

# Multimedia Content based Video Retrieval

## 1. how to run my program?

step 1: download my project and put it in a separate folder and copy video “.mpg” in same directory where code will be unzipped.

step 2: when you unzip the code you should be able to get src and all the related directories, now here create an empty directory "mkdir images" , when you run this program for the first time all the images will be stored here subsequently, if images folder is empty only then the new images will be created , else new images wont be created so there will be no issue of extra memory usage or time usage.

Step 3: I have imported "javacv" dependency from maven, to import this dependency use : "org.bytedeco:javacv-platform:1.5.7" .

step4: run the program, you will be able to see the graphical user interface on which the right side you will be able to see thumbnails of al the videos and when you click on any icon, you should be able to see a video playing.

## 2. what libraries did i use?

**javacv:** this is a wrapper class which contains the libraries required for image processing and computer vision related applications. it contains wrappers for opencv, ffmpeg and many other famous libraries. i used this library to capture frames from the video and to convert the captured frames into images.

**ffmpeg:** i used this library to capture frames from the video, i have captured image using ffmpegframegrabber() class, this helped me capture snapshot of every image. to know the frame number of these captured frames i have used ffmpegframenumbers().

**jav2Dframeconverter:** i have used this function to convert the video to frames ex: convert(video.mpg)

**bufferedImage:** to convert the captured frames into images i have used this library, where i am taking the scaled instance of every image to display.

**imageio.write():** this method is used to take the buffered image created from jav2Dframeconverter and the save it as a .png file on hard disk.

**timer from util package:** this class is used to schedule a thread that can be used in future , this class can be used to schedule a task to run one time or to run from time to time. i specifically used this class to display all the frames in the form of a video.

**timertask:** it is an abstract class that can implement runnable interface and we need to extend this class to create our own timertask that can be scheduled using javatimerclass

## 3. what are the functionalities of the classes i used?

**CBVR.java:** in this class i took the .mpg video as input and captured every frame using .grabimage() method provided by grabber. this method captures only the frames , so i converted these frames into buffered images and calculated intensity of each image. using these intensities i have created an array named SDs which stores the SD values of all the 4000 frames which consists of 3999 SDs. using these SDs i have calculated the mean and standard deviation values which are used to evaluate the Tb and Ts values. using these Ts and Tb values i have captured the cuts and stored the Ce values in cutEnds array and calculated the gradual transitions of the input video and stores the Fs and Fe values in the gradtransitions array. using these cuts and ends i have created a shorts array which will be sorted order and play the frames from one value of the array to the next value of the shorts array.

**FrameIntensity.java:** in this class i have calculated intensity of every image by extracting the RGB values and placing them in their respective bins by making use of intensity formula. in this class i have used two methods namely frametoframeintesnity() and calculateSDs(). frametoframeintesnity() is used to calculated the SD between two images, this method calculates the absolute difference between two frames. calculateSDs() this method is used to pass the frames one by one starting from first frame to frame 3999.

**StretchIcon.java:** this class takes an image as input , usually when we display any image, the image takes the size of the frame and the remaining part of the image is not displayed in the icon. So, to display the complete image without losing any of its features, i have used this stretchIcon class which adjusts the size of the image(height and width) according to the size of the frame which is used to display the icon.

**slideshow.java:** in this class i have used timer and timertask libraries to display an array of images in the form of a video. it contains a method named start which takes start and end as parameters, these parameters are the Ce and Fs+1 values obtained, so what this class does is it loads the images into an array named images. we have a runnable timer task which implements a run method, this run method takes current image as an input and paints the image on the Jpanel and updates the upcoming images and paints it. The timer task that we have created is scheduled using a timer class , while creating this schedule we have created a period because  $1000\text{ms}/40\text{ms} = 25\text{fps}$  , which is the actual fps rate of the original video.

This is a log printed by the program which says the input video runs at 25fps speed:

*Stream #0:0[0x1e0]: Video: mpeg1video, yuv420p(tv), 352x288 [SAR 1:1 DAR 11:9], 1150 kb/s, 25 fps, 25 tbr, 90k tbn to 40ms*

#### challenges/issues face :

- while trying to import libraries from maven i have faced multiple issues to install javacv and xuggler libraries. for javacv i had to try multiple versions to make sure that my environment supports a specific version.
- to display videos i have initially tried to implement the video display part by using vlsj library, basically what happens is if i have vlc media player on my laptop, then this library will use that media player to display the video directly on the frame that we are currently using, vlsj embeds the frame on the jpanel with the video obtained from vlc media player. but for some reason mac was not supporting that library
- i have also tried xuggler library to implement the display of the videos, but mac was not supporting any version of the xuggler , i have tried to install all the dependencies on maven and tried to resolve it but none of them worked. i tried to implement xugler using jar files as well, but that didnt work on mac os as well. i used xuggler version 5.4 which few of my other classmates used, they have implemented the usage of this jar on other OS so it seems like xuggler worked for them. i spent almost one complete day to find out a proper library which can help me in displaying the images but seems like it didnt work. I ended up using javacv and util libraries to get my work done.

the Ce and Fs+1 values that i got are:

Ce: Cuts

1091  
1112  
1575  
1618  
1926  
2332  
2584  
3008  
3050  
3532

4358

4484

Fs+1 values are:

1865

2406

2676

3200

3551

3624

3765

3838

3928

4042

4299

4561

4608

4776

4986