A REPORT

ON

ANDROID APP DEVELOPMENT FOR SOIL TESTING AND FRUITS QUALITY INSPECTION

BY

BHARAT ARORA

2014B3A7629G

AT

CSIR-CENTRAL ELECTRONICS ENGNEERING RESEARCH INSTITUTE

CSIR-CEERI, Pilani

A Practice School-I station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

(JULY, 2016)

A REPORT

ON

ANDROID APP DEVELOPMENT FOR SOIL TESTING AND FRUITS QUALITY INSPECTION

BY

BHARAT ARORA

2014B3A7629G

Prepared in partial fulfilment of the

Practice School-I Course

AT

CSIR-CENTRAL ELECTRONICS ENGNEERING RESEARCH INSTITUTE

CSIR-CEERI, Pilani

A Practice School-I station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

Acknowledgement

I would like to express my special thanks of gratitude to Prof. Santanu Chaudhury, the Director of CSIR-Central Electronics Engineering Research Institute to have a collaboration with BITS Pilani for the Practice School I programme and providing the opportunity to students to work in one of the premier research institute of our nation.

I would like to express my gratitude to Dr. Satyam Srivastav, head of Water Technology Lab, CSIR-CEERI for his guidance and support during the project. I would like to thanks Mr. Pawan Sharma, faculty coordinator of the PS1 station CSIR-CEERI of BITS Pilani, for his guidance while selection of projects.

I would like to extend my thanks to all the people of Practice School Division of BITS Pilani who work hard day and night to make PS1 programme successful.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

PILANI (RAJASTHAN)

Practice School Division

Station: CSIR-Central Electronics Engineering Research Institute **Centre**: Pilani, Rajasthan

Duration: 8 Weeks **Date of Start**: 23/05/2016

Date of Submission: 12/0672016

Project Title: Android app development for testing soil and fruits quality inspection.

ID No. : 2014B3A7629G

Name: Bharat Arora

Discipline of the student: MSc. (Hons.) Economics Science and B.E. (Hons.) Computer Science

Name and Designation Of the expert: Dr. Satyam Srivastava (In-charge of Water Technology Lab)

Name of the PS Faculty: Mr Pawan Sharma

Keywords: API, Inheritance, Multithreading, Exception Handling

Project Areas: Image processing, Android App Development

Abstract

Current changes in global environment has led to the increased requirement for crop care for the farmers. Most of the developed countries has done this through the inclusion of the advanced technology in the farming methods but farmer in developing countries like India are still lack behind in terms of modern equipment because most of them are imported. In India this has reached to more alarming stage due to lack of awareness among farmer about the use of fertiliser and frequent droughts. This report presents a step towards solving these problems through application of rapidly growing technologies like image processing and android smartphones. A prototype of two basic app “Soil Scanner” and “Fruit analyser” is presented in it. Former is developed to solve the problem (like time lags between soil submission and soil health card generation) associated with “Soil Health Card scheme” and later is another independent app to access the quality of fruits and vegetables. It gives information about various important quality parameters such as degree of freshness, storage time, firmness, shelf life and early warning of different disease.

Signature of Student **Signature of PS Faculty**

Date **Date**

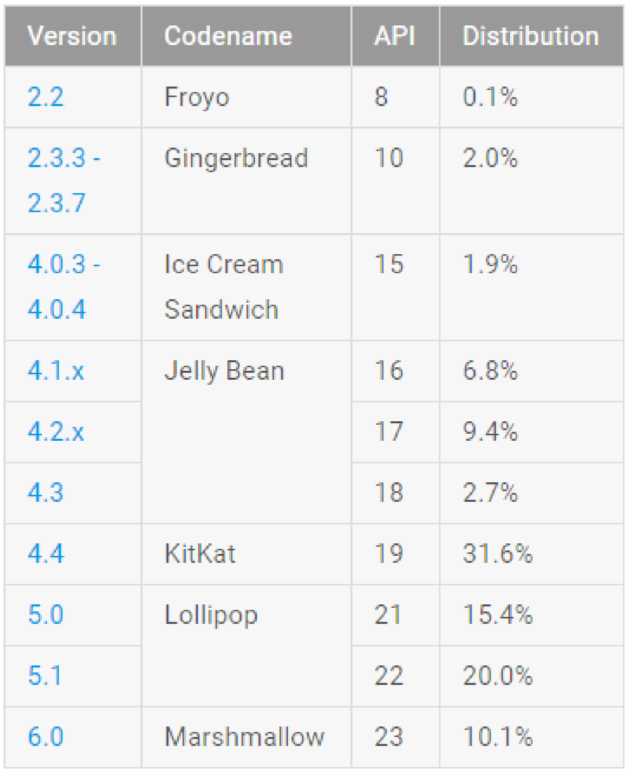
Table of Contents

1 Introduction

In recent years, agriculture sector has grabbed a lot of attention by government of India. Due to increased mortality rate among the farmers because of crop failures. Various reasons have been offered to explain these suicides like frequent draughts, unfavourable government policy and lack of modern equipment. So, Indian government has launched a lot of schemes to help farmers. One of these scheme is soil health card scheme. But there are some issues associated with it like sample collection by improper methods and delays in generation of soil heath card. So, to solve these issues a handy system is required which can be generated through combination of image processing and android. In the recent past, developing countries have experienced major technological advancements including high smartphone penetration. So assuming, smartphones are very easily accessible, prototype of two apps is generated. First can be used for inspection of soil properties and second can be used for quality inspection of fruits and vegetables.

2 Introduction to Android

Android is an open source and Linux-based Operating System for mobile devices, cameras, televisions and watches etc. Android was developed by the Open Handset Alliance, led by Google, and other companies. The below given table gives a cumulative estimate of the number of mobile devices running different API level of android.



Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.9%
4.1.x	Jelly Bean	16	6.8%
4.2.x		17	9.4%
4.3		18	2.7%
4.4	KitKat	19	31.6%
5.0	Lollipop	21	15.4%
5.1		22	20.0%
6.0	Marshmallow	23	10.1%

Figure 1 Percentage of devices running on different API Levels

Currently most of the latest devices are running Lollipop 5.1.1 and some of the devices has got updates for marshmallow. So the appropriate targeted Sdk level and minimum Sdk level for developing an app should be 23 and 15, so that app can run 99.9% of devices.

2.1 Android architecture

Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram.

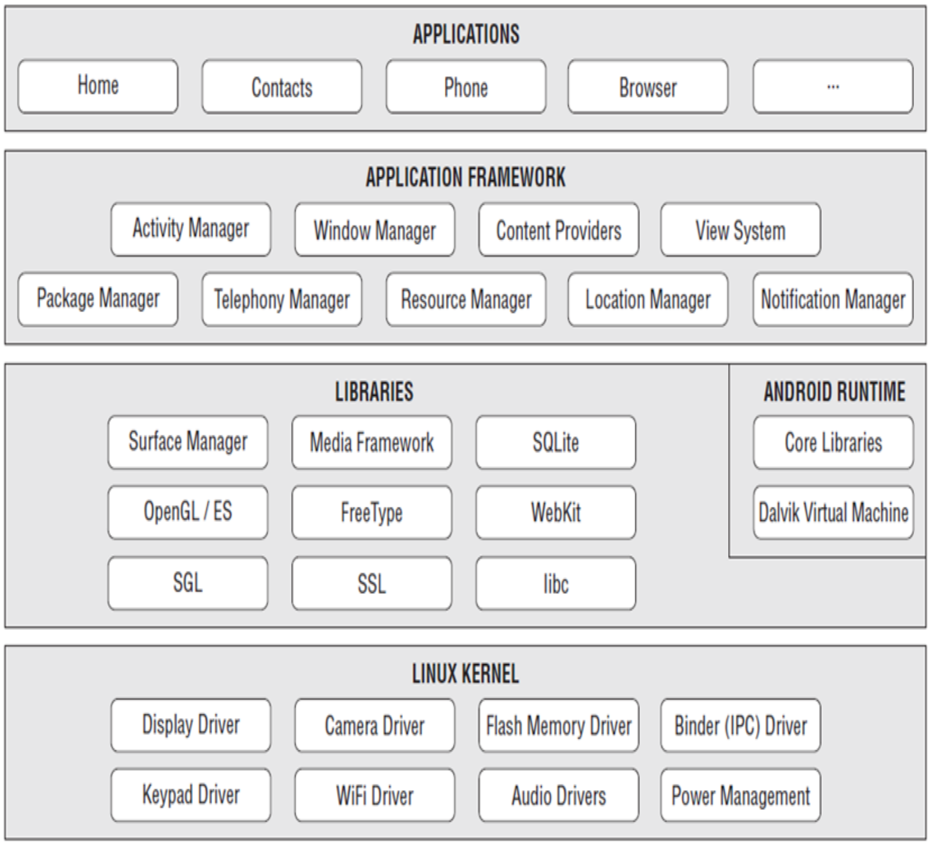


Figure 2 Android Architecture

Bottom layer contains Linux kernel which provides a level of abstraction between the device hardware. It has all the essential hardware drivers. There are libraries for storage and sharing of application data, libraries to play and record audio and video and libraries responsible for Internet security etc. On top of system libraries there are application framework libraries and those that facilitate user interface building, graphics drawing and database access. Top most layer contain our application which use all these previously given resources to work perfectly.

3 Introduction to Image Processing

3.1 Digital Image Possessing

Image processing is method that deals with the conversion of image into digital form and then performing some manipulation on it, to extract some useful information in it which can cannot be detected under normal human vision

3.2 Image

A digital image is nothing but a two dimensional matrix of signals. It is mathematical function f(x, y) of x and y are the two co-ordinates horizontally and vertically where the value of f(x, y) at any point is gives the pixel value at that point of an image.

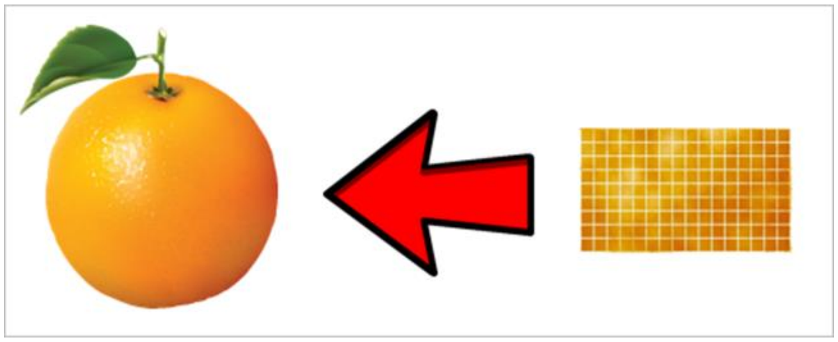


Figure 3 Matrix form of Image

3.3 Types of Image

3.3.1 Binary Image

The binary image contain only two pixel values 0 and 1. It requires 1 bit of memory per pixel. Binary images have a format of PBM (Portable bit map).

3.3.2 Grayscale Image

A grayscale image has 256 different shades of colours in it. It requires 8 bit of memory per pixel .The format of these images are PGM (Portable Grey Map).

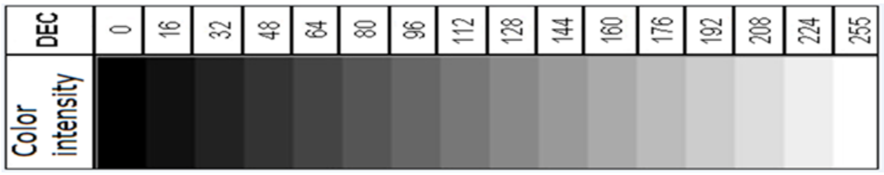


Figure 4 Different intensities grayscale

3.3.3 RGB Image

Each pixel of RGB Image is has red, green and blue colour intensity value. It requires 24 bit of memory, 8 bit for each of R, G and B. The format of these images are BMP (Bitmap).

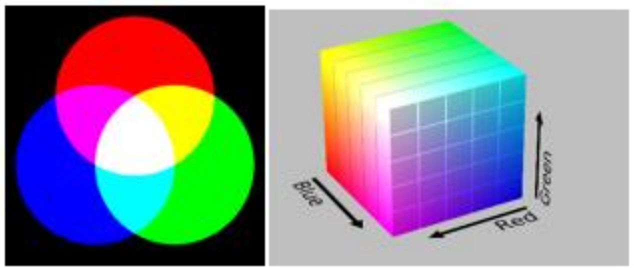


Figure 5 RGB cube

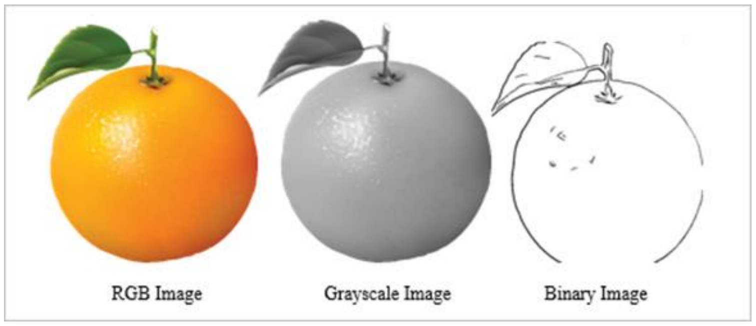


Figure 6 Types of Image

3.4 Image Matrix

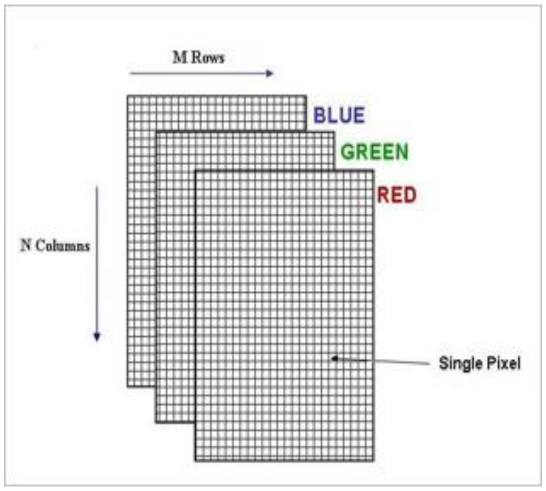


Figure 7 Image Matrix

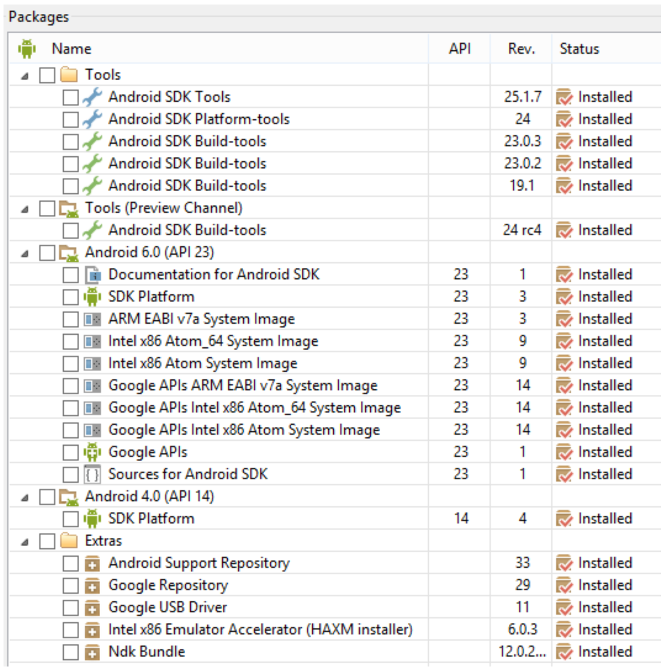
It is matrix in which each element of matrix represent the intensity of color falling in that area. RGB Image matrix has three value corresponding to a pixel but grayscale has only one. Each value ranges from 0-255 in a 8 bit grayscale image.

4 Setting up Android Studio for development

These are the steps to setup environment in windows 8.1. For any other operating system please refer to stackoverflow.com.

4.1 Installing Android Studio

1. Download latest java jdk from http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html according to system your requirement.
2. Install jdk.
3. Copy the path of java jdk. (e.g. C:\Program Files\Java\jdk1.8.0_91 this will be the default path of your application.)
4. Go to Computer>>properties>>Advanced System Setting >> Environment variables >> New (under User variable) >>put Variable name: JAVA_HOME and Variable path: jdk path. Click Ok.
5. Download android studio from https://developer.android.com/studio/index.html
6. Install android studio (uncheck the start android studio option at Finish).
7. Run android studio first time as administrator.
8. It will download some essential component. (It is just a onetime thing).
9. Before making your first project. Click configure >> SDK Manager >> Launch Standalone Sdk Manager>> update the following package. Make sure that you android Sdk path do not contain any spaces not even in user name.



Packages

Name	API	Rev.	Status
Tools			
Android SDK Tools		25.1.7	Installed
Android SDK Platform-tools		24	Installed
Android SDK Build-tools		23.0.3	Installed
Android SDK Build-tools		23.0.2	Installed
Android SDK Build-tools		19.1	Installed
Tools (Preview Channel)			
Android SDK Build-tools		24 rc4	Installed
Android 6.0 (API 23)			
Documentation for Android SDK	23	1	Installed
SDK Platform	23	3	Installed
ARM EABI v7a System Image	23	3	Installed
Intel x86 Atom_64 System Image	23	9	Installed
Intel x86 Atom System Image	23	9	Installed
Google APIs ARM EABI v7a System Image	23	14	Installed
Google APIs Intel x86 Atom_64 System Image	23	14	Installed
Google APIs Intel x86 Atom System Image	23	14	Installed
Google APIs	23	1	Installed
Sources for Android SDK	23	1	Installed
Android 4.0 (API 14)			
SDK Platform	14	4	Installed
Extras			
Android Support Repository		33	Installed
Google Repository		29	Installed
Google USB Driver		11	Installed
Intel x86 Emulator Accelerator (HAXM installer)		6.0.3	Installed
Ndk Bundle		12.0.2...	Installed

Figure 8 Required Packages

10. After update reopen the android studio to make sure it android studio is aware of update.
11. It is preferred you run your app on real device. For that you need to setup.
12. Install OEM driver for your phone on laptop from the following link https://developer.android.com/studio/run/oem-usb.html.
13. In your phone go to Setting >> About Phone >> tap 7 times on Build number to enable developer options.
14. Go back and open developer options, Turn on developer options and enable usb debugging.
15. It is also preferred to check stay awake options to prevent getting your phone in offline mode from android studio while using it.
16. After connecting through usb change usb option to Camera (PTP).

4.2 Integrating opencv with Android Studio

1. Download “opencv for android” from http://opencv.org/downloads.html. (Project is done using opencv for android 2.4.10.)
2. Unzip the directory and place the folder somewhere the absolute path do not contain any spaces.
3. In your android studio project go to Menu: /File/New/Import Module: Source-directory of unzipped director >> Sdk >> java. Click Ok.
4. You will get some errors. (But *you also get an error message saying failed to find target with hash string 'android-14'.... This happens because the build.gradle file in the OpenCV zip file you downloaded says to compile using android API version 14, which by default you don't have with Android Studio v1.4.1.*). Click on the link provided in the errors and download it.
5. Go to Menu:/File/Project Structure >> app >> Dependencies >> click on + button on right-top side >> Module dependencies >> select opencv >> Ok.
6. Correct following in build.gradle (Module: openCVLibrary2410) by looking at build.gradle (Module: app). And sync gradle again.
   a. compileSDKVersion
   b. buildToolsVersion
   c. minSdkVersion
   d. targetSdkVersion

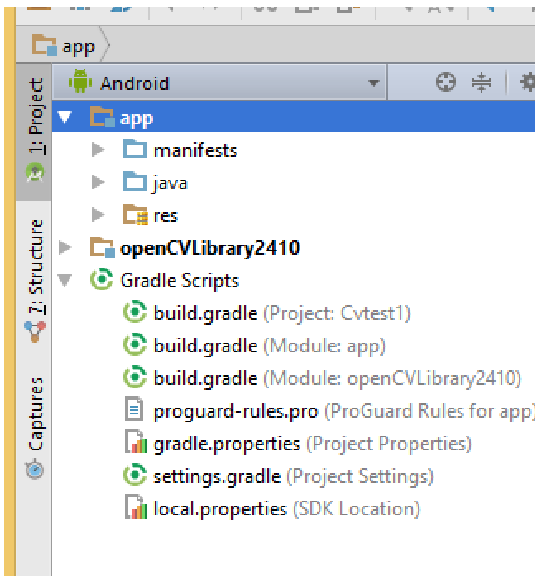

app
Android
app
manifests
java
res
openCVLibrary2410
Gradle Scripts
build.gradle (Project: Cvtest1)
build.gradle (Module: app)
build.gradle (Module: openCVLibrary2410)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)

Figure 9 Gradle Scripts

7. Copy {unzip-dir}/Sdk/native/libs into cvtest1/OpenCVLibrary310/src/main/ and rename the libs folder into jniLibs.(right click >> refractor >> rename)
8. Always add the following function to load the library before you use it.

```
if (!OpenCVLoader.initDebug()) {
    Log.e(this.getClass().getSimpleName(), "  OpenCVLoader.initDebug(), not working.");
} else {
    Log.d(this.getClass().getSimpleName(), "  OpenCVLoader.initDebug(), working.");
}
```

Figure 10 Code to Load OpenCV

Now android studio is ready to use with integrated opencv.

5 Developing an App

5.1 Soil Scanner

It is app to identify essential properties of soil based on feature of captured image from mobile phones camera. Properties like pH, nutrient content shows a very a high correlation with soil Color. There are various research conducted in the past to verify this correlation. Before developing an app, data collection is done from these research papers and other secondary sources to map features of image to soil characteristics.

Image processing is computationally very intensive process and it uses lot of resources of computers/smartphones. Hence, a lot of existing smartphones will not be able to do complete process in one step. So, complete process is divided it small sub process to enhance the user experience while using app. Following are the steps

1. Loading opencv library
2. Getting user information
3. Getting sample image from phone
4. Processing image for pH & Nutrient calculation
5. Displaying the results
6. Verifying results manually.
7. Processing image for characteristic matching
8. Displaying the characters
9. Verifying results manually.

5.1.1 Loading OpenCV library

Android app takes some time to start up, especially when it uses some other libraries like opencv. So to avoid time lags, a splash screen is created with a progress bar to display the start-up progress. It is also essential to indicate the branding and getting the user trust, So CSIR-CEERI logo is displayed in the splash screen.

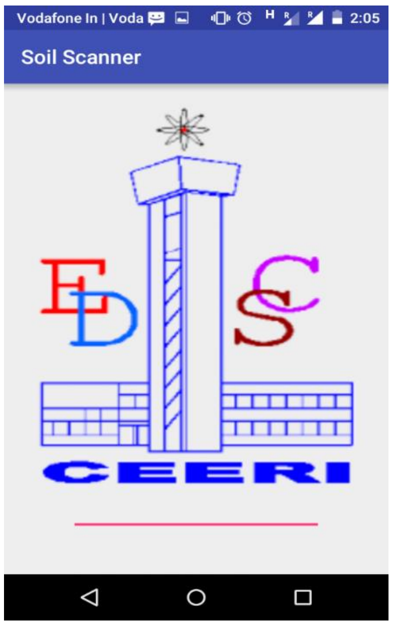


Figure 11 Splash activity

5.1.2 Getting user information

Getting user credential is very essential for a developer to get a statistic of number of people actively using the app and add more features depending upon the geographical location of users. Only 4 basic credential are asked to save user time. In upcoming versions, “Village” credential will also be removed with Google location finder, to minimize the efforts by user and “Crop type” will be provided with suggestions.

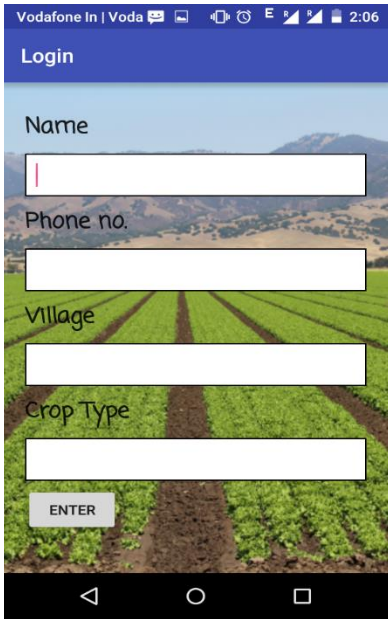


Figure 12 Login Activity

5.1.3 Getting sample image from phone

User is provided with an option to capture a new image of soil sample from their camera application, or open an existing image from gallery. User is provided with the switch to change the input methods. A lot of standard images has been used to notify the different state of activity. So that people who are not easy with android can also use this app easily. Below given image display different state of activity, first is with switch OFF and second is with switch ON and third is when image is selected respectively. Next button which was disabled till now to avoid any bugs, will get enable now. See appendix A for complete code.

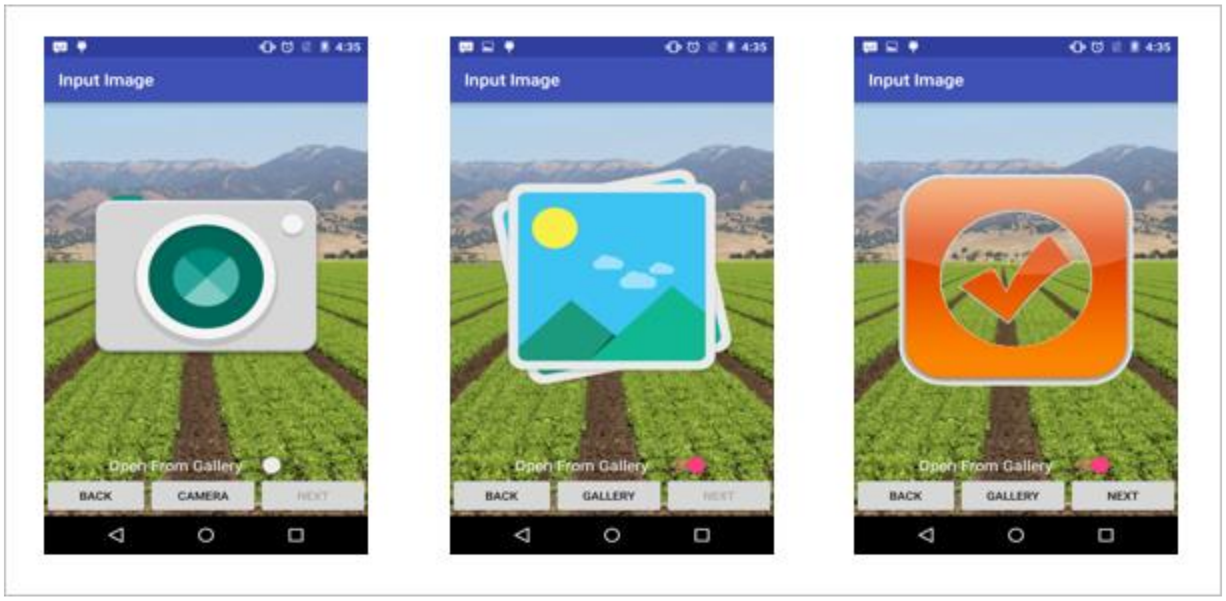


Figure 13 Three different state of input activity

5.1.4 Processing image for pH and Nutrient calculation

Very few button is used on each activity, so that user without any knowledge of it can also use it. These buttons changes there state after competition of their respective task. Below given images display the three different state of the activity for Color matching. Data for 50 different colors in RGB format with their respective pH values is stored in app. Color matching is done is CIELAB Color space which is specially designed to approximate human vision. According to Wikipedia: A Lab Color space is a Color-opponent space with dimension L for lightness and a and b for the Color-opponent dimensions, based on nonlinearly compressed (e.g. CIE XYZ) coordinates.

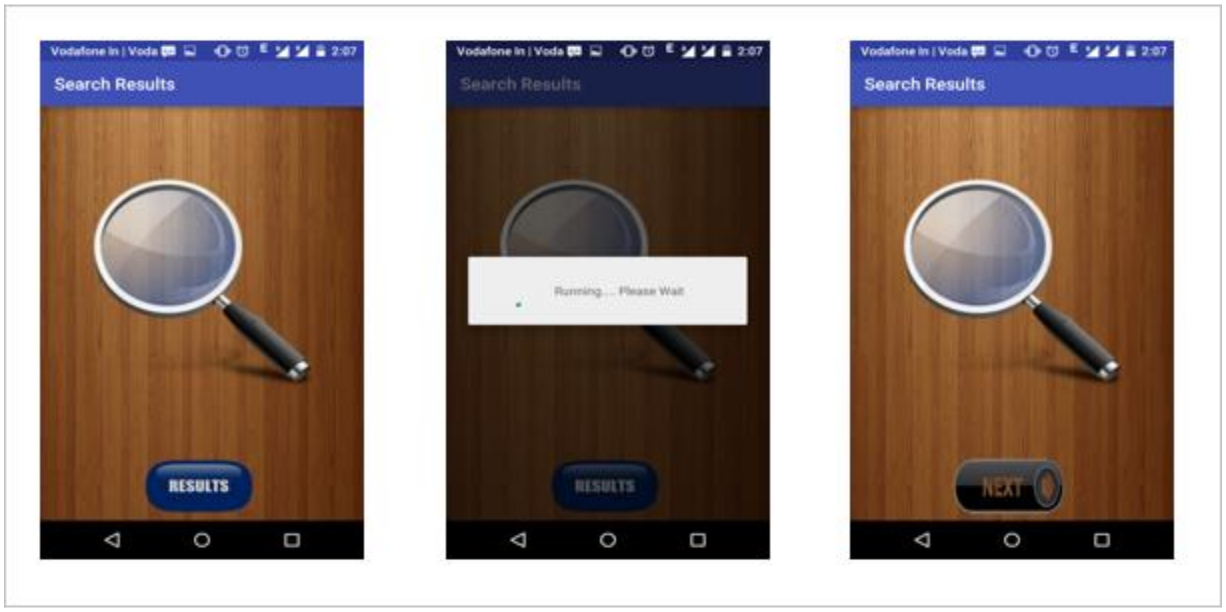


Figure 14 Different states of processing activity

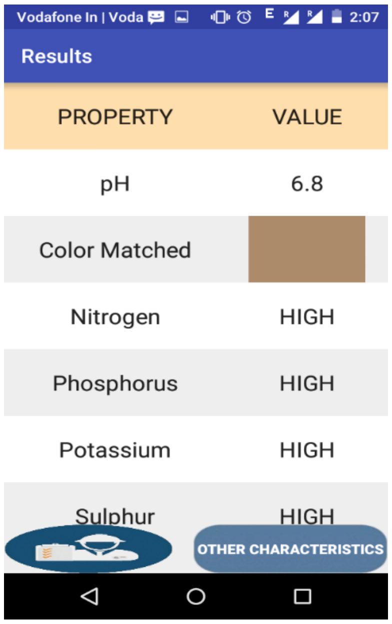
Following algorithm is used for Color matching.

1. Average intensity values for R, G, and B channels is calculated.
2. Average RGB values from image and trained data is converted into CIELAB Color space.
3. Euclidean distance between average value and every point from trained data is calculated.
4. Index of point having minimum distance is stored in a variable.
5. This index represent the best matched colour out of those 50 colours.
6. This index is displayed on the screen for a few seconds and also transferred to next activity to display the result

See appendix B for complete code of this activity.

5.1.5 Displaying the results

Results are displayed very simple table format which display pH value of the closest Color matched, Color matched and quantity of essential nutrients in categorical format with three categories high, medium, low.



PROPERTY	VALUE
pH	6.8
Color Matched	🟫
Nitrogen	HIGH
Phosphorus	HIGH
Potassium	HIGH
Sulphur	HIGH

Figure 15 Results of Processing

5.1.6 Verifying the results manually

On left-bottom a Button for manually verifying the results is provided. On pressing Manual-verification button, activity with two strips on top and bottom representing average RGB will be displayed along with the 50 trained value in a scrollable format. Index of Color matched will be highlighted through black Color. By scrolling user can easily find out if there is any other colour which best match the average RGB values or not.

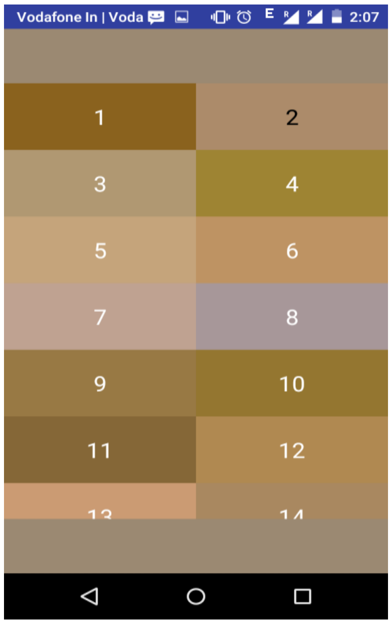
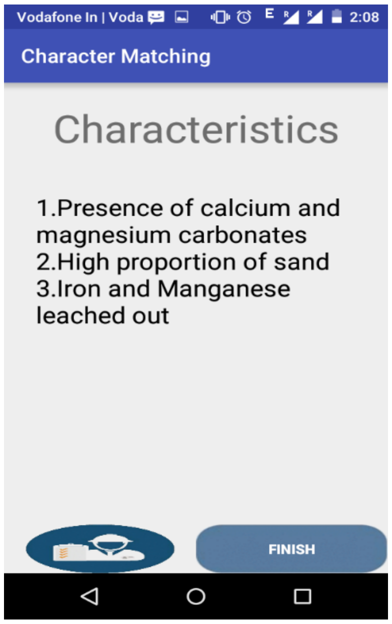


Figure 16 Activity to verify color matched manually

5.1.7 Processing image for characteristic matching

On right-bottom another button is provided to calculate the other characteristic. On pressing it, Color matching will be performed again but with a new data base and the result will be shown as given and On pressing finish user will be transferred back to login screen.



Characteristics

1.Presence of calcium and magnesium carbonates
2.High proportion of sand
3.Iron and Manganese leached out

Figure 17 Activity to display result of character matching

5.2 Fruit Analyser

It is an app to inspect the quality of fruits and vegetables. It can be used for any kind of fruits or vegetable given its quality can be checked through an image only. It is not pre-trained for every kind of fruit but an option is provided to train it. It contains a database only for tomato by default. For training the app, take image of different sample you have, calculate their result through the app and save them in the app itself. Put an optimum threshold by analysing the change in colour of fruit by changing the proportion of different channel (channel a and b in CIELAB space).

Complete process of analysing the fruit is divided into following steps.

1. Getting a sample image from phone.
2. Doing Background subtraction
3. Calculating of percentage of colours in RGB channel.
4. ~~Displaying results*~~
5. Finding spots and over ripened areas.

Note: *Collection of data is still in progress, so results activity is disabled till data collection get completed.

5.2.1 Getting a sample image

Code for inputting an image is same as that for previous app. Check appendix A and section 5.1.1. for it.

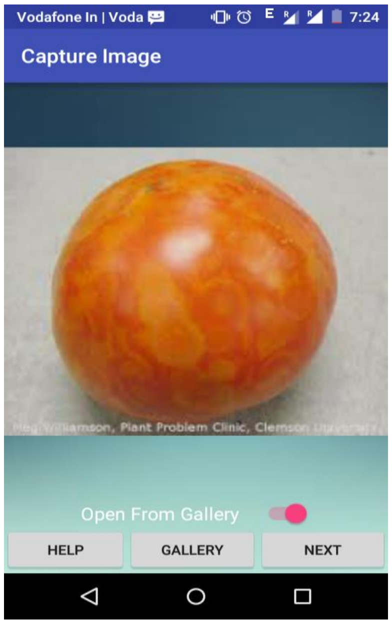


Figure 18 Activity to capture image

5.2.2 Doing Background Subtraction

Background subtraction is done to extract useful feature of image and to remove redundant information as well. It is essential to have a uniform background to do background subtraction because it is not a real time background subtraction. See appendix C for complete code.

Following are the steps to do background subtraction.

1. Image is converted into HSV colour space
2. Image is split into H,S and V channel
3. Histogram of each channel is calculated.
4. Frequencies below a optimum threshold are removed
5. Remaining information is stored as new image.

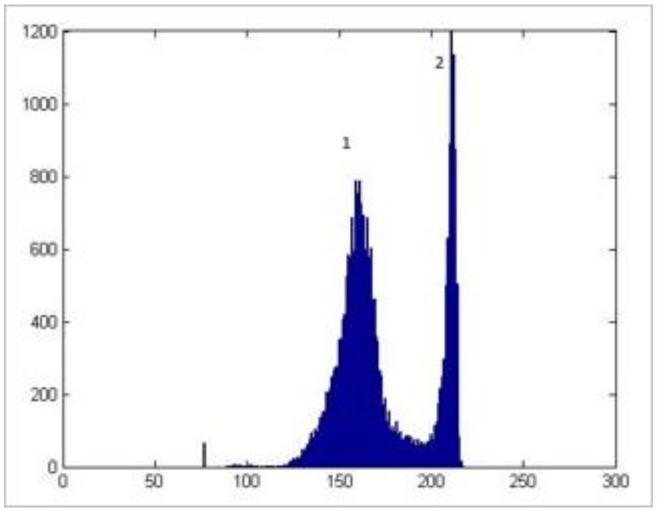


Figure 19 Histogram of object with background

Activity is provided with invert image switch so that if image is not focused (background cover more area of image) result will be wrong. It is preferred to ON that switch to get desired result.

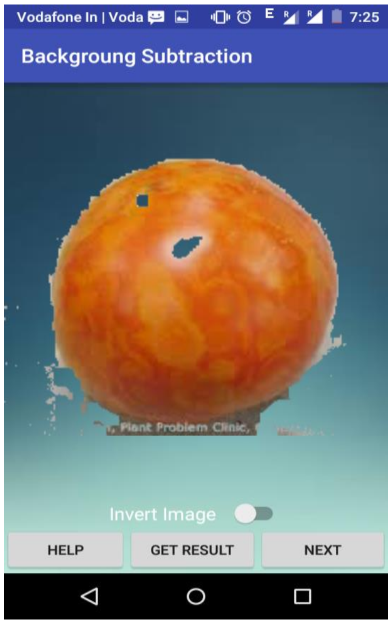


Figure 20 Activity to subtract background

5.2.3 Calculating RGB percentage

RGB percentage means the percentage of red, green and blue colour in respective channel.

Following algorithm is used to calculate the percentages

1. Image is splatted into R, G and B channel
2. Mean intensity of channel is calculated.
3. Intensities above that mean value are assigned 1 and others are assigned 0.
4. Percentage of 1 in each channel is displayed as their percentage.

These percentage are used to find ripening index of tomato.

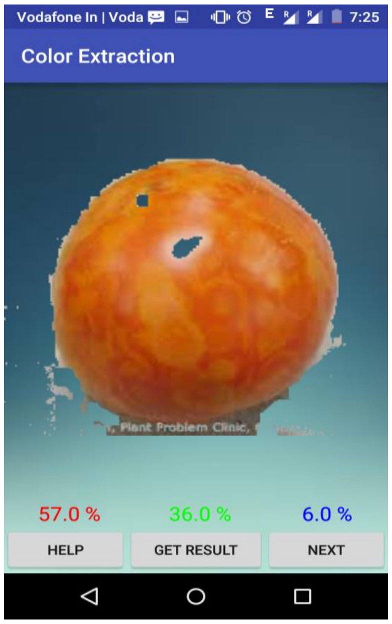


Figure 21 Activity to get RGB percentages

5.2.4 Finding spots and over ripened areas

Image after background subtraction is used to detect spot and over ripen areas. For this standard edge detection algorithm is used called “canny edge detection”. Canny edge detection determine the edges based on the change in gradient of grey scale image. Region with change in gradient above a provided threshold will be treated as edge or strong edge. This algorithm need one additional threshold to determine other edges or usually called weak edges. These are the edges in which change in gradient is small but are directly connected to strong edges. See appendix D for complete code.

Following are the steps to do canny edge detection.

1. Convert RGB image into grey scale
2. Blur it minimise noise.
3. Calculate an optimum threshold from mean and standard deviation of image
4. Use standard canny edge detection opencv function with calculated threshold

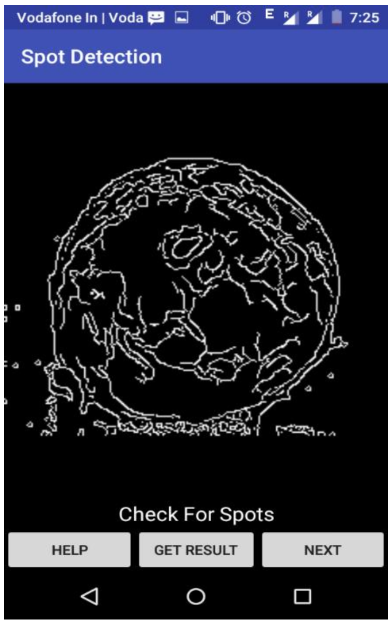


Figure 22 Activity to detect spots and over ripen areas

6 Conclusions

Image processing and android are the most rapidly growing technologies. Image processing has application in various aspects of business and day to day to life. But this technology is still limited to businesses because of unavailability of high computational power devices to individuals. But increasing smartphone market is empowering the individuals with high power devices. Hence development of app based on image processing will help individuals to make their life easier in many spheres. One of that sphere is Agriculture, where quality inspection is primary need weather it is soil or fruits and considering the situation of farmers in India it is the need of the hour.

A lot of research has been done in past to correlate properties of agricultural product with their image features. The data from these research when combined with the power of Android and Image processing can generate miraculous results for farmers as well as for society.

7 Appendices

7.1 Appendix A

Code to Get a sample Image

```
package com.example.bharatarora.cvtest1;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import android.support.v7.app.ActionBarActivity;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.ImageView;
import android.widget.Switch;
import android.widget.Toast;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;


public class InputImage extends ActionBarActivity implements View.OnClickListener {

    protected static final int CAMERA_REQUEST = 100;
    protected static final int SELECT_PICTURE = 1;
    protected Uri selectedImageUri;
    public static File mediaStorageDir;
    protected Uri fileUri;
    protected static String selectedImagePath;
    protected File OutputFile = null;
    Boolean switchStatus=false;

    ImageView input_imageView;
    Button input_button_next, input_button_back, input_button_result;
    Switch input_switch;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.input);
        input_initialise();
        input_setToListen();
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.input_button_back:
                Intent help_activity=new Intent(InputImage.this,InputImage_help.class);
                startActivity(help_activity);
                break;

            case R.id.input_button_next:
                input_next_activity();
                break;

            case R.id.input_button_reset:
                if(switchStatus)
                    open_gallery();
                else
                    open_camera();
                break;
        }
    }

    private void input_initialise() {
        input_button_back = (Button) findViewById(R.id.input_button_back);
        input_button_next = (Button) findViewById(R.id.input_button_next);
        input_button_result = (Button) findViewById(R.id.input_button_reset);
        input_switch =(Switch)findViewById(R.id.InputImage_switch);
        input_imageView = (ImageView) findViewById(R.id.input_imageView);
        input_button_next.setEnabled(false);
    }

    private void input_setToListen() {
        input_button_back.setOnClickListener(this);
        input_button_next.setOnClickListener(this);
        input_button_result.setOnClickListener(this);
        input_switch.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
                if(isChecked){
                    switchStatus=true;
                    input_imageView.setImageDrawable(getResources().getDrawable(R.drawable.gallery));
                    input_button_result.setText("GALLERY");
                }else {
                    switchStatus=false;
                    input_imageView.setImageDrawable(getResources().getDrawable(R.drawable.camera));
                    input_button_result.setText("CAMERA");
                }
            }
        });
    }

    private void input_next_activity() {
        Intent activity = new Intent(InputImage.this, BackSubtraction.class);
        startActivity(activity);
    }

    public String getPath(Uri uri) {
        String[] projection = { MediaStore.Images.Media.DATA };
        Cursor cursor = managedQuery(uri, projection, null, null, null);
        int column_index = cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
        cursor.moveToFirst();
        return cursor.getString(column_index);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        try {
            if (requestCode == CAMERA_REQUEST) {
                if (resultCode == RESULT_OK) {
                    Toast.makeText(getApplicationContext(), "done", Toast.LENGTH_LONG).show();
                    Bitmap photo = MediaStore.Images.Media.getBitmap(this.getContentResolver(), fileUri);
                    input_imageView.setImageBitmap(photo);

                    input_button_result.setEnabled(true);
                    input_button_next.setEnabled(true);
                }
            }
            if (requestCode == SELECT_PICTURE) {
                if (resultCode == RESULT_OK) {
                    selectedImageUri = data.getData();
                    selectedImagePath = getPath(selectedImageUri);
                    input_imageView.setImageURI(selectedImageUri);
                    new loadStuffFromGallery().execute(selectedImagePath);

                    input_button_result.setEnabled(true);
                    input_button_next.setEnabled(true);
                }
            }
        } catch (NullPointerException e) {
            Log.d("Async","Failed");
            e.printStackTrace();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void open_gallery(){
        Toast.makeText(getApplicationContext(), "Please open through Gallery", Toast.LENGTH_LONG).show();
        try {
            Intent galleryIntent = new Intent(Intent.ACTION_PICK, MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
            startActivityForResult(galleryIntent,SELECT_PICTURE);
        } catch (Exception e) {
            Log.d("Just exception","failed");
        }
    }

    public void open_camera(){
        Toast.makeText(getApplicationContext(), "Camera Started", Toast.LENGTH_LONG).show();
        try {
            Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
            fileUri=Uri.fromFile(getOutputMediaFile_Camera());
            cameraIntent.putExtra(MediaStore.EXTRA_OUTPUT, fileUri);
            startActivityForResult(cameraIntent, CAMERA_REQUEST);
        } catch (RuntimeException e) {
            e.printStackTrace();
        }
    }



    public static File getOutputMediaFile_Camera() {
        mediaStorageDir = new File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES), "MyCameraApp");
        if (!mediaStorageDir.exists()) {
            if (!mediaStorageDir.mkdir()) {
                Log.d("MyCameraApp", "Failed to create");
            }
        }
        File mediaFile = new File(mediaStorageDir.getPath() + File.separator + "input.jpg");
        return mediaFile;
    }



    public class loadStuffFromGallery extends AsyncTask<String,Integer,Void>{

        ProgressDialog progressDialog;
        protected void onPreExecute(){
            progressDialog=new ProgressDialog(InputImage.this);
            progressDialog.setMax(100);
            progressDialog.setProgress(0);
            progressDialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
            progressDialog.show();
            progressDialog.setCancelable(true);
        }


        @Override
        protected Void doInBackground(String... params) {

            new Thread(new Runnable() {
                @Override
                public void run() {
                    for(int i=0;i<20;i++) {
                        try {
                            Thread.sleep(100);
                        } catch (InterruptedException e) {
                            Log.d("Interrupted", "failed");
                        }
                        publishProgress(5);
                    }
                }
            });

            //taking input to copy
            File inputFile = new File(params[0]);
            FileInputStream fileInputStream = null;
            try {
                fileInputStream = new FileInputStream(inputFile);
            } catch (FileNotFoundException e) {
                Log.d("input", "failed:  file not found exception");
            }

            //setting output location
            mediaStorageDir = new File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES), "MyCameraApp");
            if (!mediaStorageDir.exists()) {
                if (!mediaStorageDir.mkdir()) {
                    Log.d("MyCameraApp", "Failed to create");
                }
            }

            OutputFile = new File(mediaStorageDir.getPath() + File.separator + "input"+ ".jpg");
            FileOutputStream fileOutputStream = null;
            try {
                fileOutputStream = new FileOutputStream(OutputFile);
            } catch (FileNotFoundException e) {
                Log.d("output", "failed: unable to copy image");
            }

            //copying the file
            byte[] array= new byte[0];
            try {
                array = new byte[fileInputStream.available()];
            } catch (IOException e) {
                e.printStackTrace();
            }
            try {
                while ((fileInputStream.read(array)) != -1){
                    fileOutputStream.write(array);
                }


            } catch (IOException e) {
                Log.d("writing", "failed");
            } catch (NullPointerException e) {
                Log.d("writing", "failed");
            }

            fileUri=Uri.fromFile(OutputFile);
            return null;
        }

        protected void onProgressUpdate(Integer... progress){
            progressDialog.incrementProgressBy(progress[0]);
```

```
            }

            protected void onPostExecute(Void result){
                progressDialog.incrementProgressBy(100 - progressDialog.getProgress());
                progressDialog.dismiss();

            }
        }

        @Override
        protected void onPause() {
            super.onPause();
        }
        @Override
        protected void onDestroy() {
            super.onDestroy();
        }
        @Override
        protected void onStart() {
            super.onStart();
        }
        @Override
        protected void onResume() {
            super.onResume();
        }
        @Override
        protected void onStop() {
            super.onStop();
        }
        @Override
        protected void onRestart() {
            super.onRestart();
        }
    }
```

7.2 Appendix B

Code for colour matching

```
package com.example.bharatarora.soilscanner;

import android.annotation.TargetApi;
import android.app.NotificationManager;
import android.app.ProgressDialog;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.support.v7.app.ActionBarActivity;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;

import org.opencv.core.Mat;
import org.opencv.core.Size;
import org.opencv.highgui.Highgui;
import org.opencv.imgproc.Imgproc;

import java.io.File;

public class Color_Matching extends ActionBarActivity implements View.OnClickListener {

    public Bitmap matching_bitmap;
    int min_position;
    int toast_colorMatched=0;
    double min_distance=0.0;
    float[] cielab_mean=new float[3];
    int r_mean;
    int g_mean;
    int b_mean;

    int r_min;
    int g_min;
    int b_min;

    private static Boolean resultStatus=false;

    public double[] ph={
            7.05, 6.80, 6.63, 6.64, 8.35, 7.35,
            7.35, 7.49, 7.38, 7.40, 7.25, 7.30,
            7.83, 7.59, 7.51, 7.20, 7.36, 7.35,
            7.38, 7.44, 7.68, 7.57, 7.50, 7.22,
            7.96, 7.99, 7.99, 6.90, 7.56, 6.75,
            6.63, 6.43, 7.90, 7.04, 6.90, 6.48,
            5.52, 6.96, 6.62, 6.42, 6.80, 7.09,
            5.58, 5.52, 6.06, 6.53, 6.58, 6.50,
            6.45, 7.24};

     public int[][] rgb_array={
            {138,98,30},{172,139,106},{176,152,114},{158,132,51},{197,164,123},{190,147,99},
            {191,162,145},{167,151,153},{152,121,68},{148,118,48},{133,103,55},{176,137,81},
            {203,155,115},{169,136,96},{175,134,102},{162,128,88},{162,131,72},{168,128,76},
            {156,119,66},{163,131,93},{182,146,110},{175,138,93},{207,186,157},{186,144,94},
            {227,209,173},{187,155,117},{226,186,125},{196,146,109},{203,168,140},{155,137,89},
            {157,134,80},{179,129,67},{230,181,123},{208,161,119},{196,150,101},{176,139,69},
            {182,136,50},{128,105,27},{185,155,91},{152,122,52},{229,210,152},{255,220,162},
            {176,169,123},{173,128,43},{169,145,85},{189,164,110},{167,154,99},{186,154,97},
            {164,160,113},{158,150,111}};
```

```
ImageView matching_imageVIew;
Button matching_result;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.color_matching);
    matching_initialise();
    matching_result.setOnClickListener(this);
}

public void matching_initialise(){
    matching_imageVIew =(ImageView)findViewById(R.id.matching_imageView);
    matching_result = (Button)findViewById(R.id.matching_button_result);
    try{
        matching_bitmap = BitmapFactory.decodeFile(InputImage.mediaStorageDir.getPath() + File.separator + "input"+ ".jpg");
    }catch (Exception e){
        e.printStackTrace();
    }
}

public void matching_algo(){
    new doColorMatching().execute();
}

@TargetApi(Build.VERSION_CODES.JELLY_BEAN)
@Override
public void onClick(View v) {
    switch (v.getId()){
        case R.id.matching_button_result:
            if(!resultStatus)
                matching_algo();
            else{
                Bundle bundle=new Bundle();
                bundle.putDouble("ph",ph[min_position]);
                bundle.putInt("min_position",min_position);
                bundle.putInt("mean_color",Color.argb( Color.alpha(matching_bitmap.getPixel(0,0)),r_mean,g_mean,b_mean));
                bundle.putInt("matched_color",Color.argb( Color.alpha(matching_bitmap.getPixel(0,0)),r_min,g_min,b_min));
                Intent next_intent=new Intent(Color_Matching.this,Results.class);
                next_intent.putExtras(bundle);
                startActivity(next_intent);
            }

            break;

    }
}

public class doColorMatching extends AsyncTask<Void,Void,Void>{

    ProgressDialog progressDialog;

    protected void onPreExecute(){
        progressDialog=new ProgressDialog(Color_Matching.this);
        progressDialog.setMessage("Running.... Please Wait");
        progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
        progressDialog.show();
        progressDialog.setCancelable(false);
    }

    @Override
    protected Void doInBackground(Void... params) {

        Mat input= Highgui.imread(InputImage.mediaStorageDir.getPath()+ File.separator + "input"+ ".jpg");
        Mat im_blur=new Mat();
        Imgproc.GaussianBlur(input,im_blur,new Size(9,9),10.0);
        Highgui.imwrite(InputImage.mediaStorageDir.getPath() + File.separator + "im_blur"+".jpg",im_blur);
        try{
            matching_bitmap = BitmapFactory.decodeFile(InputImage.mediaStorageDir.getPath() + File.separator + "im_blur"+ ".jpg");
        }catch (Exception e){
            e.printStackTrace();
        }
        long r_sum=0;
        long b_sum=0;
        long g_sum=0;

        long total=matching_bitmap.getWidth()*matching_bitmap.getHeight();
        for(int i=0;i<matching_bitmap.getWidth();i++){
            for(int j=0;j<matching_bitmap.getHeight();j++){
                int p=matching_bitmap.getPixel(i,j);
                int r=Color.red(p);
                int g=Color.green(p);
                int b=Color.blue(p);
                r_sum=r_sum+r;
                g_sum=g_sum+g;
                b_sum=b_sum+b;
            }
        }
        r_mean=(int)(r_sum/total);
        g_mean=(int)(g_sum/total);
        b_mean=(int)(b_sum/total);

        rgb2lab(r_mean,g_mean,b_mean,cielab_mean);

        double weighted_distance;

        float[] cielab=new float[3];
        int r0=rgb_array[0][0];
        int g0=rgb_array[0][1];
        int b0=rgb_array[0][2];
        rgb2lab(r0,g0,b0,cielab);
        min_position=0;
        min_distance = Math.pow( Math.pow(cielab[0]-cielab_mean[0],2)+Math.pow(cielab[1]-cielab_mean[1],2)+ Math.pow(cielab[2]-
cielab_mean[2],2),0.5);
        for(int i=0;i<50;i++){
            int r=rgb_array[i][0];
            int g=rgb_array[i][1];
            int b=rgb_array[i][2];
            rgb2lab(r,g,b,cielab);
            weighted_distance = Math.pow( Math.pow(cielab[0]-cielab_mean[0],2)+Math.pow(cielab[1]-cielab_mean[1],2)+ Math.pow(cielab[2]-
cielab_mean[2],2),0.5);
            if(min_distance>weighted_distance){
                min_distance=weighted_distance;
                min_position=i;
            }
        }

        r_min=rgb_array[min_position][0];
        g_min=rgb_array[min_position][1];
        b_min=rgb_array[min_position][2];

        return null;
    }

    public void rgb2lab(int R, int G, int B, float[] lab) {
        //http://www.brucelindbloom.com

        float r, g, b; //input rgb in range of [0,1]
        float  X, Y, Z; //output XYZ

        // RGB to XYZ
        r = R/255.f; //R 0..1
        g = G/255.f; //G 0..1
        b = B/255.f; //B 0..1

        // assuming sRGB (D65)
        if (r <= 0.04045)
```

```
            r = r/12.92f;
        else
            r = (float) Math.pow((r+0.055)/1.055,2.4);

        if (g <= 0.04045)
            g = g/12.92f;
        else
            g = (float) Math.pow((g+0.055)/1.055,2.4);

        if (b <= 0.04045)
            b = b/12.92f;
        else
            b = (float) Math.pow((b+0.055)/1.055,2.4);


        X =  0.412453f*r     + 0.357580f*g + 0.180423f*b;
        Y =  0.212671f*r     + 0.715160f*g + 0.072169f*b;
        Z =  0.019334f*r     + 0.119193f*g + 0.950227f*b;

        float xr, yr, zr;
        float Xr = 0.95047f;  // reference white D50
        float Yr = 1.0000f;
        float Zr = 1.08883f;
        float eps = 216.f/24389.f;
        float fx, fy, fz;
        float k = 24389.f/27.f;

        // XYZ to Lab
        xr = X/Xr;
        yr = Y/Yr;
        zr = Z/Zr;

        if ( xr > eps )
            fx =  (float) Math.pow(xr, 1/3.);
        else
            fx = (float) ((k * xr + 16.) / 116.);

        if ( yr > eps )
            fy =  (float) Math.pow(yr, 1/3.);
        else
            fy = (float) ((k * yr + 16.) / 116.);

        if ( zr > eps )
            fz =  (float) Math.pow(zr, 1/3.);
        else
            fz = (float) ((k * zr + 16.) / 116);

        float Ls, as, bs;

        if(yr > eps)
            Ls = ( 116 * fy ) - 16;
        else
            Ls=903.3f*fy;

        as = 500*(fx-fy);
        bs = 200*(fy-fz);

        lab[0] = Ls;
        lab[1] = as;
        lab[2] = bs;
    }

    @TargetApi(Build.VERSION_CODES.JELLY_BEAN)
    protected void onPostExecute(Void result){
        progressDialog.dismiss();
        resultStatus=true;
        matching_result.setBackground(getResources().getDrawable(R.drawable.ne));
        toast_colorMatched=min_position+1;
        Toast.makeText(getApplicationContext(),"Color Number : "+toast_colorMatched,Toast.LENGTH_LONG).show();
    }
    }

    @TargetApi(Build.VERSION_CODES.JELLY_BEAN)
    @Override
    protected void onResume() {
        super.onResume();
        matching_result.setBackground(getResources().getDrawable(R.drawable.result1));
        resultStatus=false;
    }
}
```

7.3 Appendix C

Code for Background subtraction

```
package com.example.bharatarora.cvtest1;

import android.app.ProgressDialog;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Environment;
import android.support.v7.app.ActionBarActivity;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.ImageView;
import android.widget.Switch;

import org.opencv.android.OpenCVLoader;
import org.opencv.core.Core;
import org.opencv.core.CvType;
import org.opencv.core.Mat;
import org.opencv.core.MatOfFloat;
import org.opencv.core.MatOfInt;
import org.opencv.core.Point;
import org.opencv.core.Scalar;
import org.opencv.core.Size;
import org.opencv.highgui.Highgui;
import org.opencv.imgproc.Imgproc;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class BackSubtraction extends ActionBarActivity implements View.OnClickListener{


    Bitmap bitmap,operation;
    Boolean switchStatus=false;

    Button BackSubtraction_back,BackSubtraction_result,BackSubtraction_next;
    ImageView BackSubtraction_imageView;
    Switch BackSubtraction_switch;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.back_subtraction);
        if (!OpenCVLoader.initDebug()) {
            Log.e(this.getClass().getSimpleName(), "  OpenCVLoader.initDebug(), not working.");
        } else {
            Log.d(this.getClass().getSimpleName(), "  OpenCVLoader.initDebug(), working.");
        }
        BackSubtraction_initialise();
        BackSubtraction_setToLiten();

    }

    public void BackSubtraction_initialise(){
        BackSubtraction_back = (Button)findViewById(R.id.BackSubtraction_button_back);
        BackSubtraction_result = (Button)findViewById(R.id.BackSubtraction_button_result);
        BackSubtraction_next= (Button)findViewById(R.id.BackSubtraction_button_next);
        BackSubtraction_imageView=(ImageView)findViewById(R.id.BackSubtraction_imageView);
        BackSubtraction_switch = (Switch)findViewById(R.id.BackSubtraction_switch);
        BackSubtraction_next.setEnabled(false);

    }

    public void BackSubtraction_setToLiten(){
        BackSubtraction_next.setOnClickListener(this);
        BackSubtraction_back.setOnClickListener(this);
        BackSubtraction_result.setOnClickListener(this);
        BackSubtraction_switch.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
                if(isChecked){
                    switchStatus=true;
                }else {
                    switchStatus=false;
                }
            }
        });
        BackSubtraction_switch.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                BackSubtraction_switch.setChecked(switchStatus);
            }
        });
    }

    public void doSubtraction(){
        new doSubtraction().execute(switchStatus);
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()){
            case R.id.BackSubtraction_button_back:
                Intent help_activity=new Intent(BackSubtraction.this,BackSubtraction_help.class);
                startActivity(help_activity);
                break;
            case R.id.BackSubtraction_button_result:
                doSubtraction();
                BackSubtraction_next.setEnabled(true);
                break;
            case R.id.BackSubtraction_button_next:
                Intent next_activity = new Intent(BackSubtraction.this, Extract.class);
                startActivity(next_activity);
                break;
        }
    }

    public class doSubtraction extends AsyncTask<Boolean,Void,Void>{

        ProgressDialog progressDialog;
        protected void onPreExecute(){
            progressDialog=new ProgressDialog(BackSubtraction.this);
            progressDialog.setIndeterminate(true);
            progressDialog.setMessage("Running.... Please Wait");
            progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
            progressDialog.show();
            progressDialog.setCancelable(true);

        }

        @Override
        protected Void doInBackground(Boolean... params) {
            Mat image = Highgui.imread(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES).getPath()+
File.separator+"MyCameraApp"+ File.separator +"input.jpg");
            List<Mat> hsvPlanes=new ArrayList<>();
            Mat hsvImg= new Mat();
            Mat thresholdImg=new Mat();

            hsvImg.create(image.size(), CvType.CV_8U);

            Imgproc.cvtColor(image,hsvImg,Imgproc.COLOR_BGR2HSV);
            Core.split(hsvImg,hsvPlanes);

            List<Mat> hue = new ArrayList<>();
            Mat hist_hue =new Mat();
            MatOfInt histSize=new MatOfInt(180);
            double average=0.0;
            hue.add(hsvPlanes.get(0));
            Imgproc.calcHist(hue,new MatOfInt(0),new Mat(),hist_hue,histSize,new MatOfFloat(0,179));
            for (int h=0;h<180;h++){
                average+=(hist_hue.get(h,0)[0]*h);
            }
            double threshValue = average / hsvImg.size().height / hsvImg.size().width;
            if(params[0]==true)
                Imgproc.threshold(hsvPlanes.get(0),thresholdImg,threshValue,179.0,Imgproc.THRESH_BINARY);
            else
                Imgproc.threshold(hsvPlanes.get(0),thresholdImg,threshValue,179.0,Imgproc.THRESH_BINARY_INV);

            Imgproc.blur(thresholdImg,thresholdImg,new Size(5,5));
            Imgproc.dilate(thresholdImg,thresholdImg,new Mat(),new Point(-1,-1),1);
            Imgproc.erode(thresholdImg,thresholdImg,new Mat(),new Point(-1,-1),3);
            Imgproc.threshold(thresholdImg,thresholdImg,threshValue,179.0,Imgproc.THRESH_BINARY);
            Mat foreground=new Mat(image.size(),CvType.CV_8UC3,new Scalar(255,255,255));
            image.copyTo(foreground,thresholdImg);

Highgui.imwrite(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES).getPath()+File.separator+"MyCameraApp"+
File.separator +"foreground.jpg", foreground);

            try {
                bitmap = BitmapFactory.decodeFile(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES) +
File.separator+"MyCameraApp"+ File.separator +"foreground.jpg");
            } catch (Exception e) {
                e.printStackTrace();
            }
            operation= Bitmap.createBitmap(bitmap.getWidth(),bitmap.getHeight(),bitmap.getConfig());
            for(int i=0; i<bitmap.getWidth(); i++) {
                for (int j = 0; j < bitmap.getHeight(); j++) {
                    int p = bitmap.getPixel(i, j);
                    int r = Color.red(p);
                    int g = Color.green(p);
                    int b = Color.blue(p);
                    int alpha = Color.alpha(p);
                    if(r>170&&g>170&&b>170)
                        operation.setPixel(i, j, Color.argb(1, 0, 0, 0));
                    else
                        operation.setPixel(i, j, Color.argb(alpha, r,g,b));
                }
            }
            FileOutputStream fileOutputStream=null;
            try{
                fileOutputStream=new FileOutputStream(InputImage.mediaStorageDir.getPath() + File.separator + "fore"+ ".png");
                operation.compress(Bitmap.CompressFormat.PNG,100,fileOutputStream);
            }catch (IOException e){
                e.printStackTrace();
            }finally {
                if(fileOutputStream!=null){
                    try {
                        fileOutputStream.close();
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                }
            }
```

```
            return null;
        }
        protected void onProgressUpdate(Void... progress){
        }

        protected void onPostExecute(Void result){
            progressDialog.incrementProgressBy(100 - progressDialog.getProgress());
            progressDialog.dismiss();
            BackSubtraction_imageView.setImageBitmap(operation);
        }
    }

    @Override
    protected void onPause() {
        super.onPause();
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
    }
    @Override
    protected void onStart() {
        super.onStart();
    }
    @Override
    protected void onResume() {
        super.onResume();
    }
    @Override
    protected void onStop() {
        super.onStop();
    }
    @Override
    protected void onRestart() {
        super.onRestart();
    }
}
```

7.4 Appendix D

Code for edge detection.

```
package com.example.bharatarora.cvtest1;

import android.app.ProgressDialog;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Environment;
import android.support.v7.app.ActionBarActivity;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;

import org.opencv.android.OpenCVLoader;
import org.opencv.android.Utils;
import org.opencv.core.CvException;
import org.opencv.core.Mat;
import org.opencv.core.Size;
import org.opencv.highgui.Highgui;
import org.opencv.imgproc.Imgproc;
import java.io.File;

public class EdgeDetection extends ActionBarActivity implements View.OnClickListener{

    int threshold1 = 70;
    int threshold2 = 100;
    Bitmap bitmap;
    int count=0;
    long mat_face;
    int mat_channel;
    int mean=0;

    ImageView edgeDetection_imageView;
    Button edgeDetection_back,edgeDetection_result,edgeDetection_next;
    TextView edgeDetection_textView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.edge_detection);

        if (!OpenCVLoader.initDebug()) {
            Log.e(this.getClass().getSimpleName(), "  OpenCVLoader.initDebug(), not working.");
        } else {
            Log.d(this.getClass().getSimpleName(), "  OpenCVLoader.initDebug(), working.");
        }
        edgeDetection_initialise();
        setEdgeDetection_setListen();
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()){
            case R.id.edgeDetection_button_back:
                Intent help_activity=new Intent(EdgeDetection.this,EdgeDtetection_help.class);
                startActivity(help_activity);
                break;
            case R.id.edgeDetection_button_next:
                Intent next_activity=new Intent(EdgeDetection.this,Results.class);
                startActivity(next_activity);
                break;
            case R.id.edgeDetection_button_result:
                doEdgeDetection();
                edgeDetection_next.setEnabled(true);
                break;
        }
    }

    public void edgeDetection_initialise(){
        edgeDetection_textView =(TextView) findViewById(R.id.edgeDetection_textView);
        edgeDetection_back = (Button)findViewById(R.id.edgeDetection_button_back);
        edgeDetection_next= (Button)findViewById(R.id.edgeDetection_button_next);
        edgeDetection_result=(Button)findViewById((R.id.edgeDetection_button_result));
        edgeDetection_imageView = (ImageView)findViewById(R.id.edgeDetection_imageView);

        edgeDetection_next.setEnabled(false);
    }

    public void setEdgeDetection_setListen(){
        edgeDetection_next.setOnClickListener(this);
        edgeDetection_back.setOnClickListener(this);
        edgeDetection_result.setOnClickListener(this);
    }

    public void doEdgeDetection(){
        new doEdgeDetectionClass().execute();
    }

    public class doEdgeDetectionClass extends AsyncTask<Void,Void,Bitmap>{

        ProgressDialog progressDialog;
        protected void onPreExecute(){
            progressDialog=new ProgressDialog(EdgeDetection.this);
            progressDialog.setMessage("Running.... Please Wait");
            progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
            progressDialog.show();
            progressDialog.setCancelable(true);

        }

        @Override
        protected Bitmap doInBackground(Void... params) {
            Mat image =
Highgui.imread(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES).getPath()+File.separator+"MyCameraApp"+
File.separator +"fore.png");
            Mat im_gray = new Mat();
            Imgproc.cvtColor(image,im_gray,Imgproc.COLOR_BGR2GRAY);
            Mat im_blur = new Mat();
            Imgproc.blur(im_gray,im_blur,new Size(3,3));
            Mat im_canny =new Mat();

            byte[] buff=new byte[(int)im_blur.total()*im_blur.channels()];
            im_blur.get(0,0,buff);
            long sum=0;

            mat_face=im_blur.total();
            mat_channel=im_blur.channels();
            count=(int)mat_face*mat_channel;
            for (int i=0;i<(int)mat_face*mat_channel;i++){
                if((int)buff[i]==0)
                    count=count-1 ;

                sum=sum +(long)buff[i];
            }
            mean=(int)(sum/(long)count);
            threshold1=(int) (mean);
            threshold2=(int)(mean*4);
            Imgproc.Canny(im_blur, im_canny, threshold1, threshold2);

Highgui.imwrite(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES).getPath()+File.separator+"MyCameraApp"+
File.separator +"fore_canny.jpg", im_canny);
            bitmap = Bitmap.createBitmap(image.width(),image.height(), Bitmap.Config.ARGB_8888);
            try{
                Utils.matToBitmap(im_canny,bitmap);
            }catch (CvException e){
                e.printStackTrace();
            }
            return bitmap;
        }

        protected void onPostExecute(Bitmap result){
            edgeDetection_imageView.setImageBitmap(result);
            progressDialog.dismiss();
        }
    }

    @Override
    protected void onPause() {
        super.onPause();
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
    }
    @Override
    protected void onStart() {
        super.onStart();
    }
    @Override
    protected void onResume() {
        super.onResume();
    }
    @Override
    protected void onStop() {
        super.onStop();
    }
    @Override
    protected void onRestart() {
        super.onRestart();
    }
}
```

8 References

1. http://docs.opencv.org/java/2.4.9/
2. https://docs.oracle.com/javase/7/docs/api/overview-summary.html
3. https://developer.android.com/index.html
4. http://www.easyrgb.com/
5. opencv.org
6. www.google.com
7. www.javatpoint.com/android-tutorial
8. www.mybringback.com
9. www.stackoverflow.com
10. www.tutorialspoint.com/dip/
11. www.udacity.com
12. www.wikipedia.com

9 Glossary

HSV – Hue saturation and Value

Channel – Image with single intensity component of image.

CIELAB - space- It is Color space to approximate Human Vision.

Histogram - A 2–D graph representing frequency of each intensity level of particular channel.