**Assignment 4- Option 2**

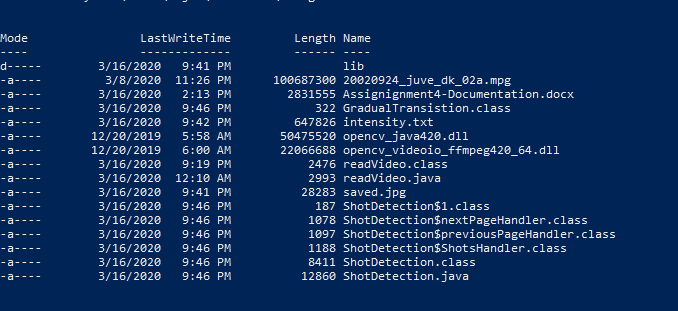
## I have used Windows 10 and windows power shell to run the program.

## I have submitted a zip file Assignment4.zip which contains all the executable files. Executable files are retrieved by doing

1. javac -cp ".;./lib/\*" .\readVideo.java
2. java -cp ".;./lib/\*" readVideo
3. javac -cp ".;./lib/\*" .\ShotDetection.java

By this we get all the executable files into the zip folder. There is also “video file” inside the zip.

**Zip folder contains:**



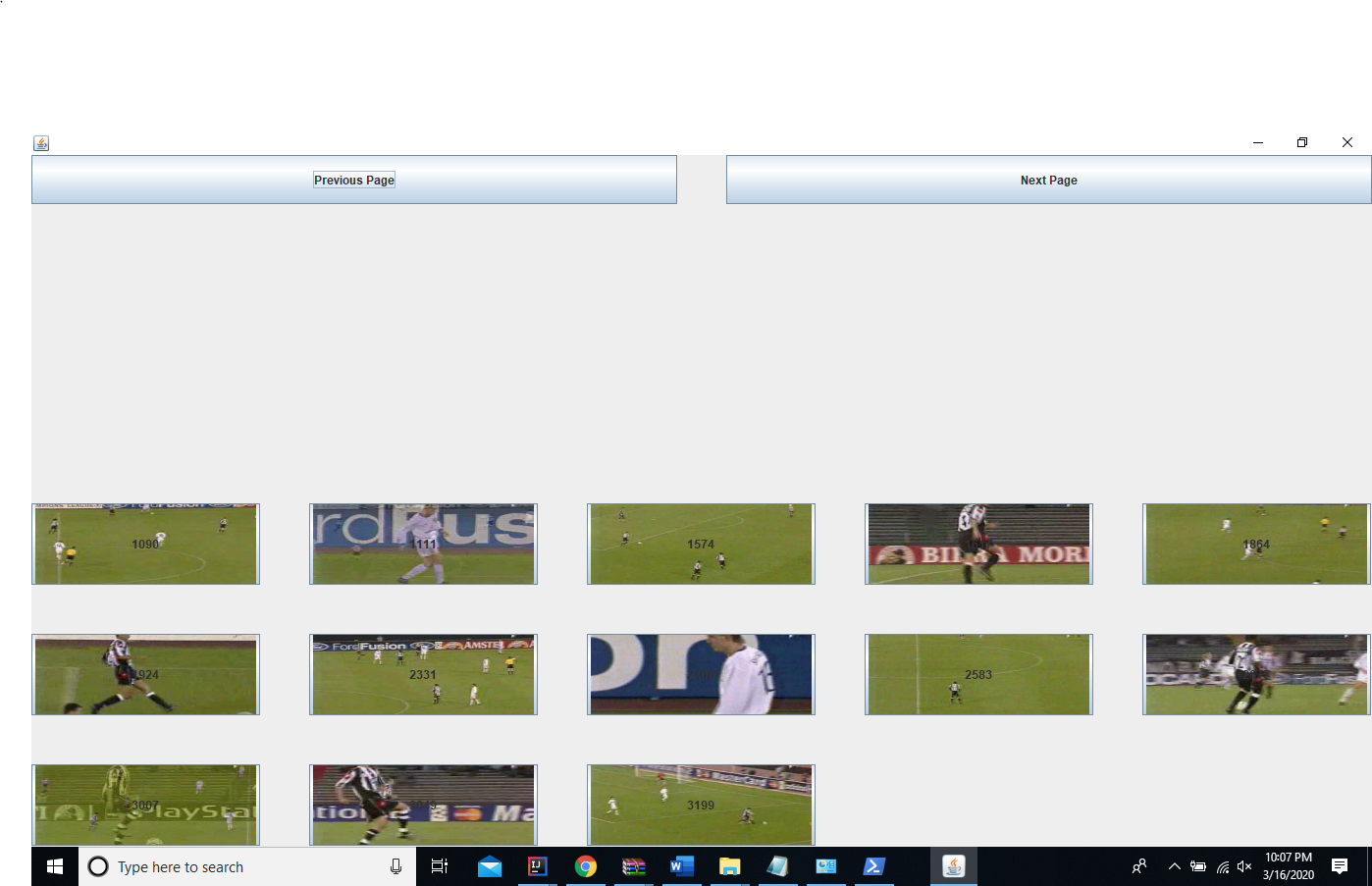
**How to run the Program:**

**Step1: Download the zip file and in power shell please get into the downloaded directory.**

In powershell or commandline run the command **java -cp ".;./lib/\*" ShotDetection**



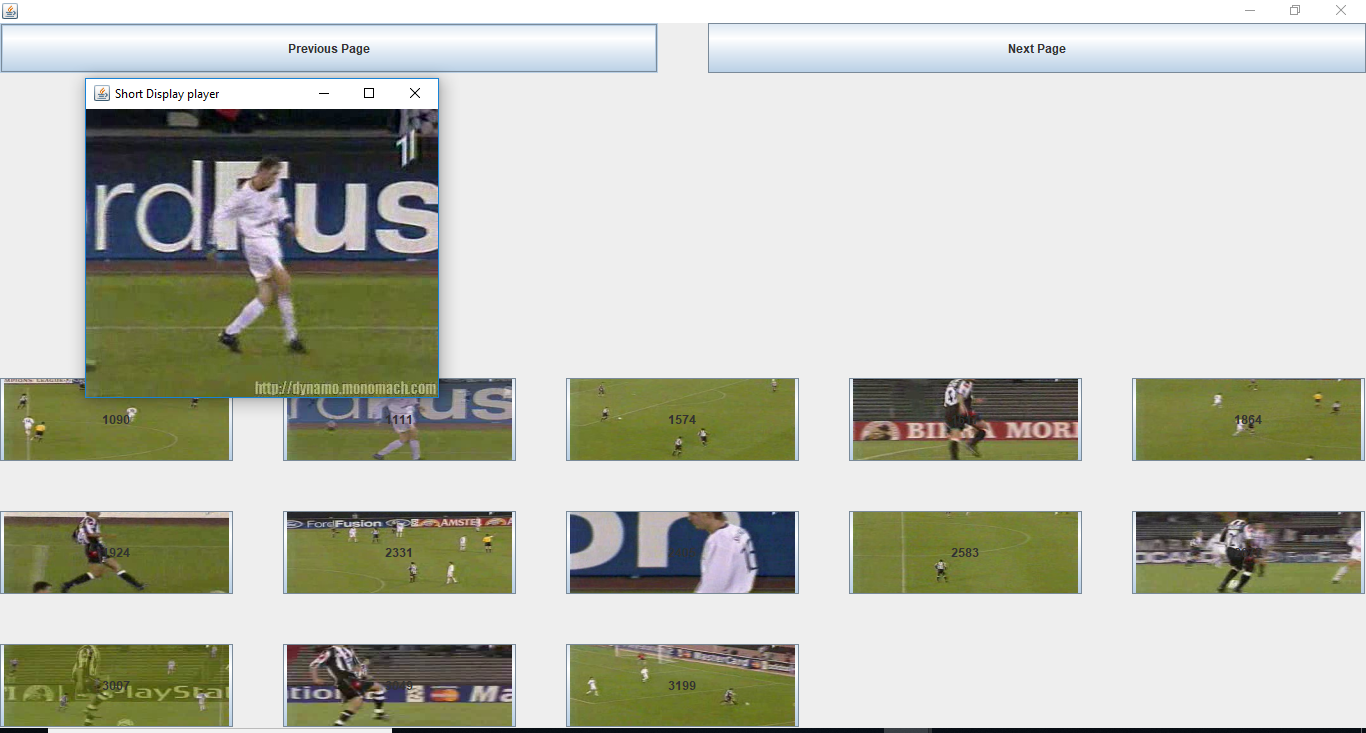
This allows the executable to run and display the required page. Once this step is done, we get a window which contains all the buttons which are the video shots.



**Step2:** Once this window is popped up we can click on any button. These buttons are videos when clicked it opens a new window and plays the shot

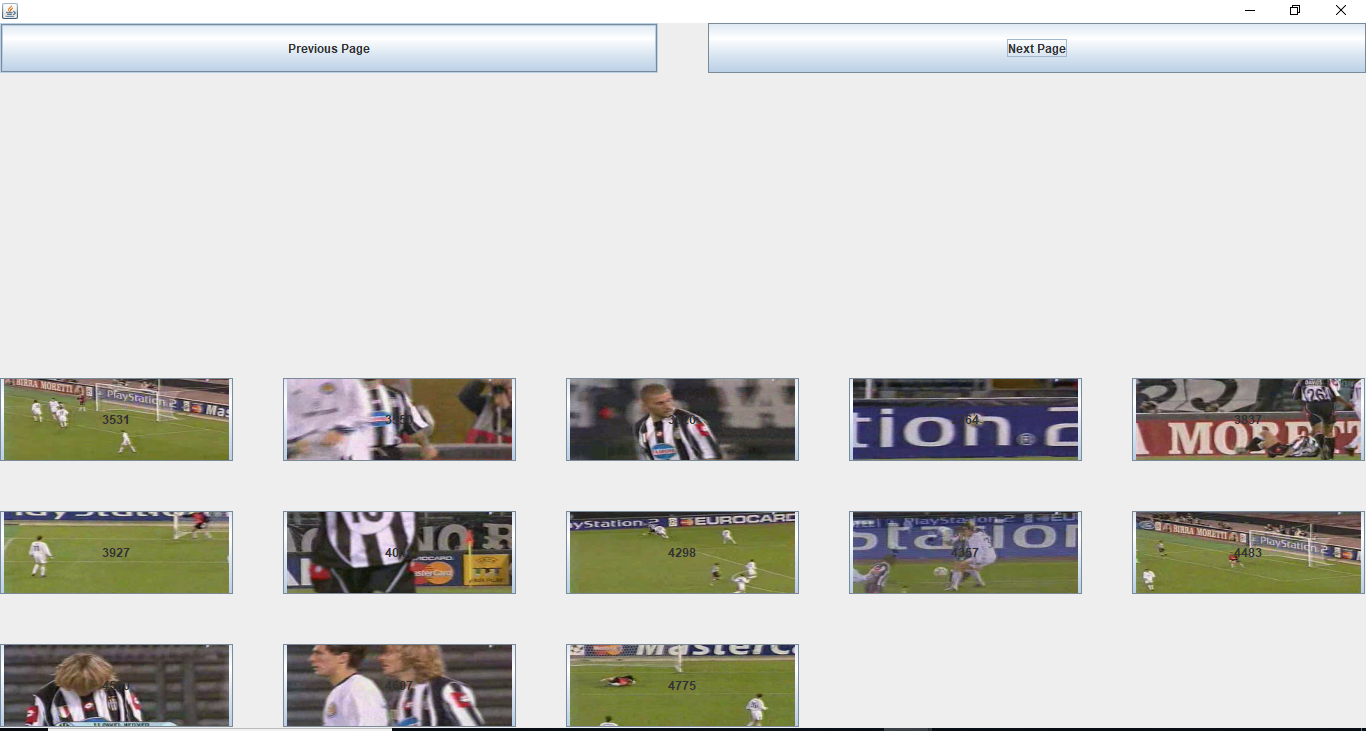
**Example 1:**

**Screenshot when button 1 is clicked:**



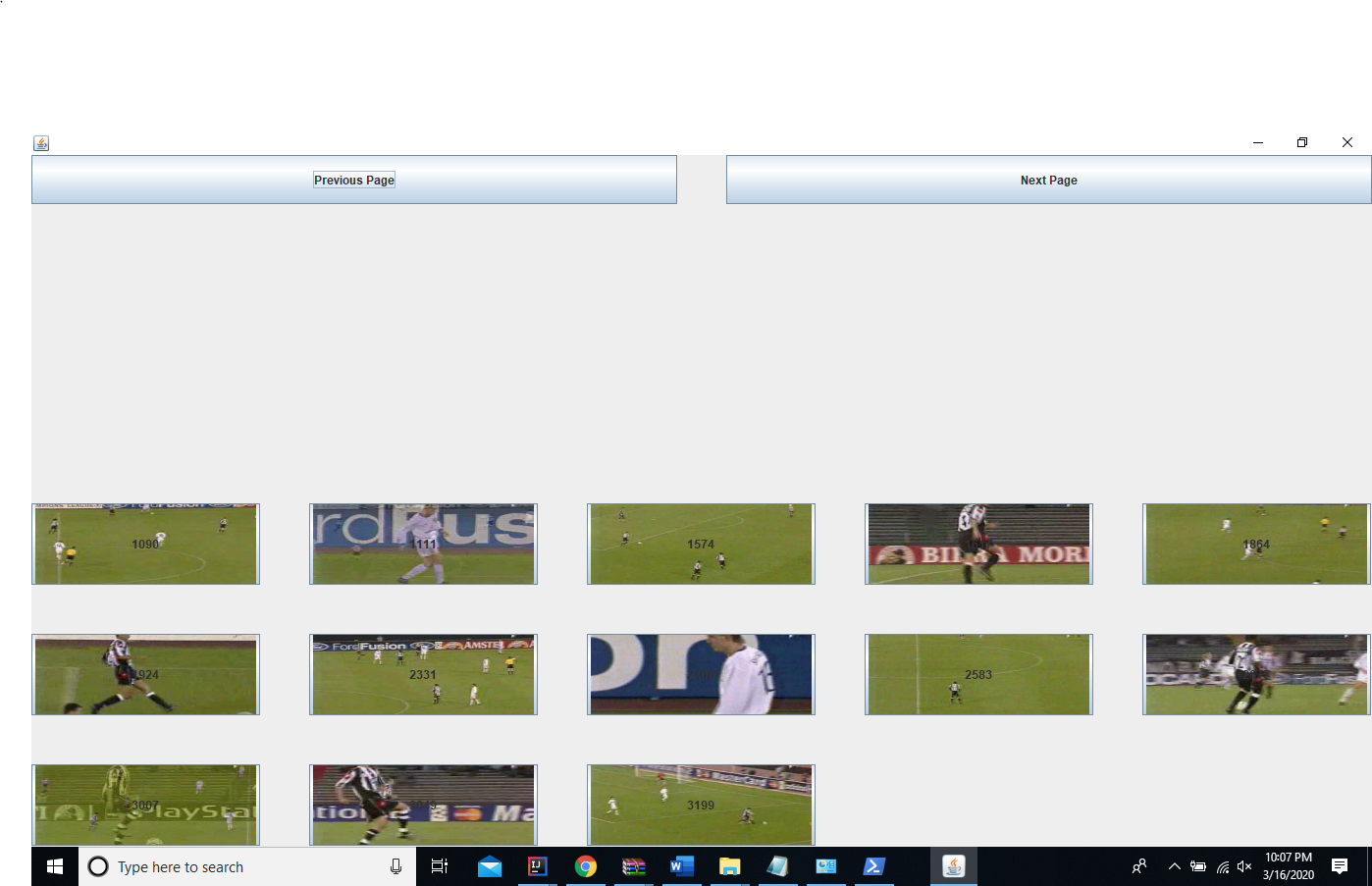
**The new window popped up played the video shot when button 1 is clicked.**

**Step 3: clicking on Next Page takes to other set of shots.**

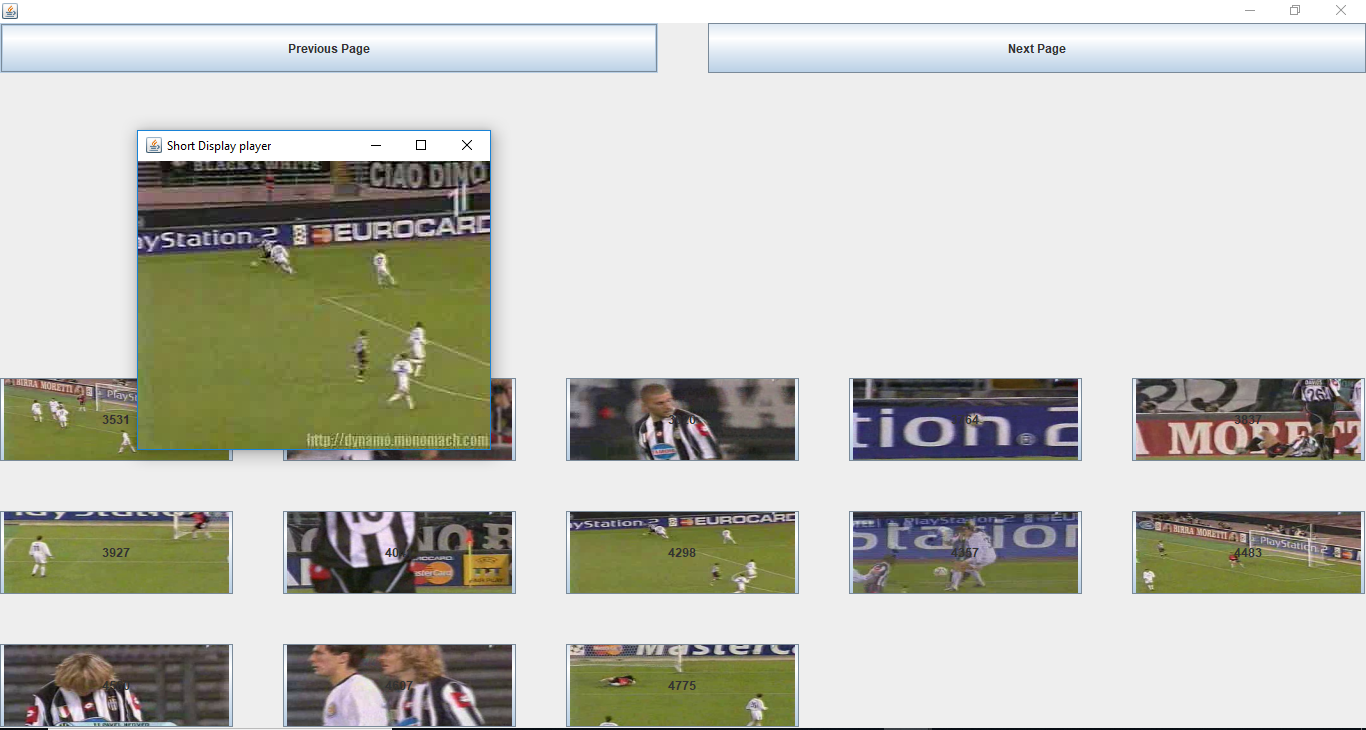


**You can observe all the buttons have changed from previous page since I have clicked Next Button.**

**Step 4: clicking on Previous Page button it takes to the previous set of shots.**



**Example Screenshot when button 7 is clicked:**

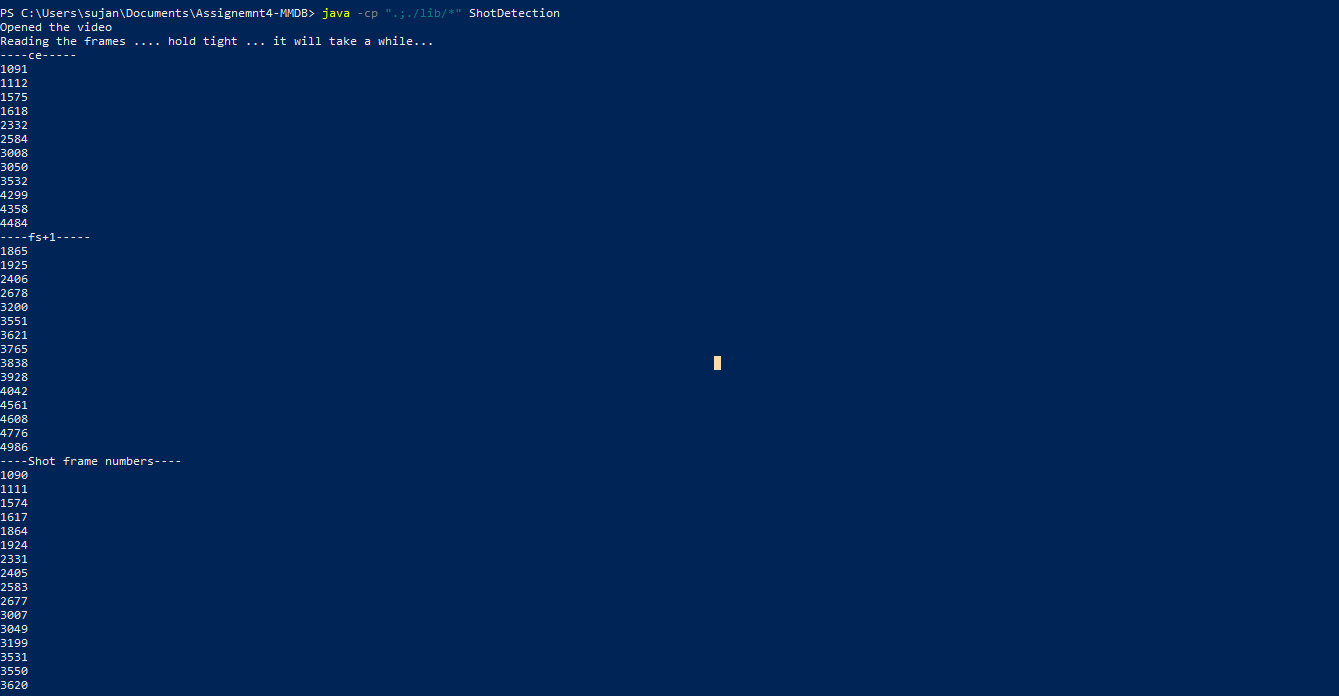


**We can observe there isa new window pop up which shows that the video played.**

**Functionalities in the GUI:**

1. I have created my own GUI. Wherein the GUI contains all the shot buttons which has the respective frame numbers of a particular shot (start frame number-end frame number) on the buttons.
2. Once we click any button then the respective video would play in a pop-up window.
3. Once the video has been played close button would work to close that shot video and give a chance to play any other shot required.
4. There are previous and Next page buttons to navigate all the pages.
5. There is an image on the button which is the starting frame image of that particular shot.
6. There is also the starting frame number on the button with black digits on top of the image.

**Screenshot of ce, fs+1 and the shot frames (all cuts and gradual transitions sorted)**





**Libraries and methods Used:**

1. **Apache Commons Math: (commons-math3-3.6.1.jar)**

* **Commons Math** is a library of lightweight, self-contained mathematics and statistics components addressing the most common problems not available in the Java programming language or Commons Lang.
* I have used this library to calculate the Standard Deviation.

1. **Javacpp and Javacv: (javacpp-1.5.2.jar, javacv-1.5.2.jar)**

* **JavaCPP** provides efficient access to native C++ inside Java. It is used to produce complete interfaces to OpenCV.
* **JavaCV** uses wrappers from the [JavaCPP Presets](https://github.com/bytedeco/javacpp-presets) of commonly used libraries by researchers in the field of computer vision ([OpenCV](http://opencv.org/), [FFmpeg](http://ffmpeg.org/), [libdc1394](http://damien.douxchamps.net/ieee1394/libdc1394/), [PGR FlyCapture](https://www.ptgrey.com/flycapture-sdk), [OpenKinect](http://openkinect.org/), [librealsense](https://github.com/IntelRealSense/librealsense), [CLPS3EyeDriver](https://codelaboratories.com/downloads/), [videoInput](http://muonics.net/school/spring05/videoInput/), [ARToolKitPlus](https://launchpad.net/artoolkitplus), [flandmark](http://cmp.felk.cvut.cz/~uricamic/flandmark/), [Leptonica](http://www.leptonica.org/), and [Tesseract](https://github.com/tesseract-ocr/tesseract)) and provides utility classes to make their functionality easier to use on the Java platform
* I have used this library in order to display the shots from given frame number to stop frame number (gradual transitions). By the use of method- showImage(Image) which takes in image as input.
* CanvasFrame method is used as part of this library, to open a new window to play the shot, when a shot button is clicked on GUI.

1. **OpenCV: (opencv-420.jar)**

* **OpenCV** (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications.
* I have used OpenCV to read the video file, using the method- Video Capture which takes in the path of the video location.
* Used ImgCodecs to save a particular frame as an image to output file. Saving a frame to image and saving to output file is to check if the exact image has been retrieved when compared to virtual dub.
* Used Mat object- In OpenCV the main matrix class is called **Mat** and is contained in the OpenCV-namespace **cv**. I have used it to store each frame detected by video capture.