AI Based Lecture Summarization Tool

Team No. 9

1 Problem Statement

The onset of COVID-19 pandemic has devastated several economies and industries with the education industry being particularly hard it. There has been a marked shift from offline classroom based learning to online learning methods using applications such as Zoom, Microsoft Teams and Google Hangouts. The lecture videos are often uploaded on cloud service providers so that students can access them whenever required.

Despite the many advantages of online learning, it faces some severe disadvantages such as:

- Lack of accessibility due to internet connectivity issues
- Low attention span of students as compared to physical learning in classroom settings [1]
- Uploaded videos are difficult to revise as compared to class notes

As part of our end semester project in Software Engineering, we aim to address a few of the above problems by using AI to automatically summarize lecture videos. Our solution would leverage the latest advancements in machine learning and computer vision to make it very easy for users to revise using the uploaded videos.

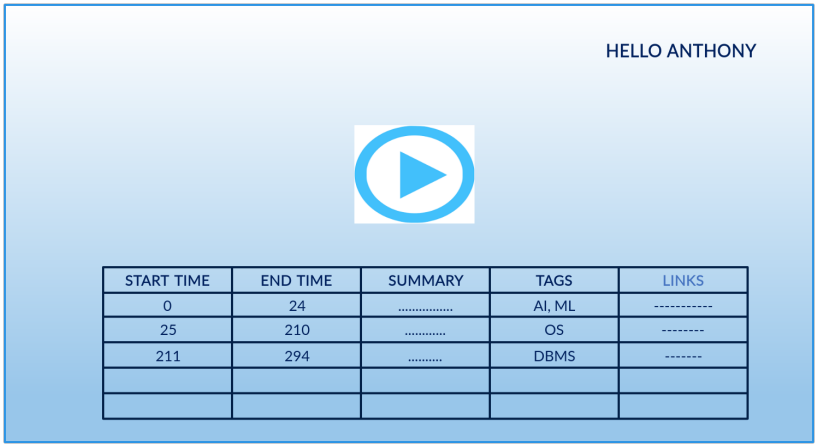

Figure 1: User Interface of website

In essence, we do the following:

1. Given a video (uploaded by the users), we split it into several meaningful short sections/clips based on visual cues
2. For each clip, we generate the captions of the lecture video using deep learning models for speech recognition
3. We generate keywords and a summary for each of the above clips using extractive and descriptive summarization tools
4. The short clips are then indexed using the corresponding keywords and can be searched by the user using a web UI

Figure 1 provides a brief description of our product. The user uploads the required lecture video and Figure 1 shows the results of our algorithm. The user is shown a list of short clips in a chronological order with the corresponding start and end times, keywords and link to the video respectively. There is also a search interface, which can be used to search for a particular clip based on the required keywords.

This project thus provides a method to summarize videos effectively and makes revision of online videos easier. Moreover, the search feature will allow students to pick out parts of video lectures specific to a given topic, and save them the extra manual effort of going through the whole lectures.

2 Solution

We will develop our software using a client-server architecture and the user will interact with our product using a web application.

- The client side will provide the user with a UI where they will be able to login, and add, search & summarize the videos
- The server would deal with all the client side requests such as:
  - Provide authentication methods for login
  - Summarize a given uploaded video
  - Return the most relevant clips for a given search query

The task of summarizing a video by splitting it into meaningful portions is a very difficult task. Our solution uses a combination of image/video processing, machine learning and summarization techniques to solve the problem in a fast and efficient manner. The solution provided below is inspired from a M.Tech Thesis project at IIT Kharagpur [2]. Figure 2 shows the overall architecture of our algorithm. We describe briefly the details of each individual module below.

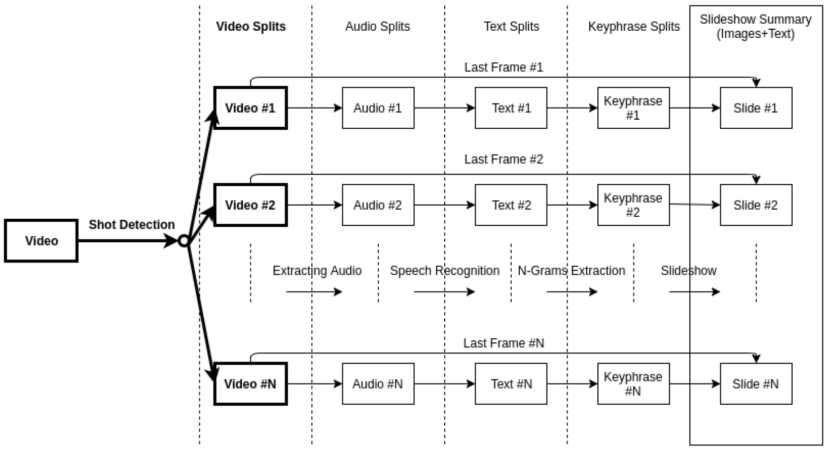

Figure 2: Architecture diagram of our algorithm

2.1 Shot Detection

Shot detection refers to automated detection of transitions between shots in a given video. In the case of video lectures, this would help in splitting the long videos (usually ¿ 1 hour) into several short segments that are meaningful. There are several open-source shot detection methods available that use thresholding algorithms with different scoring metrics. In our case, we intent to use the PySceneDetect library for the same but we will create an interface so that if need be, the work can be extended to use other libraries as well.

The input for this step is a single large video and the output would be multiple smaller video clips which are meaningful.

2.2 Audio Extraction

In this step, we simply extract the audio out of each individual video clips using tools such as FFMPEG.

2.3 Speech Recognition

Speech recognition refers to the task of converting the audio into corresponding text. There are several methods to do the same, but in recent years deep learning based models are outperforming the traditional methods significantly. As these models are difficult to train and test on our computers, we utilize the APIs provided by companies such as Google, Microsoft and Amazon.

We intend to create an interface through which we utilize the APIs for transcribing the given speech/audio. At present, we aim to use the Google Speech Recognition API for the same due to availability of free credits.

2.4 Summarization

Automatic text summarization is the task of producing a concise and fluent summary while preserving key information content and overall meaning. There are two types of summarization:

- Extractive Summarization
- Abstractive Summarization

In extractive summarization, we identify the important sentences from the original text and extract only those from the text. This is a naive form of summarization but can be done easily using NLP based methods and python libraries such as NLTK. We also aim to identify keywords for each video clip using NLP techniques such as summarization and TF-IDF analysis.

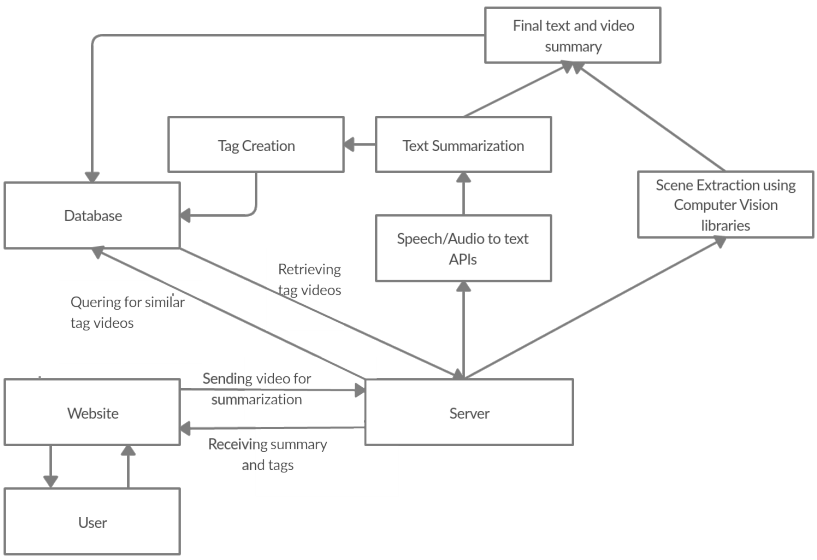

Figure 3: High Level System Diagram

2.5 Search

The main aim of the project is to provide the user with a means to search for the relevant portions of a video when revising a particular concept. Thus, we intend to index each segment with the keywords that were generated in the previous section and provide a tag based search UI for the user.

3 Design Documents

Below, we have provided a high level design of our project which consists of the individual modules along with the interfaces that we aim to implement. We show the High Level Design of the whole

system in Figure 3. We also describe the details of our summarization algorithm in Figure 4. Figure 4 describes the high level design of our server based algorithm/API.

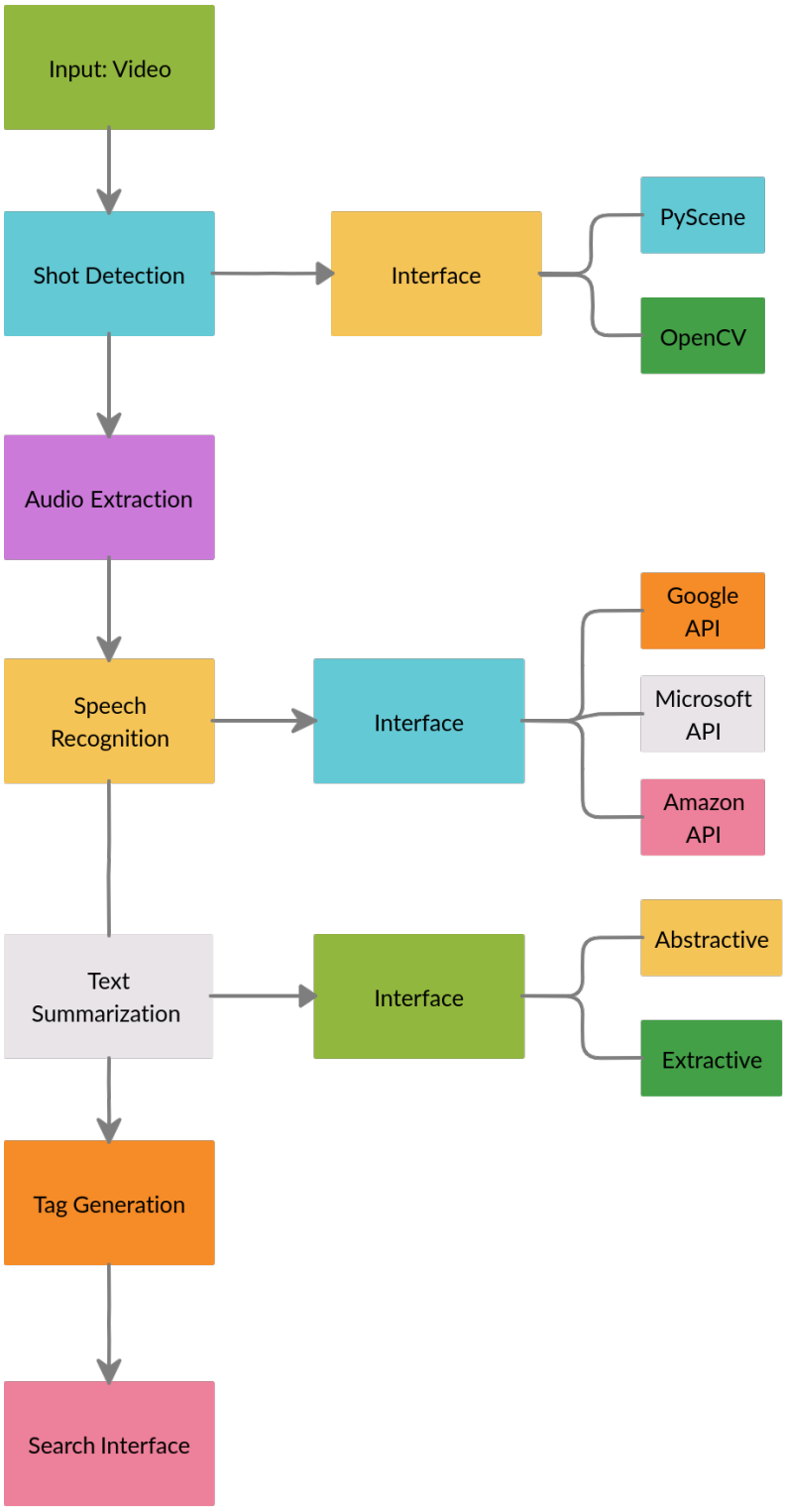

Figure 4: High Level Design of our algorithm

4 Timeline and deliverables

We aim to use the various methodologies taught in class such as Software Development Life Cycle and tracking methods such as Backlogs/SCRUM to manage the project and present the deliverables

on time.

References

[1] Nitza Geri, Amir Winer, and Beni Zaks. A learning analytics approach for evaluating the impact of interactivity in online video lectures on the attention span of students. *Interdisciplinary Journal of E-Learning and Learning Objects*, 13(1):215–228, 2017.

[2] Shamik Sural Shreyans Pagariya. Video summarization. *M.Tech Project Report*, 2018.