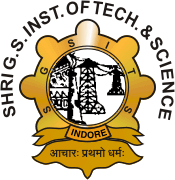
**"Video Summarizer: A tool for summarizing educational videos"**



**A project report of Phase-I submitted to**

**Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal**

**towards partial fulfilment of the**

**degree of**

**Bachelor of Technology**

**in**

**Computer Engineering**

**Guided By:**

**Submitted By:**

0801CS201009 – Akshat Gupta

0801CS201012 – Anekant Shingai

0801CS201024 – Avni Sharma

0801CS201027 – Chevi Parsai

0801CS201068 – Pragati Gupta

Ms. Himani Mishra

Assistant Professor

Department of Computer Engg.

Dr. Vandan Tewari

Professor

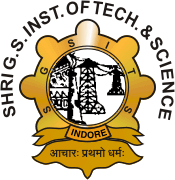
Department of Computer Engg.

**DEPARTMENT OF COMPUTER ENGINEERING**

**SHRI G.S. INSTITUTE OF TECHNOLOGY AND SCIENCE, INDORE(M.P.)**

**2023-2024**

**SHRI G.S. INSTITUTE OF TECHNOLOGY AND SCIENCE, INDORE(M.P.)**



**RECOMMENDATION**

The project report of Phase-I entitled "Video Summarizer: A tool for summarizing educational videos" submitted by 0801CS201009 –Akshat Gupta, 0801CS201012 –Anekant Shingai, 0801CS201024 –Avni Sharma, 0801CS201027 –Chevi Parsai, 0801CS201068 –Pragati Gupta, students of B.E. III year in the session 2023-2024, towards partial fulfilment of the degree of Bachelor of Technology in Computer Engineering of Rajiv Gandhi Proudyogiki VishwaVidhyalaya, Bhopal is a satisfactory account of their work.

**Dr. Vandan Tewari**

**Head**

Department of Computer Engg.

**Dean (Academics)**

**S.G.S.I.T.S. Indore**

**Ms. Himani Mishra**

Project Guide

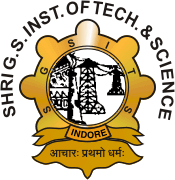
Department of Computer Engg.

**Dr. Vandan Tewari**

Project Co-Guide

Department of Computer Engg.

**SHRI G.S. INSTITUTE OF TECHNOLOGY AND SCIENCE, INDORE(M.P.)**



**CERTIFICATE**

The project report of Phase-I entitled "Video Summarizer: A tool for summarizing educational videos" submitted by 0801CS201009 -Akshat Gupta, 0801CS201012 –Anekant Shingai, 0801CS201024 –Avni Sharma, 0801CS201027 –Chevi Parsai, 0801CS201068 –Pragati Gupta, students of B.E. III year in the session 2023-2024, towards partial fulfilment of the degree of Bachelor of Technology in Computer Engineering of Rajiv Gandhi Proudyogiki VishwaVidhyalaya, Bhopal is a satisfactory account of their work.

**Internal Examiner External Examiner**

**Date:**

**ACKNOWLEDGEMENT**

We express our profound sense of gratitude to our project guide Ms. Himani Mishra and Co-Guide Dr. Vandan Tewari who had advised us to do project work entitled “Video Summarizer: A tool for summarizing educational videos”. Their continuous support and motivation always made us deliver our best. Having such guidance has always been an amazing experience which is a valuable gift for an engineer to progress in his/her life.

We are also grateful to Dr. Vandan Tewari, Head, Department of Computer Engineering and Dr. Rakesh Saxena, Director, S.G.S.I.T.S., Indore for providing us with numerous facilities and academic environment during the study.

We sincerely wish to express, our gratefulness to all the members of staff of S.G.S.I.T.S. who have extended their cooperation at all times and have contributed in their way in developing the project.

The successful completion of a project is not an individual effort. It is an outcome of the cumulative number of people, each having their importance to the objective. We express love and respect towards our parents and all family members who are our strength in everything we do.

With a blend of gratitude, pleasure, and great satisfaction we convey our indebtedness to all those who have directly or indirectly contributed to the successful completion of our project work.

0801CS201009 – Akshat Gupta \_\_\_\_\_\_\_\_\_\_\_\_\_

0801CS201012 – Anekant Singhai \_\_\_\_\_\_\_\_\_\_\_\_\_

0801CS201024 – Avni Sharma \_\_\_\_\_\_\_\_\_\_\_\_\_

0801CS201027 – Chevi Parsai \_\_\_\_\_\_\_\_\_\_\_\_\_

0801CS201068 – Pragati Gupta \_\_\_\_\_\_\_\_\_\_\_\_\_

**ABSTRACT**

Watching videos as a good way of gaining information but the length of videos can be long and sometimes, they may not contain the relevant information. To save this time and effort we create a summary from the video by extracting its audio and image and then combining those to obtain summary.

The main functionality of this application is to utilize different machine learning algorithms and natural language processing to generate and accurate summary from the audio and images extracted from the video.

Video summarization is a process of condensing the information conveyed in a video or sequence of images into a concise and informative text summary. With the explosion of online video content, it has become increasingly important to quickly and efficiently extract the key information from these sources. Video and image summarization tools use advanced technologies such as computer vision, machine learning, and natural language processing to analyse the visual and auditory content and generate a summary that captures the essential points and highlights. These tools are useful for a range of applications, including news reporting, educational content, and marketing videos, as they allow users to quickly and easily understand the most important aspects of a video or image sequence without having to watch or view the entire content.

**CONTENTS**

**Recommendation**

**Certificate**

**Acknowledgements**

**Abstract**

1. **Introduction**
   1. Preamble
   2. Need of the project
   3. Problem Statement
   4. Objectives
   5. Proposed Approach
   6. Organization of the Reports
2. **Background Study**
   1. Tools & Technologies
      1. OpenCV
      2. MoviePy
      3. Speech Recognition
      4. NLTK (Natural Language Toolkit)
      5. Tesseract
      6. PyTorch or Tensorflow
      7. Numpy
      8. Pandas
      9. Matplotlib or seaborn
   2. Existing Systems
3. **Literature Review**
   1. Audio Extraction
   2. Frame Extraction
   3. Visual Features Extraction
   4. Frame Change Detection
   5. Text Preprocessing
   6. Data Pre-Processing
   7. Model Input Preparation
   8. Summarization Model
   9. Fusion and Output Generation
4. **Analysis** 
   1. Detailed Problem Statements
   2. Requirement Analysis
      1. Functional Requirements
      2. Non-Functional Requirements
      3. Functional Requirements
      4. Resource Requirements
   3. Feasibility Study
5. **Design**
   1. System Architecture
   2. Design Diagrams
      1. Activity Diagrams
      2. Sequence Diagrams
      3. Data Flow Diagrams
   3. Algorithms Designed
6. **Conclusion**
7. **References**

**LIST OF FIGURES**

1.1 Comparison Chart Existing Tools

4.1 Use Case Diagram

5.1 Activity Diagram: Working Model

5.2 Sequence Diagrams

5.3 Dataflow Diagram Level 0

5.4 Dataflow Diagram Level 1

**Chapter 1**

**INTRODUCTION**

In this chapter, there is a brief introduction about the project giving the need for the project, the problem being solved, objectives, and also the approach to be used.

**1.1 Preamble**

Watching videos to gain information can be time taking as the length of videos can be large. There are times when after watching the entire video we may realize that it was not relevant. To save this amount of time and effort we create a summary from the video by extracting its audio and image and then combining those to obtain summary. By using an efficient tool to generate summary we can read the summary to decide whether to watch the video or not. This will also save the users time and resources that would otherwise be required to watch the entire video.

Video summarization is a process of condensing the information conveyed in a video or sequence of images into a concise and informative text summary. With the explosion of online video content, it has become increasingly important to quickly and efficiently extract the key information from these sources. Video summarization tools use advanced technologies such as computer vision, machine learning, and natural language processing to analyse the visual and auditory content and generate a summary that captures the essential points and highlights. These tools are useful for a range of applications, including news reporting, educational content, and marketing videos, as they allow users to quickly and easily understand the most important aspects of a video or image sequence without having to watch or view the entire content.

**1.2 Need of the Project**

In the recent times peoples are switching to the Internet, rather than reading a book to understanding the key concept of the topic. But watching videos can be time taking especially when the length of video is large. To address this issue, we need a tool that summarises the content of the video so that user can read it and decide whether he/she should watch the video or not.

**1.3 Problem Statement**

To develop and implement a Video Summarization tool which generates summary using both the audio and images extracted from the video and combining the text extracted from both of these using machine learning algorithms to obtain the final text summary.

**1.4 Objectives**

The main objective of this project is to make available following functionalities for the application –

1. Generate a summary for the given input video.
2. Help the user know about the relevance and usefulness of video.

**1.5 Proposed Approach**

The approach aims at dividing the video into video segments using OpenCV and CNN and then extracting audio and images/frames from the segment. From the extracted audio we use Natural Language Processing libraries to convert it to text. The text from the image is extracted using EasyOCR. These texts are then combined according to timestamp using Recurrent Neural Network. The obtained text is summarised using Bi-RNN.

**1.6 Organization of the Report**

The brief summary about the organization of report is given.

* Chapter 1: A brief introduction about the project giving the need for the project, the problem it is going to solve, objectives, and also the approach to be used.
* Chapter 2: It provides a background study of the project which includes Tools and Technologies used and the comparison of various video summarisers.
* Chapter 3: This chapter describes the literature survey done to study the concepts of the existing system.
* Chapter 4: This Chapter describes the detailed problem statement of the project work. It deals with a detailed analysis of the project with the functional requirements and non-functional requirements, system components, use case analysis and the feasibility study.
* Chapter 5: This chapter focuses on design details comprising of the basic architecture of the system and working of various modules of the system. Along with this the algorithm designed has also been discussed in this chapter.
* All the referred resources are added in the References chapter.

**Chapter 2**

**BACKGROUND STUDY**

In this chapter, the background information of the project is presented. The first section contains the brief description of the main technology domain in which our project falls, i.e., Image and Audio Processing. The tools and technologies, different frameworks, APIs required to build this project are specified in the next section. The last section of this chapter covers the existing application that has been developed in document creation and cleaning and used by people to perform academic day-to-day tasks.

1. Audio extraction: It is to extract audio from the given video.
2. Image processing: It is for changing contrast and other quality characteristics.
3. Image analysis: It involves measurements and calculation of various parameters.
4. Edge Detection: It is an image processing technique for finding the boundaries of objects within images.
5. Text generation: It is the process to extract text from audio and images.
6. Combining text: It is the process to combine the text from audio and images.
7. Summary Generation: It is the process of generating the summary from the text.

**2.1 TOOLS AND TECHNOLOGIES**

**2.1.1 Open CV**

OpenCV is a cross-platform library using which users can develop real-time computer vision applications.

It is a huge open-source library for computer vision, machine learning, and image processing. It supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. It is used to Identify image pattern and its various features such as vector space and perform mathematical operations on these features.

Features of OpenCV Library:

* Read and write images
* Capture and save videos
* Process images (filter, transform)
* Perform feature detection
* Detect specific objects such as faces, eyes, cars, in the videos or images.
* Analyse the video, i.e., estimate the motion in it, subtract the background, and track objects in it.

**2.1.2 MoviePy:**

A Python library for video editing that can be used for video file loading, video clipping, and video concatenation.

The key features of the MoviePy Python library-

* Video file I/O: Easy reading and writing of video files in various formats.
* Clip creation and editing: Creation and manipulation of video clips with resizing, cropping, and rotating capabilities.
* Video effects and transitions: Application of built-in effects and transitions to enhance videos.
* Compositing and mixing: Overlaying multiple clips, images, or text and mixing audio tracks.
* Image and audio processing: Image resizing, cropping, and filtering, as well as audio extraction and manipulation.

These features make MoviePy a versatile tool for video editing and manipulation tasks in Python.

**2.1.3 Speech Recognition:**

A speech recognition library that can be used for audio processing, including converting audio to text using speech recognition algorithms.

**2.1.4 NLTK (Natural Language Toolkit):**

A powerful natural language processing library in Python that can be used for text analysis tasks, such as text tokenization, text normalization, and text summarization.

Key features of NLTK are –

* Natural language processing
* Language modelling
* Syntax and parsing

These features make NLTK a powerful library for natural language processing tasks in Python, providing a wide range of tools and resources to analyze and process textual data.

**2.1.5 Tesseract:**

An optical character recognition (OCR) engine developed by Google that can be used for text recognition in image frames extracted from the video.

Key features -

* Language support
* Image preprocessing
* Customization and training
* Open-source and cross-platform

These features make the Tesseract Python library a popular choice for implementing OCR functionality in Python applications, providing accurate text extraction from images and scanned documents in a variety of languages.

**2.1.6 PyTorch or Tensorflow:**

Popular deep learning libraries that can be used for training and deploying machine learning models for video analysis tasks, such as video summarization, scene detection, or keyframe extraction.

**2.1.7 Numpy:**

A powerful numerical computing library in Python that can be used for various mathematical operations, such as array manipulation, matrix operations, and statistical analysis, which may be useful for feature extraction or data processing in video summarization.

**2.1.8 Pandas:**

A data manipulation library in Python that can be used for data analysis and manipulation, such as reading and writing data in different formats (e.g., CSV, Excel), data cleaning, and data aggregation, which may be useful for handling metadata or annotations related to video frames.

**2.1.9 Matplotlib or Seaborn:**

Popular data visualization libraries in Python that can be used for visualizing data, such as plotting graphs, charts, or heatmaps, which may be useful for visualizing video summary results or other analysis outcomes.

**2.2 EXISTING SYSTEMS**

A survey has been conducted on various summary generation applications to identify the key functionalities they are offering-

1. **Upword**

Upword is a powerful tool for AI note-taking and summarization that helps you on all websites, not just YouTube. Though it summarizes the videos, these summaries are not as short compared to what other tools provide. For instance, it provided a 485-word summary of an 11-minute YouTube video, cutting it by 77%. Although this ensures that important points don’t get lost, you’ll have to spend greater time reading summaries. Besides the summary, it also displays a time-stamped transcript on the right side. You can use these timestamps to jump to any part of the video. This is a paid tool and costs approx. $5 per month.

TECHNOLOGY USED – Artificial Intelligence

1. **YoutubeDigest**

YoutubeDigest Generates summaries for youtube videos using ChatGPT. YouTubeDigest can create summaries for YouTube videos in different forms. By default, it provides a summary in five bullet points. However, you can change settings to get summaries as titled sections, single paragraphs, or articles.

There is an option to limit the number of words in summary. This is available as Chrome /Firefox /Brave extension.

TECHNOLOGY USED - Generative Pre – trained transformers

LIMITATION- It can only summarize those videos that has a transcript provided by Youtube.

1. **Wordtune**

Wordtune gives us the full rundown on any video you need, including summaries and important timestamps. This is a paid tool and costs approx. $10 per month. This is available as Chrome /Firefox /Brave extension.

TECHNOLOGY USED - Language Modelling

LIMITATION - It can only summarize the videos with subtitles

1. **Summarize.tech: AI-powered video summaries**

This tools gives a summary of any long YouTube video, like a lecture, live event or a government meeting. It is powered by ChatGPT.

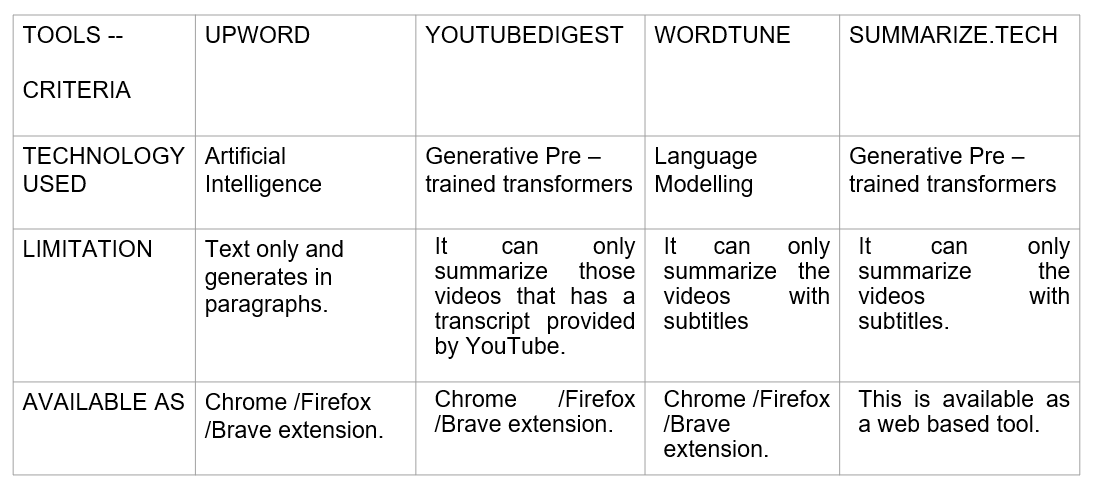
This is available as a web tool. It generated a 320 words summary for a 11 minute video.

TECHNOLOGY USED - Generative Pre – trained transformers

LIMITATION - It can only summarize the videos with subtitles.

**COMPARISION OF VARIOUS TOOLS-**

**Table 1**



**Chapter 3**

**LITERATURE REVIEW**

1. **Audio Extraction:**

The audio is extracted from the video using python libraries such as MoviePY. The extracted audio is then further used to generate text from it.

1. **Frame Extraction:**

Extract frames from the video at a specified frame rate. This can be done using video processing libraries or APIs, such as OpenCV in Python or FFmpeg. The extracted frames will be used as input for further processing.

1. **Visual Feature Extraction:**

Extract relevant visual features from the frames. This can include extracting text using OCR techniques, detecting objects or regions of interest (ROI) using object detection algorithms like YOLO or Faster R-CNN, or extracting visual descriptors like color histograms or deep features using pre-trained CNNs (Convolutional Neural Networks). These visual features will be used as input for the summarization model.

1. **Frame Change Detection:**

Implement frame change detection techniques to identify keyframes or frames with significant visual changes. This can include comparing histograms, computing frame differences, or using deep learning-based approaches. Frame change detection can help identify scenes or content shifts in the video.

**Key frame detection:**

There are several techniques that can be used to detect visual changes or keyframes in a video for frame extraction. Here are some common methods:

* **Image Differencing:** Image differencing involves subtracting one frame from another to obtain the differences or changes between them. Frames with higher differences, which indicate significant visual changes, can be selected as keyframes. This method is simple and can be effective in detecting keyframes in videos with relatively simple and static scenes. [5]
* **Motion Detection:** Motion detection involves analyzing the motion or movement in consecutive frames of a video to identify frames with significant motion. Techniques such as optical flow, background subtraction, or frame differencing can be used to detect motion in a video. Frames with higher motion can be selected as keyframes, as they indicate significant visual changes or scenes with motion [6]
* **Scene Change Detection:** Scene change detection algorithms automatically detect changes in the video content, such as transitions between scenes, cuts, or fades. These algorithms analyze features such as color histograms, edge information, or motion vectors to identify frames where significant visual changes occur. Frames detected as scene changes can be selected as keyframes for summarization. [7]
* **Visual Feature-Based Methods:** Visual feature-based methods involve extracting visual features, such as text, objects, or visual patterns, from frames and using them to detect keyframes. Techniques such as Optical Character Recognition (OCR) can be used to extract text from frames, object detection algorithms can be used to detect objects or regions of interest in frames, and pre-trained Convolutional Neural Networks (CNNs) can be used to extract high-level visual features from frames. Frames with relevant visual features can be selected as keyframes for summarization

1. **Text Processing:**

If text is extracted from the frames, apply natural language processing (NLP) techniques to process the text, such as text summarization, keyword extraction, or sentiment analysis. This can help in generating summaries that include relevant information from the extracted text.

1. **Data Pre-processing:**

Pre-process the extracted visual features, text, and other relevant data to prepare it for input to the summarization model. This may involve normalizing or scaling the data, encoding categorical variables, or handling missing values.

1. **Model Input Preparation:**

Combine the processed visual features, text, and other relevant data to create the input data for the summarization model. This can involve concatenating or combining different features, and organizing the data in a suitable format for the model's input requirements.

1. **Summarization Model:**

Train or use a pre-trained machine learning or deep learning model for generating video summaries. The model can take the processed frames, extracted visual features, text, and other relevant data as input and produce a summary in text format as output.

1. **Fusion and Output Generation:**

Combine the audio-based summary (generated separately from the audio processing step) with the video-based summary generated from the summarization model. This can be done using various fusion mechanisms, such as concatenation, weighted fusion, or attention mechanisms, to create a cohesive summary output in text format.

**Chapter 4**

**ANALYSIS**

In this chapter, a detailed analysis of the project has been discussed. It includes requirement analysis. i.e., functional requirements and non-functional requirements of the proposed system, various hardware and software requirements.

**4.1 DETAILED PROBLEM STATEMENT**

To develop an application which takes video / link of a video as input and the extracts audio and images from it. It then uses these images to create a summary using their content and creates a summary form the audio by first translating it to text using NLP and then summarizing that text. The summary generated from audio and video will be combined to create the required summary.

**4.2 REQUIREMENT ANALYSIS**

In this section the requirements of the system are specified both functional and use-case analysis, followed by non-functional requirement and resource requirements.

**4.2.1 Functional Requirements**

A functional requirement specifies the functionality of a system or one of its subsystems. It also depends upon the type of software, expected users and the type of system where the software is used. Functional user requirements may be high level statements of what the system should do but functional system requirement should also describe clearly about the system services in detail.

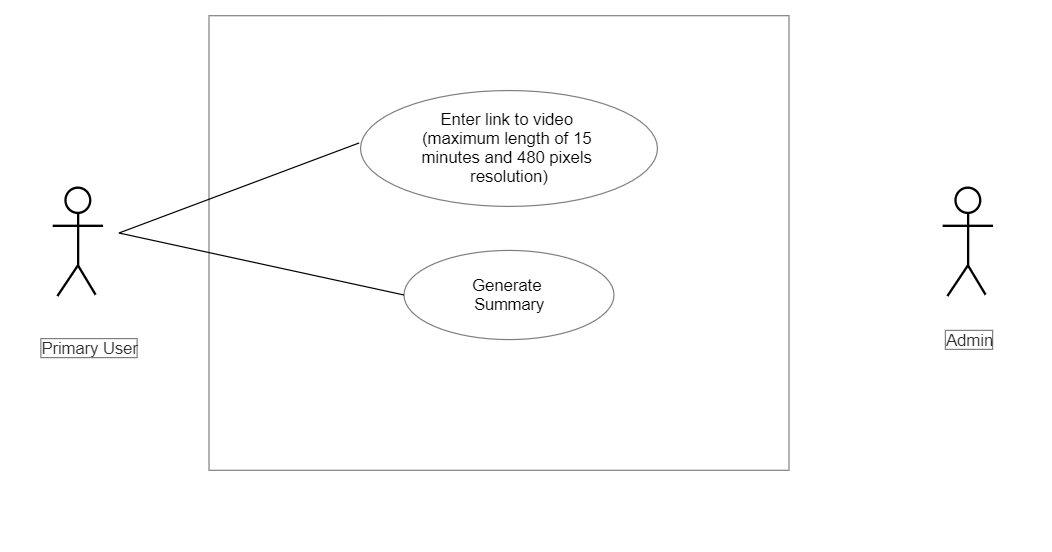
Functional requirements as follows:

1. Enter link to video (maximum length of 15 minutes and 480 pixels resolution)
2. Generate Summary
   * 1. **Use Case Analysis**

**Table 2**

|  |  |
| --- | --- |
| **Use case 1**  Actors Involved  Description  Pre-condition  Post condition  Main Flow | Enter link to video  User  Link to video must be provided and video must be of maximum length of 15 minutes and 480 pixels resolution)  None  None  The link is taken and video is processed to generate the summary |
| **Use Case 2**  Actors Involved  Description  Pre-condition  Post condition  Main Flow | Generate Summary  User  The summary is generated by our ml model / system and displayed to the user.  Video link must be provided  None  The video is processed using machine learning models such as RNN, NLP etc. |

**Use Case Diagram**

****

**4.2.3 Non-Functional Requirements**

Requirements, which are not related to functional aspect of software, fall into this category. They are implicit or expected characteristics of a software.

The major non-functional requirements of the system are:

• Availability: The application is available 24\*7.

• Reliability: The summary generated must be useful and correct

• Usability: The application is user-friendly. Anyone can use the application easily.

• Accessibility: The application is accessible to all.

• Performance: The system response time should be adequate.

**4.2.4 Resource Requirements**

System requirements are the configuration that a system must have in order for a hardware or software application to run smoothly and efficiently. Failure to meet these requirements can result in performance problems.

**Software Requirement**

* Programming Language: Python is a popular choice due to its extensive libraries and frameworks for machine learning, computer vision, and audio processing.
* Development Environment: IDEs like PyCharm, Jupyter Notebook, or Visual Studio Code can facilitate code development, debugging, and project organization.
* Machine Learning Frameworks: Libraries such as TensorFlow, PyTorch, or Keras provide a wide range of tools and pre-trained models for developing and training machine learning models.
* Computer Vision Libraries: OpenCV is a widely-used library for image and video processing tasks, including frame extraction, image differencing, object detection, and scene segmentation.
* Audio Processing Libraries: Libraries like MoviePy or PyAudio offer functions and utilities for audio analysis, feature extraction, and signal processing tasks.
* Natural Language Processing (NLP) Libraries: NLP libraries like spaCy or NLTK can be utilized for text processing tasks, such as extracting text from frames, language modeling, or sentiment analysis if applicable.
* Version Control: Tools like Git and hosting platforms like GitHub or GitLab can facilitate collaborative development, version control, and project management.
* Data Storage: Depending on the size and complexity of the dataset, you may need a database management system like MySQL, PostgreSQL, or MongoDB to store and retrieve video metadata, extracted features, or summary results.

**Hardware Requirement**

* Processor (CPU): A powerful multi-core processor is recommended to handle the computational load efficiently. Processors with higher clock speeds and multiple cores can significantly speed up the processing of frames, audio, and feature extraction.
* Graphics Processing Unit (GPU): GPUs can accelerate certain computations, particularly those related to deep learning models and parallel processing. Having a GPU with CUDA support can significantly speed up the computations.
* Random Access Memory (RAM): Sufficient RAM is important to store and manipulate frames, audio segments, and feature vectors. The amount of RAM required depends on the size of the video dataset and the complexity of the processing tasks. At least 8GB or more of RAM is required for handling medium to large video collections.
* Storage: Adequate storage is needed to store video files, extracted frames, audio segments, and any intermediate or final results. The storage capacity will depend on the size of the video dataset and the desired retention period. Consider using high-capacity hard drives or solid-state drives (SSDs) for faster read/write speeds.
* Network: If your project involves accessing video data from remote sources or streaming videos, a stable and reliable network connection is required.
* GPU Memory (if using GPU): Deep learning models, especially those based on Convolutional Neural Networks (CNNs), may require significant GPU memory for training and inference. Ensure that the GPU memory is sufficient to accommodate the models and the size of the video frames being processed.
* Cooling System: Video processing tasks, especially when involving deep learning models or computationally intensive algorithms, can generate a significant amount of heat. Adequate cooling mechanisms, such as fans or liquid cooling, are essential to prevent overheating and ensure the stability and longevity of the hardware.

**4.3 FEASIBILITY STUDY**

The main objectives of feasibility include the assessment of recruitment capability and resulting sample characteristics, data collection procedures and outcome measures, acceptability of the intervention and study procedures, resources and ability to manage and implement the study and intervention, and preliminary evaluation of participant responses to the intervention. This includes the analysis that takes all of the software’s relevant factors into account—including economic, technical, and scheduling considerations—to ascertain the likelihood of completing the project successfully. Feasibility study is essential for the business development of the software.

1. **Technical Feasibility**

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system.

1. **Economic Feasibility**

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

1. **Legal Feasibility**

This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws, data protection acts or social media laws. Let’s say an organization wants to construct a new office building in a specific location. A feasibility study might reveal the organization’s ideal location isn’t zoned for that type of business. That organization has just saved considerable time and effort by learning that their project was not feasible right from the beginning.

1. **Operational Feasibility**

This assessment involves undertaking a study to analyse and determine whether and how well the organization’s needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

1. **Scheduling Feasibility**

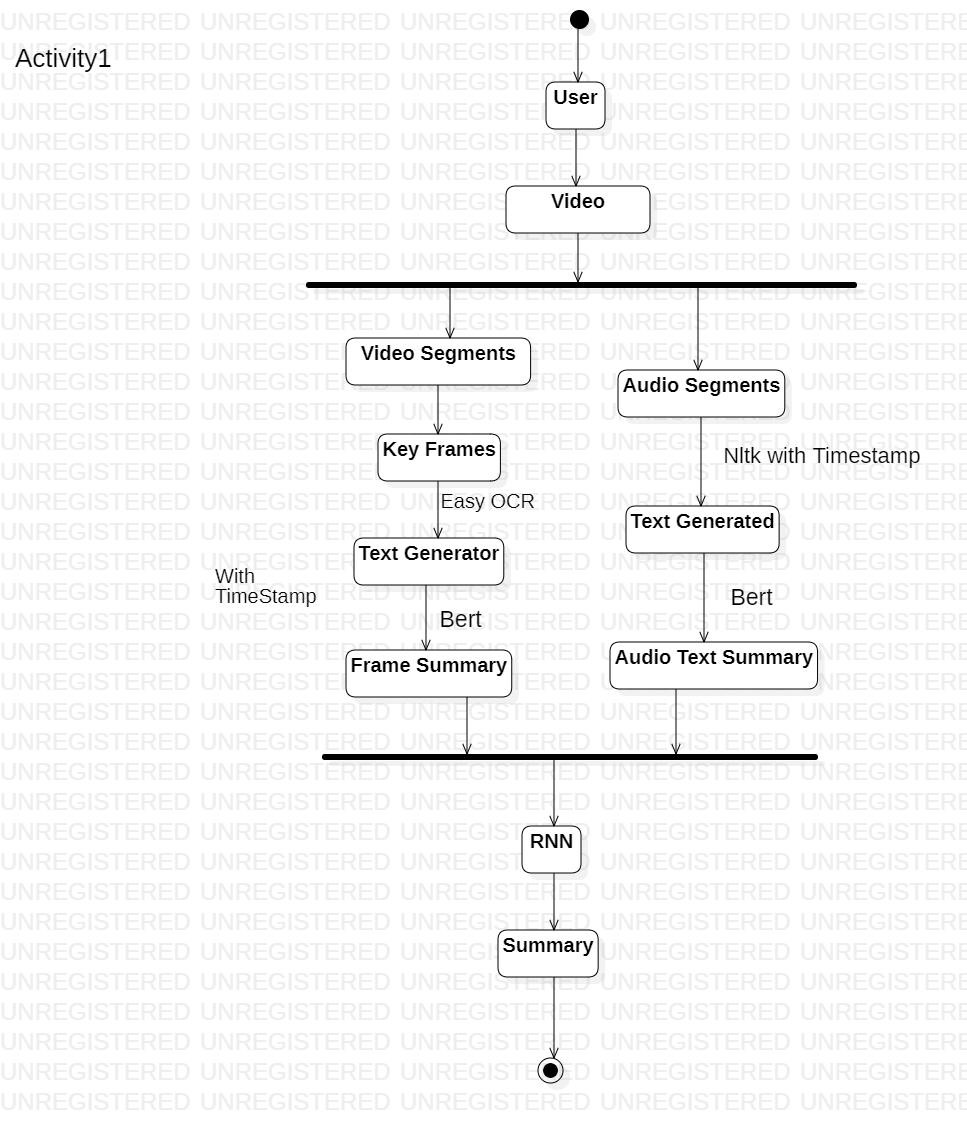
This assessment is the most important for project success; after all, a project will fail if not completed on time. In scheduling feasibility, team estimates how much time the project will take to complete. Being a mobile application, no launch cost is associated with it. All the frameworks and API used are freeware. No cost will be charged from potential customers. From this it is clear that the project is financially feasible. This project is a web application which will be developed using HTML, CSS, Python, OpenCV and other libraries and APIs available freely. Thus, the project is technically feasible. The current mode of operation provide response on time accurately and meets all the requirements which is specified; hence it is operationally feasible. The proposed project meets the legal threshold for completion, which includes permission of data from the user which is used for building requirements, confirms to legal and is ethical, hence it is legally feasible. The project is completed as per the decided schedule for project completion within the time, hence it is scheduling feasible.

**Chapter 5**

**DESIGN**

**5.1 DESIGN DIAGRAMS**

**5.1.1 Activity Diagrams (Figure 1)**

****

****

**5.1.2 Sequence Diagrams (Figure 2)**

****

* + 1. **Data Flow Diagrams (Figure 3)**

**Data flow diagram level 1**

****

**Data flow diagram level 1**

****

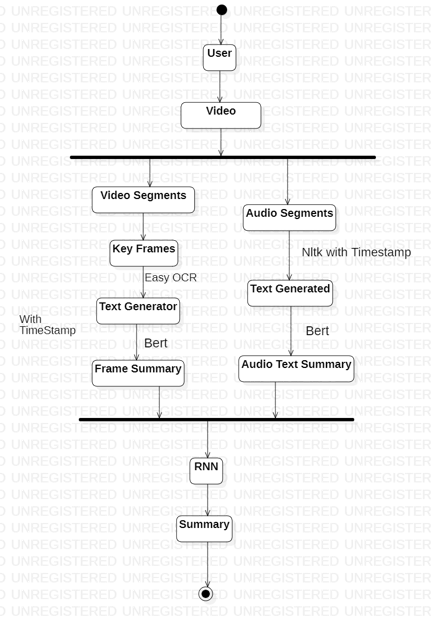
**5.2 ALGORITHMS DESIGNED**

Video summarization is the process of condensing a video into a concise and meaningful summary. The goal is to capture the most important frames and content of the video.

**Steps in Methodology:**

1. Video Segmentation
2. Audio Extraction
3. Frame Extraction
4. Feature Extraction
5. Importance Scoring
6. Keyframe Selection
7. Text Extraction
8. Combining the text
9. Summary Generation

**ALGORITHM (Figure 6 )**



**DETAILED STEPS -**

**STEP 1. Video Segmentation**

• Divide the video into fixed lengths segments.

• This can be done using python libraries such as OpenCV

**STEP 2. Audio Extraction**

• The next step is to extract audio from the video segments using MoviePy library.

• The audio extracted can then be used to convert to text using NLP (Natural Language Processing).

**STEP 3. Frame Extraction**

• Frames are extracted from the video using the OpenCV library.

• Instead of capturing frames at a fixed rate, changes in frames are detected using a threshold approach.

• Frames with significant changes exceeding the threshold are selected for further processing and analysis.

**STEP 4. Feature Extraction**

• Visual features are extracted from the selected frames.

• Object detection techniques, such as YOLO or Faster R-CNN, are employed to identify and localize objects within the frames.

• Image classification models, like ResNet or VGG, are used to classify the content of the frames into specific categories.

• Text recognition algorithms, such as Tesseract, are applied to extract text from frames containing textual information.

**STEP 5. Importance Scoring**

• Importance scores are assigned to frames based on the extracted visual features.

• Machine learning algorithms, such as SVM or random forests, can be trained to predict the importance of frames.

• Heuristics based on the presence of key objects, text, or significant visual changes can also be used to calculate the importance scores.

**STEP 6. Keyframe Selection**

• Keyframes are selected from the pool of frames based on their importance scores.

• Frames with higher importance scores are considered as keyframes.

• Keyframes represent the most important visual content or significant changes in the video.

**STEP 7. Text Extraction**

• Optical character recognition (OCR) techniques are applied to extract text from frames.

• OCR libraries like Tesseract or Google Cloud Vision API are used to recognize and extract text information.

• Extracted text can be utilized for generating text-based summaries or captions for the video.

**STEP 8. Combine the text**

• The next step is to combine the text generated from the images and audio.

• To combine this text we use RNN (Recurrent neural network)

**STEP 9. Summary generation**

• A machine learning model, such as an extractive or abstractive summarization model, is employed to generate the video summary.

• The combined text is used as input to the summarization model.

• The model considers visual content, importance scores, and textual information to generate a concise and meaningful summary of the video.

**CONCLUSION**

• Video summarization is a challenging task that involves frame extraction, feature extraction, importance scoring, keyframe selection, text extraction, and summarization model.

• The proposed methodology provides a systematic approach for generating video summaries.

• Further improvements can be made by exploring advanced techniques, incorporating audio analysis, or integrating user preferences into the summarization process.

**REFERENCES**

1. <https://www.youtubedigest.app/>
2. <https://app.wordtune.com/read>
3. <https://www.summarize.tech/>
4. <https://app.upword.ai/>
5. <https://pyimagesearch.com/2017/06/19/image-difference-with-opencv-and-python/>
6. <https://pyimagesearch.com/2015/05/25/basic-motion-detection-and-tracking-with-python-and-opencv/>
7. <https://towardsdatascience.com/change-detection-using-siamese-networks-fc2935fff82>