Bansilal Ramnath Agarwal Charitable Trust’s

**Vishwakarma University**

Pune – 411 048

**Deployment Document**

**On**

**VIDEO SUMMARIZATION**

###### **Submitted By**

**Pritesh Deepak Singhvi 201700101 (1780030053)**

**Paras Satish Karnawat 201700064 (1780030016)**

**Rinki Rajendra Ray 201700092 (1780030044)**

**Ritu Rajendra Ray 201700093 (1780030045)**

**Sumit Kishor Jeswani 201700060 (1780030012)**

**Bachelor of Technology (Computer Engineering)**

**School of Science and Technology**

**Year 2017 – 2021**

#### BRACT’s

Vishwakarma University

Survey No. 2/3/4 Kondhwa(Bk.) , Pune – 411048

Department of Computer Engineering

C E R T I F I C A T E

This is to certify that the project entitled

**VIDEO SUMMARIZATION**

Submitted by

Pritesh Deepak Singhvi

Paras Satish Karnawat

Rinki Rajendra Ray

Ritu Rajendra Ray

Sumit Kishor Jeswani

is a bonafide work carried out by them under the supervision of Prof. Prasad Gokhale and it is approved for the partial fulfillment of the requirement of Vishwakarma University for the award of the degree of Bachelor of Engineering (Computer Engineering) in the academic year 2017 - 2021.

Prof. Prasad Gokhale Head of Department

(Project Guide) (Computer Engineering)

|  |  |  |  |
| --- | --- | --- | --- |
| Table of Contents | | | |
| 1 |  |  | **Introduction** |
|  | 1.1 |  | Purpose |
|  | 1.2 |  | Scope |
| 2 |  |  | **Test Report** |
| 3 |  |  | **Project Major Module Pseudo Code** |
| 4 |  |  | **Software Installation and Configuration** |
| 5 |  |  | **Future Scope** |

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# **1. Introduction**

## **1.1. Purpose**

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.

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.

## **1.2. Scope**

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 5 secs 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.

# 

# **3. Test Report**

3.1 Test Report 1

|  |  |  |
| --- | --- | --- |
| **INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** |
| Install the application | Successfully Installed | Successfully Installed |
| Install the application without python install | Successfully Installed | Python not found |
| Install the required libraries | Successfully Installed | Successfully Installed |
| Run the application | Application windows appeared | Application windows appeared |

3.2 Test Report 2

|  |  |  |
| --- | --- | --- |
| **INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** |
| User clicks on generate button without setting the required fields | No video file generated with error message | Can't generated file please try again |
| User clicks on generate button with setting all the required fields | Video file generated | Video file generated |
| User generates sports type video with setting movie field | Sports video generated | Sports video generated instead of movie video |
| User generates movie type video with setting sports field | Movie video generated | Movie video generated  instead of sports video |

3.3 Test Report 3

|  |  |  |
| --- | --- | --- |
| **INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** |
| Downloading video from given youtube URL | Successfully downloaded | Successfully downloaded |
| Downloading video from given youtube URL without internet connection | Not downloaded with error message | No internet connection found, try again |
| User gives right path to text field | Generates the summary | Generates the summary |
| User gives wrong path to text field | Error message | File Not found |

# **4. Project Major Module Pseudo Code**

## **Module 1: Sports Summarization**

1. Input the video
2. Extract audio from the video source.
3. Break the audio into chunks having a duration of 5 seconds.
4. Calculate short time energy for every chunk
5. For each chunk, *ci* in all chunks

If *short\_time\_energy(ci)* >= *threshold*

Store the time of this chunk, ci into excitement chunks, *E*

1. For each excitement chunk, *ei* in excitement chunks
   1. Clip the video from start time to finish time as saved in *ei*
   2. Add the clip to the merging list, *M*.
2. Merge the clips in the list, *M*.

## **Module 2: Movie Summarization**

1. Input the video
2. Fetch the number of cores in the hardware
3. Divide the video frames into chunks for each core
4. For each chunk *ci* in core `*i*`
   1. For each frame *f* in ci
      1. Compute the histogram for *f* in HSV colorspace (default)
      2. Save the histogram into a list, *H*
   2. Save the list, *H* for combining later
5. Merge the histograms computed by every core.
6. Set *correlations* = []; *correlations[0]* = 0
7. For *i = 1* to number\_of\_frames(video)
   1. Compute the correlation between *i*’th and (*i - 1)*’th histogram and store in *correlations[i]*

If *correlations[i] <= threshold*

* + 1. Compute the start and end time of the scene from frame rate and frame numbers *i* and *i - 1*, respectively
    2. Store the start and end time as tuple *(s, e)* in *S*

1. Apply 0-1 Knapsack algorithm with scene durations as weights, scene histogram intensity (mean) as value and duration of the summary as capacity.
2. Merge the selected scenes.

# **5. Software Installation and Configuration**

*Pre-Installation Requirements*

The following prerequisites and requirements must be satisfied in order for the to install successfully.

*Hardware Requirements*

* Minimum Hardware Requirements
  + CPU: Windows 8 or higher or UNIX with kernel version above 5.x
  + Processor: Intel Core i3 (6th gen) or equivalent
  + Processor speed: 2.8 GHz or faster
  + Random access memory (RAM): 4gb
  + Hard disk capacity: 200mb
  + GPU: Intel(R) UHD Graphics 530
  + Internet: Internet connection required for software activation
* Recommended Hardware Requirements
  + CPU: Windows 10 or any UNIX OS with kernel version above 5.x
  + Processor: Intel Core i5(8th gen) or higher
  + Processor speed: 3.3GHz
  + Random access memory (RAM): 8gb or higher
  + Hard disk capacity: 200mb
  + GPU: nVidia GeForce GTX 1050 or equivalent
  + Internet: Internet connection required for software activation

*Software requirement*

* Python = 3.5.x or higher
* Pip = 3.x
* K-Codec Lite Pack = Basic version or higher

4.1 **Installation**

1. Download the files from [here](https://drive.google.com/drive/folders/1HWSorpeRAMoFkdXctpjE_icB3eQ7UkgE?usp=sharing).
2. Download K-Codec Lite Pack from [here](https://files3.codecguide.com/K-Lite_Codec_Pack_1595_Standard.exe).
3. Run install.bat on Windows or install.sh on Unix OS file to install all the required libraries and softwares
4. Run app.bat on Windows or run.sh on Unix OS to run the application

4.2 **Troubleshooting**

1. Python not installed : [Click Here](https://phoenixnap.com/kb/how-to-install-python-3-windows)
2. Not enough space : Try to install on different drive or upgrade your hard drive

# **6. Future Scope**

There are some areas we can further explore to improve comprehensive video understanding in the future. First of all, we didn’t benefit from action and scene connection in both recognition and summarization tasks, which leaves us room to better utilize this prior knowledge. Second, as we re-formalize video summarization in the recommender framework, some state-of-the-art recommendation technologies can be introduced such as Collaborative Filtering.

While most of current work on sports summarization aims to detect highlights or play scenes only, in this paper, we have proposed a more unified framework for sports video summarization. In particular, we have described the main reasons why play or highlights alone are not sufficient to support a wide range of requirements. In addition, it is identified that break scenes, such as preparation of a free kick, ceremonies and commentaries should still be retained to support future queries.Hence, for future work, we need to design an automated method for deciding the thresholds, based on the desired precision and recall, so that the algorithms can be fully automated.