“YOUTUBE VIDEO SUMMARIZER”

Submitted in partial fulfillment of the requirements of the degree

BACHELOR OF ENGINEERING

IN

COMPUTER ENGINEERING

By

1) Rahul R. Chaurasia	**Roll No 08**
2) Pratiksha P. Kirolkar	**Roll No 36**
3) Minal R. More	**Roll No 43**

Name of the Mentor

Prof. S. A. Awate

Department of Computer Engineering
A.C. Patil collage of Engineering Kharghar,
Navi Mumbai
University of Mumbai

(AY 2021-22)

CERTIFICATE

This is to certify that the Mini Project entitled **"YouTube Video Summarizer"** is a bonafide work of **Pratiksha Kirolkar(201042001), Minal More(201042005), Rahul Chaurasia (201042016),** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Bachelor of Engineering"** in **"Computer Engineering" .**

(Prof. S. A. Awate)
Mentor

(Dr. M. M. Deshpande)
Head of Department

(Dr. V. N. Pawar)
Principal

Mini Project Approval

This Mini Project entitled **"YouTube Video summarizer"** by **Pratiksha Kirolkar (201042001), Minal More (201042005), Rahul Chaurasia (201042016),** is approved for the degree of **Bachelor of Engineering** in **Computer Engineering.**

Examiners

1……
(Internal Examiner Name & Sign)

2……
(External Examiner name & Sign)

Date:

Place:

Contents

ABSTRACT

This project discusses a YouTube video summarizer using python language. A project created for YouTube video summarization since, lots of videos are uploaded every single day around the world which are very long, and if you want to find some useful information from those videos, it's an almost impossible task to achieve since, most of the videos contain so much useless buffer content, because of this you have to watch the whole video which probably wastes a lot of time and efforts.

So, generating the summaries of those video transcripts will save your lots of time and you will quickly gather more useful and important information from it, which will surely save your efforts and time to watch the whole video.

The project which we have made extracts the summaries from video transcripts and looks for important keywords from it to create a short summary.

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to several individuals and organizations for supporting us throughout the project.

First, we wish to express our sincere gratitude to our supervisor Prof. S. A. Awate for her enthusiasm, patience, insightful comments, helpful information, practical advice and unceasing ideas that have always helped us tremendously in our research and writing of this thesis.

The Immense knowledge, profound experience, and professional expertise in quality control has enabled us to complete this project successfully. Without her support and guidance, this project would not have been possible.

Thanks for all your encouragement!

List of Figures

List of Tables:

1.Introduction

1.1 Introduction

Enormous number of videos are being created and shared on the Internet throughout the day. It has become really difficult to spend time in watching such videos which may have a longer duration than expected. By summarizing the captions or transcript of any YouTube video, it is able to pull the most important information and condense it into a small paragraph. Reading this paragraph would take a tiny fraction of the total length of the video, while still providing the most important points to the user.

Due to advances in video streaming and expansion of low-cost storage media, digital video has become an important factor in education, entertainment, and commerce. Consequently, there has been a great interest in designing and building systems that organize and search video data based on its content. In addition to search capabilities, such systems should be able to derive intuitive and compact data representations so that users may easily and quickly browse through the whole database or through the results of a query. Such representations rapidly provide the user with information about the contents of the particular sequence being examined while preserving the essential message.

Video summarization is a task where a video sequence is reduced to a small number of still images called keyframes, sometimes called storyboard or thumbnails extraction, or a shorter video sequence composed of key shots, also called video skim or dynamic summaries. The keyframes or key shots need to convey most of key information contained in the original video. This task is similar to a lossy video compression, where the building block is a video frame. In this paper we focus solely on the key shots-based video summarization.

1.2 Motivation

The main motivation to make the YouTube Video Summarizer is to save time. To get the concept of the video rather than watch the entire video for a tremendous time.

There are countless occurrences where YouTube videos lure users in with eye catching titles and thumbnails, and the next thing they know, half the day has passed. There are several workarounds to saving time already, such as watching the video on double speed, however this only reduces the watch time by half.

This is where YouTube video summarizer comes in, by summarizing the captions or transcript of any given video, it is able to pull the most important information and condense it into a small paragraph, and reading this paragraph would take a tiny fraction of the total amount of time required to watch the entire video, while still providing the most important points to the user.

1.3 Problem Statement & Objectives

We spend a considerable amount of time on YouTube which is also a primary source of knowledge because it contains lecture videos from primer institutes. Sometimes, we watch an hour-long video and try to grab the important ideas from it. What if we can let a system do it for us by just providing the link of the video. Understand the important ideas from the video and summarize it in a way which will allow you to look back and revisit the key ideas without

watching the video all over again.

Objectives:

- You Tube Video Summarization aims to generate a short synopsis that summarizes the video content by selecting its most informative and important parts.
- To save time by showing a brief on what topics are covered in the video.
- Our project uses python techniques to extract the summary of the video. Its always difficult to run these codes for a common user. What if we have a GUI which allows us to just paste in the link of the YouTube video and the application does all the hard lifting for us and saves the summary in the text format in the desired location.

1.4 Organization of the Report

Chapter 1

This chapter include initiation of this project where we will analyze problem, project motivation and objectives.

This will be categorized as follows:

1. Introduction

2. Motivation

3. Problem statement

4. Objectives

Chapter 2

This chapter is all about the analysis of existing systems, the problems and flaws of this system, the reference we used to make this project and further reading.

This will be further categorized as:

1. Survey of existing system

2. Limitation of existing system

3. Mini project contribution

Chapter 3

Here, we analyze the architecture and the implementation of our software, it also looks at the possible obstacles and the actual obstacles faced along with our test runs and conclusion.

This will be further categorized into:
1. Introduction
2. Architecture/ Framework
3. Algorithm of application
4. Details on software and hardware used
5. Test criteria
6. Test results
7. Conclusion
8. Further work

2. Literature Survey

The existing system for video summarization takes an input from the user and creates summary. The aim of this project is to create an application which generates a text summary which can be easily understood by user.
In the existing system, Srinivas et al., 2016 proposed an improved method for video summarization techniques. The aim is to get a summary content of a video which is interesting to the viewer and representing the whole video.

2.1 Survey of Existing system /SRS

The previous work of this already exists. In existing system, they created the summarization only in one language.

- Over the last decade, several YouTube summarizers tools have been developed and it is still receiving increased interest from both educators and researchers.
- In this project, there is a limitation of user input like only selected you tube video link can be access.
- There is no audio effect and limitations in languages.

Sr. No	Title of Review Paper	Year	Review
1	Visualization Tools of Data Structures Algorithms	2014	This study provides better understanding of visualization tool and provides knowledge to the beginners who want to work in visualization tools.
2	Large scale deep neural network acoustic modeling with semi-supervised training data for YouTube video transcription	2013	YouTube videos are extremely challenging for automatic speech recognition systems. Standard adapted Gaussian Mixture Model (GMM) based acoustic models can have word error rates above 50%, making this one of the most difficult reported tasks.

Table 1: Literature survey

2.2 Limitation Existing System or Research Gap

In some of the existing applications, there is only an option to generate a text summary, while there is no feature for changing the language of the generated summary, as well as you cannot listen to it. Some the extensions don't provide the functionality while others don't work properly.
Limitations:
1) Most of the available software are extensions which are not available for all browsers.
2) No text to speech conversion
3) Does not provide the ability to change languages.
4) Can crash a browser if they take up too much resources.

2.3 Mini project contribution

Our project is a small attempt for developing YouTube Video Summarizer using python. Firstly, we generated set of algorithms that we want to implement in this project with the help of many existing system and their limitations. While keeping the design simple and not including too many features user can easily understand the working of transcript and summarize the video into text without having to give their attention elsewhere.
It gives the complete satisfaction of knowing that the machine doesn't affect the success of Project. Project will help to enhance the skills.

3. Proposed System

3.1 Introduction

You tube video summarizer will help in creating short and meaningful summary for YouTube video. The main idea behind this project is to save time. Rather than watching the entire video it will be great to create meaningful summary.

The below diagram shows the working of project.

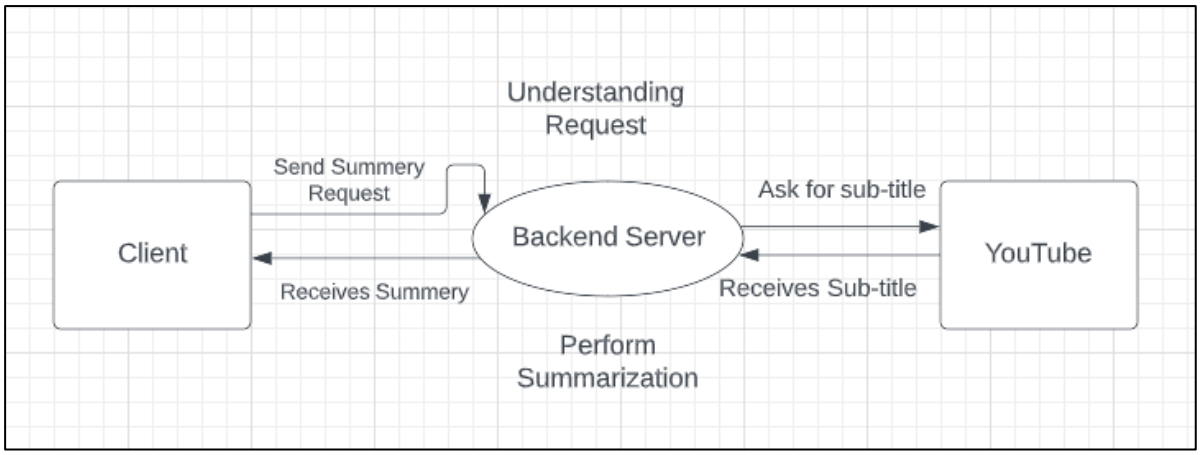

Figure 1: Demonstration

To create summary, it is necessary to take a YouTube video with caption.

3.2 Architecture And Framework

Use case diagram

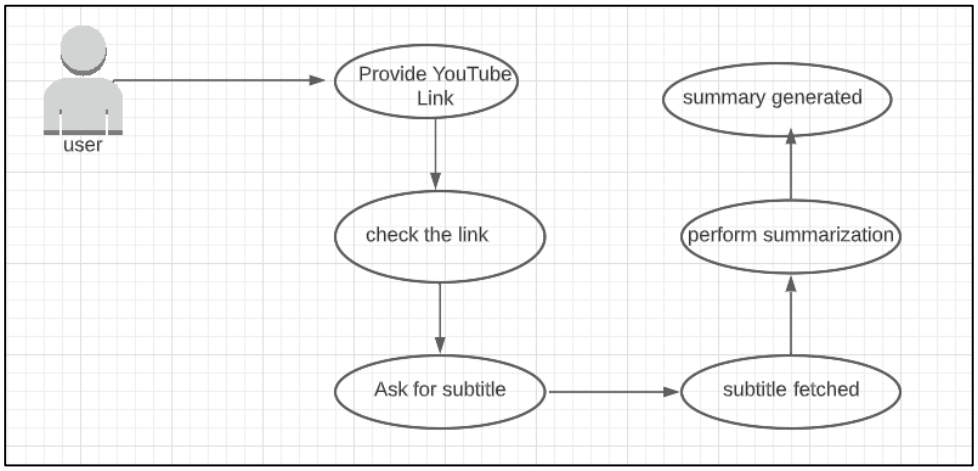

Figure 2: Use Case

3.3. Algorithm and process design

Algorithm

- Extract transcript using ytapi
- Summarize the transcript using pipelines.
- Translate the summary into various dialects using google translator
- Audio output using gTTS
- For UI we have used tkinter

Flowchart

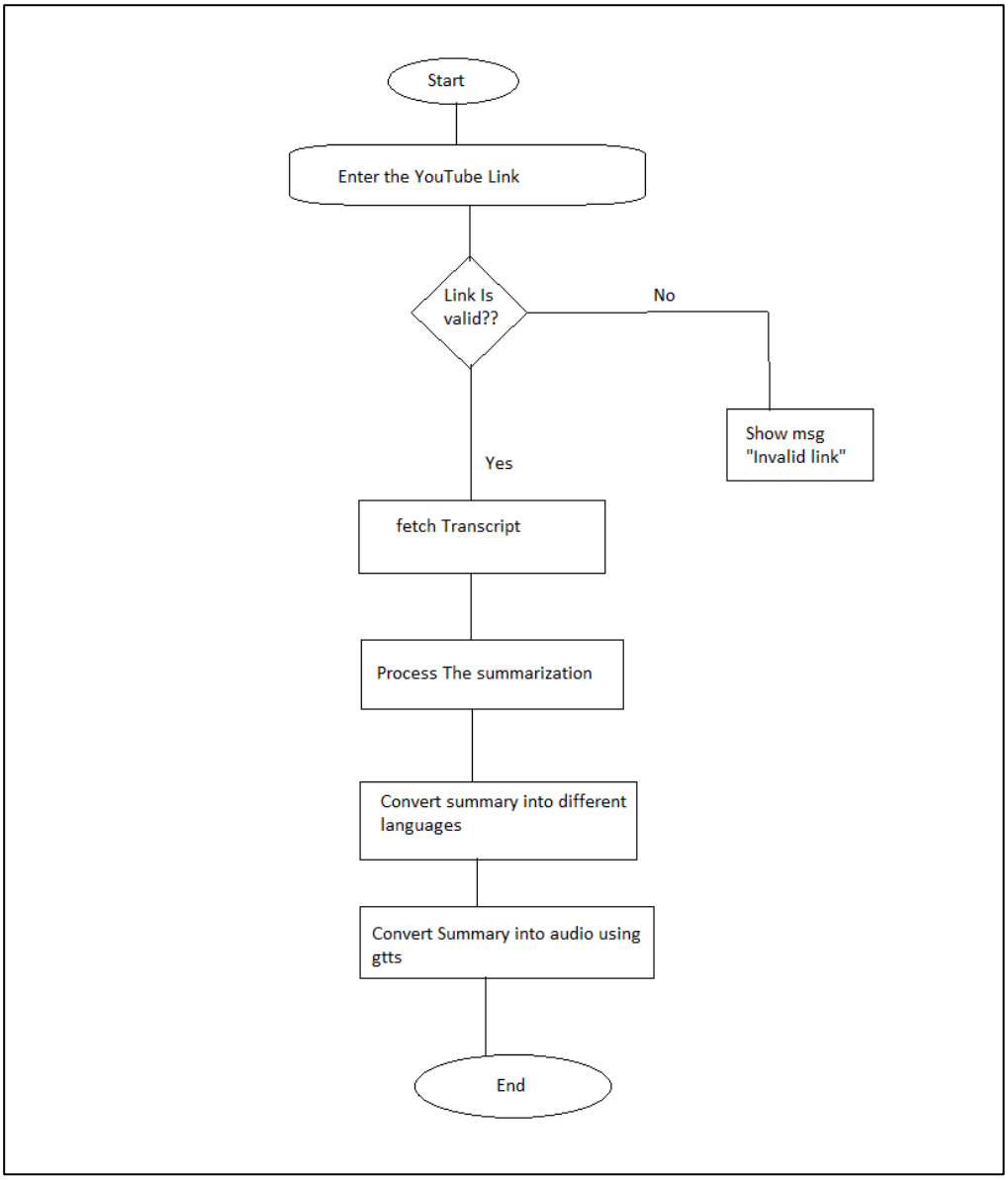

Figure 3: Flowchart

Flowchart to check whether the link is valid or not

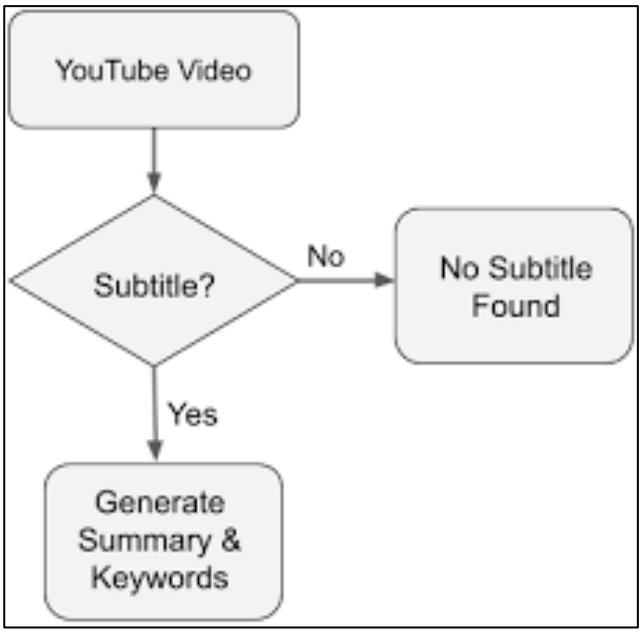

Figure 4: Flowchart for Link

3.4 Details of Hardware & Software

Software Requirements:

Operating System - Windows 10
Language - python
Browser - Any of Mozilla, Opera, Chrome etc.
Software Development Kit – Jupyter notebook, Google Colab

Hardware Requirements:

Processor – Intel Core i3
RAM – 8GB

3.5 Test Cases (Procedure and Expected Results)

Test case ID	Test Case Objective	Step ID	Test Data	Expected Result	Actual Result	Status
TC 01	To test the YouTube Link	1)	Provide YouTube Video Link	Link Should be Valid	Valid Link	Pass
TC 02	To test the YouTube Link	1)	Provide invalid YouTube Video Link	Msg should be display "Invalid Link"	"Invalid Link"	Pass
TC 03	To test the summarization functionality	1)	Click on Required action	Summarization should be performed	Video summarized	Pass
TC 04	To test the summarization functionality with YouTube video which does not have caption	1)	Enter a video link which does not have caption	Summarization should not be performed	Summarization unable to performed.	Pass
TC 05	To test the translate language functionality	1)	Click on different language	Summary should be translated into selected language	Summary displayed in selected language	Pass
TC 06	To test the translate audio functionality	1)	Click on play button	Summary should be played into audio format	Summary played into audio format	Pass

Table 2: Test Cases

3.6 Code Testing and Result

Testing objectives:

Unit testing is the testing of the individual components (units) of the software. Unit testing is conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.
When developing a strategy for unit testing, there are three basic organizational approaches that can be taken. These are top down, bottom up and isolation.
In our case of a YouTube video summarizer, we simply use a top-down approach.
The project is divided into to two main parts i.e.: the summarization part and the GUI part. For the summarization part the user has to enter a link to the YouTube video from which the transcript will be fetched and the summarization will be performed of the YouTube video if it exists otherwise, it will throw an error. For the GUI part a GUI is created using tkinter which will fetch the summary generated in the first part and then provide text-to-speech using google text-to-speech services and the functionality to change languages will be implemented as well.
Here are the essential software testing steps every software engineer should perform before showing their work to someone else.

1. Basic functionality testing

Begin by making sure that every button on every screen works. Also need to ensure that you can enter simple text into each field without crashing the software. If the feature is designed to be accessed by way of an API, you need to run tests to make sure that the basic API functionality works before submitting it for more intensive testing. If your basic functionality testing detects something that doesn't work, that's fine.

2. Code Review

Another pair of eyes looking at the source code can uncover a lot of problems. If your coding methodology requires peer review, perform this step before you hand the code over for testing. Remember to do your basic functionality testing before the code review, though.

3. Static Code Analysis

There are tools that can perform analysis on source code or bytecode without executing it. These static code analysis tools can look for many weaknesses in the source code, such as security vulnerabilities and potential concurrency issues. Use static code analysis tools to enforce coding standards, and configure those tools to run automatically as part of the build.

4. Unit Testing

We will write unit tests to make sure that the unit (be it a method, class, or component) is working as expected and test across a range of valid and invalid inputs. In a continuous integration environment, unit tests should run every time you commit a change to the source code repository, and you should run them on your development machine as well. Some teams have coverage goals for their unit tests and will fail a build if the unit tests aren't extensive enough.

Result:

```
In [7]: original_text = final_tra1
        summary_text = summarization(original_text)[0]['summary_text']
        print("Summary:", summary_text)

Token indices sequence length is longer than the specified maximum sequence length for this model (1442 > 512). Running this se
quence through the model will result in indexing errors

Summary: if you have created a ui for an application and you want to show that application inside an actual mobile phone then t
his website is very useful in creating mockups . you can either press get started or create a mockup on the website you can sel
ect the particular template and download the mock-up that's it . the website has numerous mobile phones like iphone google pixe
l and whatnot it has tablets it has monitors it has tv screens and even smart watches .
```

Figure 5: Summary Generated

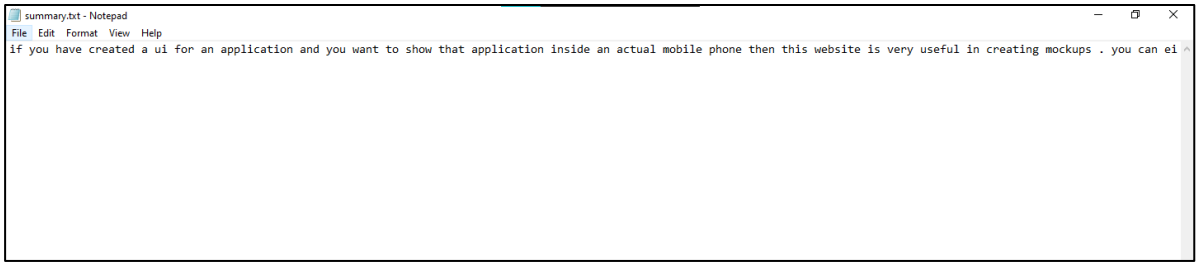

Figure 6: Summary.txt

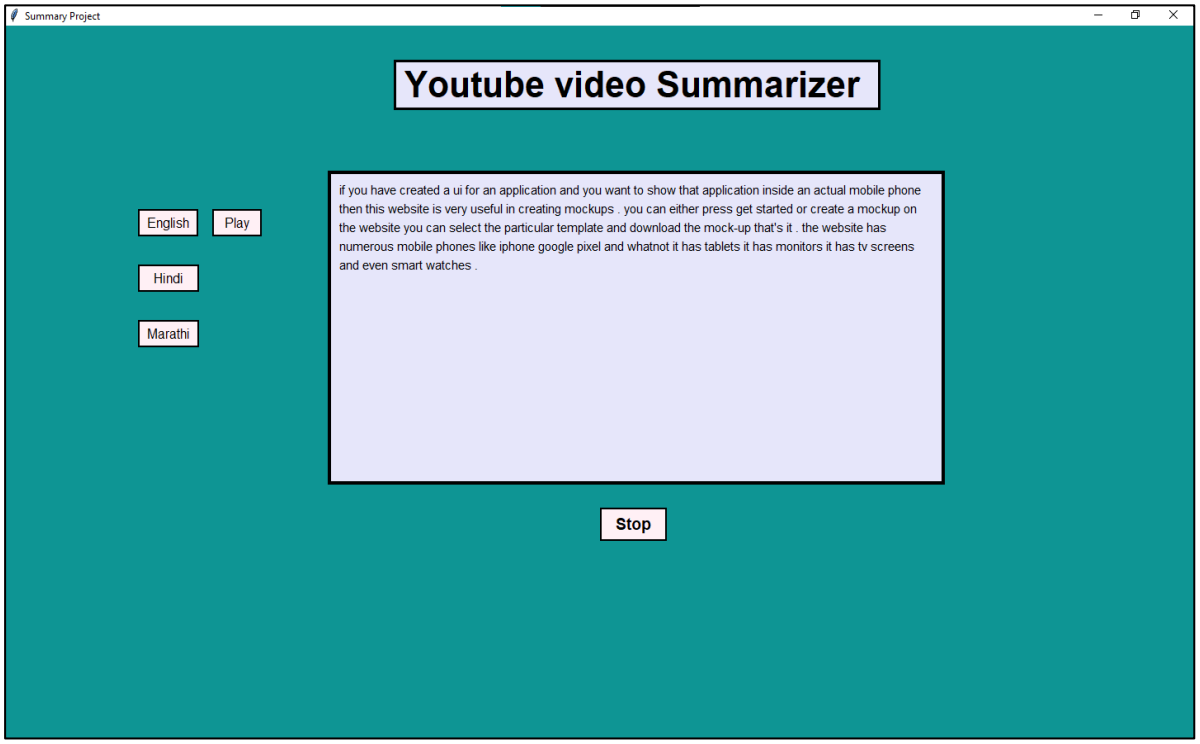

Figure 7: GUI

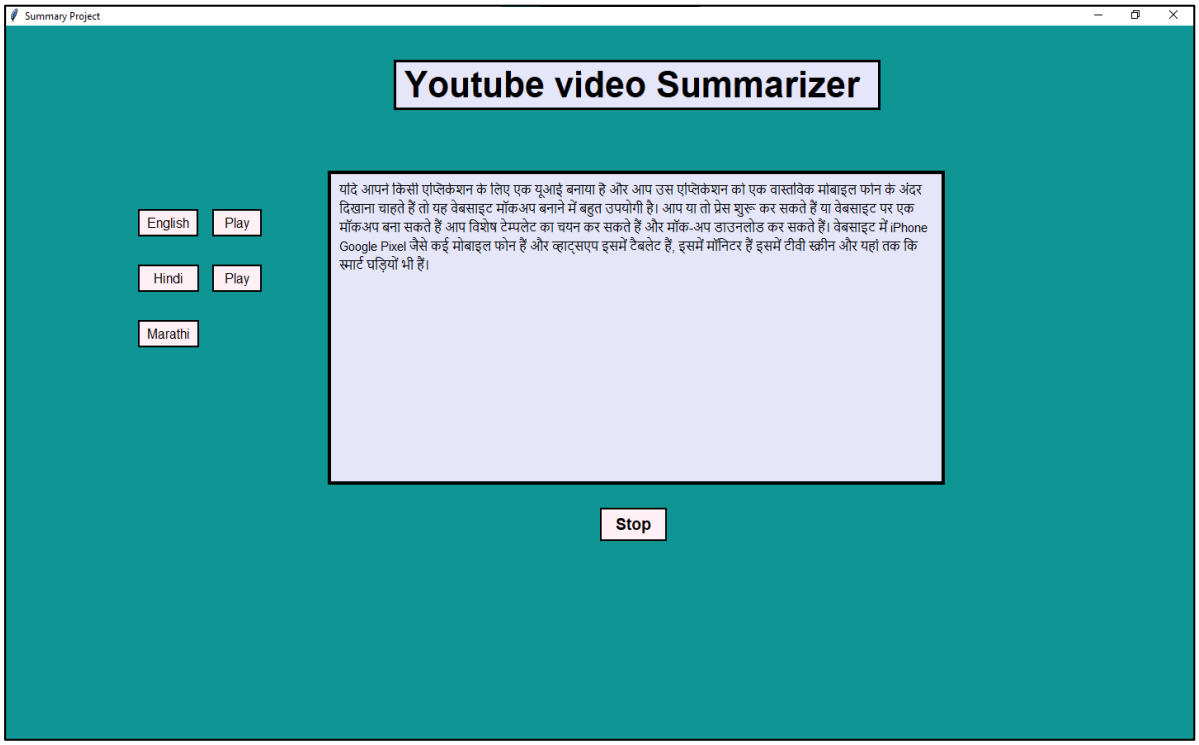

Figure 8: Conversion language (Hindi)

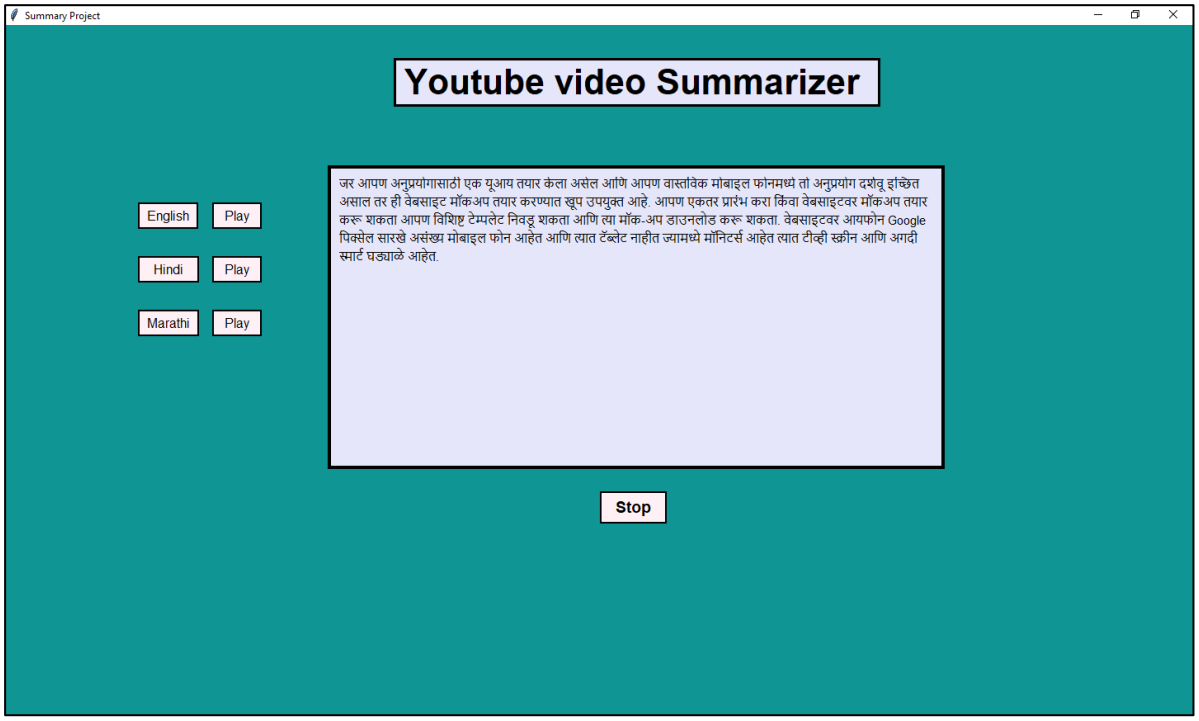

Figure 9: Conversion of language (Marathi)

3.7 Conclusion and Future work

The objective of the YouTube video summarizer is to make system which will give short summary for the YouTube video.
By doing this project we able to access summary for the huge video.

Future work:

Our project will be able to implements in future after making some changes and modifications that can be done in our project are:

1) Using different algorithms for summarization
2) By Creating a better GUI.
3) Using different languages.
4) It can also be used for YouTube Video which does not have captions.

References

- pypi.org project: youtube-transcript-api
- pythonbasics.org : transcribe-audio
- www.thepythoncode.com
- translate-text-in-python