networks. So, here we are also using deep neural networks to make changes in the audio files.

Many individuals have found it difficult to edit audio, so we created this product to make the process as quick and easy as possible. Since this project combines several technologies, including speech recognition, voice production, and text-to-speech, we will go over the prior research in each of these areas individually.

2.1 SPEECH RECOGNITION

Speech recognition is a process to recognize the speech. It basically involves the following steps:

2.1.1 Audio Format File Conversion

In the very first step, we perform conversion. Like, if the given file is already inwav format or audio format, we don't do anything with that. But if we are given the audio in any other format, then first we will have to convert that file into wav audio format. Start with inputdata that consists of audio files of the spoken speech in an audio format such as ".wav".

2.1.2 Audio Preprocessing

Audio Preprocessing refers to the methods used to manipulate and enhance audio data before its fed into an algorithm. The goal is to improve quality and usefulness of audio data for tasks that may include speech recognition, music classification or audio compression. Audio Preprocessing involves signal filtering, signal normalization, feature extraction, time-frequency representation and data augmentation.

After converting the file, the next step is audio preprocessing. In this step, we remove any background noise, increase a person's voice or frequency if it's low, tune the voice, and make it better and clearer. There can be lots of variations. Clips might be sampled at different rates or have a different number of channels. The clips will most likely have different durations. The dimensions of each audio item will be different. Since our deep learning models expectall our input items to have a similar size, we now perform some data cleaning steps to standardize the dimensions of our audio data. We resample the audio so that every item has the same sampling rate. We convert all items to the same number of channels. All items also have to be converted to the same audio duration. This involves padding the shorter sequences or truncating the longer sequences. If the quality of the audio was poor, we might enhance it by applying a noise-removal algorithm to eliminate background noise.

2.1.3 Feature Extraction

In the context of machine learning and signal processing, audio feature extraction is used to represent data in a format that can be easily processed and analyzed by algorithms.

One popular audio feature extraction method is the Mel-frequency cepstral coefficients (MFCC) which has 39 features.

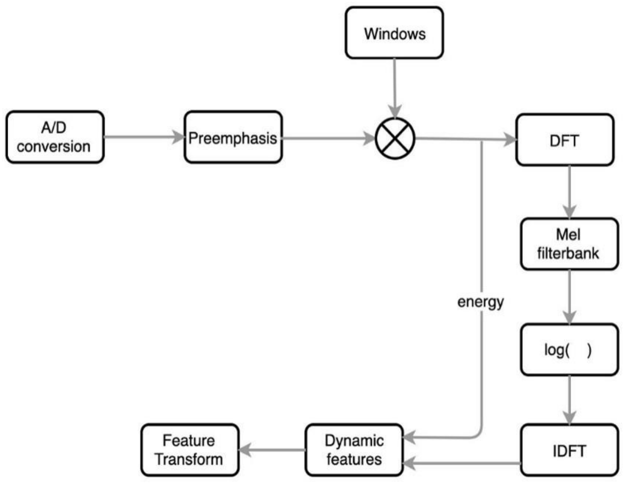


Figure 1: Feature extraction using Mel Spectrogram

The Key features include:

- Remove vocal fold excitation (F0) — the pitch information
- Make the extracted features independent.
- Adjust to how humans perceive loudness and frequency of sound.
- Capture the dynamics of phones (the context).

This raw audio is now converted to Mel Spectrograms. A Spectrogram captures the nature of the audio as an image by decomposing it into the set of frequencies that are included in it. For human speech, in particular, it sometimes helps to take one additionalstep and convert the

Mel Spectrogram into MFCC (Mel Frequency Cepstral Coefficients). MFCCs produce a compressed representation of the Mel Spectrogram by extracting only the most essential frequency coefficients, which correspond to the frequency ranges at which humans speak. We have now transformed our original raw audio file into Mel Spectrogram (or MFCC) images after data cleaning and augmentation. Technologies of LDA and HLDA are used for compression.

2.1.4 Kaldi's Acoustic and Language Model

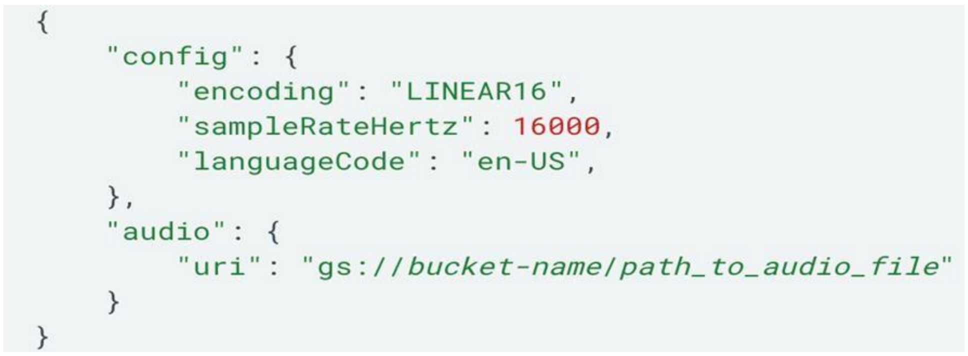
Kaldi is a state-of-the-art automatic speech recognition (ASR) toolkit, containing almost any algorithm currently used in ASR systems. You may train your own speech model using Kaldi.

Acoustic models are the statistical representations of a phoneme's acoustic information. The term 'phoneme' only loosely corresponds to the linguistic use of the term'phoneme'.

This model contains the acoustic properties of all phonemes (pronunciation of a word).Kaldi's language model works like the conditional probability of Bayes theorem using the N-grams model. It basically predicts occurrences of words coming. This model is like a finite state automaton

2.1.5 Google Speech Recognition

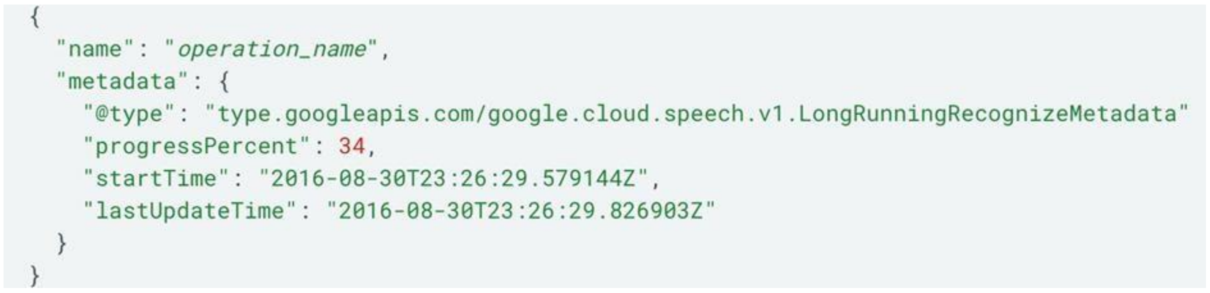
Google speech recognition is one of the tools which senses the human voice and converts it into text. This is one of the most important tools widely used now-a-days. It works in three different ways which include Synchronous, Asynchronous and Streaming recognition. In synchronous recognition, it processes the data upto 1 min or less than that. An example is shown below



```
{
    "config": {
        "encoding": "LINEAR16",
        "sampleRateHertz": 16000,
        "languageCode": "en-US",
    },
    "audio": {
        "uri": "gs://bucket-name/path_to_audio_file"
    }
}
```

Figure 2: Google Speech Recognition configurations

In asynchronous recognition, audio files of up to 480 mins can be processed. An example is given below:



```
{
  "name": "operation_name",
  "metadata": {
    "@type": "type.googleapis.com/google.cloud.speech.v1.LongRunningRecognizeMetadata"
    "progressPercent": 34,
    "startTime": "2016-08-30T23:26:29.579144Z",
    "lastUpdateTime": "2016-08-30T23:26:29.826903Z"
  }
}
```

Figure 3: Google Speech Asynchronous Recognition

The last one streaming recognition is used for real time capturing purposes. Google speech recognition is a perfect tool for use.

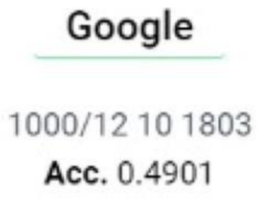
It has many benefits like:

- Adapts the speech
- work on live streaming speech recognition
- Filter the content
- work on multiple channels and their recognition
- work on the models which are highly likely to be domain specific,
- Automatically detect the language and many more

So why didn't we use it after having many benefits?

In our model, we tried to use google speech recognition. But after uploading the audio it gave the results which were not according to our expectations.

1. We gave it audio that was speech-based and spoke the mathematical numbers. As a result, the text that the Google speech recognition model gave us treated the numbers as mathematical numbers rather than language. We requested the numbers as text, but it sent us the numbers instead. The first flaw in it was this.
2. The second problem was that it didn't provide much accuracy. The accuracy was too low to work using this model.



Google

1000/12 10 1803

Acc. 0.4901

Figure 4: Google Prediction and accuracy

2.1.6 Dragon Professional Speech Recognition

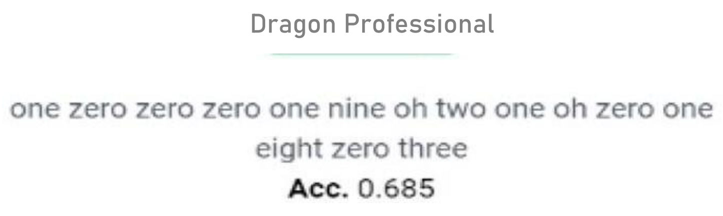
Onto the next model, which we were unable to use. Dragon Professional is a speech recognition business that use AI and creates voice assistants for businesses to use. Although it appears to be a fantastic model to work on and create assistants, using it for our project was certainly a mistake.

This model too has many benefits like:

- works fast and very accurately recognize the voice
- Platform independent
- can make customizable solutions

So why didn't we use this model?

We tried to use this model by providing it the voice. But as a result, we couldn't get the accuracy we wanted to have.



Dragon Professional

one zero zero zero one nine oh two one oh zero one eight zero three

Acc. 0.685

Figure 5: Dragon professional Prediction and accuracy

Given the picture above, you can see the accuracy which is too low to work with.

2.1.7 DeepGram Speech Recognition

As well as the above, this is also a speech recognition company. It works with AI to work on NLP and create bots, devices of IoT and many more.

It is capable of creating any type of bots, which creates interactions with humans. It is used to create devices which can interact with anyone who wants to write and speak with them. Facebook is its owner.

DeepGram also has many benefits which are given below:

- It provides support for programming language and integration
- It is not a free platform
- It is a Software as a Service platform
- It can help you visualize the experience

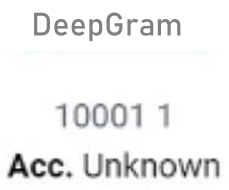
- It can create the handsfree interfaces of voice

It also had some disadvantages:

- It doesn't support the tools of integration of third party
- Considering task performance training engine checks and takes time
- It becomes really slow when load of stories increases

This company offered may benefits, but we couldn't use it in our project because:

1. It was paid
2. It didn't take our complete audio
3. It didn't provide the confidence



DeepGram

10001 1

Acc. Unknown

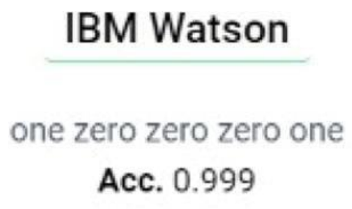
Figure 6: Deep Gram Prediction and accuracy

2.1.8 IBM Watson Speech Recognition

Similar to the methods mentioned above, it likewise turns human voice into text. For our project, this model works out fairly well. Ibm employed the native cloud solution. Artificial intelligence deep learning algorithms are required. After learning, it then applies what it has learned about grammar. Practicing with models can help with language structure. It also works with signal compositions that include audio and video. It then produces customised text transcription from speech recognition. It offers the same advantages as the methods mentioned before.

But why didn't we use this model?

This was the model which gives good accuracy but it didn't take the complete sample and didn't give the complete prediction. It maybe removes some sample pointswhich at the end resulted in wrong prediction.



IBM Watson

one zero zero zero one

Acc. 0.999

Figure 7: IBM Watson Prediction and accuracy

2.2 TEXT-TO-SPEECH (TTS)

The next phase of our project deals with the conversion of text-to-speech. There are several methods to accomplish this phase. The primary objective of this stage is to faithfully recreate the speaker's voice and create edits that are as realistic as possible.

The first important step in TTS is to **extract features of the Speaker** (with the help of speaker recognition model). This model will help us to extract the basic features of the human voice (that differentiate them with each other) i.e. frequency, pitch, high or low notes etc. The Mel Frequency Cepstral Coefficient Method serves as the basis for this model (MFCC). We will ultimately have all of the voice's features in a vector quantized form thanks to this approach, which is very beneficial for extracting features.

Now, the next phase is to synthesize or regenerate the voice according to these features. So, for this I will be explaining some of the techniques that can help us in this next.

2.2.1 Concatenative Synthesis

This is the very basic form of voice regeneration. In this technique, simply the phonemes (pronunciation of a particular word) of all the words that are appearing, are **concatenated**. After the concatenation, we will have a simple layer of audio file.

The main disadvantage of this method is that the voice that is produced is frequently not human, or more accurately, a robotic voice. Because of this, this method does not effectively aid us in completing our mission.

2.2.2 Audio Synthesis from phonemes

This method of voice regeneration is very intriguing. This approach begins by breaking down each piece of text into its individual phonemes. The segmentation model is used in this method. This model examines how phonemes are segmented, including gaps between phoneme segments, the length of a phoneme, the frequency at which a phoneme must occur in order to produce audio that is as similar to the original as feasible, and other segmentation considerations. After passing through this model, we have successfully generated the audio file.

- Despite this segmentation model, this technique also has some drawbacks. The major drawback of this technique is it is not applicable or good for creating a customized speech. (Which is major requirement of this milestone)

2.2.3 AdaSpeech

This model solves our problem very efficiently. This technique starts with the conversion of text into phonemes. The major advantage of this model is its **Acoustic Condition Modeling.** This model extracts the features of the human voice (from the video). These characteristics include vibrations, high or low notes, pitch, frequency, and others. Hence, these features aid in voice regeneration (as close as possible to the original voice). A layer of audio will then be produced after this. The voice will then be tuned in accordance with the features and variance after the variance adapter (already generated by the Acoustic Condition Model). After tuning, a Mel Decoder is used to decode the voice, and in the end, we were able to create a voice that was as near to a human voice as feasible.

Till now, due to the accuracy of the AdaSpeech Model, we have a plan to use this model in our project. It may change as we dive deeper into the project.

2.2.4 K-Means clustering

At first, we start with the very basic, naive yet very powerful approach to create the audio voice. i.e., the use of k-means clustering machine learning algorithm. Basically, it is a type of unsupervised learning, in which there is a lot of raw data with no labels. Model learns and trains on its own by making clusters of the similar data points. Similarly, we are using the same approach i.e., we are training our model with a lot of the audio files of different users and they are making the clustering have the samevoices to a certain index.
We have used the LibriSpeech Automatic Speech Recognition dataset to train the modelon K-means clustering machine learning algorithm

- **DataSet**

LibriSpeech is a collection of roughly 1,000 hours of 16kHz read English speech. The information originates from audiobooks read as part of the LibriVox project. Almost 8,000 public domain audio books have been generated by the volunteer-run LibriVox project, the majority of which are in English. The vast majority of the recordings are based on Project Gutenberg texts, which are likewise considered to be works of public interest.
After creating the multiple clusters depending on the audio features, with the closest one in single clusters. We are calculating the mean of that cluster to let's say create a mean voice (which is the mean of all of the features of the voice). Then at the end we will have the trained model, which is ready to create the audio from the text given.so, whenever a new voice is mapped, based on the multiple features of the audio. Then it will use the mean voice of the nearest cluster, to create the voice of the user.

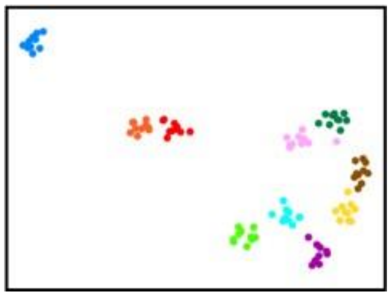


Figure 8: Voice Clustering Approach

Drawbacks:

There are some major drawbacks in the accuracy of this model, that is causing a lot of disorientation in the voice created and merged to the original voice.

- The dataset is used to train the model, which requires extensive pre-processing. Hence, if we need to retrain the model in the future, we will have to start over from scratch.
- If there are any problems with the clustering, such as noise or low pitch, such problems will also show up in the mean voice of the clustering and consequently in the final outcome.
- Producing the mean voice, which takes the average of all the audio's voice characteristics, is a fantastic idea, but it won't provide us with the accuracy of 90%. It provides us with an accuracy of 60–70% at most. due to the separation between a cluster's core and the original.
- A lot of training of the modified k-means clustering is required to handle the above problem which is computationally very expensive and requires a lot oftime to train.

To deal with this kind of issue, we have devised and used another approach to create the voice along with the extraction of the multiple important features of the voice.

2.3 DEEPFAKE

According to the last component of our solution, we are remaking the user's video in accordance with the text adjustments made. The Deepfake technologies are the sole basis for this reconstruction of the video. Deepfakes essentially involve decoding someone else's encodes using their decoder. We'll examine each of these tactics separately.

2.3.1 Emotion Classification

The method utilised to execute Deep Fakes begins with the classification of emotions, which explains "how we will genuinely react to the next post we will read" by inferring from our present facial expressions and body language. The primary objective of emotion categorization is to provide the appropriate emotion for the subsequent segment of the video.

The emotion generation procedure is a function of that person's age, gender, topic of discussion and powerful current emotions. That's why, 2 CNN models (Convolutional Neural Networks) are trained to predict age and sex while 2 other RNN models (Recurrent Neural Networks) are trained to predict the topic during the video as well as the dominant emotion. Then FOL functions are used to combine the results to introduce the newly generated emotion to the next video.

2.3.2 Auto Encoders

Auto-encoders are mostly used for **lip-syncing** Deep Fakes where the movement of the lips are matched with the respective piece of text. Auto encoders work like charm using the process of swapping.

They generate an encoder that conducts picture compression and dimension reduction on a video frame's worth of data before producing a decoder for the same movie. However, it switches the decoders after constructing two encoders and two decoders for two experimental films. In this stage, the video B is altered with relation to the original video A. Deepfakes are commonly produced with auto encoders. While the eye motions and face alignments produced by this method using its standard algorithms are unsatisfactory. Even though GANS can make it more effective.

2.3.3 GANS (Generative Adversarial Networks)

GANS has been a trending topic on Machine Learning for the past 10 years now. It is considered extremely efficient for performing learning tasks due to its capability of learning itself from even a smaller dataset of training examples

A GANS's operation is totally dependent on the cooperation of its discriminator and generator. The discriminator is first given a result or aim that needs to be attained. Random data points are generated by the generator and then classified by the discriminator. They first diverge from the desired findings, and the discriminator can immediately detect their false positives.

But from this experience the generator keeps on learning and at last, generates a hard to classify piece of object that the discriminator can't classify as real or fake. This is the point where we have a generator of our choice.

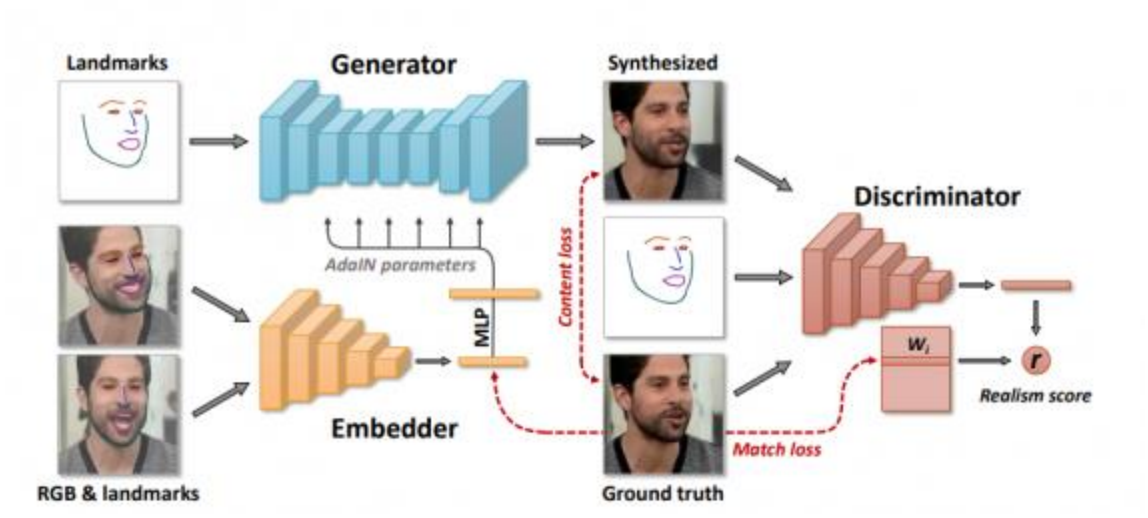


Figure 9: Deep Fakes approach with GANS

Chapter 3

PROBLEM STATEMENT

There are multiple issues in the field of media industry, especially in the field of audio/video editing.

3.1 Audiobook Creators

Editing issue affects those who create audiobooks; if there is even a minor error, you must restart from the very last sentence, and frequently from the very last paragraph, to prevent any impression of mismatched voices. Professional editors must utilise expensive software to alter the audio, leaving it essentially alone. Editing the audiobooks is incredibly laborious and exhausting for every voice actor in the world.

It will be incredibly challenging for you to edit a voice-over again in your own voice once you've already done it. You cannot write sentences in your native language and add words. Obviously, you will have to recreate it if you want to add more words and sentences, but that is still rather laborious.

While creating the voice-over, the artists usually don’t focus on quality and pronunciation of words. Which will still put them into trouble during recreation. You have to create audiobooks which are high quality and listenable for other people. But again the artists end up being in trouble because of the absence of software which can edit the voice-over in one click.

Sometimes there is no consistency in the features of audio like tone, frequency, volume, noise and pronunciation. There are so many fluctuations in the audio which lead to more problems for the artists. They compromise on quality due to cost and recreating it again and again.

Vocal quality has a big impact on voice-over. A voice-over with poor quality would be regarded favourably and might be rejected by others. The process of producing an audiobook once more would be quite taxing. It is really challenging to keep the features in line with the requirements. It's quite challenging for authors to create an audiobook from start. Their time and resources are being wasted. When we take into account the entire globe, including audiobook creators, we can observe how much time and money is wasted on them in order to replicate the audiobooks. Every laptop has resources, and by creating the appropriate software, we may use them effectively.

While creating an audiobook, you need to adjust the equipment in the right wayto cover the audio completely. When a slight inconvenience happens, you have to makethe audiobook again for just a simple mistake of a few seconds. This is again very troublesome for the creator.

Initially, as the authors are learning how to produce the audiobook. Even setting up the equipment so that the voice may be clearly recorded is beyond their comprehension. At that point, students must duplicate it repeatedly while correctly arranging the equipment. This may divert the creator from the audiobook project and reduce their interest in it, both of which are negative outcomes.

The background noise should be eliminated before starting to record an audiobook. Otherwise, it can result in a disorganised voice. When you turn your head away from the microphone, the voice changes. You will need to start the audiobook over if you use the incorrect phrase. The voice can become distorted if you turn the pagein case you are reading from the page. All these factors must be catered while creating an audiobook and without AudioPhonic you wouldn't be able to just edit the audiobook in one click.

3.2 YouTube Audience

On average, **more than 150,0000 new videos** are uploaded to YouTube every minute, adding up to around 330,000 hours of video content based on an average video of 4 mins.

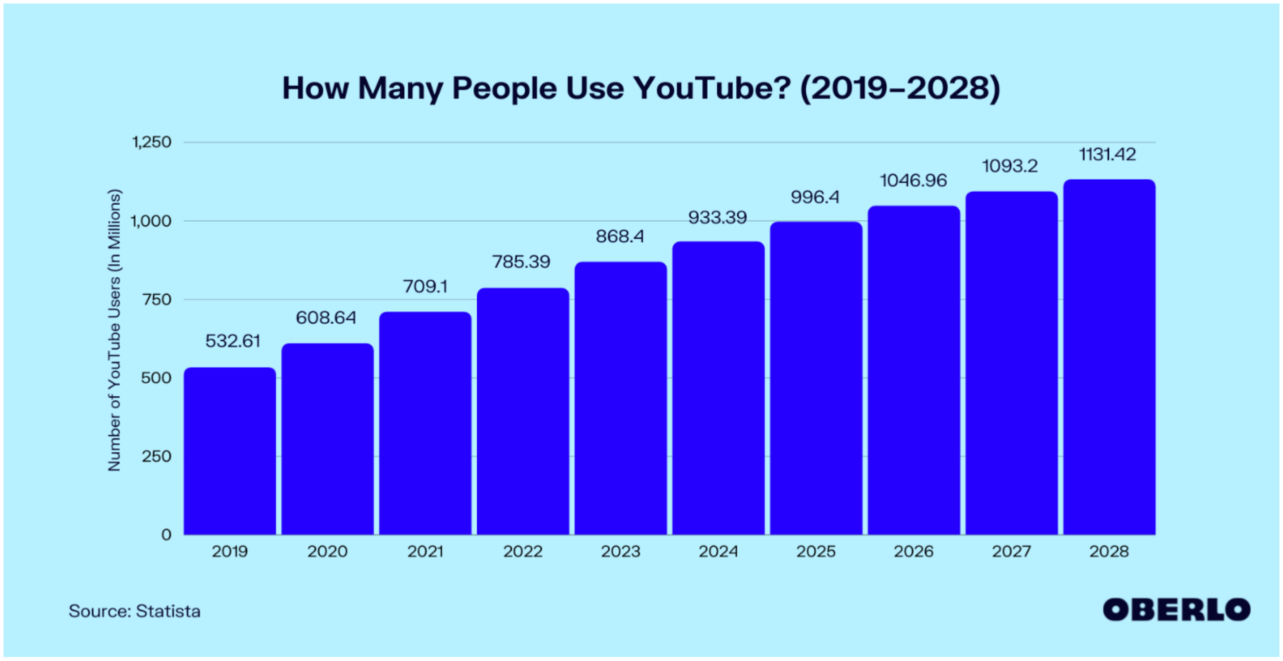


Figure 10: Countrywide Dataset of youtube channels

As per the report of Statista **2023**, YouTube channels have grown so rapidly and people are making videos frequently. By **2024**, the growth is expected to rise upto **934 million.**

YouTubers are always curious and energetic about their videos. They need to make perfect videos with the least amount of effort. But as in the present world, there isno software available which can help them edit their videos in one click. Editing softwares is available but not a single software is available for simple trimming and concatenating new clips with the previous videos.

Considering a survey of YouTubers, they feel very troublesome working with editing software available now because they have to create the videos again and again.A simple edit takes many hours.

There are various channel types, some of which are small and others of which are large, and each requires tweaking. Some videos need days or even weeks to be edited since they are small or very large. This software can solve all of these issues with a single click. Today, YouTubers don't have to stress about editing really long videos with a lot of content.

3.3 Online Learning

Lastly, with all learning activities moving online, we now need a way to deliver high-quality content despite problems with internet connectivity. This solution is also the source of this project. We can work for online text transmission and audio generation on the other end using our TTS technologies. More than 1.2 million California students are said to face internet or computer problems when learning online, according to a report. And the other networks are likewise experiencing either extremely low bandwidth or persistent connectivity issues. Thus, by giving them a solution like free Facebook for their online courses and gatherings, we may also tackle this problem.

Following COVID-19, the majority of global operations have been moved online. It could include industries like education, software development, freelancing, several factory operations, retail, offices, pharmaceuticals, the furniture industry, and many more. In this situation, when the entire educational process is moved online, teachers need good software to create high-quality movies and deliver them using little data.

Every college, university, and school has switched to online instruction. Anyone can use the software to create videos and modify them whatever they want. There are many areas in the world in which the internet is not available and people are using low data. To send videos or any other files there is a very difficult task. But using AudioPhonic we can create low bandwidth videos and send them easily.

Anyone can create videos using AudioPhonic and upload them on video platforms. So, people can watch them and learn anything from them. This is how online learning is catered in this world after covid.

Who will benefit from the System ?

The system will effectively be used and helpful for the variety of different users like:

1. Voice over artists (audio)
2. Audiobooks (audio)
3. Blog writers (audio)

Chapter 4

METHODOLOGY

One application with automated audio editing capabilities is AudioPhonic. It employs a number of interconnected modules and dependencies that act as a system. Chapter 5 discusses the system's intricate construction.

An audiovisual auto corrector called AudioPhonic operates in a very simple and uncomplicated manner. Each user can leverage the power of our application to do the following tasks in this order:

1. User will input the audio files
2. Editable transcription will be generated from audio
3. User made edits in the transcription and submit it
4. User can finally download the new version of the audio from our application

4.1 Speech-to-text

The only thing that makes speech sound to the ears is a series of air compressions and suppressions. We utilise meaningful sentences that make sense when we talk; sentences are composed of words or phrases, which are themselves composed of phonemes. As we work with sound digitally, we can't keep infinite vibrations, thus we sample the audio at a specific rate. For this project, our voice-to-text modules are sampled at a single channel sampling rate of 16000 hertz, meaning 16000 samples are collected each second. This is the optimal value for the project because anything less than it could negatively affect recognition results and anything over it could slow down performance, which is undesirable for a real-time system.

4.1.1 Kaldi Toolkit

An open source, C++-based automatic speech recognition toolbox is called Kaldi ASR. The Apache v2 licence is held by Kaldi. Kaldi was developed by Dr. Daniel Povey, an assistant professor at John Hopkins University. For voice recognition research and development, the Kaldi toolset is utilised.

Kaldi is a Linux-based program that includes pre-built models, data cleaning, and data filtering scripts that may be used for pre-training activities. The learning curve for Kaldi is

steep and setting it up is a laborious and time-consuming procedure.
Few prominent features of Kaldi are:

1. Kaldi has a wide range of support for effective implementation of linear Algebra.(i.e., back-bone of machine learning algorithm)
2. It is an open-source library. A lot of bugs and issues are resolved daily.
3. Many scripts and code bases are included that may be used for tasks other than automatic speech recognition.

4.1.2 Main components of Automatic Speech Recognition

In speech, depending on phones alone may not be adequate to reveal the contents; nonetheless, scientists are more interested in the transitional zone, or "diphones," because it may contain more data than steady waveform regions. Speech sub-phonic units are separated into three zones.

Let's look at three components of phones: the first portion is reliant on the previous part, the second part is independent of the previous part, and the third part is dependent on the first part of the first phoneme. States, or senones, are what they're called. The waveform of speech may not appear to be conveying information, but when examined at the frame level, it appears to contain important information and resemblesa sine waveform.

Audio is divided into units, phones, or frames, and specific features are retrieved for each frame, known as "feature vectors," because features are essential for every data unit. Another important concept is the recognition model. A Gaussian mixture model, which blends three waveform states, is used by Kaldi.

The acoustic model makes use of Senones' acoustic characteristics. Auditory models can be either context-dependent or context-independent. They are self-explanatory since the context dependent model considers the context of senones, whilst the latter provides themost likely vectors for each phone.

Word-to-phone mappings are included in the phonetic dictionary. The language model then decides which word follows the one that was previously detected. N-grams and finite state language models are examples of language models. The next word predictor must be extremely efficient for increased recognition, yet there are situations when the language model can be problematic, such as when recognizing names. You can find simple answers by breaking up words into smaller parts or by employing phone numbers.

4.1.3 Phases of Speech Recognition

There are multiple phases of speech recognition and a person opts for the one thatis greatly effective for their project. Same concept goes for our project as well.

Speech sampling:

A higher sample frequency or higher sampling precision are needed to get a higher identification accuracy. Similar parameter settings are used by commercial speech recognizers, which produce outstanding results. After 11,025 Hz or 8 bits, increasing the sample rate or sampling precision is not necessary.

1. **Pre-emphasis:**

Noise is present in the speech waveform that is captured and saved in digital form, and this can have a significant impact on the quality of our product. Pre-emphasis is therefore required to flatten the signal and make it less sensitive to speech finite processing. The pre-emphasis method that is most frequently employed is the first-order system. The audio signal's pre-emphasis can be described as

$$s'(n) = s(n) - 0.95s(n-1)$$

2. **Windowing:**

The signal is windowed or tapered to zero at the beginning and end of each frame in this phase to remove signal discontinuities, which is a crucial step in audio speech recognition. The usual Hamming window has the following form:

$$w(n) = 0.54 - 0.46\cos\cos(2n/(N-1))$$

3. **Discrete Fourier Transform:**

To extract that important feature of the audio files more specifically the user's voice, the Discrete Fourier Transform is applied to the signals.

4. **Mel Frequency Warping:**

Generally, the recorded voice can have a wide range of frequencies in a simple audio file. So, they should be normalized before using it to get the most accurate results possible. Here comes the Mel scale, in which the frequencies are wrapped toachieve low dimensionality.

5. **Critical Band Integration:**

The audio files contain a wide variety of features that are extracted. Each feature has a use for the user in a different way. Hence, in this instance, not all 32–40 features are useful, so we are using the average strategy rather than utilising them all (only first 13 are used). The average feature value of the user's speech in the audio file is obtained by dividing frames into dimensions (let's say 13, for example), then taking the average of each frame.

6. **Cepstral Mean Normalization and Energy Normalization:**

The mean value of the features (calculated in the previous step) is then used to normalize the energy factor of the voice features. The mean is subtracted to remove any kind of the transfer function. The 0th cepstral coefficient is normalized to remove the variation occurring because of the amplitude.

4.1.4 Acoustic Models

Speech containing words is delivered during alignment, but because the length of phonemes is unknown, Hidden Markov Models are utilized to address the problem.

Hidden Markov Models:

HMMs are statistical models that tackle problems caused by temporal alignment and uncertain phoneme length by using transition probabilities. HMMs have both visible and hidden states.

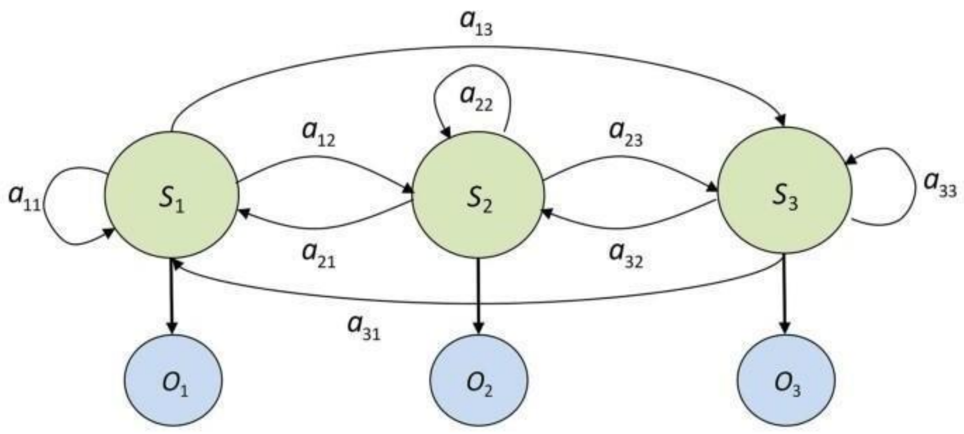


Figure 11: Hidden Markov Model

Language Model: In order to determine the possibility that a word will appear in transcriptions, the Kaldi toolset successfully applies n-gram language models. N-gram models consider the number of consecutive words that appear together when predicting the next word.

Unigram: When predicting, these types of language models use single-word probabilities that are independent of prior states.
Bigram: The likelihood of an occurrence is determined by the preceding word.
Trigram: The likelihood of an occurrence is determined by the previous two words.

4.1.5 Decoding

Once we have all of the necessary requirements for the transcript generation i.e., acoustic model, language model. We will input the audio signals to it and will have the text transcription according to the audio as output. Generally, there are two different types of decoding used in the market.

- **Online Decoder:**

  As soon as they receive audio, they begin to transcribe it. Real-time systems frequently employ online decoders. When compared to offline decoders, they perform worse. Several of the features have been computed and standardised, however because of the audio stream, it is challenging to normalise and set variance to one.
- **Offline Decoder:**

  They use audio files as input and, when compared to online decoders, exhibit better performance and results since the entire audio is available, as well as distinct borders between the phones. It's commonly employed in systems that aren't very sensitive.

Our implemented Approach:

We employ offline decoding in our project "AudioPhonic - an audiovisual auto corrector" because working on an online model and decoder would place undue strain on the system's performance. So, we adapted an offline approach, i.e. pass audio to the model to get the desired transcript.

4.1.6 Vosk

This Python module is for Vosk. An offline, open source voice recognition toolkit is called Vosk. Including English, Indian English, German, French, Spanish, Portuguese, Chinese, Russian, Turkish, Vietnamese, Italian, Catalan, Arabic, Greek, Persian, Filipino, Ukrainian, Kazakh, Swedish, Japanese, Esperanto, Hindi, and Czech, it recognises speech in more than 20 languages and dialects. More are coming. Vosk models are small (50 Mb), yet they provide continuous, massive transcription, zero-latency response with streaming API, and movable vocabulary and speaker identification. Vosk offers speech recognition for chatbots, smart home devices, and virtual assistants. Also, it can create transcripts of lectures, interviews, and movie subtitles. Vosk is capable of growing from a single Raspberry Pi or Android phone to massive clusters.

We are incorporating the Kaldi acoustic and automatic speech recognition model into our project utilising the vosk framework. The offline open source ASR toolset is called Vosk. It is built and developed mostly using the Kaldi automatic speech recognition model and acoustic model. It gives us a simple, quick, and secure way to interact with audio files and extract all of the crucial components from the Kaldi automatic speech recognition model (as detailed below).

The Kaldi model gives us a lot of other important information from the audio files that can be useful in many different ways in our project. Most of the important features that are extracted from the audio by Kaldi automatic speech recognition modelis

- **Confidence:**

  Typically, there is a probability of error if we want to utilise any kind of technology to carry out specific activities, and we must account for this possibility in order to use the system as effectively as feasible. So, there is also a problem in this case is that whenever we get the transcript from the model, there are chances that the model misjudged a certain part of audio and gives us wrong words from the audio. Long-term consequences of this will be problematic. In order to tackle this issue, Kaldi gives us faith in each word and sentence. It basically refers to how much the model believes its translation to be accurate. One of the primary reasons we chose this model was due to its accuracy in providing the english transcription from the audio files, along with a number of other variables. The accuracy of Kaldi automatic speech recognition is typically above 90% when it comes to the English language.

- **Time stamps:**

  One of the additional key aspects that the Kaldi automatic speech recognition model extracts for us is that it not only provides us with the words and phrases but also their time stamps, i.e., the exact location in the audio where each word is really pronounced (that is translated by Kaldi ASR model). The Kaldi ASR's time stamp extractions are extremely valuable and significant to us. We will benefit from being able to alter the audio files in accordance with changes the user makes to the transcript.

- **Alternates:**

  As it is already discussed in the confidence section, Kaldi gives us the word that (is according to the model) has the most confidence among all of the other words. The model also includes an extension that provides us with the additional alternative words (that is also extracted by the Kaldi automatic speech recognition model). Alternates are, to put it simply, words with a lower confidence rating than the top term.

4.1.6.1 Installation

Pip is the simplest way to install the vosk API. There is no need to compile anything.The following platforms are currently supported by us:

- Raspberry Pi running Linux on x86 64
- Windows on arm64 Linux on arm64 OSX (only x86, no M1)

We used the most recent versions of pip and Python3:

1. Python 3.5-3.9 pip version 20.3 and newer
2. Then, using pip, install vosk on Linux/Mac:
3. install vosk with pip3

Please note that pip does not fully support all systems; for example, on arm64, you must install from the released wheels:
https://github.com/alphacep/vosk-api/releases/download/v0.3.31/vosk-0.3.31-py3-none-linux aarch64.whl

If you're having difficulties installing, look at the output of the following commands and keep it handy:

pip3 —version python3

4.1.6.2 Models

Vosk includes offline models that are set up and ready to use on a variety of devices and operating systems. Each type of speech recognition applications on any IO device benefit from its scalable and portable nature. In order to create an offline real-time speech recognition tool, we employed its pre-built model based on KALDI ASR.

Our model, which resides on the machine and aids in real-time text prediction, is lightweight (70.8 MB in file size for the data we used) and prepared for deployment to any server to make the service easily accessible to anybody.

The model we used was officially documented as:

Model	Size	Word error rate/Speed	Notes	License
vosk-model-small-en-us-0.15	70.8 MB	9.85 (librispeech test-clean) 10.38 (tedlium)	Lightweight wideband model for Android and RPi	Apache 2.0

Table 1: Offline Model of Vosk used for SR

4.1.6.3 Precautions before predictions

Modern systems' accuracy is still unstable, which means that it can be very good at times and very awful at others. It's difficult to design a system that will work in all situations. There could be several explanations for this:

- The audio is of poor quality
- The system's vocabulary is inconsistent (yes, most of the systems, even end-to-end ones still use a fixed vocabulary)
- Some unexpected audio issues, such as frame loss or frame coding flaws
- Software bugs

4.2 Text-to-Speech

The text that is translated to audio should be as close to the original speech as feasible. While converting the text back to the audio file is a simple process, there is a significant issue in our instance because it will be for naught if the user alters any text (from the transcript created by the model from the audio files) and the audio produced after submission differs from the original voice. If the user alters lengthy sentences, neither the user nor the situation will be improved. It will sound like two distinct voices speaking in the same statement. So, we have to create a model that is good and accurate enough to give us the audio that has the features and characteristics of the original voice (as close as possible) to avoid any kind of disorientation in the audio files for the listener.

So, there are multiple features in the audio of a person that makes one person's voice different to another person's such as:

- energy
- frequency
- pitch
- power
- vocal cords stretch etc.

These multiple features combined with each other to make a person’s voice unique from another person’s voice. Our text-to-speech model is developed in multiple phases of improvements.

4.2.1 Speaker encoder

40-channel log-mel spectrograms with a 25ms window width and a 10ms step are used as inputs to the model. The L2-normalized hidden state of the final layer, which is a vector of 256 items, is the output. A ReLU layer precedes normalization in our system, with the purpose of making embeddings sparse and hence more easily interpretable.

The speaker encoder is trained on the Speaker verification challenge, which is one of the key ideas in speech detection. SVT is, in essence, a straightforward and common sort of biometric security in which a user's speech characteristics are used to identify them. A person is given a collection of features and templates that include all the crucial traits and traits that set one person's voice apart from another's sound. Enrollment is the name given to this template-creation procedure. By contrasting the new user's speech with the enrolled speaker embeddings of the user in the system, a specific user is recognised at runtime.

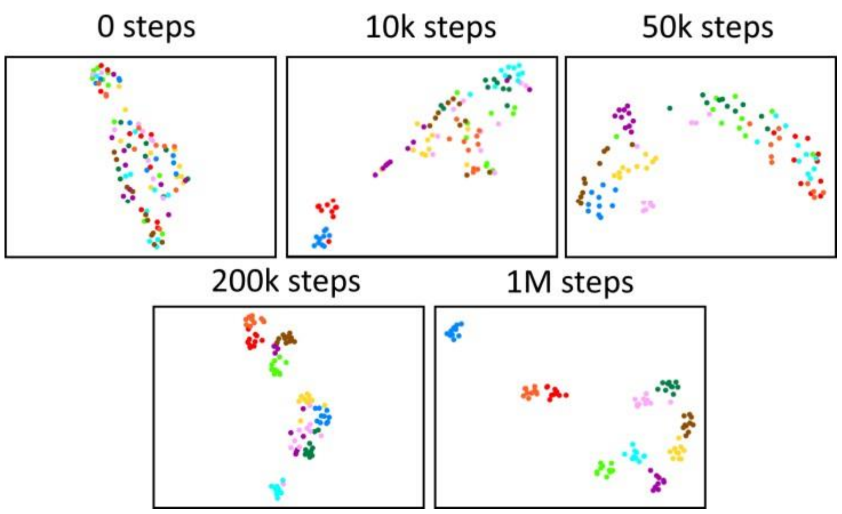


Figure 12: Comparison with clusters Embeddings in 0-1M steps

A two-by-two comparison of all speaker embeddings and nearly all of the accessible embeddings is used to determine similarity. Scaled cosine similarity is typically the recommended similarity when comparing voice embeddings.

$$S(ij,k) = w \cdot \cos(e(ij), c(k)) + b = w \cdot e(ij) \cdot ||c(k)||^2 + b$$

This model's training batch, which is sampled from the dataset's larger complete utterances, has a fixed duration of 1.6 seconds. Although there will be some change, as was mentioned above, this model will operate essentially identically to the k-means clustering technique. The measure used to determine and assess the accuracy of this model is the Equal Error Rate. Our main goal is to reduce this kind of inaccuracy. We are convinced that the speaker encoder is producing meaningful embedding, which is what our system requires, as a result of numerous experiments and outcomes.

4.2.2 Synthesizer

The synthesizer part of this model is based on the Tacotron 2 (without Wavenet).

For this purpose we have used the open-source Tensor Flow implementation of Tacotron 2.Let's talk about tacotron 2, basically it is a recurrent sequence-to-sequence model that is helpful to predict the mel spectrogram from the text given to the system. At its core,it is following the protocols of encoder-decoder architecture.

The initial embeddings of the individual characters from the text sequence produce the vectors. Because each decoder input frame is combined with the output of the previous decoder frame and sent through a pre-net, the model is autoregressive. The entire frame sequences are then run through the post-net prior to the production of the mel-spectrogram. The target mel-spectrogram contains 80 channels and was generated from a 50 ms window with a 12.5 ms step. Because of this, it has many more features than the speaker encoder.

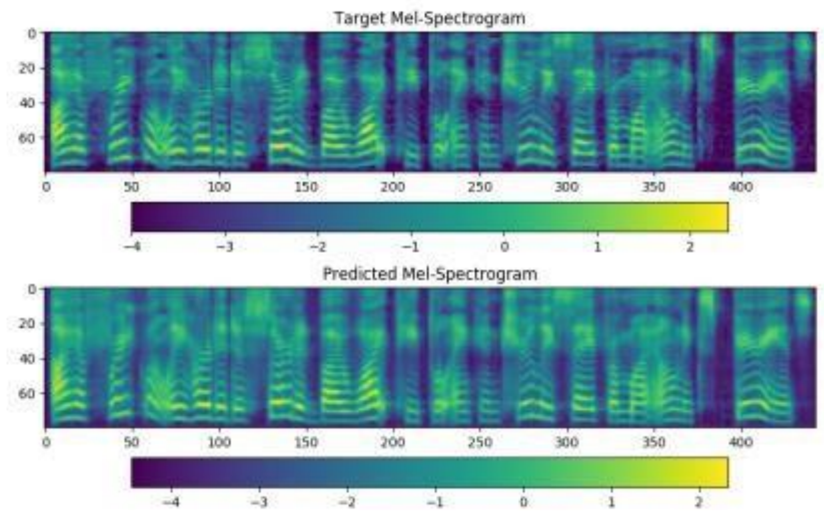


Figure 13: Mel Spectrogram of original voice and deep voice generated

One of the main objectives in this case is to minimize the L2 loss of Mel-spectrogram.

4.2.3 Vocoder

The vocoder in this model is based on the WaveNet model. Since its release, it has been a deep learning programme that is primarily focused on audio data, and it continues to be the industry standard when it comes to creating a natural voice for Text-to-Speech. It is regarded as the slowest deep learning architecture used in practical tasks despite being a crucial component of audio naturalness in deep learning. We opted for this model, as our main target is the naturalness and accuracy of the model. Time is also an important factor however, we have discovered through tests and findings that the worst case scenario—long paragraphs of text—will cause problems. The modified text in the transcript could not, in the best and average situation, be big paragraphs rather than single words. So, there won't be much of a tradeoff, if any at all, over time.

The vocoder uses K-Nearest Neighbor to find the closest embeddings in order to obtain the best and most accurate embeddings.

4.2.4 Audio integration

The second, and most crucial, stage in our project is to combine the two audio files—one of which is a newly made audio file and the other is the original audio speech. by taking into account the system's performance and the complexity of time. When making changes to the transcript, we do not use our entire model to construct the voice from the given text (from the beginning). But, in this instance, we are taking the best course of action. We can tell when text has been added, changed, or removed. Instead of giving the model the entire text to use as the basis for their voiceover, To avoid having to work on text whose audio is already available and is not modified in the transcript, we are only entering the text that is affected, i.e., either added, changed, or updated. A particular user can alter the text in three major ways. These topics are briefly explored below. The implementation chapter will go into more depth about how the system includes and addresses these problems, though.

Addition of the Text:

One of the things that a user can perform after getting the editable transcript from the audio file is addition of some words or sentences. Commonly, this case arises when a user skipped some sentences or forgot to add certain words in his speech (maybe because of the marketing purposes) or some other issues.

Deletion of the Text:

Saying certain words by mistake is also a very common issue by the voice over artist and content creators. Thus, use lightweight software to just remove audio snippets that are 1-2 seconds long. In this situation, our suggested solution—removing the words from the transcript and submitting it—is really helpful. After then, the system will handle it on its own.

Updating the Text:

Mispronunciation is a significant problem for both voice over actors and content creators. Due to technological issues, the words are occasionally not recorded clearly and contain some sort of pronunciation error. That is not a very good professional practise. Hence, for dealing with this kind of issue, our solution is also a fantastic choice.

Above are the main changes that we want or need to cater to in the process of audio integration. This is just a brief description of the problem. Now, how our problem is solving these problems more accurately by keeping in mind the complexity figure as well.

Chapter 5

DETAILED ARCHITECTURE AND DESIGN

5.1 System Architecture

5.1.1 Architecture Design Approach

Our whole project is based on the multiple modules that worked with each other as a unit to make the system work accurately . On counting, it can be dividedInto 4-5 modules i.e.,

- User will input the audio files
- Editable transcription will be generated from audio
- User made edits in the transcription and submit it
- User can finally download the new version of the video from our application

Together, these several functional modules serve as a single entity that enables the AudioPhonic to easily carry out a variety of difficult tasks while taking complexity and time into account. Python is the primary language we are utilising to develop the product, and we are using Flask, a Python package, to create the website application. As practically all of our backend and machine learning algorithms are written in Python, that is the primary justification for using the flask. We will find it extremely simple to integrate the frontend and backend because Flask is a Python package. All of the modules have their own responsibilities and tasks, have their own dedicated functionality, they are not independent in such a way, that output of one module is considered as the input for other modules. Although each module's primary functions are independent of one another and thus coherent, they do communicate with one another for input and output purposes.

5.2 Speech Recognition Architecture

In this architecture, there were a total of four modules details of which are given and show below in the diagram:

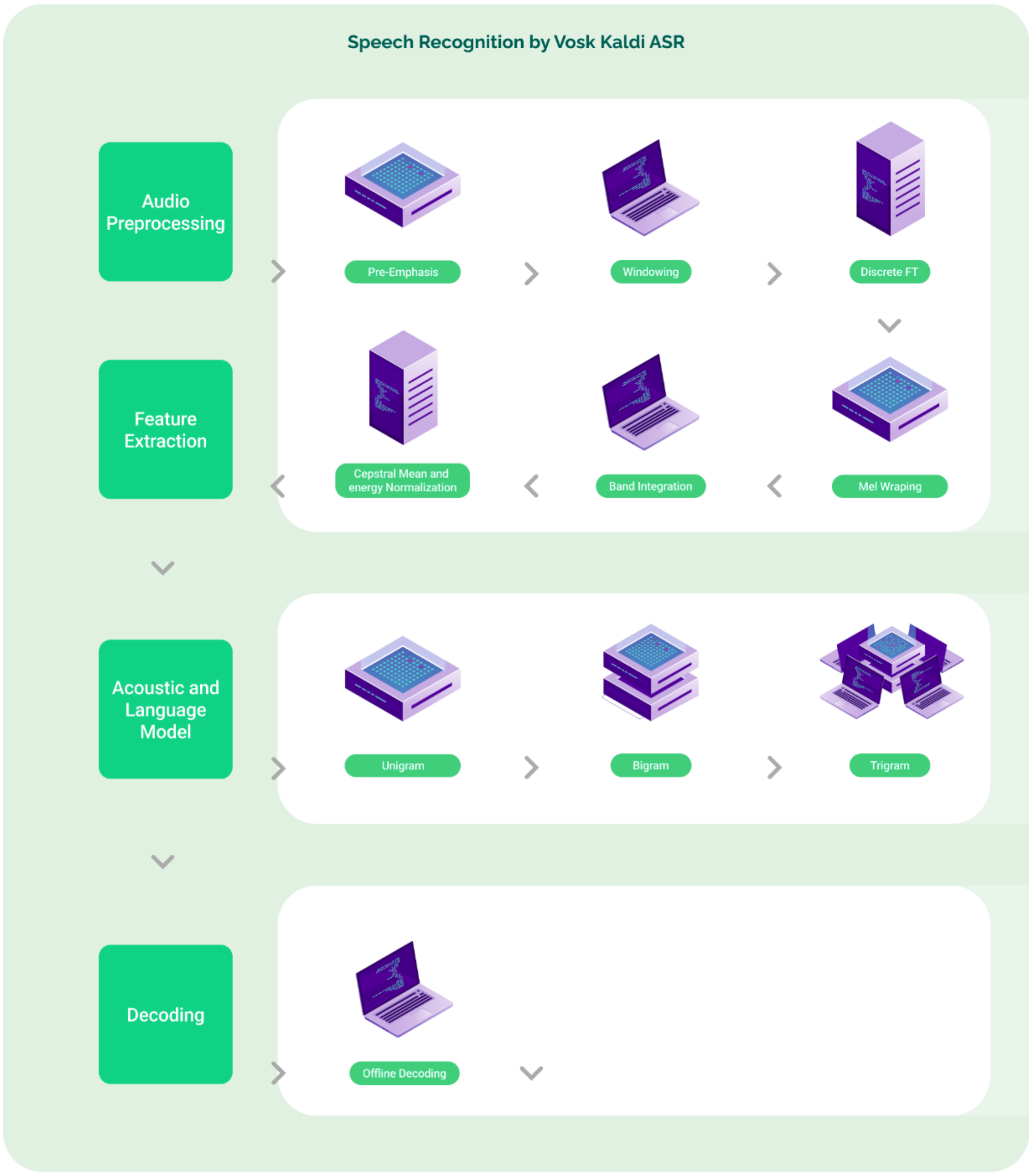


Figure 14: Architecture of Speech Recognition model

5.3 Text to Speech Architecture

In this architecture, there were a total of four modules details of which are given andshow below in the diagram:

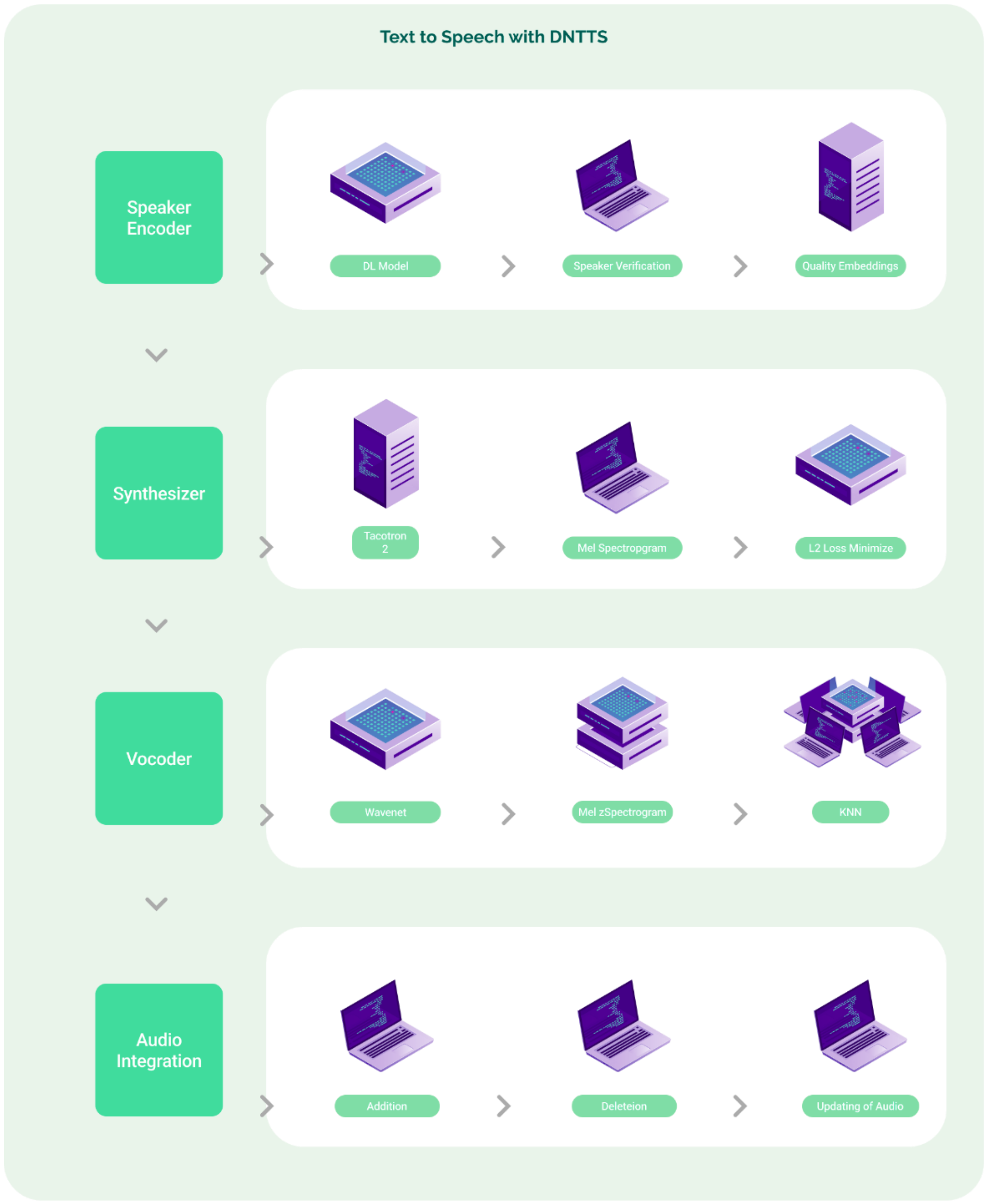


Figure 15: Architecture of text to speech model

5.4 Flask Architecture

The above two modules of speech recognition and text to speech models were fully integrated with a total of four web pages in this design. The completion of this flask web application coordinated with the completion of other independently developed modules, such as audio, sr, and rtvc.

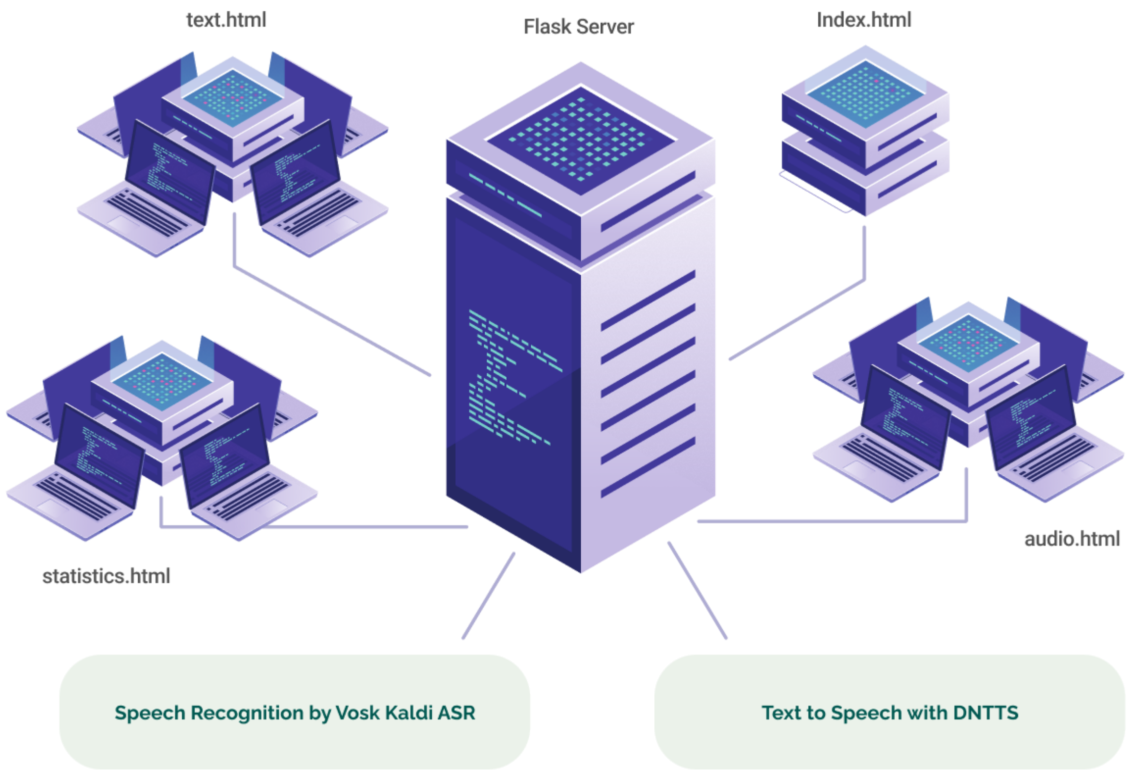

text.html
Flask Server
Index.html
audio.html
statistics.html
Speech Recognition by Vosk Kaldi ASR
Text to Speech with DNTTS

Figure 16: Architecture of flask web application model

5.5 Context Diagram

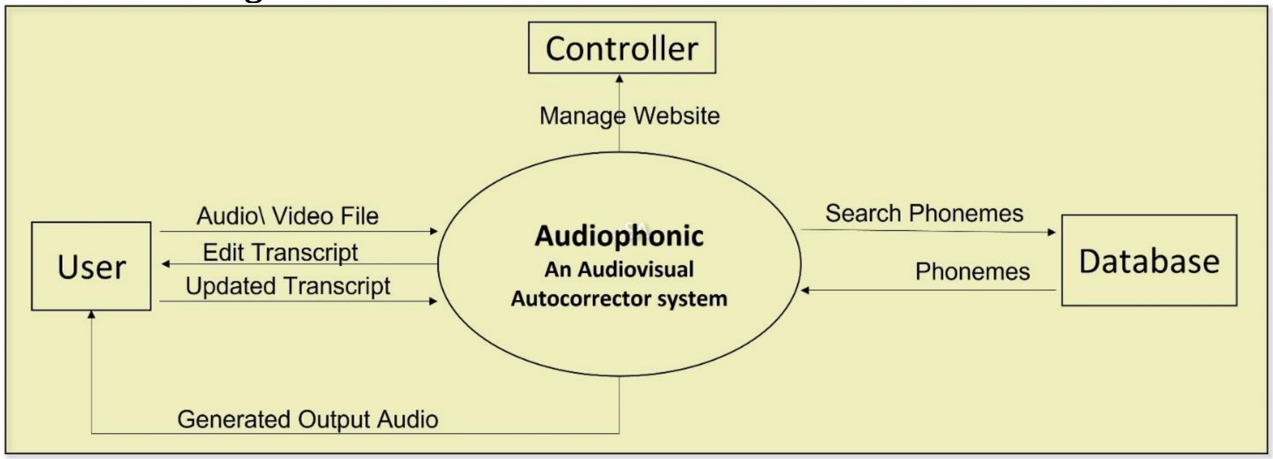


Figure 17: Audiophonic Context Diagram

5.6 Class Diagram

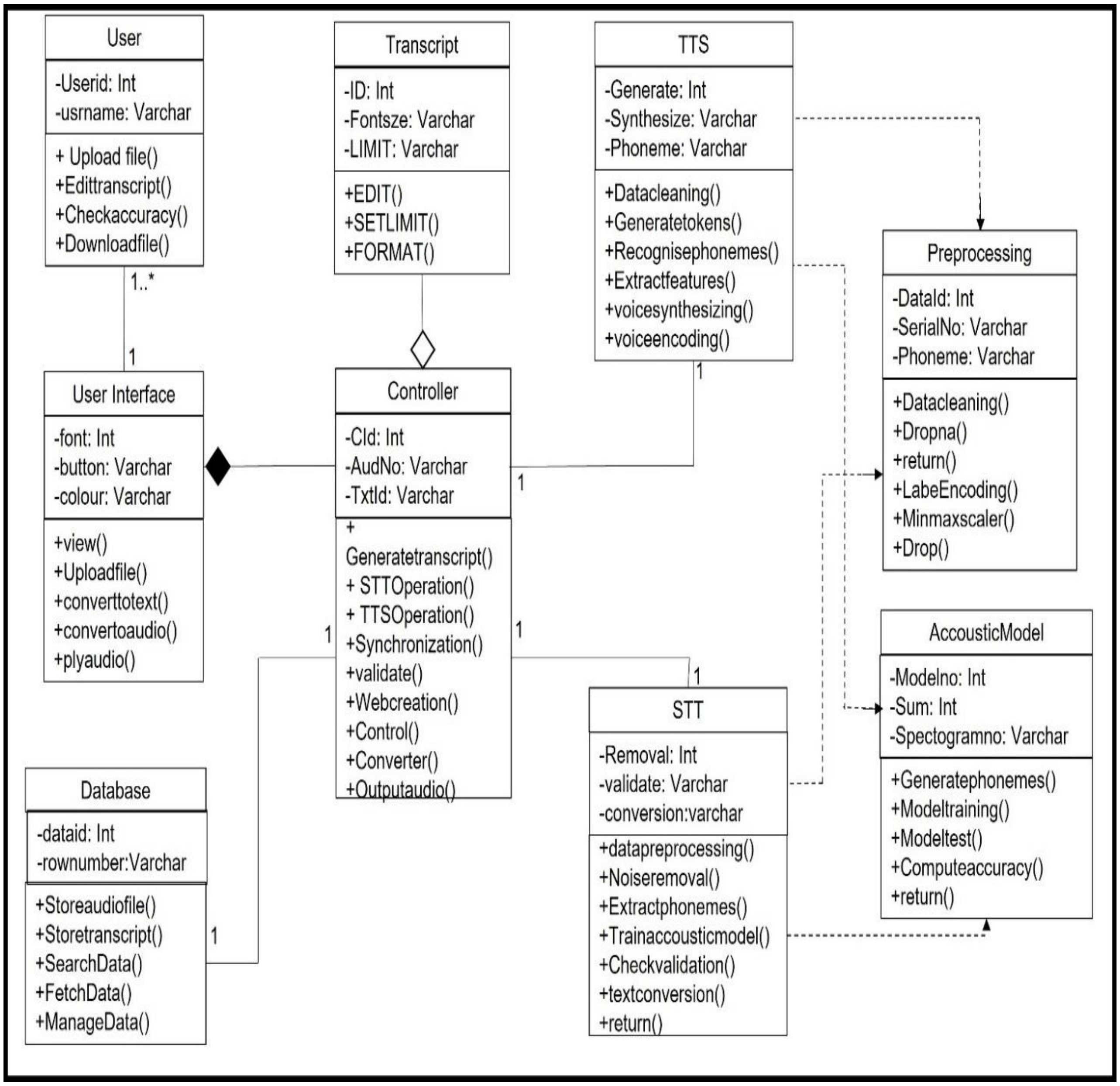


Figure 18: Audiophonic Class Diagram

5.7 UseCase Diagram

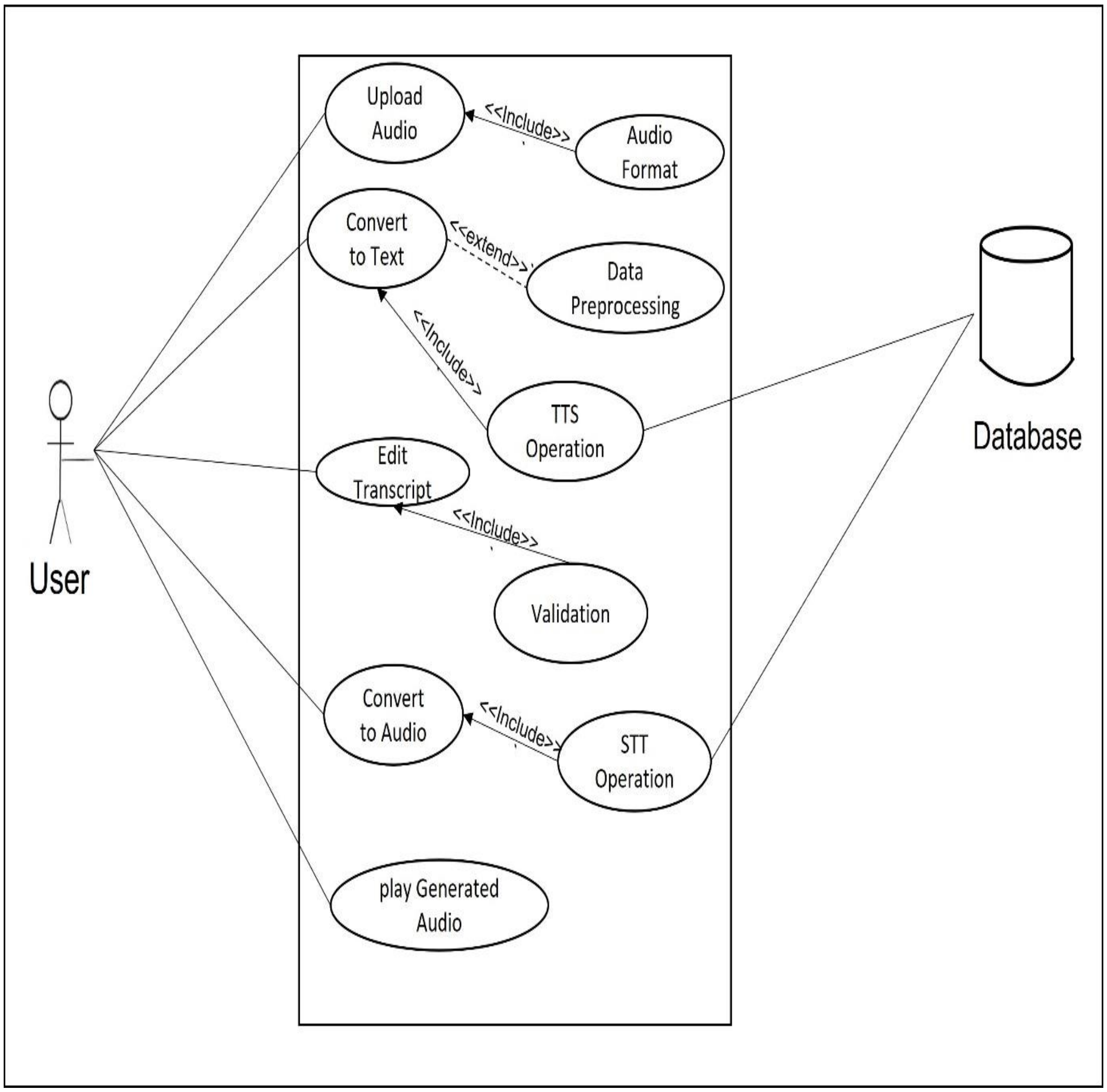


Figure 19: Audiphonic Use case Diagram

5.8 Sequence Diagram

1. Upload Audio\Video File

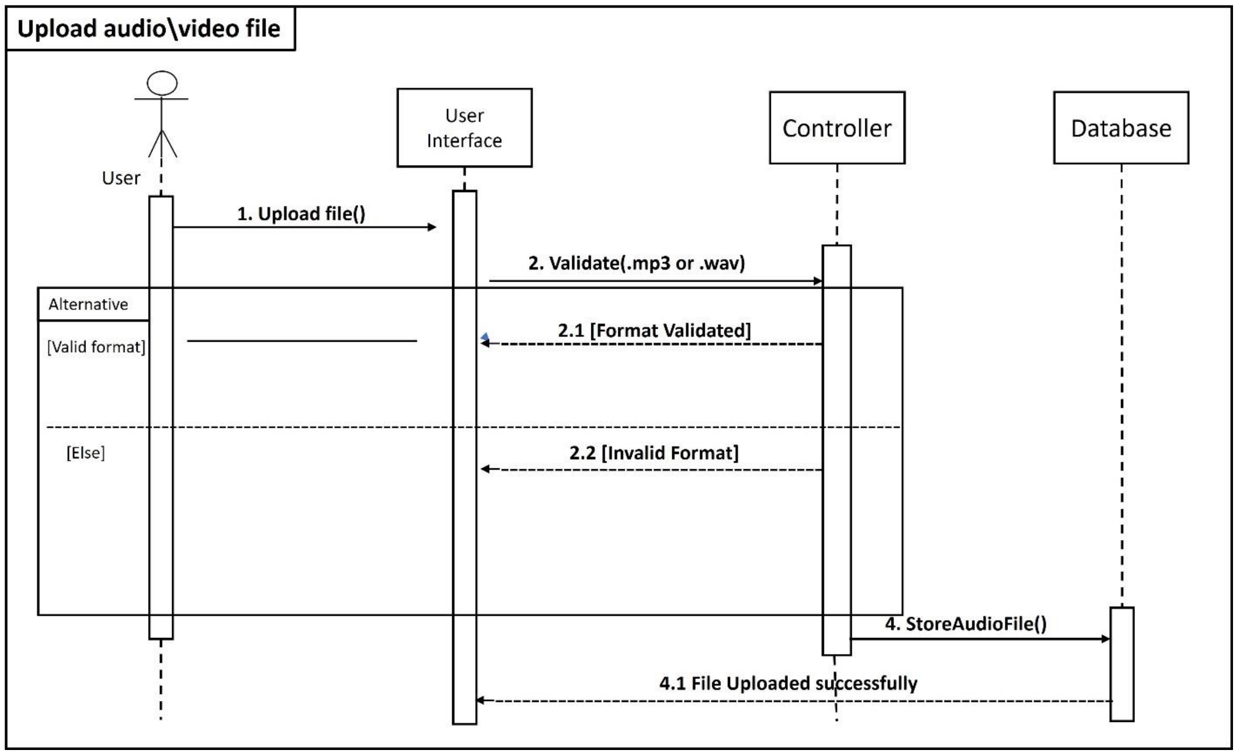


Figure 20: Upload Audio\video Sequence Diagram

2. Edit Transcript:

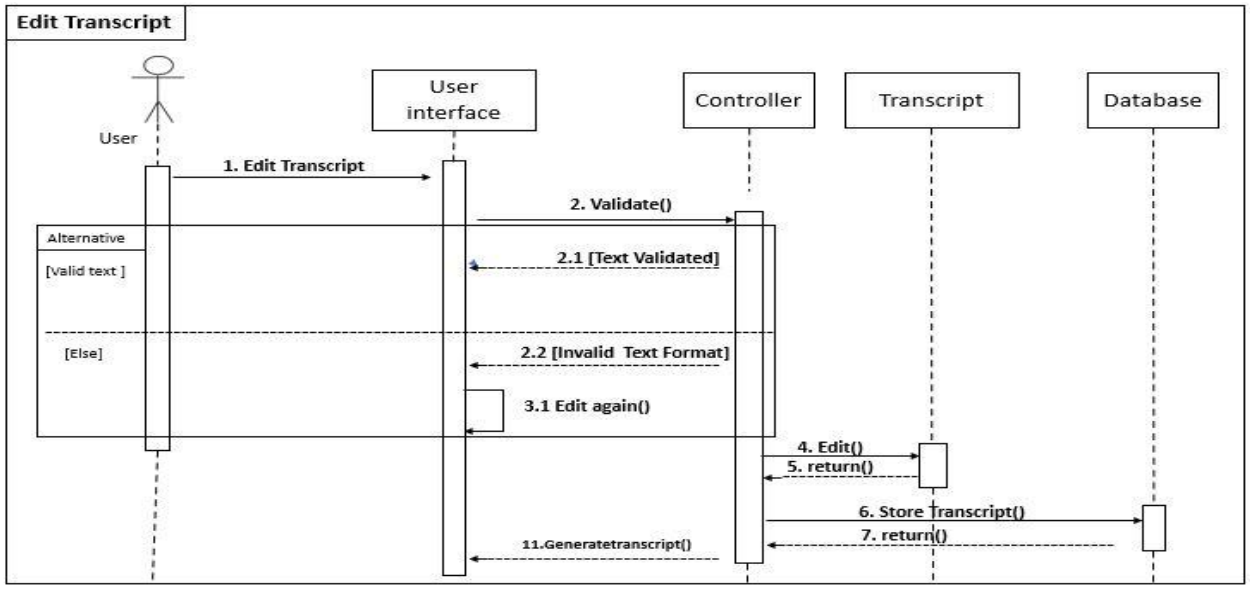


Figure 21: Edit Transcript Sequence Diagram

3. Convert to text

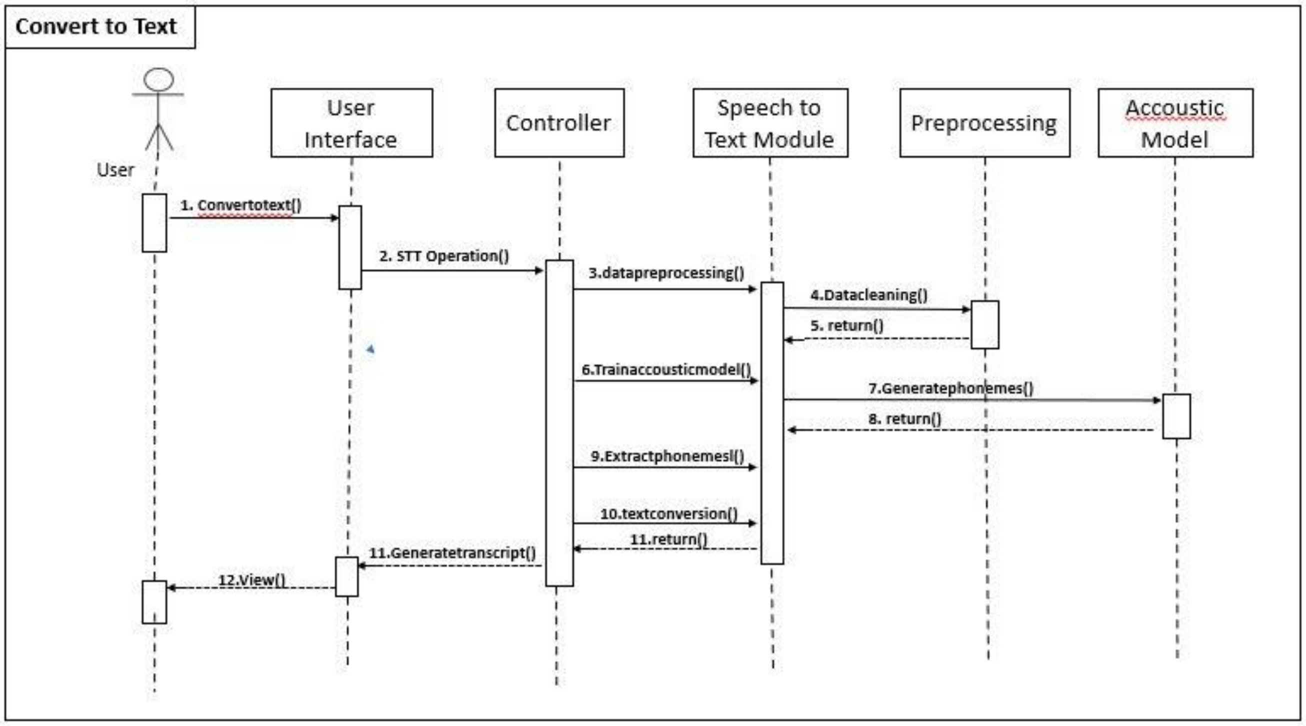


Figure 22: Convert to Text Sequence Diagram

4. Convert to Audio

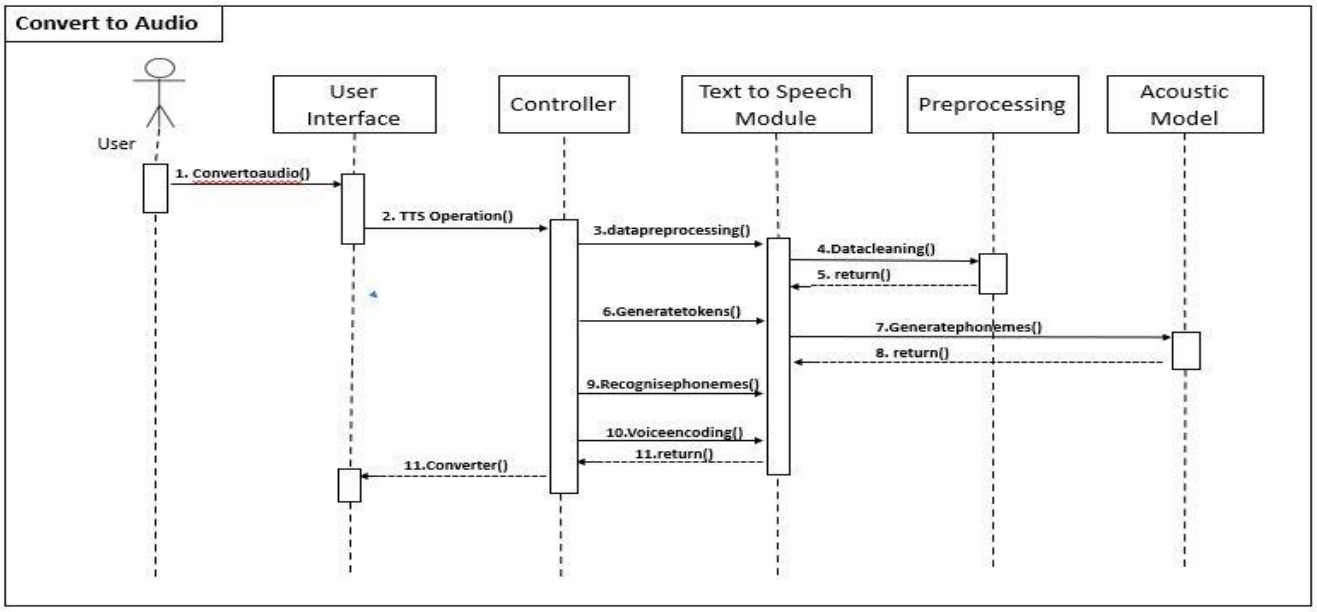


Figure 23: Convert to Audio Sequence Diagram

5. Play Generated voice

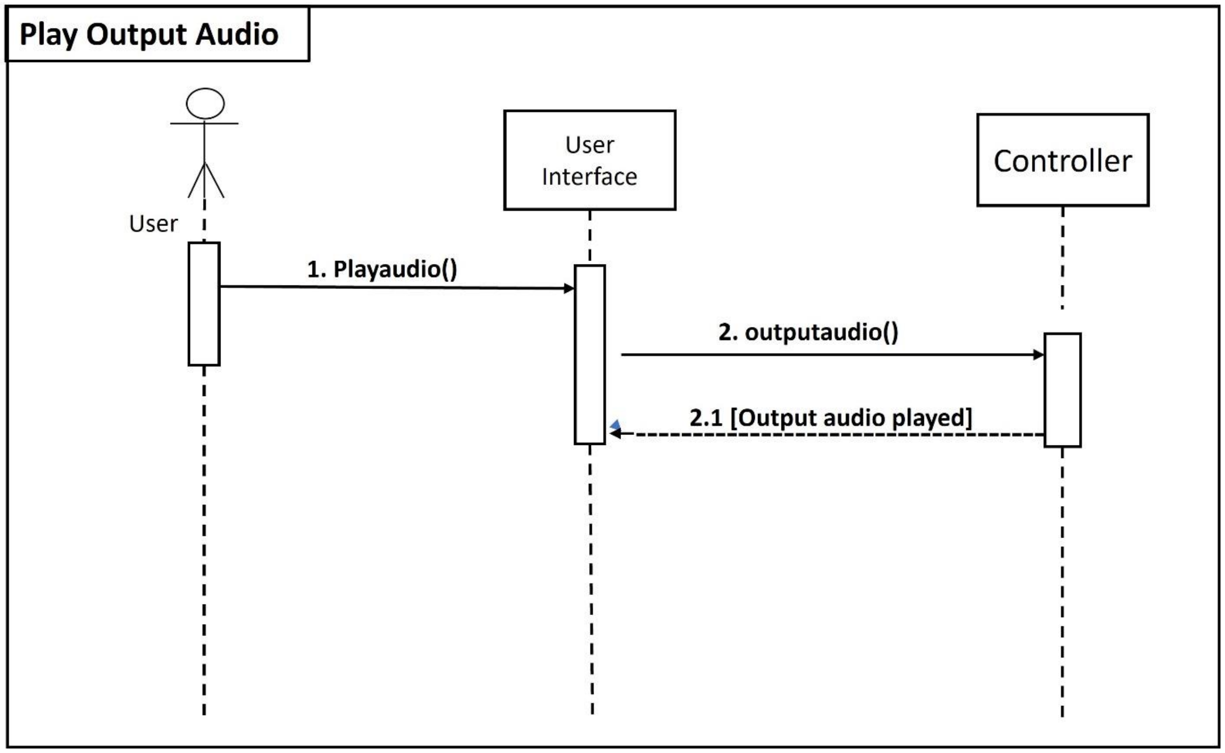


Figure 24: Play Generated voice Sequence Diagram

5.9 Activity Diagram

1. Upload File

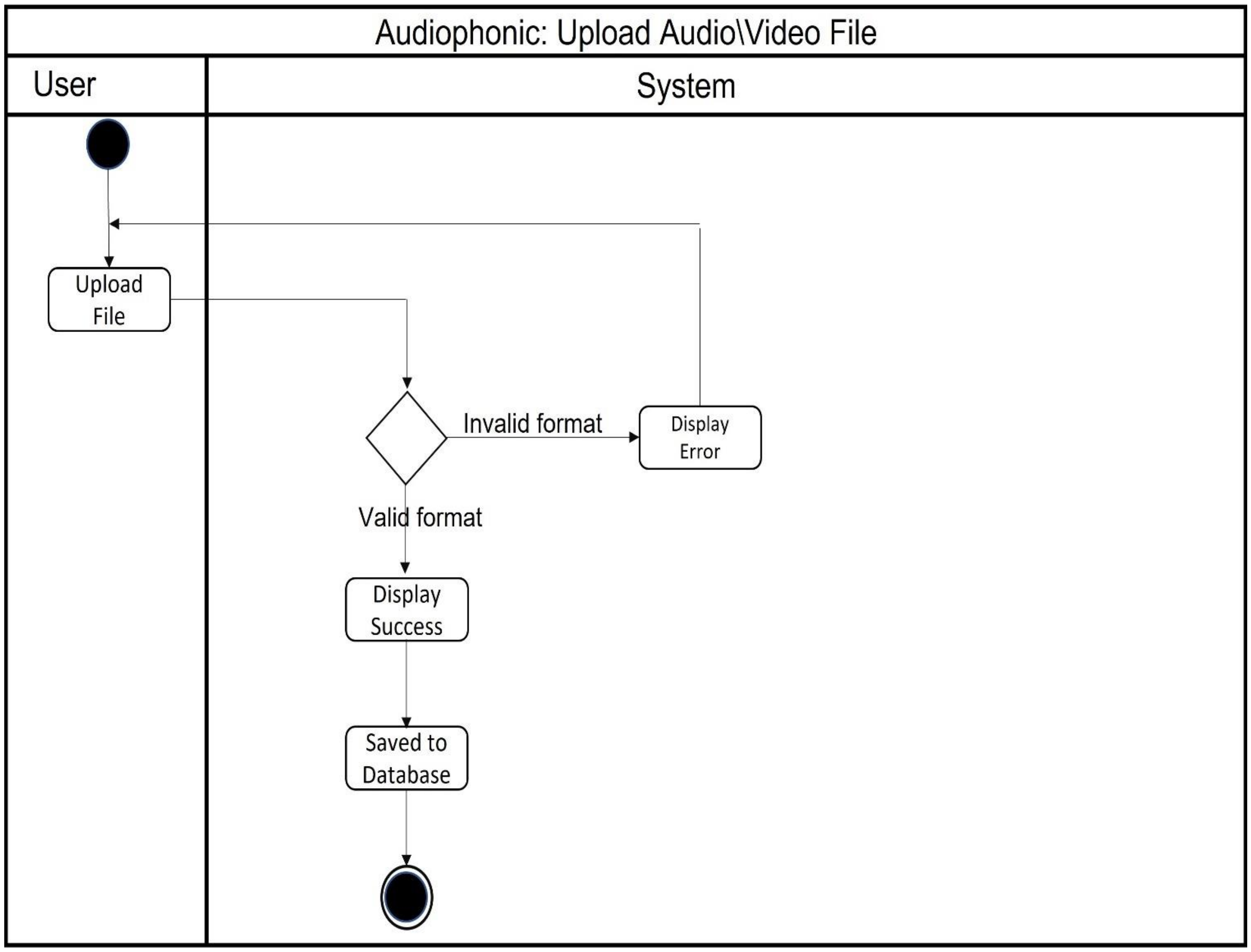


Figure 25: Upload File Activity Diagram

2. Edit Transcript

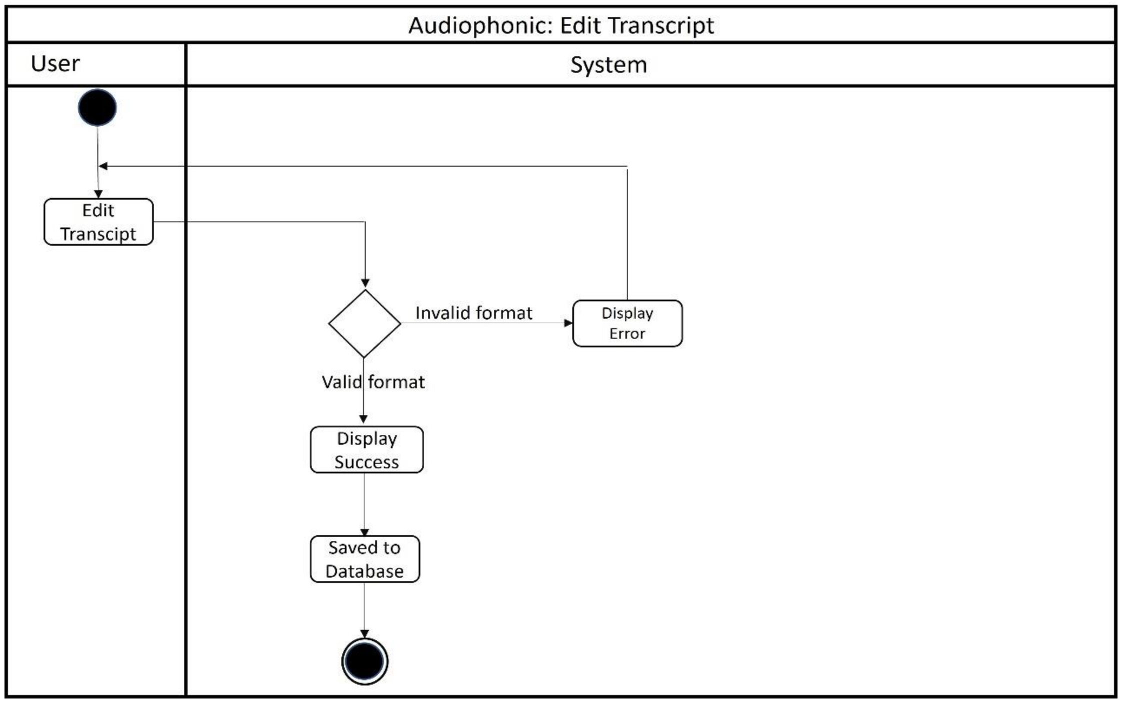


Figure 26: Edit Transcipt Activity Diagram

3. Convert to text

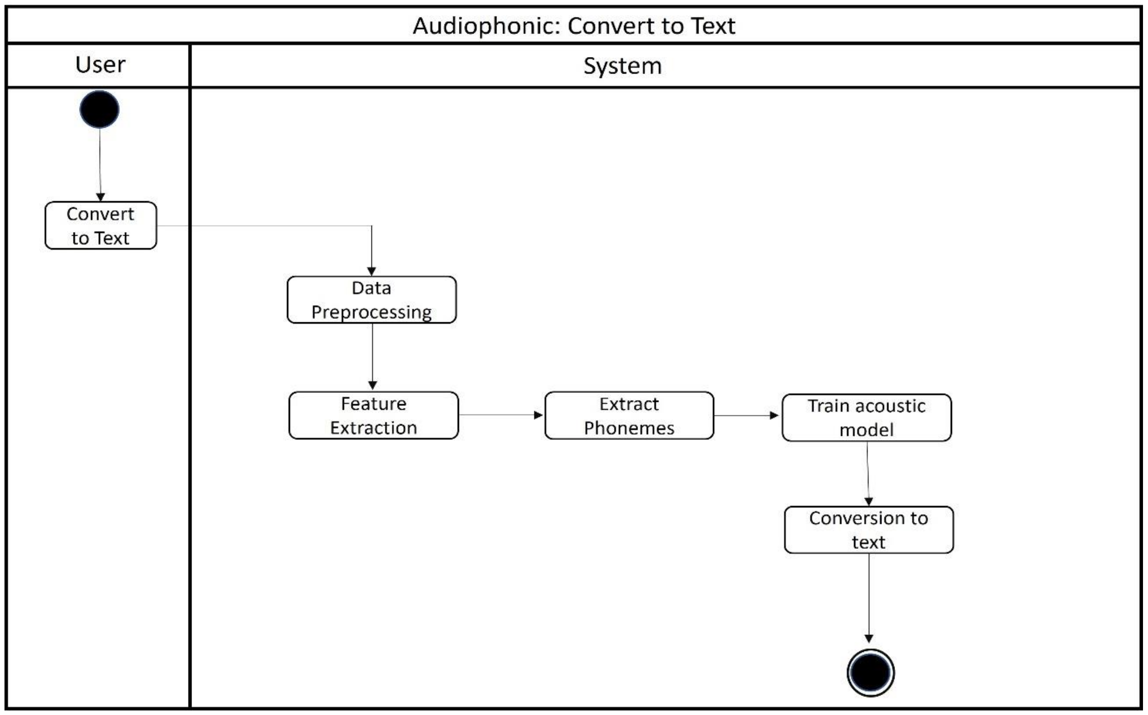


Figure 27: Convert to text Activity Diagram

4. Convert to Audio

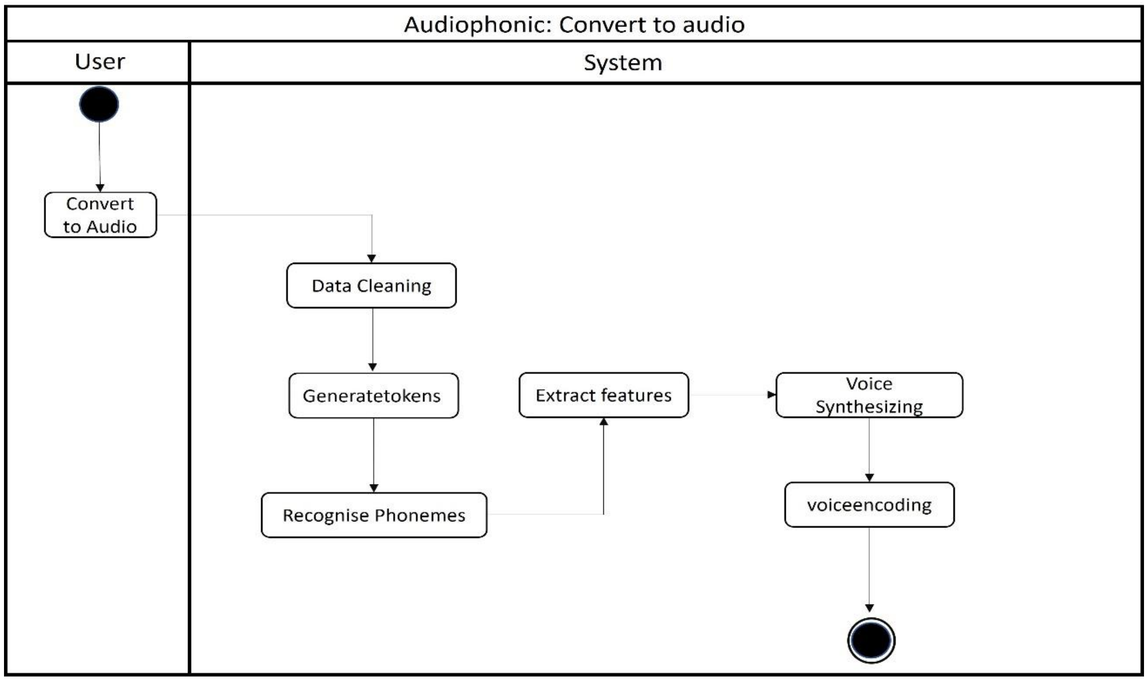


Figure 28: convert to audio Activity Diagram

5. Play Generated voice

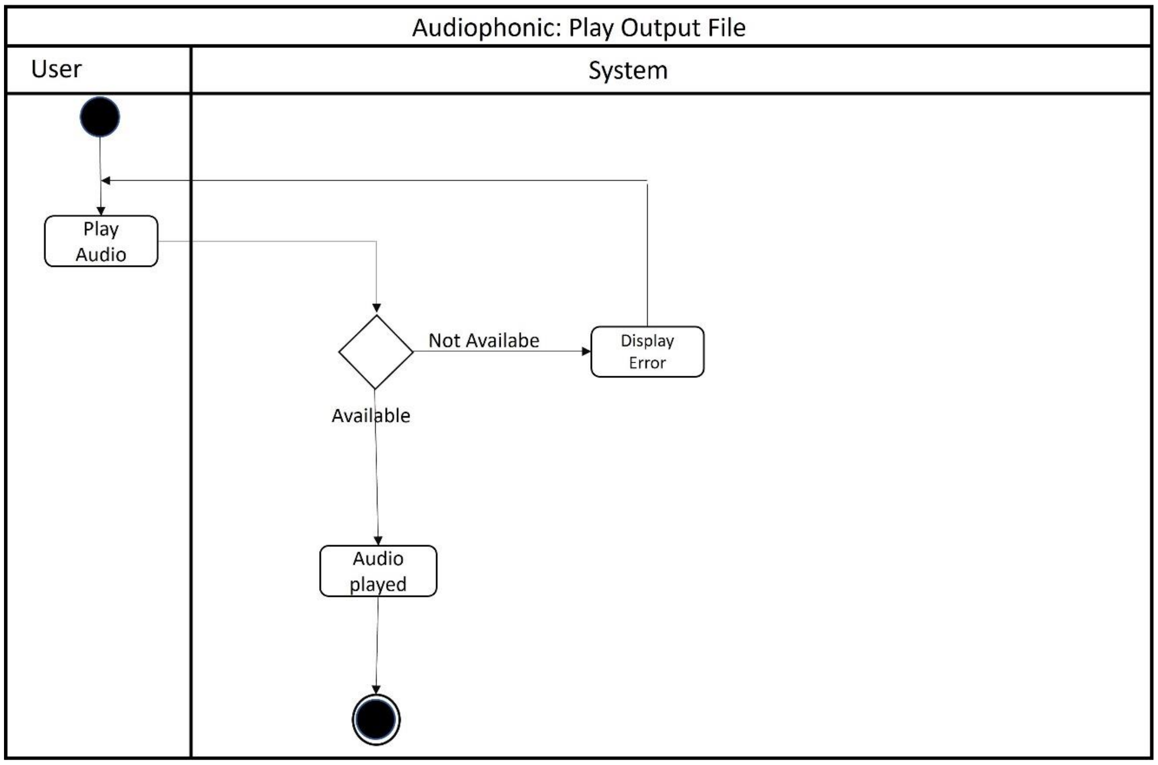


Figure 29: Play Generated voice Activity Diagram

Chapter 6

IMPLEMENTATION AND TESTING

6.1 Tools & Techniques

An audiovisual autocorrector, or AudioPhonic, is a very large project that necessitates the integration of numerous distinct technological stacks. We use a variety of open source libraries to create projects that are ready for production. The following is a list of some of the instruments and methods we employ in order to create and maintain this large project:

6.1.1 VSCode:

Visual studio code is one of the best code editors out there in the market and is widely used in the field of software engineering across the world. So, to keep up with the industry standards and protocols, we also use the Visual Studio code as our main code editor of our project.

Visual Studio Code's extensive community and support for a wide variety of libraries and languages are two of the key benefits of using it. Our primary language, Python, has a lot of support in Visual Studio Code. The working environment is excellent, and we gain from pair programming, which is one of its best advantages. Pair programming is advantageous to us during the Carona Timings. If you are unfamiliar with pair programming, it is a method of writing parallel code using online sessions. If two programmers are located in separate cities or even nations, they can simply start a session on the VScode platform and begin writing parallel code (pair programming)

6.1.2 Python

In the field of machine learning and deep learning, it is not a new name. Everybody knows about python, because of its supporting framework in the field of deep learning. Python offers a wide variety of machine learning libraries. Python can be utilised to manage the extremely complicated neural networks that serve as the foundation for machine learning and neural network algorithms, while not being the fastest language.

That is also the key reason we favour Python over other programming languages. In addition to its capability for machine learning, Python is also relatively simple to create and comprehend. Even a person who is not a coder, in my opinion, can understand the syntax of the Python languages (let alone the logic). So, it was the primary justification for choosing the Python programming language for the project's backend.

6.1.3 Flask

The finished product of our project resembles a website application. So, we choose the Flask module, a Python package useful for building Website applications using Python. Almost all of our backend coding for machine learning and neural network implementation is written in Python, as was mentioned above in the section. So, we want to adopt a front-end language that can interface with the back-end simply, effectively, and without making any mistakes.

Also, because flask is a package for the computer language Python (to create website applications). As a result, it also permits programming logic on the front end, which is useful for displaying visuals and visualisation as well as some logical rendering of the website's components.

6.1.4 Kaldi

An open source, C++-based automatic speech recognition toolbox is called Kaldi ASR. The Apache v2 licence is held by Kaldi. Kaldi was developed by Dr. Daniel Povey, an assistant professor at John Hopkins University. For voice recognition research and development, the Kaldi toolset is utilised.

Kaldi is a Linux-based program that includes pre-built models, data cleaning, and data filtering scripts that may be used for pre-training activities. The learning curve for Kaldi is steep and setting it up is a laborious and time-consuming procedure.

Few prominent features of Kaldi are:

a. Kaldi has a wide range of support for effective implementation of linear Algebra. (i.e., back-bone of machine learning algorithm)
b. It is an open-source library. A lot of bugs and issues are resolved daily.
c. Many scripts and code bases are included that may be used for tasks other than automatic speech recognition.

Detailed documentation of the Kaldi Toolkit and it’s acoustic model can be viewed on its official website (Kaldi ASR)

6.1.5 Google Colab

It ranks among Google's top research products, in my opinion. It enables users to execute, run, and work together on coding from many locations. We need a very powerful machine to train and test our model because our project involves a lot of machine learning and deep neural networks. One of our models takes two to three hours to train on average. The model's complexity is one of the causes, but my major theory for why training takes so long is because our computer is unable to do these challenging jobs.

Google Collab also provides us with the state-of-the-art platform to run complexdeep neural networks with minimum time possible due to their method of resource allocation to the individual users.

6.1.6 PyDub

One of the key components of our project, as covered above, is audio integration, which involves inserting the newly produced audio section where the old voice should go. We have the benefit of working with audio files in Python thanks to the Python module or library known as PyDub. If I separate the activities on the audio file, then they just consist of splitting, merging, and editing the audio files, and Pydub provides us with the really simple architecture and mask to conduct all of these tasks on the audio files with little to no effort.

6.1.7 Matplot Library

Our users will mostly see our results and conclusions through graphics and visualisation. You will notice that a variety of various graphics are employed to present the results on the page if you follow the basic flow of our programme. For the purpose of developing static or animated visualisations for any kind of application, Matplot is a fairly extensive library. It offers us a very user-friendly interface or set of tools. You only need to provide the data and many options for constructing the graph, and it will then provide the graph for us. Very accurate, precise and colorful graphs are the main aspects of the Matplot library.

6.1.8 Github

It is known in the field of software engineering, that if you are a software engineer, then one should know about the working flow with Github. Github is still usedin the industry, more precisely an industry standard to collaborate on open source projects.

We are soon to graduate and will become software developers. As a result, we also use Github as the platform for our project's cooperation so that you may become familiar with the Github working environment and how this collaboration operates. Working with Github is very good and helpful.

6.2 General Flow

Our programme has a very straightforward and broad flow that everyone can understand due to its simplicity and convenience. The user uploads the audio or video material that he wishes to alter on our platform (the website application for AudioPhonic) in a straightforward narrative manner. After uploading, the user can listen to the audio (which was taken from the video) and, if desired, edit the video/audio file. The user cannot alter the file once it has been submitted to the system. After submitting, some background processes (i.e., machine learning and deep neural networks) will run to make the transcription from audio to subtitles. The user will then see and be able to edit the transcript. Every term that the user wants to edit can be added, removed, or updated. The user will submit the transcript once more to the user after making the necessary changes. Following that, the user can view some graphics and visualisation on the site. In essence, it will display the various aspects of the human voice—more precisely, the voice of the user. However, in the background, the system will be working on the different parts of the audio and implementing different kinds of machine learning and deep neural networks algorithms to convert the changed text to the audio. The audio will be displayed to the user after this conversion is complete, and they will be able to download and listen to it (if the user is satisfied with the results)
The overflow of our application in the form infographics is displayed below. This will better help the reader or user to understand the flow of our application.

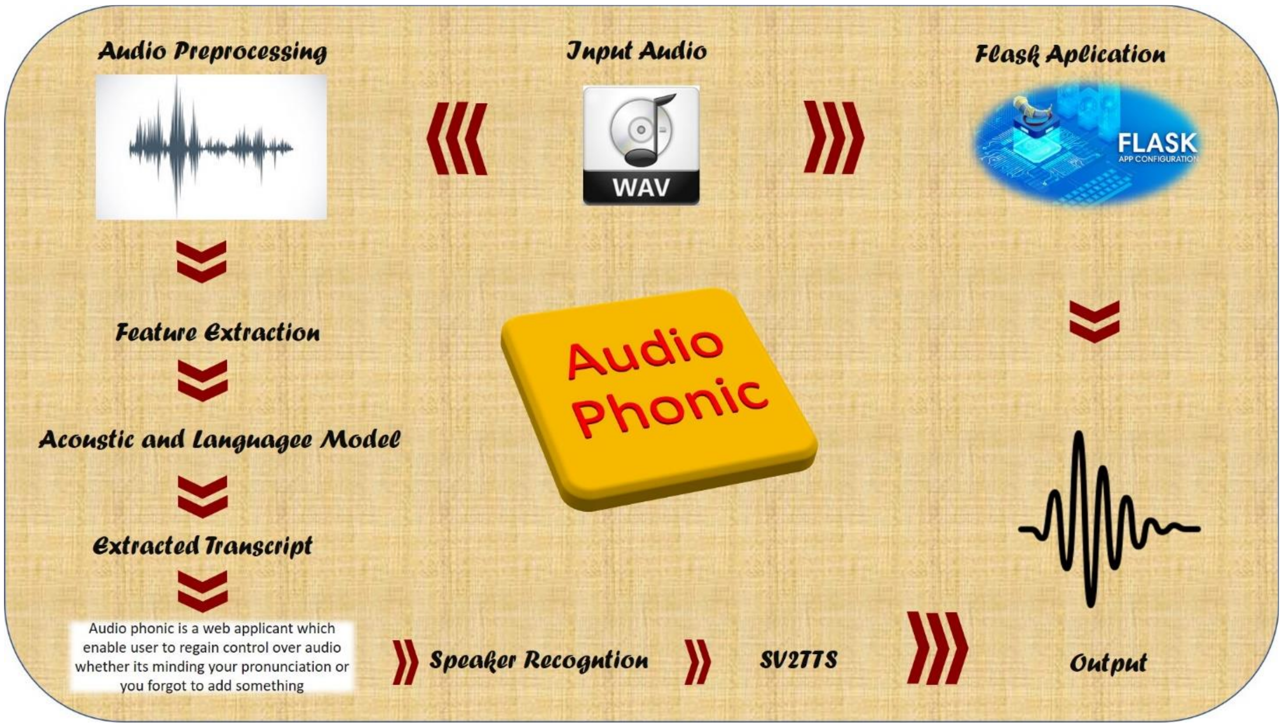

Audio Preprocessing
Input Audio
WAV
Flask Aplication
FLASK
APP CONFIGURATION
Feature Extraction
Audio Phonic
Acoustic and Languagee Model
Extracted Transcript
Audio phonic is a web applicant which enable user to regain control over audio whether its minding your pronunciation or you forgot to add something
Speaker Recogntion
SV2TTS
Output

Figure 30: Understanding the flow of AudioPhonic

6.3 Testing of Product

The product's testing is just as crucial as the original software. Since this software combines several distinct neural networks to function, accuracy of the final product is also a significant problem that needs to be addressed. Because using this product in real time was its original intent. So, using actual customers to test this product will be highly beneficial to us.

6.3.1 Developer testing

Our approach, called AudioPhonic, consists of several separate independent modules that interact with one another to function as a whole. Because some of the modules are significantly dissimilar from one another, integration is also crucial. so that they can collaborate in the most productive way imaginable. We begin by having the product's developer independently test each of the modules. This is a crucial stage in the development of our solution because these modules serve as the foundation for our project. Any issue in any of the modules, can cause major problems in thelong run.

6.3.2 Beta Testing

After the product's overall module testing. It is now time to test the product with actual customers. We go on to the next stage of our testing, which involves evaluating the product by real users in real-time scenarios, once we are certain that every module is functioning accurately and producing satisfactory results (nearly 75–80% accuracy). In testing this product and handling the product's noise, we impose two restrictions on the users. The findings from our product may not be accurate because of the noise. There are many various masking techniques that we employ to reduce audio noise, but if there is still too much noise, there will be problems.. Keeping in mind these constraints and testing the product on the real users.

Our product is tested by 20 different users that can access our product temporarily. There are multiple standards that we set for the testing of the product i.e., accuracy, performance of the system etc. The overall rating by the users is 8/10. Theusers are quite satisfied by the services of our solution.

6.3.3. Accuracy

Several things affect our product's accuracy. the amount of words that the user adds or edits, followed by the audio file and any noise it contains. The accuracy in cases of deletion is greater than 90%. The quantity of words that users alter affects the product's accuracy when it comes to word addition and update, though.

Number of Words changed	Accuracy
1-2 words	80% - 85%
10-11 words	70% - 75%

Table 2: Accuracy with respect to words

6.3.4 Performance

The effectiveness of the application must also be taken into consideration. We assess the performance depending on the various environments when measuring the performance. As a result of the time and resources needed to develop and execute the model, the performance of our system currently depends on the users' computers. Comparatively speaking, low-spec machines will require more time to complete the same task than high-spec equipment.

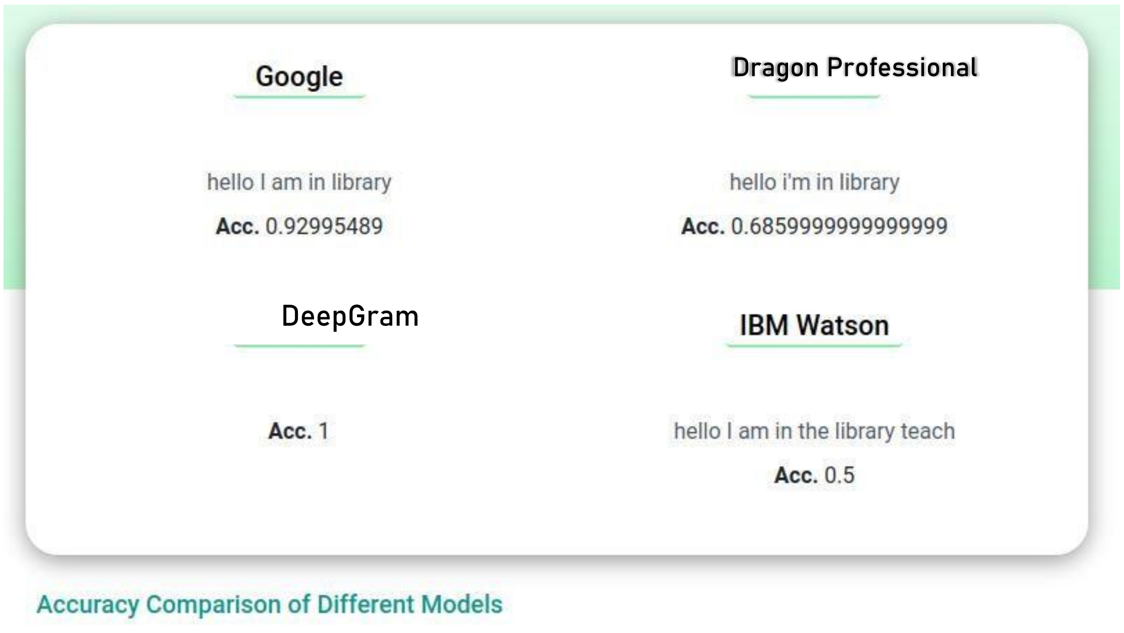
System Specs of User machine	Performance (time to run in seconds)
CPU: core i7, 10th generation GPU: GTX 1660Ti (6 GB)	It takes 40-45 seconds to completely execute the task of deep voice generation
CPU: core i5, 7th generation GPU: Intel Iris Graphics 645 (1.5 GB)	It took 70-80 seconds to completely execute the task of deep voice generation
CPU: core i5, 2nd generation GPU: Intel HD Graphics	It took 105-120 seconds to completely execute the task of deep voice generation

Table 3: Performance comparisons on different machines

Chapter 7

RESULT & DISCUSSIONS

Following the thorough testing and implementation of our product. It is now time to compile the many outcomes that were produced at various stages of our product. I'll give a flow explanation of all the outcomes in the context of our product. The graphic below shows the outcomes of voice recognition as they were taken from various free source speech-to-text converters. There, the translation and their accuracy are also shown. There are various reasons why these converters shouldn't be used. It can be the accuracy in some instances or the paid version of the product in others.


Google

hello I am in library

Acc. 0.92995489

Dragon Professional

hello i'm in library

Acc. 0.6859999999999999

DeepGram

Acc. 1

IBM Watson

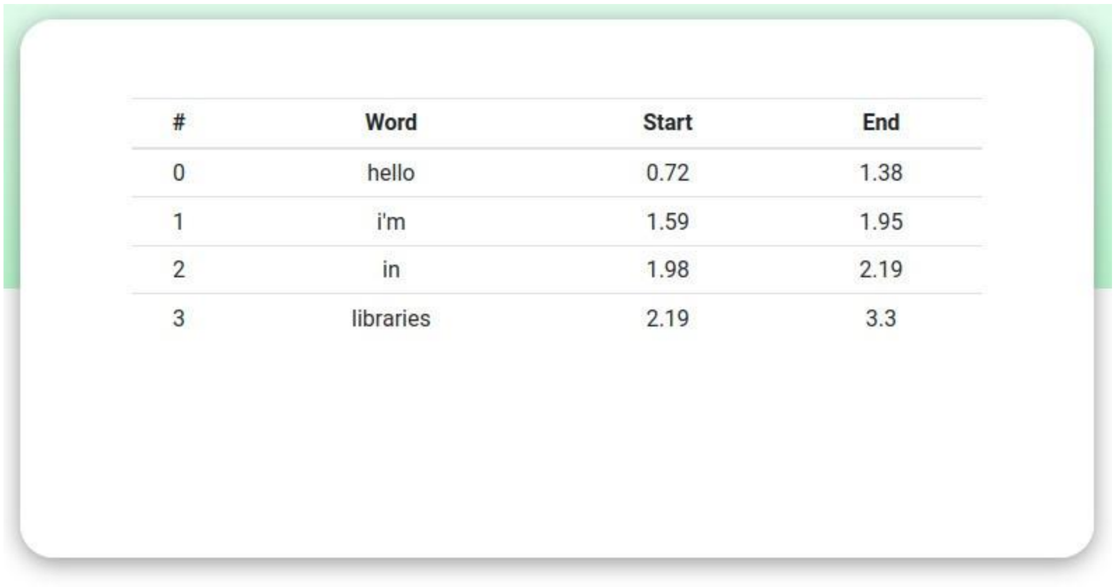
hello I am in the library teach

Acc. 0.5

Accuracy Comparison of Different Models

Figure 31: Comparisons of accuracy of different SR models

Although Google's accuracy is very high (i.e., 92%), the free version only provides us with 5–6 words, not the entire text. The premium version is necessary to get the complete voice to text. The fundamental problem with IBM Watson, DeepGram, and Dragon Professional is accuracy. Although they occasionally produce outcomes that are adequate, the majority of the time they don't. They also abandon them all in favour of Kaldi's automatic speech recognition and acoustic model for this reason. The results of the Kaldi model are discussed in the following section of the conclusion. Different words are at different locations in the audio and our main task requires the location of these words in the audio, because we have to perform certain tasks on the audio depending on their location as well as commands made by the users. The image from our product (flow) that displays the precise timestamp—that is, the beginning and end times of the word in the audio file—is shown below. The term is displayed in a table together with its start and end timestamps. It is another element of AudioPhonic that is incredibly beneficial to both consumers and developers.



#	Word	Start	End
0	hello	0.72	1.38
1	i'm	1.59	1.95
2	in	1.98	2.19
3	libraries	2.19	3.3

Figure 32: Extracted words from speech and their respective start and end times

Chapter 8

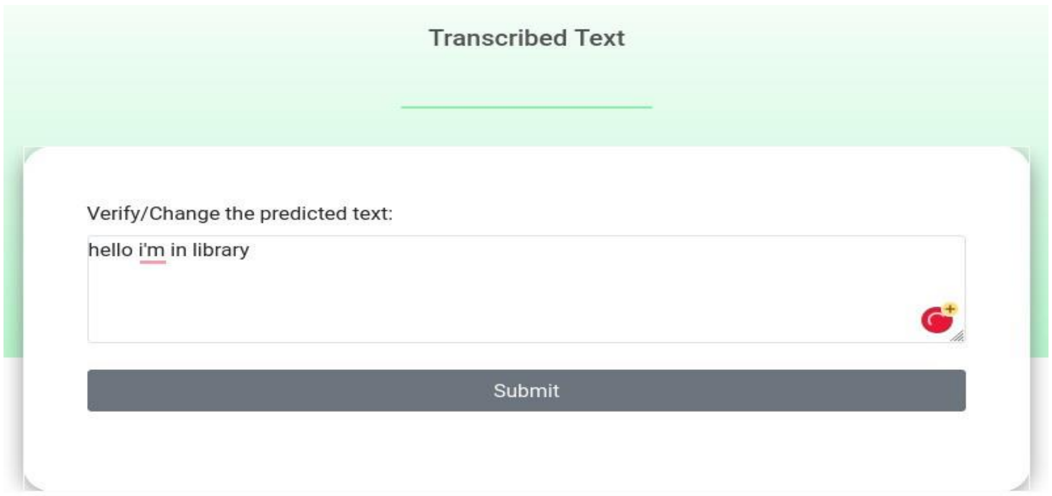
CONCLUSION & FUTURE WORK

8.1 Conclusion

The entire project is functioning successfully on the provided samples overall, although there are some improvements that might be made. We give our project the audio sample. The speech is converted into text via the web app. The user is then presented with the text. Change the wording as necessary. As the software converts the text into speech by taking into account the changes made to the text, the changes will be reflected in the speech. The largest flexibility of this app is that the speech will be delivered in real-time using a real person's voice. This is how our programme functions, and it gives us rather accurate results. The following list of speech recognition and text-to-speech conclusions.

8.1.1 Speech Recognition

One of the early stages of our product is speech recognition, and the major objective here is to transform the audio to text. Finally, we are generating transcripts using Kaldi's artificial speech and acoustic model. The Kaldi's ASR model's output is displayed in the image below. The many chapters up above go into great detail about the benefits and drawbacks of Kaldi's model. The text in the graphic, which is changeable, is produced from the audio file. Users can edit this text in any way (add, remove, or update), and after a few steps, the changes will be audible in the user's audio.


Transcribed Text

Verify/Change the predicted text:

hello i'm in library

Figure 33: Transcription of text from speech using SR models

8.1.2 Text-to-Speech

after sending the system the revised transcript. The system's primary responsibility after that is to translate the text back into speech such that it has the effect of a revised transcript. The online application interface is shown in the image below at the moment the voice is formed. If you take a close look from top to bottom. The user-added word (in the transcript) appears at the beginning at the time stamp (here the time is 2.19s). The button to restart is located after this. If there are any mistakes, it will return the user to the task's beginning.

Then finally, there is the audio player, that the user can run to listen to their edits on the original audio. User can also download the audio from the system, if the user is satisfied with the results from our product.

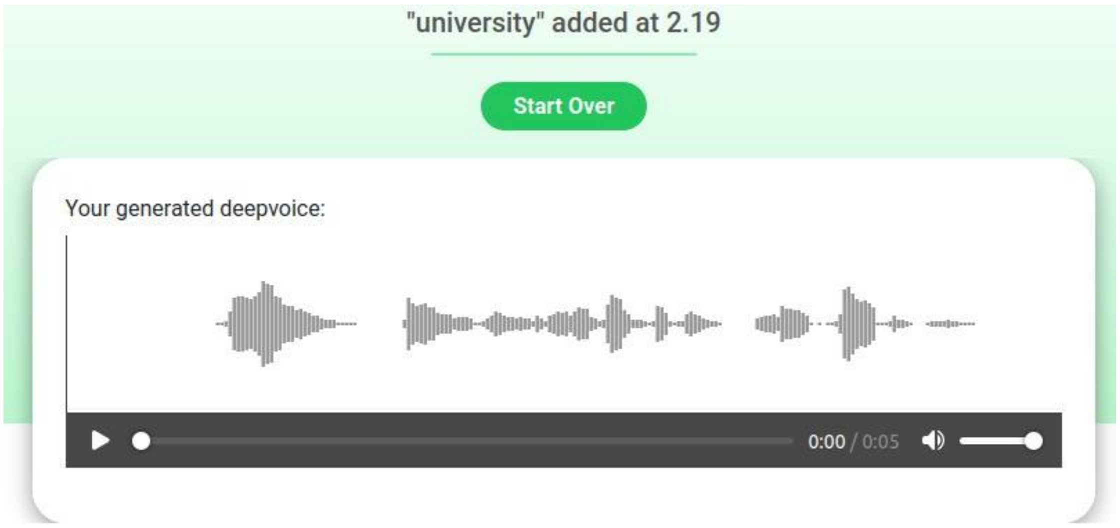


Figure 34: Generated deep voice combined with original audio

8.2 Future Works

Our software has a tonne of features that can be added in the future. Our programme already has a lot of features that function fairly well with audio. similar to translating both speech and text. We are considering introducing a few more excellent features in the future.

Writing about future works, we have made three more features which can be implemented. The three includes:

8.2.1 DeepFakes

The most significant and distinctive technique of the current period is deep fake. This method is feared by a lot of people since anyone can abuse it in any situation. However in contrast to that, there are people in the world who use this strategy effectively and believe it to be a good one. Deep learning models and artificial intelligence are both very effectively used in deep fakes. You can utilise deep fake as much as you want for creative purposes.
Deep fakes employ speech and image manipulation techniques. any image or speech, then converting it to a different format. Deep fakes generally use the generative adversarial networks. They can be used to create any synthetic audio and video.

With GANs, many individuals have produced synthetic audio and video. You can actually make a lengthy audio or video of your choice in addition to offering a tiny sample. Simply provide the persona's original audio and video, and the models will produce the audio and video for you. With GANs, you can even produce phoney audio and video of any individual. It uses the video after placing it into a neural network to create deep fakes.

As you may have seen in videos featuring actors and actresses' faces but which they very certainly did not make, this practise can lead to disaster.

Switching the faces of the people in the films can give the viewer a highly false impression.

This method also offers a lot of advantages. like how the globe has effectively become a small town. Regarding the film industry, it can free up the time of the actors and they are not need to be there during the film's filming. The shooting doesn't need to be recorded and filmed again.

Voice synthesis allows for the preservation of an actor's voice while also translating it into several languages. We set a goal for adding this feature, which will enable the creation of deep fakes. It can be used to alter videos by any creator or editor. If the creator neglected to include something in the video, he or she only needs to amend the text and the changes will be reflected in the video.

8.2.2 Bandwidth Challenges

This may be a really useful function in our software. We will be able to deliver theratf file and merely the words to the recipient using this function, via the internet. It will turn them into a voice after transferring both items to the user on the other computer. When little data is needed if there isn't much internet access, this can easily be done with relatively little bandwidth. It wouldn't require a lot of information, and people could readily learn items supplied to them via the internet.

8.2.3 Efficiency

The project works well on small samples. But when we upload a large sample it creates a problem. Like it will train the model and give the result for a small sample and less words. On the contrary, when you add more words, it will reduce the accuracy by like **10-20%** which is a bad thing. This thing needs to be changed and thisis in our future work.

The app is now platform-dependent, which is the second issue. It operates differently while operating on Windows, for example. It functions differently when running on a Mac. Hence, it is now platform-dependent. When we put it online or in the cloud, it will function flawlessly on a variety of devices.

Another issue is that the app's accuracy decreases when background noise is present. So it won't be an issue if you're sitting somewhere with very little background noise. As soon as there is background noise, accuracy will be reduced because the text won't be correctly extracted.

Taking into account that our web application is currently inefficient and requires some time to process the audio before converting. By creating a mobile app, we want to increase its viability. To increase the app's efficiency, we will make it compatible with local devices and more effective models. Any app that is operating locally on the device works considerably more quickly than when it is offline. It becomes more efficient by utilising effective models and operating offline on devices.

chapter 9

REFERENCES

[1] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, The HTK Book (for version 3.4). Cambridge University Engineering Department, 2009.

[2] A. Lee, T. Kawahara, and K. Shikano, “Julius – an open source real time large vocabulary recognition engine,” in EUROSPEECH, 2001, pp. 1691–1694.

[3] W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, and J. Woelfel, “Sphinx-4: A flexible open source framework for speech recognition,” Sun Microsystems Inc., Technical Report SML1 TR2004-0811, 2004.

[4] B. Guven, “extracting speech from video using python”, Aug. 2020. Available: https://towardsdatascience.com/extracting-speech-from-video-using-python- f0ec7e312d38

[5] D. Povey, “The kaldi speech recognition toolkit,” 2011. Available: https://infoscience.epfl.ch/record/192584

[6] Yury Pinsky, “Tomato, tomahto. Google Home now supports multiple users,” https://www.blog.google/products/assistant/tomato-tomahtogoogle-home-now- supports-multiple-users, 2017.

[7] Mihai Matei, “Voice match will allow Google Home to recognize your voice,” https://www.androidheadlines.com/2017/10/voice-matchwill-allow-google-home-to-recognize-your-voice.html, 2017.

[8] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio, “Understanding the exploding gradient problem,” CoRR, vol. abs/1211.5063, 2012.

[9] L. Wan, “Generalized end-to-end loss for speaker verification,” Nov. 2020. Available: https://arxiv.org/pdf/1710.10467.pdf

[10] Amodei, Dario, Anubhai, Rishita, Battenberg, Eric, Case, Carl, Casper, Jared, Catanzaro, Bryan, Chen, Jingdong, Chrzanowski, Mike, Coates, Adam, Diamos, Greg, et al. Deep speech 2: End-to-end speech recognition in English and mandarin. arXiv preprint arXiv:1512.02595, 2015. Boersma, Paulus Petrus Gerardus et al. Praat, a system for doing phonetics by computer. Glot international, 5, 2002.

[11] Bradbury, James, Merity, Stephen, Xiong, Caiming, and Socher, Richard. Quasi- recurrent neural networks. arXiv preprint arXiv:1611.01576, 2016.

[12] Amin, M., Shah, B., Sharif, A., Tanveer, T., Kim, K.L., Anwar, S.: Android malware detection through generative adversarial networks. Trans. Emerg. Telecommun. Technol. (2019). https://doi.org/10.1002/ett.3675

[13] L. Graziani, “Generating facial expressions associated with text,” Lecture Notes in Computer Science, vol. 12396, Oct 2020.. Available: https://link.springer.com/chapter/10.1007/978-3-030-61609-0_49

[14] S. Huang, “Meta-TTS:Meta-Learning for few short speaker adaptive text-to- speech,” ACM Transactions on Audio, Speech, and Language Processing, vol. 30, pp. 1558–1570, 2022. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9756900

[15] https://www.statista.com/