*Project Report*

# **On**

**Video Summarization**

**Submitted for partial fulfilment of requirement for the degree of**

# BACHELOR OF ENGINEERING

**(Computer Science and Engineering)**

## Submitted By Mr. Karan Kurhade, Mr. Aniket Eche, Mr. Anmol Agrawal, Mr. Chirag Kashikar.

**Under the Guidance of**

## Prof. Dr. S.R. Gupta



**Department of Computer Science & Engineering,**

**Prof. Ram Meghe Institute of Technology & Research, Badnera**

**2020-2021**



**Department of Computer Science & Engineering**

**Prof. Ram Meghe Institute of Technology &**

**Research, Badnera**

**2020 -**

**2021**

CERTIFICATE

*This is to certify that the Project (8KS07) entitled*

**Video Summarization**

*is a bonafide work and it is submitted to the*

*S*

*a*

*n*

*t*

*G*

*a*

*d*

*g*

*e*

*B*

*a*

*b*

*a*

*A*

*m*

*r*

*a*

*v*

*a*

*t*

*i*

*U*

*n*

*i*

*v*

*e*

*r*

*s*

*i*

*t*

*y*

*,*

*A*

*m*

*r*

*a*

*v*

*a*

*t*

*i*

*By*

**Mr. Karan Kurhade**

**Mr. Aniket Eche**

**Mr. Anmol Agrawal**

**Mr. Chirag Kashikar**

*in the partial fulfillment of the requirement for the degree of*

*B*

*a*

*c*

*h*

*e*

*l*

*o*

*r*

*o*

*f*

*E*

*n*

*g*

*i*

*n*

*e*

*e*

*r*

*i*

*n*

*g*

*in*

*C*

*o*

*m*

*p*

*u*

*t*

*e*

*r*

*S*

*c*

*i*

*e*

*n*

*c*

*e*

*&*

*E*

*n*

*g*

*i*

*n*

*e*

*e*

*r*

*i*

*n*

*g*

*,*

*during the academic year*

*20*

*20*

*-*

*2021*

*under my guidance*

.

Dr. S. R. Gupta

*D*

*r*

*.*

*G*

*.*

*R*

*.*

*B*

*a*

*m*

*n*

*o*

*t*

*e*

*Guide*

*Department of Computer Sci.& Engg.*

*Prof. Ram Meghe Institute Of Technology &*

*Research, Badnera*

*Head,*

*Department of Computer Sci.& Engg.*

*Prof. Ram Meghe Institute Of Technology &*

*Research, Badnera*

*External Examiner*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

**ACKNOWLEDGEMENT**

With great pleasure we hereby acknowledge the help given to us by various individuals throughout the project. This Project itself is an acknowledgement to the inspiration, drive and technical assistance contributed by many individuals. This project would have never seen the light of this day without the help and guidance we have received

We would like to express our profound thanks to Dr. S.R. Gupta for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project. We would also thank the faculties of the Department of Computer Science & Engineering, for their kind co-operation and encouragement which help us in completion of this project. We owe an incalculable debt to all staffs of the Department of Computer Science & Engineering for their direct and indirect help.

Our thanks and appreciations also go to our colleague in developing the project and people who have willingly helped us out with their abilities.

We extend our heartfelt thanks to our parents, friends and well-wishers for their support and timely help. Last but not the least; we thank the God Almighty for guiding us in every step of the way.

Karan Kurhade \_\_\_\_\_\_\_\_\_\_\_\_\_

Aniket Eche \_\_\_\_\_\_\_\_\_\_\_\_

Anmol Agrawal \_\_\_\_\_\_\_\_\_\_\_\_\_

Chirag Kashikar \_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| **Table of Contents** | | | |
| 1 |  |  | **Introductions** |
|  | 1.1 |  | Overview |
|  | 1.2 |  | Motivation |
|  | 1.3 |  | Definitions, Acronyms, and Abbreviations |
|  | 1.4 |  | Project Objective |
| 2 |  |  | **Literature Review** |
|  | 2.1 |  | Background |
|  | 2.2 |  | What is video? |
|  | 2.3 |  | Video Summarization |
|  | 2.4 |  | Methods of Video Summarization |
|  | 2.5 |  | Recent Work |
|  | 2.6 |  | Summary and Discussion |
| 3 |  |  | **System Analysis** |
|  | 3.1 |  | System Architecture |
|  | 3.2 |  | System Modules |
|  | 3.3 |  | User Requirements |
|  | 3.4 |  | Functional Requirements |
|  |  | 3.4.1 | General Requirements |
|  |  | 3.4.2 | Video to Summary Requirements |
|  |  | 3.4.3 | Presentation Requirements |
|  | 3.5 |  | Non-functional Requirements |
|  | 3.6 |  | Constraints and Assumptions |
|  | 3.7 |  | Data Flow Diagram |
| 4 |  |  | **Proposed System Design** |
|  | 4.1 |  | UML Diagrams |
|  |  | 4.1.1 | Use Case Diagram |
|  |  | 4.1.2 | Class Diagram |
|  |  | 4.1.3 | Component Diagram |
|  |  | 4.1.4 | Sequence Diagram |
| 5 |  |  | **System Implementation and Testing** |
|  | 5.1 |  | Test Objective |
|  | 5.2 |  | Detailed Testing Strategy |
|  |  | 5.2.1 | Unit Testing |
|  |  | 5.2.2 | White Box Testing |
|  |  | 5.2.3 | Black Box Testing |
|  |  | 5.2.4 | Pass/Fail Criteria |
|  | 5.3 |  | Test Cases |
|  | 5.4 |  | Test Tools |
| 6 |  |  | **Planning** |
| 7 |  |  | **Conclusion** |
| 8 |  |  | **References** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **List of Figures**   |  |  |  | | --- | --- | --- | | **Sr. No.** | **Figure Name** | **Page No.** | | 1 | System Architecture | 9 | | 2 | Data Flow Diagram | 12 | | 5 | Use Case Diagram | 13 | | 6 | Class Diagram | 14 | | 7 | Component Diagram | 17 | | 8 | Sequence Diagram | 18 |     **List of Tables**   |  |  |  | | --- | --- | --- | | **Sr. No.** | **Table Title** | **Page No.** | | 1 | Use Case Diagram Table | 13 | | 2 | Project Planning | 21 | |
|  |
|  |
|  |
|  |
|  |

**ABSTRACT**

A piece of fake news is a news containing misinformation and hoaxes which are

directed in a particular fashion or manner to defame or to misinform and mislead someone. In

today

’

s world with more and more people being dependent on media especially the digital

media. Indeed, this has become a big market for media houses and other beneficiaries who

are involved in politics, government affairs, and antisocial activities. The fake news is created

and disseminated for all sorts of purposes. The publicity and increased viewership are the

prime objective of each and every media outlet for which

they even are putting their

reputation at stake. This fake news market is growing day by day and now has become a

threat to the harmony and the very peace of our society. Many instances have been reported

which showed the use of fake news through social platforms purely for the purpose to incite a

crowed and to encourage them to commit illegal acts including serious violence. It is needless

to emphasize that there is a definite need for a mechanism that can detect fake news and

separate it from the factual one. The mechanism, certainly, needs to be fast and accurate. The

mechanism should be able to differentiate the fake news from a genuine one and should also

bring it to the notice of concerned authorities. The proposed project is based on this very

concept with the help of Machine Learning to study the multiple sets of data and to act as a

filter which checks, separates and report the fake information from factual one as a desired

output.

### 1. INTRODUCTION

## 1.1. Overview

Video summarization is a very important tool where people can use it to get the main idea and the important scenes without watching the full original video.

The system is designed to help the users to just glance at the content of the video quickly, and decide whether to watch the video or not. For sports, looking at the highlights of the match, users can decide whether the match is interesting or not, many sports fans will be able to benefit from a summarized version of sports video which is available from anywhere. Video content can be accessed by using either a top-down approach or a bottom-up approach. The top-down approach, that is, video browsing, is useful when we need to get an essence of the content. The bottom-up approach, that is, video retrieval, is useful when we know exactly what we are looking for in the content. In video summarization, what essence the summary should capture depends on whether the content is scripted or not. Because scripted content, such as news, drama, and movie, is carefully structured as a sequence of semantic units, one can get its essence by enabling a traversal through representative items from these semantic units. Hence, table of contents (ToC)-based video browsing caters to summarization of scripted content. For instance, a news video composed of a sequence of stories can be summarized/browsed using a key-frame representation for each of the shots in a story. However, summarization of unscripted content, such as surveillance and sports), requires a “highlights” extraction framework that only captures remarkable events that constitute the summary.

## 1.2. Motivation

Video summarization is a very important tool where people can use it to get the main idea and the important scenes without watching the full original video.

The system is designed to help the users to just glance at the content of the video quickly, and decide whether to watch the video or not. For sports, looking at the highlights of the match, users can decide whether the match is interesting or not, many sports fans will be able to benefit from a summarized version of sports video which is available from anywhere.

## 1.3. Definitions, Acronyms, and Abbreviations

Video Summarization is a process of creating & presenting a meaningful abstract view of the entire video within a short period of time. The Purpose of video summarization is to speed up browsing of a large collection of video data, and achieve efficient access and representation of the video content. By watching the summary, users can make quick decisions on the usefulness of the video.

For movie summarization, the system is extracting each frame and comparing the image intensity using image histogram, when it will detect any change in the intensity, it merges the frames to get the desired summarized video. For highlights of sports, a threshold has been set based on audience volume, then the system will extract the frames in every five seconds and merge the frames which have crossed the threshold to get the desired highlight of any sport.

We have not used machine learning for video summarization because training the sheer volume of video data is time-consuming, which is not efficient for users.

End users include:

1. Sport enthusiasts who would like to see a game within a short amount of time looking for the key moments.
2. Movie enthusiasts who would like to take a glimpse of what the movie contains to decide for themselves whether to watch the complete movie or not.
3. Detecting key scenes helps the editors to ease out their work of highlights or trailer generation.

Video summarization aims to generate a short synopsis that summarizes the video content by selecting its most informative and important parts. The produced summary is usually composed of a set of representative video frames (a.k.a. video key-frames), or video fragments (a.k.a. video key-fragments) that have been stitched in chronological order to form a shorter video. The former type of a video summary is known as video storyboard, and the latter type is known as video skim. One advantage of video skims over static sets of frames is the ability to include audio and motion elements that offer a more natural story narration and potentially enhance the expressiveness and the amount of information conveyed by the video summary. Furthermore, it is often more entertaining and interesting for the viewer to watch a skim rather than a slide show of frames. On the other hand, storyboards are not restricted by timing or synchronization issues and, therefore, they offer more flexibility in terms of data organization for browsing and navigation purposes, . A high-level representation of the typical deep-learning based video summarization pipeline is depicted. The first step of the analysis involves the representation of the visual content of the video with the help of feature vectors. Most commonly, such vectors are extracted at the frame-level, for all frames or for a subset of them selected via a frame sampling strategy, e.g., processing 2 frames per second. In this way, the extracted feature vectors store information at a very detailed level and capture the dynamics of the visual content that are of high significance when selecting the video parts that form the summary. Typically, in most deep-learning based video summarization techniques the visual content of the video frames is represented by deep feature vectors extracted with the help of pre-trained neural networks.

## 1.4. Project Objectives

Summarizing news videos automatically allows us to quickly look out for the important patterns shown in the news. Generating a trailer of a movie. Moreover, the need for surveillance has increased significantly due to increase in the demand of security and highlights of sports video recordings automatically are some of the engrossing applications of video summarization.

It helps in efficient storage, quick browsing, and retrieval of large collections of video data without losing important aspects

# **2. Literature Review**

## 2.1. Background

In July 2015, YouTube revealed that it receives over 400 hours of video content every single minute, which translate s to 65.7 years’ worth of content uploaded every day 1 . Since then, we are experiencing an even stronger engagement of consumers with both online video platforms and devices (e.g., smart-phones, wearables etc.) that carry powerful video recording sensors and allow instant uploading of the captured video on the Web. According to newer estimates2, YouTube now receives 500 hours of video per minute; and YouTube is just one of the many video hosting platforms (e.g., Daily Motion, Vimeo), social networks (e.g., Facebook, Twitter, Instagram), and online repositories of media and news E-organizations that host large volumes of video content. So, how is it possible for someone to efficiently navigate within endless collections of videos, and find the video content that s/he is looking for? The answer to this question comes not only from video retrieval technologies but also from technologies for automatic video summarization. The latter allow generating a concise synopsis that conveys the important parts of the full-length video. Given the plethora of video content on the Web, effective video summarization facilitates viewers’ browsing of and navigation in large video collections, thus increasing viewers’ engagement and content consumption. The application domain of automatic video summarization is wide and includes (but is not limited to) the use of such technologies by media organizations (after integrating such techniques into their content management systems), to allow effective indexing, browsing, retrieval and promotion of their media assets; and video sharing platforms, to improve viewing experience, enhance viewers’ engagement and increase content consumption. In addition, video summarization that is tailored to the requirements of particular content presentation scenarios can be used for e.g., generating trailers or teasers of movie s and episodes of a TV series; presenting the highlights of an event (e.g., a sports game, a music band performance, or a public debate); and creating a video synopsis with the main activities that took place over e.g., the last 24hrs of recordings of a surveillance camera, for time-efficient progress monitoring or security purposes.

## 2.2. What is Video?

Video is an electronic medium for the recording, copying, playback, broadcasting, and display of moving visual media. Video was first developed for mechanical television systems, which were quickly replaced by cathode ray tube (CRT) systems which were later replaced by flat panel displays of several types.

Video systems vary in display resolution, aspect ratio, refresh rate, colour capabilities and other qualities. Analog and digital variants exist and can be carried on a variety of media, including radio broadcast, magnetic tape, optical discs, computer files, and network streaming.

Video technology was first developed for mechanical television systems, which were quickly replaced by cathode ray tube (CRT) television systems, but several new technologies for video display devices have since been invented. Video was originally exclusively a live technology. Charles Ginsburg led an Ampex research team developing one of the first practical video tape recorders (VTR). In 1951 the first VTR captured live images from television cameras by writing the camera's electrical signal onto magnetic videotape.

## 2.3. Video Summarization

Video summarization technologies aim to create a concise and complete synopsis by selecting the most informative parts of the video content. Several approaches have been developed over the last couple of decades and the current state of the art is represented by methods that rely on modern deep neural network architectures. This work focuses on the recent advances in the area and provides a comprehensive survey of the existing deep-learning-based methods for generic video summarization. After presenting the motivation behind the development of technologies for video summarization, we formulate the video summarization task and discuss the main characteristics of a typical deep-learning-based analysis pipeline. Then, we suggest a taxonomy of the existing algorithms and provide a systematic review of the relevant literature that shows the evolution of the deep-learning-based video summarization technologies and leads to suggestions for future developments. We then report on protocols for the objective evaluation of video summarization algorithms and we compare the performance of several deep-learning-based approaches. Based on the outcomes of these comparisons, as well as some documented considerations about the suitability of evaluation protocols, we indicate potential future research directions.

## 2.4. Methods of Video Summarization

Video summarization can be categorized into two forms:

* **Static video summarization (keyframing)**
* **Dynamic video summarization (video skimming)**

Static video summaries are composed of a set of keyframes extracted from the original video, while dynamic video summaries are composed of a set of shots and are produced taking into account the similarity or domain-specific relationships among all video shots.

One advantage of a video skim over a keyframe set is the ability to include audio and motion elements that potentially enhance both the expressiveness and the amount of information conveyed by the summary. In addition, it’s often more entertaining and interesting to watch a skim than a slide show of keyframes.

On the other hand, keyframe sets are not restricted by any timing or synchronization issues, and therefore, they offer much more flexibility in terms of organization for browsing and navigation purposes, in comparison to a strict sequential display of video skims.

## 2.5. Recent Work

The volume of video data generated has seen an exponential growth over the years and video summarization has emerged as a process that can facilitate efficient storage, quick browsing, indexing, fast retrieval and quick sharing of the content. In view of the vast literature available on different aspects of video summarization approaches and techniques, a need has arisen to summarize and organize various recent research findings, future research focus and trends, challenges, performance measures and evaluation and datasets for testing and validations. This paper investigates into the existing video summarization frameworks and presents a comprehensive view of the existing approaches and techniques. It highlights the recent advances in the techniques and discusses the paradigm shift that has occurred over the last two decades in the area, leading to considerable improvement. Attempts are made to consolidate the most significant findings right from the basic summarization structure to the classification of summarization techniques and noteworthy contributions in the area. Additionally, the existing datasets categorized domain-wise for the purpose of video summarization and evaluation are enumerated. The present study would be helpful in: assimilating important research findings and data for ready reference, identifying groundwork and exploring potential directions for further research.

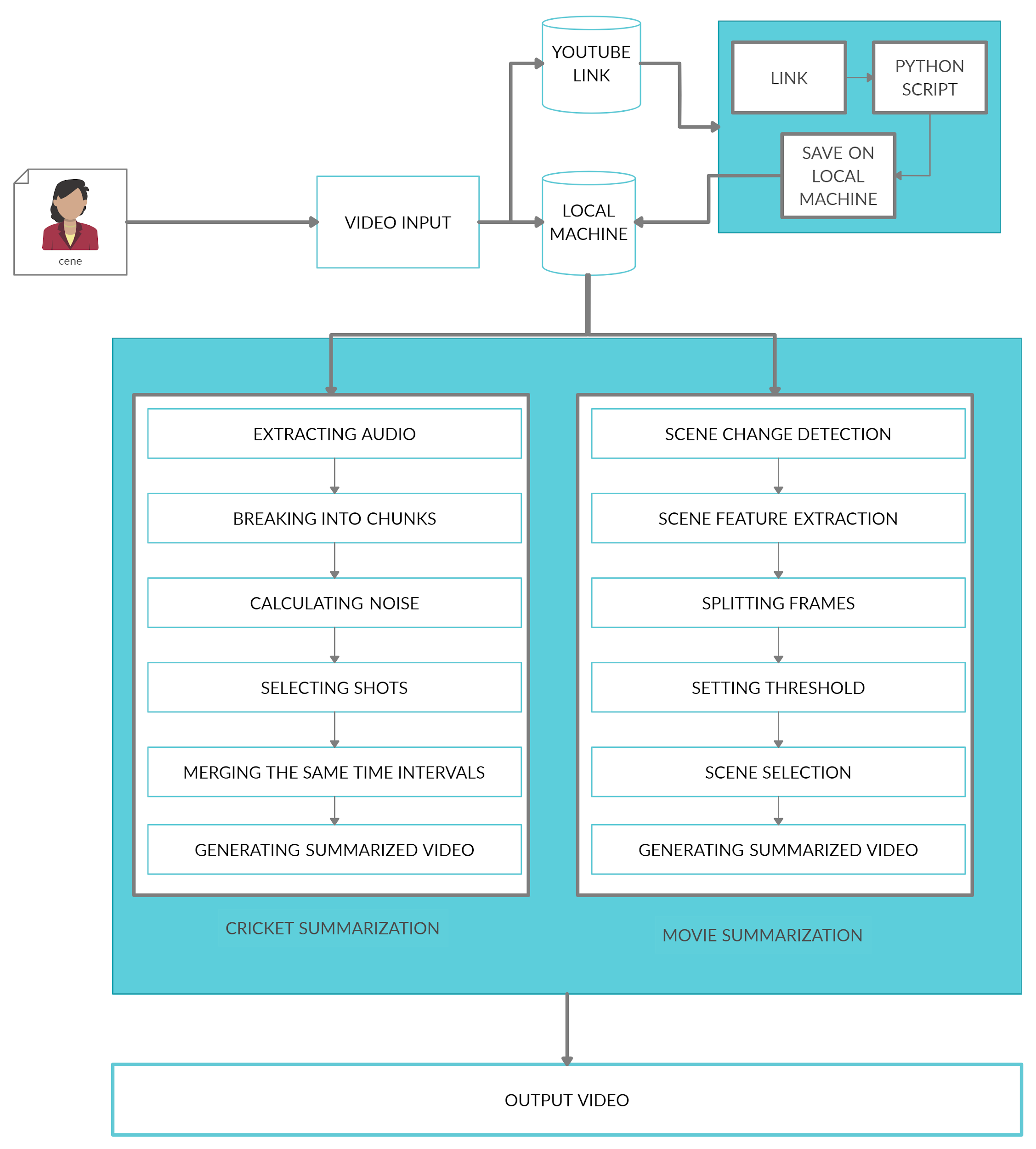
## 2.6. Summary & Discussion

## These techniques are just the start of a new era in deep learning technology when it comes to video summarization. Many advances will be made in the near future to create and optimize the best summaries based on the audience, delivery medium, and intent of summarization. Together, with efforts across the industry, we’ll make video summarization highly-scalable, reliable, and incredibly efficient.

***3. System Analysis***

## 

## 3.1. System Architecture



Our system follows the two-tier architecture. First tier consists of GUI and the second tier consists of linguistic components.

1. GUI: This component in our project deals with the interface for the user where the user can choose the video input from the local storage or can provide a youtube link.
2. Linguistic components: The linguistic component is the block where the actual processing of our project is done. This module is represented for creating summarized video. It has two parts, cricket-based summarization and movie-based summarization. In the cricket part it generates highlights of that match whereas in the movie part it generates trailer of that movie.

## 

## 3.2. System Modules

There are 4 modules in this project. They are -

1. User Interface: This module contains a user interface in which the user select the input type of the video and type of summarization (movie/cricket).
2. Input Type: This module contains how the input file will be uploaded. It can be used from a local machine or from a youtube video link which the video can be downloaded by running a python script and saving on the local machine.
3. Summarization Process:
   1. Movie Summarization: This module will generate trailer of that movie video by using feature extraction and key frame algorithm.
   2. Cricket Summarization: This module will generate highlights of that video by using the short time energy.
4. Formation of Output: This module will save the output video on the local machine.

## 3.3. User Requirement

1. Presentation: The user should be able to select the category for which they want the summary and should be able to copy the URL from YouTube to watch the summary of the video of their choice.
2. Clarity: The user should be able to understand the interface and perform the tasks above easily and completely.
3. Accessibility: User can access the system anytime.

## 

## 3.4. Functional Requirements

### 3.4.1. General Requirements

1. The system will be able to extract key frames from the video and present the most informative or interesting materials for potential users from the FlagFile and delete it after its use.
2. The system will allow users different actions according to their choice, that is, Movies,Cricket Highlights or YouTube video.
3. The shortening process should be consistent and transparent to users.

### 

### 3.4.2. Video to Summary Requirements

1. The system will receive the choice of video from the user and transform it into a Summary.
2. The system will provide options which include Movies,Cricket, Youtube URL and browse from Computer as input.
3. The system will provide segment selection and fast-forwarding.
4. The output summary is usually composed of a set of keyframes or video clips extracted from the original video with some editing process.

### 3.4.3. Presentation requirements

1. The system will present the output of the video by selecting and presenting the most informative materials without losing any important aspects.

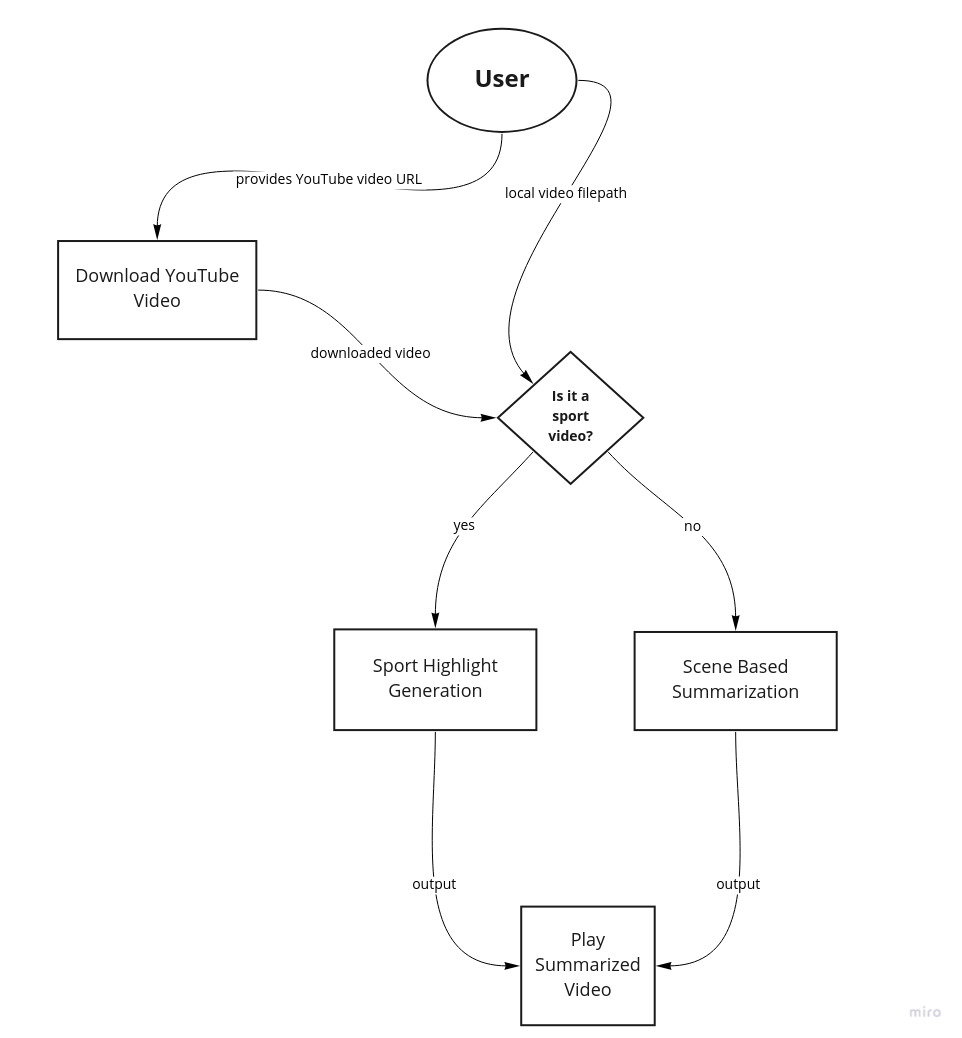
## 3.5. Non-Functional Requirements

1. Speed: Speed depends upon the length of the video file provided by the user, more the duration, more frames to process.
2. Scalability: The system should provide efficient storage, and retrieval of video summary.
3. Availability: The system should be available 24 hours.

## 3.6. Constraints and Assumptions

1. Constraints
   1. The speed of the system is dependent on the number of cores of the hardware the system is running on.
2. Assumptions
   1. Users must have Python 3.8 installed on their computer in order for the system to run.

## 3.7. Data Flow Diagram



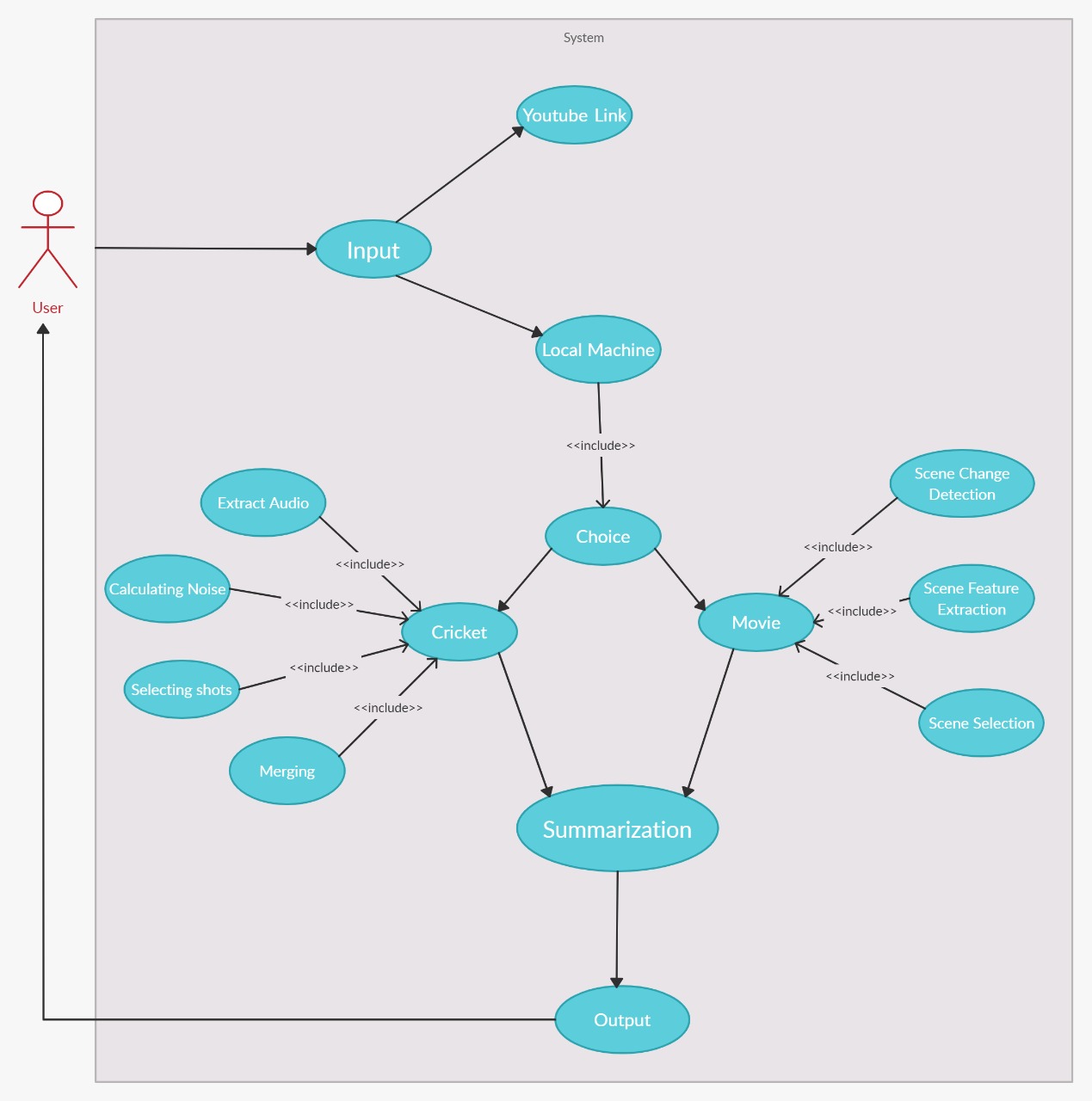
**4. *Proposed System Design***

## 

## 4.1. UML Diagram

### 

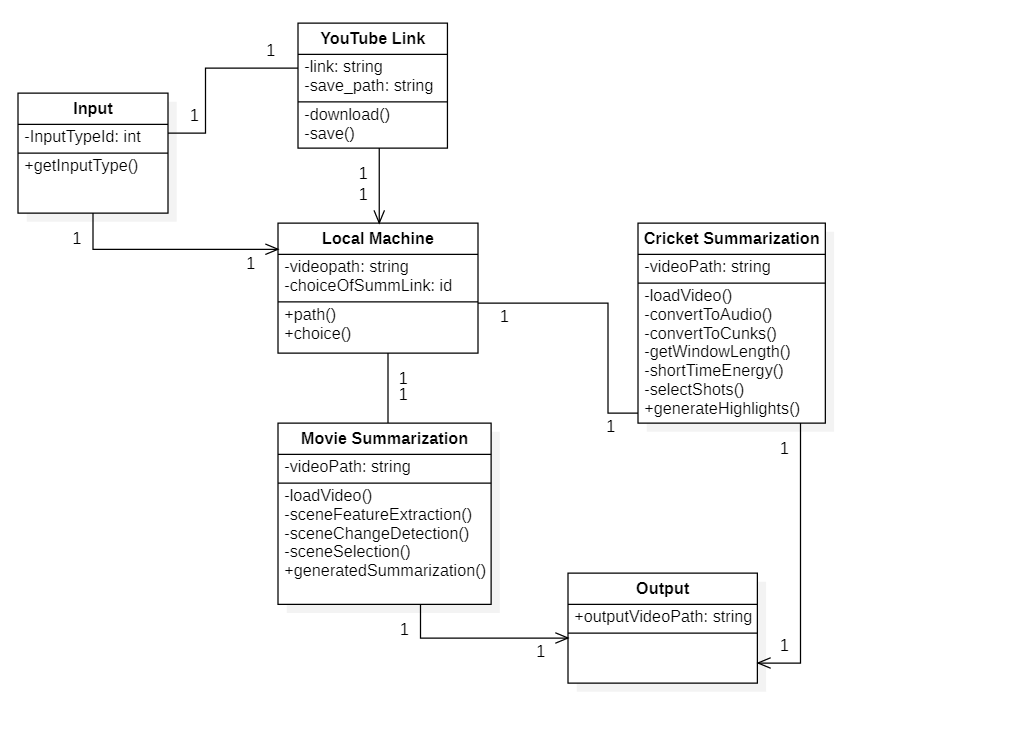
### 4.1.1. Use Case Diagram



|  |  |
| --- | --- |
| Title | Video Summarization |
| Description | Generate Summary of the video given as an input by the user. |
| Primary Actor | User |
| Preconditions | User should be able to browse from Local machine and should be able to copy Youtube URL. |
| Post conditions | User will provide the video for summarization either from the local machine or Youtube link. |
| Main Success Scenario | * User provide their choice of video for summarization. * For Cricket, the script performs Audio Extraction, Noise Calculation, shots selection, and then merging. * For Movie, the script performs Scene Change Detection, Scene Feature Extraction, and Scene Selection. * Then Summarization of the video takes place and displayed to the user. |
| Frequency of Use | User can retrieve the Summary of video any number of times. |

### 

### 4.1.2. Class Diagram



#### CRC 1

|  |  |
| --- | --- |
| **Class Name** | Input |
| **Class Type** | USER |
| **Characteristics** | Provide Input management |
| **Superclass** | None |
| **Subclass** | * Youtube Link * Local Machine |
| **Variables** | * InputTypeID |
| **Services** | Provide what should be input type |
| **Responsibilities** | **Collaborators** |
| ● getInputType() | System |

#### CRC 2

|  |  |
| --- | --- |
| **Class Name** | Local Machine |
| **Class Type** | System |
| **Characteristics** | Load the file. |
| **Superclass** | Input |
| **Subclass** | * Cricket Summarization * Movie Summarization |
| **Variables** | * VideoPath |
| **Services** | Choice of the Summarization User wants to do. |
| **Responsibilities** | **Collaborators** |
| * path() * choice() | User |

#### CRC 3

|  |  |
| --- | --- |
| **Class Name** | Cricket Summarization |
| **Class Type** | System |
| **Characteristics** | Load the file. |
| **Superclass** | Local Machine |
| **Subclass** | * Output |
| **Variables** | * VideoPath |
| **Services** | * Extract the Audio. * Calculate the noise. * Break into chunks. * Calculate the short time energy. * Selection of shots. * Generation video summarization. |
| **Responsibilities** | **Collaborators** |
| * loadVideo() * convertToAudio() * convertToChunks() * getWindowLength() * shortTimeEnergy() * selectShots() * generateHighlights() | System |

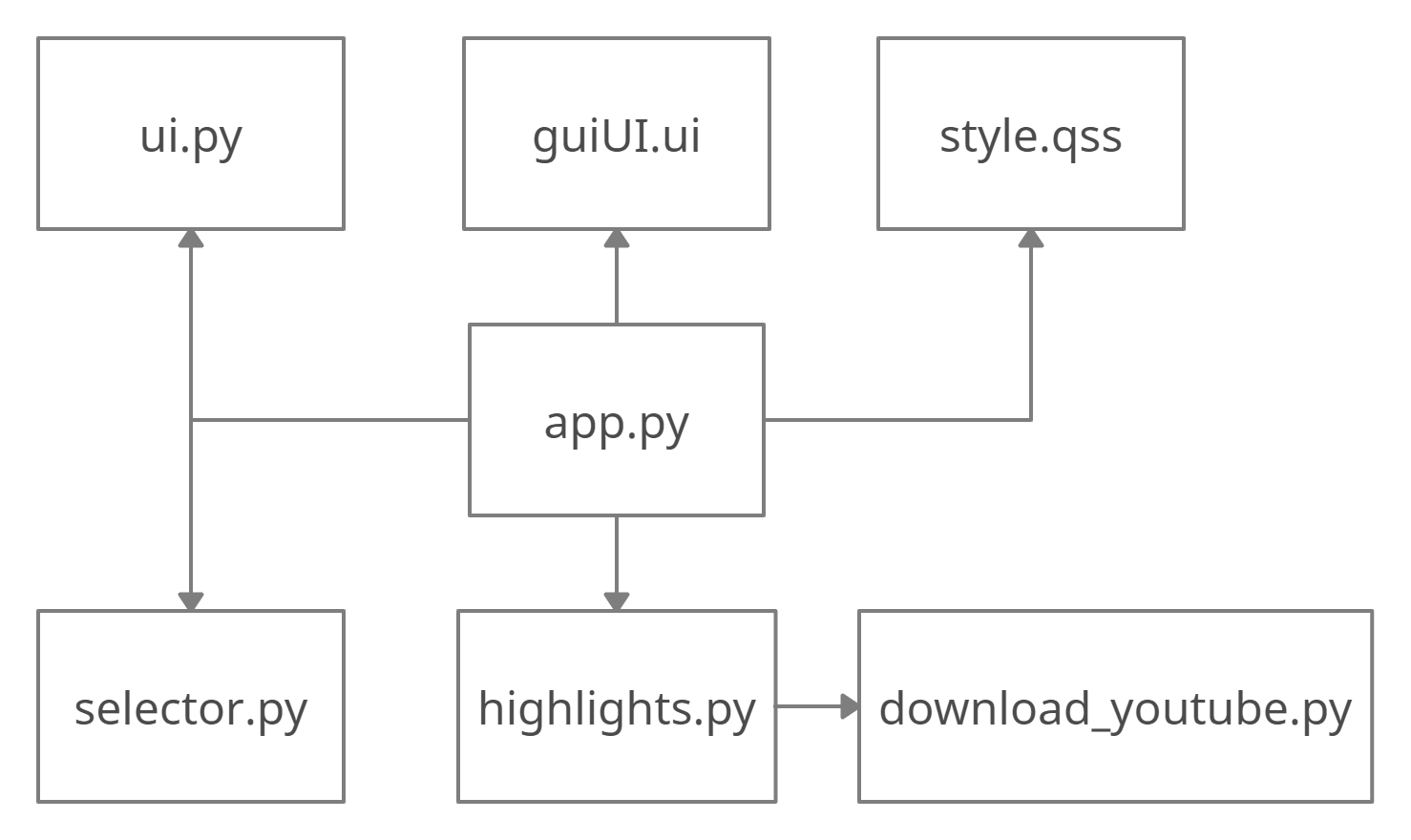
#### CRC 4

|  |  |
| --- | --- |
| **Class Name** | Movie Summarization |
| **Class Type** | System |
| **Characteristics** | Load the file. |
| **Superclass** | Local Machine |
| **Subclass** | * Output |
| **Variables** | * VideoPath |
| **Services** | * Load the video. * Extract the feature from scenes. * Detect the change of scenes. * Select the scenes. * Generated Video Summarization. |
| **Responsibilities** | **Collaborators** |
| * loadVideo() * sceneFeatureExtraction() * sceneChangeDetection() * sceneSelection() * generateSummarization() | System |

## 

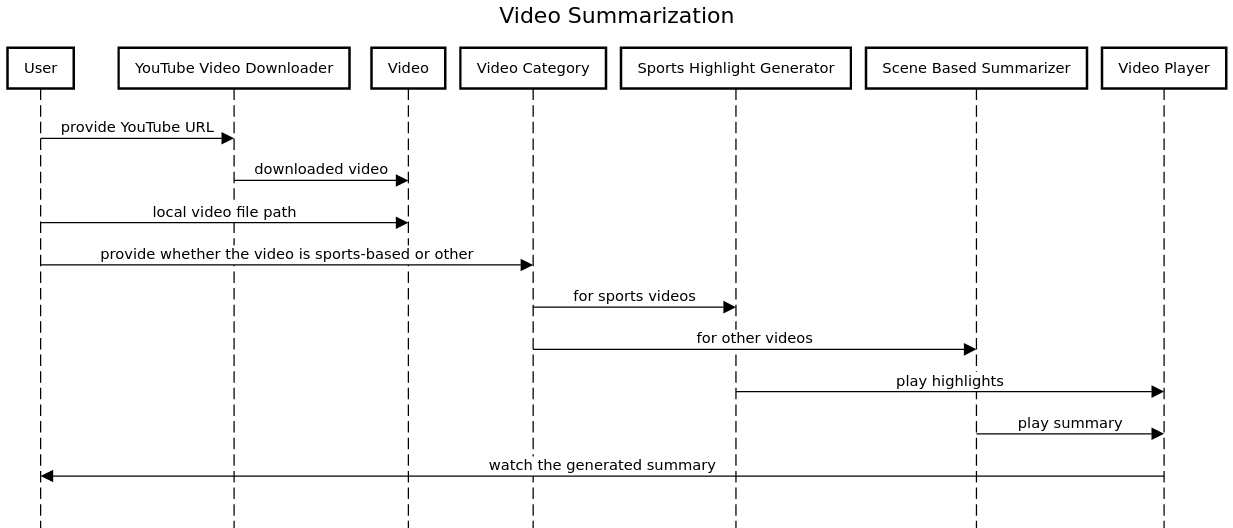
### 

### 4.1.3. Component Diagram



### 

### 4.1.4. Sequence Diagram



**5. Testing**

## 5.1. Test Objective

The objective of this document is to expand on the test plan and provide specific information needed to actually perform the necessary tests. By providing detailed test information, we hope to reduce the probability of overlooking items and improve test coverage. Testers will be able to use each test case provided in this document to move forward and begin testing. Test results will be logged in a database and a complete bug report generated for each test failure.

## 

## 5.2. Detailed Testing Strategy

### 5.2.1. Unit Testing

Unit Testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the programs correctness.

### 5.2.2. White Box Testing

In white box testing, the UI is bypassed. Inputs and outputs are tested directly at the code level and the results are compared against specifications. This form of testing ignores the function of the program under test and will focus only on its code and the structure of that code. The test cases that have been generated shall cause each condition to be executed at least once. To ensure this happens, we are applying Basis Path Testing. Because the functionality of the program is relatively simple, this method will be feasible to apply.

### 5.2.3. Black Box Testing

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. We have decided to perform Equivalence Partitioning and Boundary Value Analysis testing on our application. The Equivalent Partitioning will be performed at both the unit test level and the system test level. Boundary Value analysis will only be done at the system test level. In considering the inputs for our equivalence testing, the following types will be used:

* Legal input values – Test values within boundaries of the specification equivalence classes. This shall be input data the program expects and is programmed to transform into usable values.
* Illegal input values – Test equivalence classes outside the boundaries of the specification. This shall be input data the program may be presented, but that will not produce any meaningful output.

The equivalence partitioning technique is a test case selection technique in which the test designer examines the input space defined for the unit under test and seeks to find sets of input that are, or should be, processed identically. Black box testing will be performed by the test team. All procedural steps have been included to assist the team in executing the various tests.

### 5.2.4. Pass/Fail Criteria

This section will include the master list of both white box and black box tests which will be used to track the progress of the testing. A test will be considered a failure if the expected result or output is not achieved. A bug report will be filled out for each failure and will be submitted to the development team for correction. After the bug has been fixed, the test case will be repeated.

## 5.3. Test Cases

### 5.3.1. Install desktop application

|  |  |
| --- | --- |
| Name | Install desktop application |
| Summary | The desktop application is installed on device. |
| Users | All users |
| Pre-conditions | The device has python and pip installed. |
| Basic Course of Events | 1. The user accesses the installation file.  2. The mobile application is installed on the device. |
| Input | N/A |
| Expected output | The app is ready to be used. |

### 5.3.2. Desktop Application

|  |  |
| --- | --- |
| Name | Desktop application |
| Summary | The user accesses the desktop app to do summarization. |
| Users | All users |
| Pre-conditions | The app is installed on the desktop. |
| Basic Course of Events | 1. The user enters the required information shown on the screen and clicks on the generate button.  2. The video file gets generated and saved on a local machine. |
| Input | N/A |
| Expected output | Video file is saved on the local machine. |

### 

### 5.3.3. YouTube Video Download

|  |  |
| --- | --- |
| Name | YouTube video download |
| Summary | The video is downloaded on the machine. |
| Users | All users |
| Pre-conditions | The device has internet connection. |
| Basic Course of Events | 1. Select proper fields. 2. Give proper URL. |
| Input | https://www.youtube.com/watch?v=QhGXjJSg2jA |
| Expected output | The video is downloaded. |

### 5.3.4. Summarization

|  |  |
| --- | --- |
| Name | Summarization |
| Summary | Summary of videos. |
| Users | All users |
| Pre-conditions | The app is installed on the desktop. |
| Basic Course of Events | 1. Select the required fields. 2. Enter the proper path. |
| Input | c://video//test.mp4 |
| Expected output | The summary of the video has been generated. |

## 5.4. Test tools

1. Pyqt5 for desktop application
2. Moviepy for video formatting
3. Pytube for youtube video downloading.

# **6. Planning**

|  |  |  |
| --- | --- | --- |
| Time Period | Activity | Comments |
| Week 1 | Requirement Gathering | * Requirement gathering was to be done through searching on inter- net and taking the ideas, sharing the views among group members. |
| Week 2 | Planning | * Planning was done by reviewing IEEE papers. |
| Week 3 | Design | * Designing UML diagrams. |
| Week 4 | Frontend User Interface | * Designing application from UML diagrams. |
| Week 5 | Frontend Coding | * Coding for frontend in Python |
| Week 6 - 10 | Backend Coding | * Coding for Backend in Python |
| Week 11 | Test Cases | * Creating test cases. |
| Week 12 | Unit and Functional Testing | * Carrying out unit and functional testing. |
| Week 13-17 | Implementation of Additional Features | * Implementing additional features. |
| Week 18 | Debugging | * Debugging the code. |
| Week 19 -20 | Final Testing | * Final Testing. |

# 

.

# **7. Conclusion**

Video summarization is a very important tool where people can use it to get the main idea and the important scenes without watching the full original video.

The system is designed to help the users to just glance at the content of the video quickly, and decide whether to watch the video or not. For sports, looking at the highlights of the match, users can decide whether the match is interesting or not, many sports fans will be able to benefit from a summarized version of sports video which is available from anywhere. In this work we provided a systematic review of the deep learning-based video summarization landscape. This review allowed to discuss how the summarization technology has evolved over the last years and what is the potential for the future, as well as to raise awareness to the relevant community with respect to promising future directions and open issues. The main conclusions of this study are outlined in the following paragraphs. Concerning the summarization performance, the best performing supervised methods thus far learn frames’ importance by modeling the variable-range temporal dependency among video frames/fragments with the help of Recurrent Neural Networks and tailored attention mechanisms.

**References**

[1] ‘**Video summarization using deep semantic features**,’ M. Otani, Y. Nakashima, E. Rahtu, J. Heikkila, and N. Yokoya, in Proc. Asian Conference on Computer Vision, 2016.

[2] **‘Structure Analysis of Sports Video Using Domain Models.’** Zhong, D. and S.-F. Chang.. in IEEE ICME 2001. 2001. Tokyo, Japan.

[3] **‘Event detection and summarization in sports video’.** Li, B. and M. Ibrahim Sezan. in IEEE Workshop on Content-Based Access of Image and Video Libraries, 2001

[4] **Analytics Vidhya.**

[www.analyticsvidhya.com/blog/2019/09/guide-automatic-highlight-generation-python-without-machine-learning/](http://www.analyticsvidhya.com/blog/2019/09/guide-automatic-highlight-generation-python-without-machine-learning/)

[5] **GitHub.** <https://github.com/aravindpai/Cricket-Highlights-Generation>

[6] **‘Scene change detection using multiple histograms for motion-compensated frame rate up-conversion,’.** S.-J. Kang, S. I. Cho, S. Yoo, and Y. H. Kim in J. Display Technol., vol. 8, no. 3, pp. 121–126, Mar. 2012.

[7] **‘Fast scene change detection for personal video recorder.’** J.-R. Kim, S. Suh, and S. Sull,

IEEE Trans. Consum. Electron., vol. 49, no. 3, pp. 683–688, Aug. 2003.

[8] **GitHub.** <https://github.com/Breakthrough/PySceneDetect>