**Project 3**

**CS6825: Computer Vision**

**Section 0 Introduction and Proposal:**

**App Name: Netrikan**

By Gowrishankar Dakshnamurthy

NetId: QM9576

Email ID:gdakshnamurthy@horizon.csueastbay.edu

**Goals of Application:**

**The application is designed to help a visually challenged person pick up an object from a table or a flat surface in front of the person. The application is now restricted to find black colored box. The person will reach for the object as shown in input image. Then the voice system will guide the user towards the object.**

**Input image**



**Algorithms:**

**1) Continuously scan the picture for both the object and hand**

**2) The hand and the object are detected using OpenCV color blob detection method**

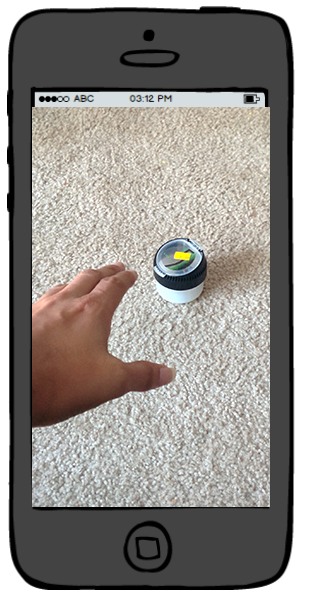
1. **The application looks for the already defaulted color of hand and the box (black) on the frame.**
2. **Any segments in the frame matching with the exact same color, the program will draw a contour over the segments matching.**
3. **The segment which has a large area will be highlighted by forming a green polygon outline around its border.**

**3) Check if any side the polygons formed by hand are on the same vertical axis as polygon formed by object.**

**4) If not ask the user to move left or right in accordance to the position of the hand to the box, till the tip of the line coincides with the objects edge coordinates.**

**5) If coincides ask the user to move hand forward till he gets the object.**

**GUI Input Interface:**

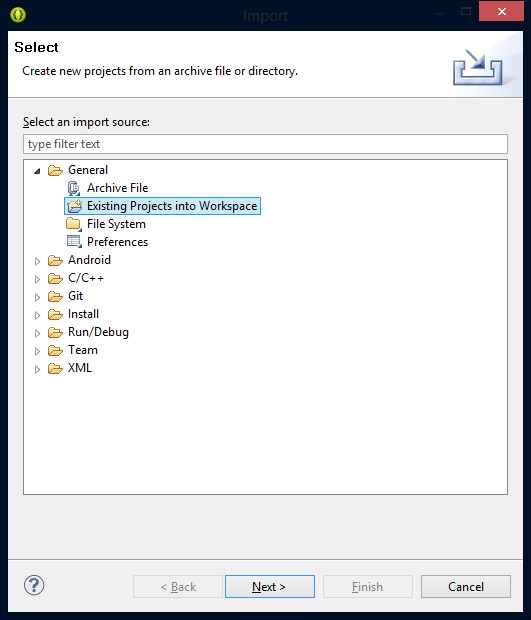
****

**Output**

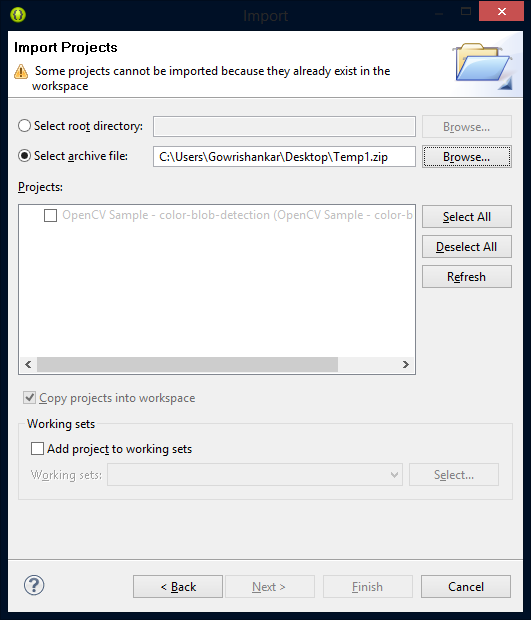
**Voice output**

**Section 1 Execution Instructions:**

*Instructions for me to download and run your code. YOU NEED to show me screen shots of you doing this from your uploaded blackboard code.....this forces you to make sure that I can run your code. You MUST have the following screenshots AND give description on what to do: screenshot 1.1 = screen shot of your files uploaded to Project 1 turn in folder on blackboard*



*After downloading the zip file from blackboard in to a specific destination in your computer, open Eclipse, go to file-> Import, select “Existing project in to workspace” option and click Next.*



*Click on select archive file radio button and browse for the downloaded zip file by clicking on browse button and select the zip file then press finish. The project would import as OpenCV sample Color-blob-detection. Run the project as Android Application in a Device.*

**Section 2 Code Description**

**ColorBlobDetector.java :**

The part of getting the scalar value from the screen and getting the proper color in HSV format and getting the contours

**public void setHsvColor(Scalar hsvColor):**

In this function, the scalar parameter passed to the function is analyzed and manipulated with to get the color value of that scalar parameter in HSV format.

**public Mat getSpectrum()**

Used to get the value of HSV format color

**void process(Mat rgbaImage)**

In this function, the contour is drawn in the frame. It draws the contour for the largest color blob detected in the frame matching the color In HSV format.

**List<MatOfPoint> getContours()**

Used to get the contours drawn in the frame.

ColorBlobDetectionActivity.java:

This is the main activity page of the application where the activity methods and some process call methods are handled.

**public** **void** onCameraViewStarted(**int** width, **int** height)

Gets called when camera is started, Instantiates some value related to manipulation of the frame.

**public** **void** onCameraViewStopped()

Gets called when camera is stopped, Releases mRgba.

**boolean** onTouch(View v, MotionEvent event)

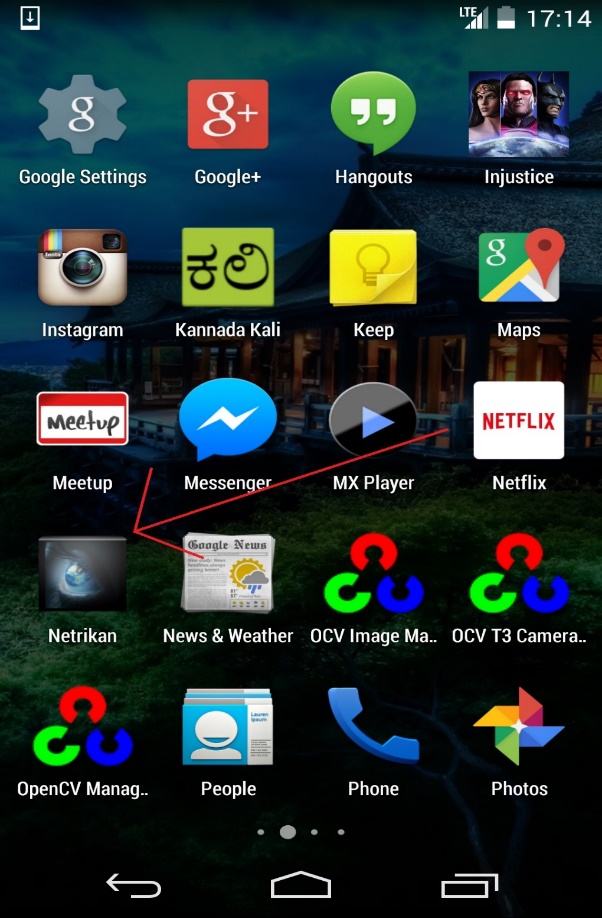
Gets the coordinates of the touched space, turns the scalar value in to HSV format and gives it to detector so that it can use the value to detect color blob.

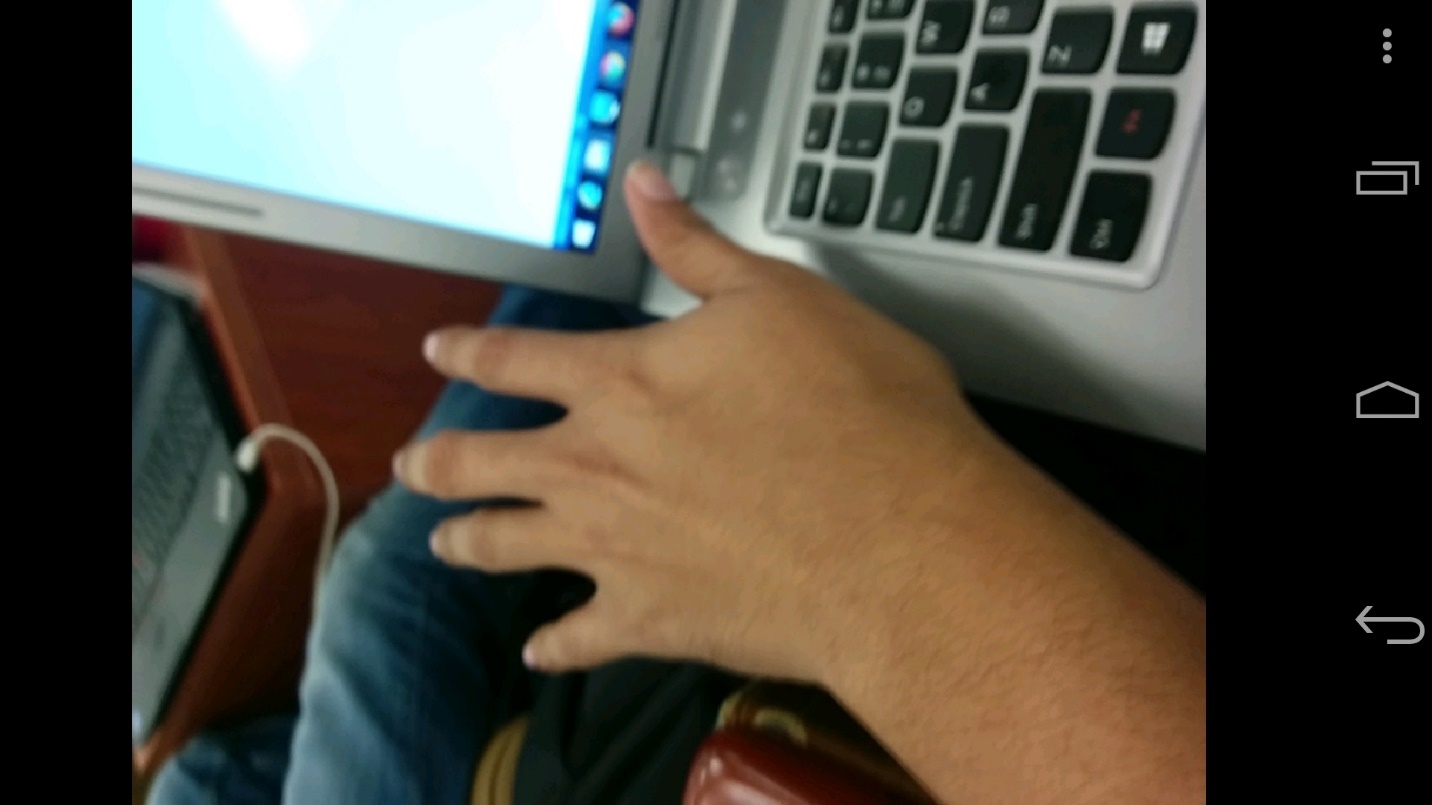
Mat onCameraFrame(CvCameraViewFrame inputFrame)

In this function, each frame is processed and send to the display screen. The manipulation of the frame is done by calling the process function in ColorBlobDetector Class.

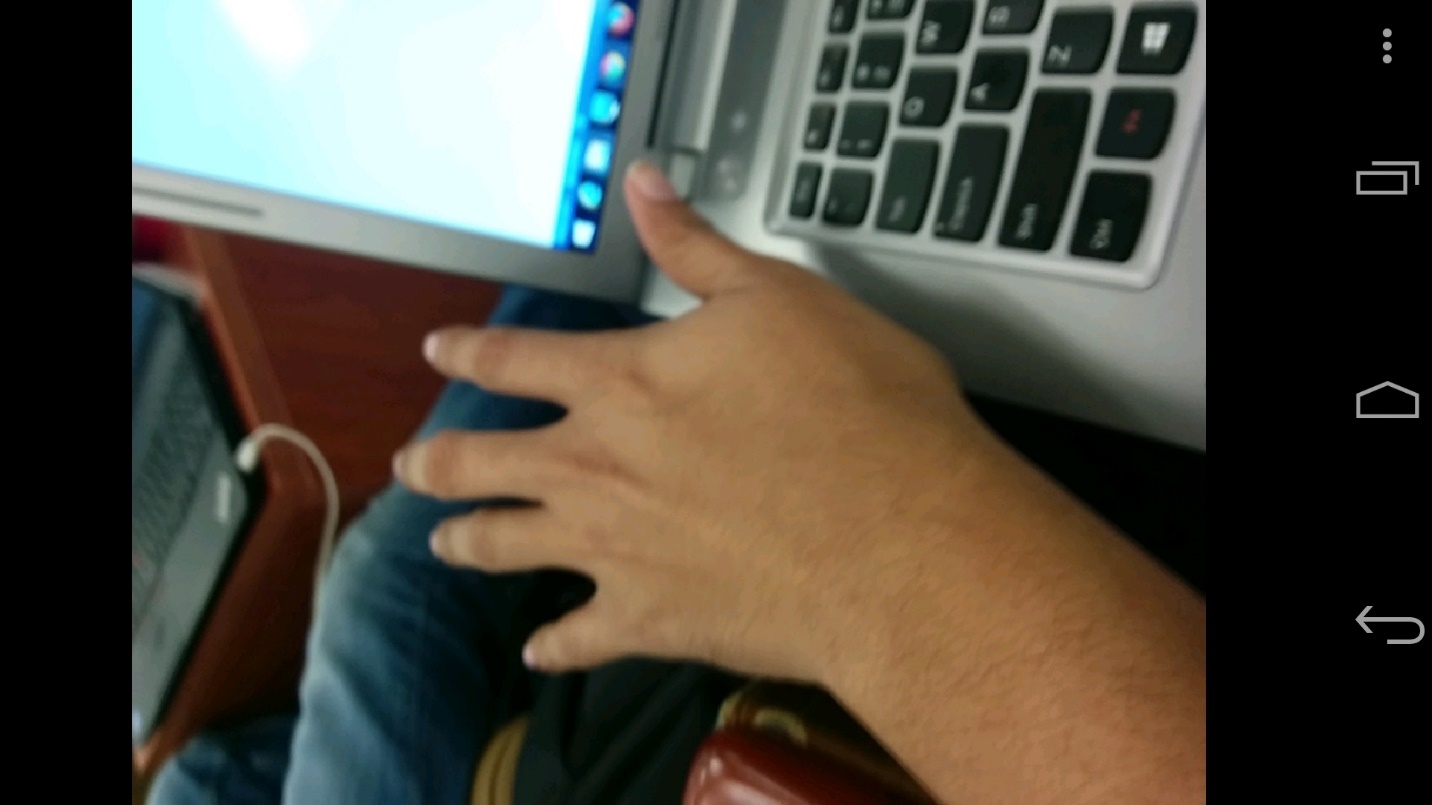
**Section 3 Testing**:

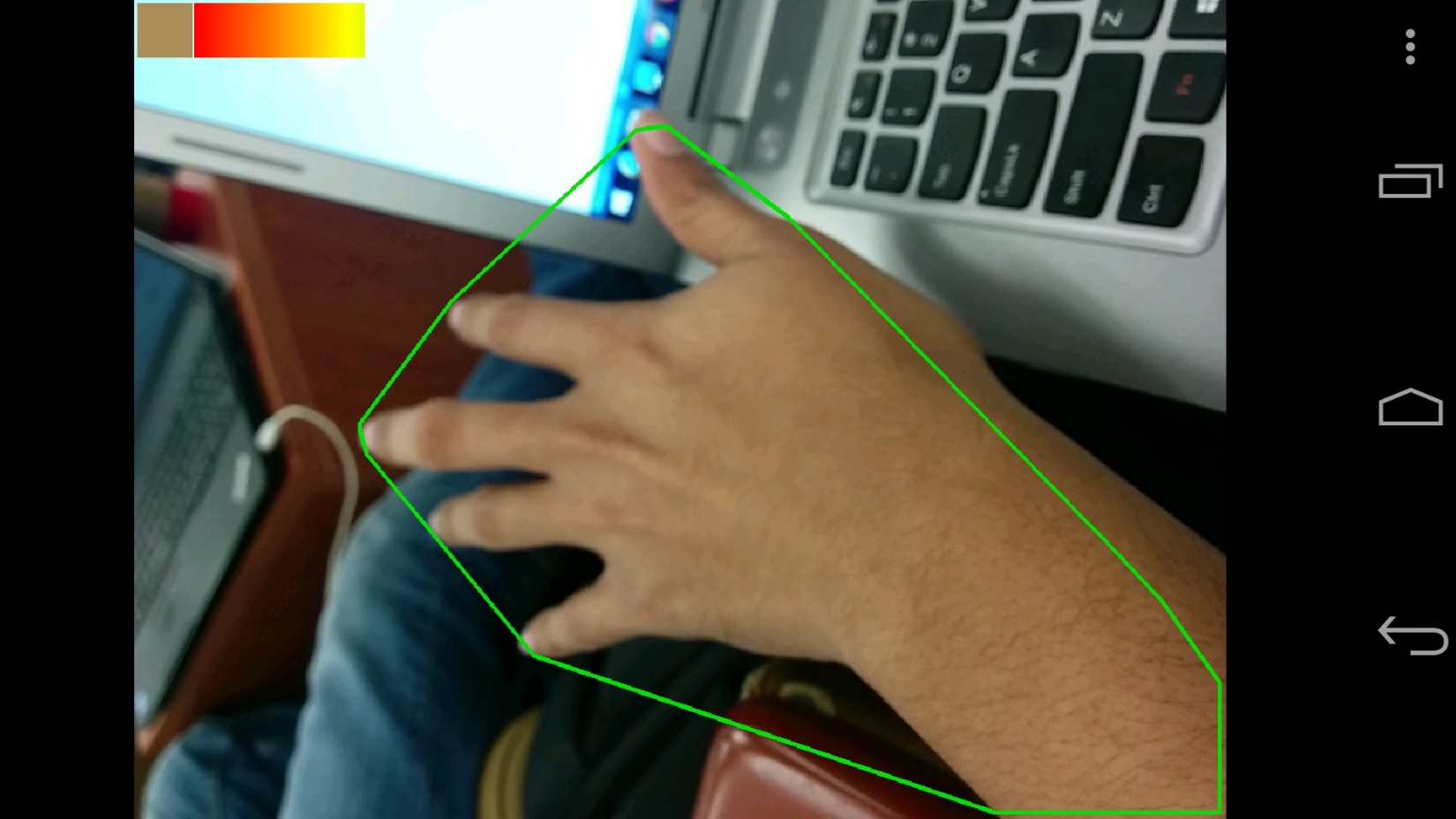
**section 3.1: starting application**

  
screenshot 3.1a= showing icon and resulting starting GUI.

  
screenshot 3.1b= the image being viewed in your application that was just loaded.

**section 3.2:**

  
screenshot 3.2a = screen shot of active image in your application you are going to process

  
screenshot 3.2b = screen shot of showing results of running your program.

**Section 4 Comments**

The application as such is not completed. The first phase of the application has been completed and still the application has to go through 3-5 more phase to make this more accurate and saleable.

**Section 5 YouTube URL**

Presentation video of Project Document:

<http://youtu.be/NPWhwIHEjQ4>

Demo Video of Application:

<http://youtu.be/z0yX83Kq4LY>