**EEE508 DIGITAL IMAGE AND VIDEO PROCESSING AND COMPRESSION**

**PROJECT**

KELVIN THOMAS MATHEW (1211037370)

**OBJECTIVE**

Motion detection and brightness adjustments are implemented in real time but they require higher processing or separate devices. It would always be highly useful to implement these concepts for our handheld devices to be used instantly in times of need. Also motion detection can perform better in improved lighting settings.

**FEATURES ADDED**

Along with the coordinates and color estimation of the touch point, we have implemented two new features.

* Motion Detection
* Brightness Adjustment

Motion Detection is the process of detecting motion or movement in a steady video source like surveillance cameras. In our case we are using a phone camera placed stationary to detect any motion. In the app, when a motion is detected it will display “Motion: Detected” till the frame becomes stationary or no motion occurs.

Brightness adjustment is the process of changing the image or video brightness based on the lighting condition. In our case, we are using a phone camera to test it. Based on the lighting condition of the frame the camera captures, the brightness increases or decreases. If the brightness is high or the lighting condition is too much, the brightness is reduced for better view. In case of dim lighting the brightness is increased to enable better viewing in low light conditions.

**APPLICATIONS**

Motion Detection is widely used in surveillance to detect the motion of objects in various environments. It can also be used for our day to day activities for convenience like pathway lighting, hand dryers, automatic doors, etc.

In video capture its necessary to adjust the brightness of the frames captured to make better viewing of the images and video. Some objects are better captured and can be identified in a slightly modified lighting condition. In low light conditions, its required to get improve the brightness to increase the chance of better capture. Sometimes the excessive lighting can cause glares and the objects get masked. Decreasing the brightness slightly would result in better visualization.

**APP FEATURES**

The app starts of as shown below.



Everything is at its default settings, the coordinates are set to 0 and color is red. There are two buttons provided to choose one of the two extended features. By default, its set to brightness adjustment as shown above, the mode is displayed in red.



If Motion Detection is selected, it will display “Motion: ” when there is no movement, “Motion: Detected” when there is movement in the frame.



If the screen is touched, the corresponding coordinates and the color of the object where the touch was performed will be displayed as shown above.

**EXECUTION STEPS**

Please note that the **minSDKversion was set to 21** considering the fact that real time testing could only be performed on an android phone with Android 5.1 Lollipop (APK 21).

The APP (APP\_build.gradle), opencv (opencv\_build.gradle), OpenCVLibrary320 (OpenCVLibrary320\_build.gradle) gradle files have been attached for reference.

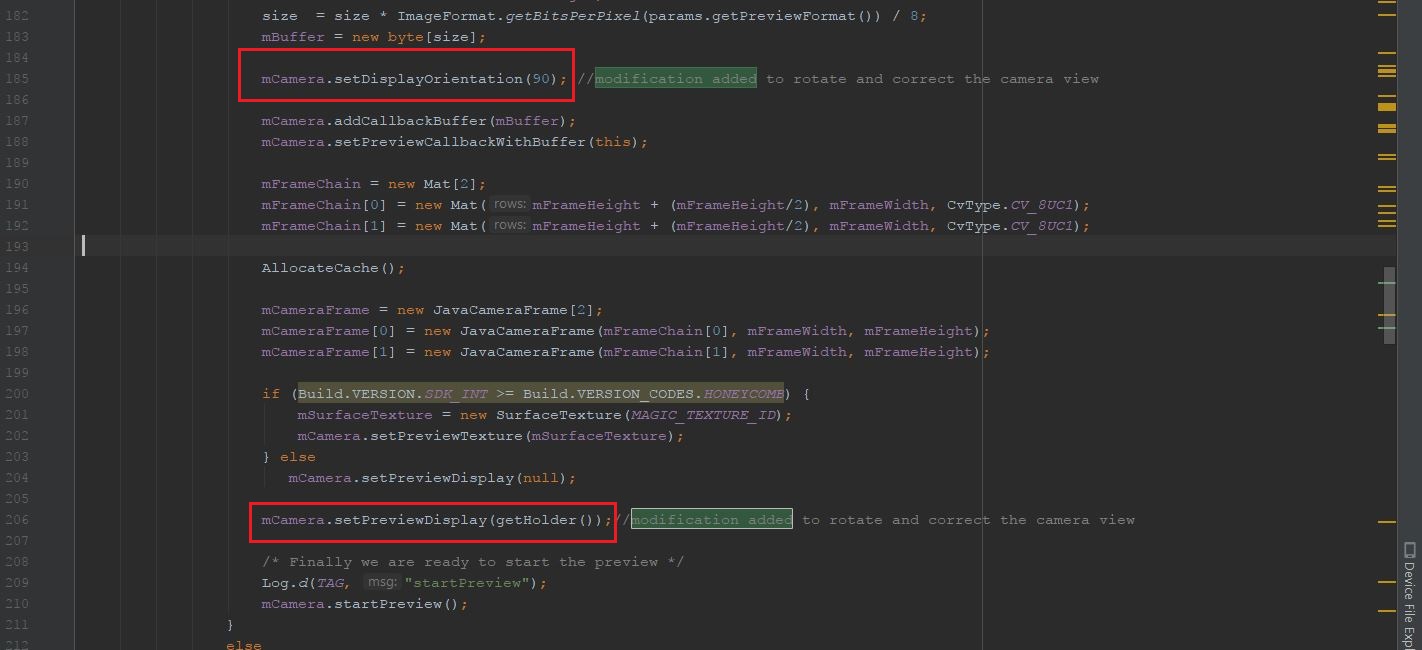
The code changes for PART A is attached as “PART A\_activity\_main.xml”.

The UI code changes for PART B & C are attached as “PART B&C\_activity\_main.xml”.

Two lines of code were added to correct the camera display orientation as shown below in JavaCameraView.java file. The camera was initially displaying 90 degree rotated version when tested on webcam and on the phone. As per the suggestion from the link (<https://stackoverflow.com/questions/14816166/rotate-camera-preview-to-portrait-android-opencv-camera>) we have the below two lines.

mCamera.setDisplayOrientation(90);

mCamera.setPreviewDisplay(getHolder());

****

The main code changes have been added to MainActivity.java which has been attached.

Once all these changes have been made, run the app which will build the gradle and launch the app.

**SAMPLE INPUT AND OUTPUT**

The sample execution in real time is attached.

PART B (Coordinates & Color).mp4 shows the coordinates and color of the touch.

PART C (Motion Detection & Brightness Adjustment).mp4 shows motion detection and brightness adjustment.