

AGRIPHONE

TEAM: 20

**MADHURI GADHAVE
ASHWINI JOGDAND
POURNIMA AMBAVALE**

TABLE OF CONTENTS

1. Introduction.....	
2. Problem Statement.....	
3. Requirements.....	
4. Implementation.....	
5. Testing Strategy and Data.....	
6. Discussion of System.....	
7. Future Work.....	
8. Conclusion.....	
9. References.....	

1. INTRODUCTION:

There is need to increase agriculture products in the market as human population is increasing day by day in India. The concern of this project is to use the modern technology in agriculture to increase the product quality and quantity.

A lot of research is going on related to greenhouse automation. But all of these research focus on various parameters in greenhouse such as temperature and humidity. The current image processing practices are done for harvesting of fruits using agricultural robot. There is little concentration on image processing practices for greenhouse crop monitoring.

Therefore, this project focuses on image processing for monitoring greenhouse crop status. The monitoring of ripeness degree of tomatoes in greenhouse is first step towards the crop health status monitoring using android phone for image processing and GSM (Group Special Mobile or Global System for Mobile Communication) technology. For some critical plants such as vegetables and flowers, they need 24 hours attention from human so that the quantities and qualities of the plant can be controlled [1]. Thus, manual supervision plays important role in crop health status monitoring.

With the help of android phone this task can be automated. The human interaction in this case is reduced. In this project, we have taken a tomato fruit as case study. We consider three main stages of ripening for tomatoes. The three stages are unripened (or raw), medium ripened and fully ripened or over ripened.

2. PROBLEM STATEMENT:

- Designing and developing application for processing images of fruits on greenhouse plants to detect ripeness degree.
- Automatically capture photos and save them in the SD card for further processing.
- There will be alarm which user needs to set for capturing photos automatically.
- The phone will process photo once it get saved and then generate results accordingly.
- The results are send to user through SMS.

3. REQUIREMENTS:

Ø Hardware requirements:

The hardware used for the development of the project is:

PROCESSOR : PENTIUM IV 2.40GHz OR HIGHER

RAM : 0.99 RAM

HARD DISK : 160 GB

MOBILE DEVICE : ANDROID BASED PHONE WITH AT LEAST 2MP
CAMERA AND SD CARD

Ø Software requirements:

The software used for the development of the project is:

OPERATING SYSTEM : Windows 7, Windows Vista, Windows XP with
or later operating system (32 or 64 bit)

ECLIPSE IDE : Eclipse IDE for Java Developers

JAVA : JDK 6

ADT Plug-ins and Android SDK

4. IMPLEMENTATION DETAILS:

The Following are the prominent features of the above discussed system...

- Using android based mobile as remote monitoring device for greenhouse plants,
- Use of GSM and SIM technology to send SMS to user regarding plant status,
- Up to 20cm range,
- Automatic working of android based mobile device, once time is set.

The major tasks performed in this system are:

- Setting the alarm to capture photo at specific time of the day.
- Automatic saving and processing of images captured by phone camera.
- Generating the appropriate results from processing of images.
- Sending the SMS to the user about the result.

I. CODE FOR ALARM SERVICE TO CAPTURE PHOTO AUTOMATICALLY:

```
package com.agribot.automaticcapture;

import android.app.Service;
import android.content.Intent;
import android.hardware.Camera;
import android.os.IBinder;
import android.view.LayoutInflater;
import android.widget.Toast;

public class MyAlarmService extends Service {

    Camera camera;
    Preview preview;
    LayoutInflater inflater;

    @Override
    public void onCreate() {

        // Toast.makeText(this, "MyAlarmService.onCreate()",
        Toast.LENGTH_LONG).show();
    }

    @Override
    public IBinder onBind(Intent intent) {
        // TODO Auto-generated method stub
        Toast.makeText(this, "MyAlarmService.onBind()",
        Toast.LENGTH_LONG).show();
        return null;
    }

    @Override
    public void onDestroy() {
        // TODO Auto-generated method stub
        super.onDestroy();
        Toast.makeText(this, "MyAlarmService.onDestroy()",
        Toast.LENGTH_LONG).show();
    }

    @Override
    public void onStart(Intent intent, int startId) {

        Toast.makeText(this, "MyAlarmService.onStart()",
        Toast.LENGTH_SHORT).show();
        Intent intent1 = new Intent(getBaseContext(),
        CapturePhoto.class);
```

```

        intent1.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        getApplicationContext().startActivity(intent1);
        super.onStart(intent, startId);
    }

    @Override
    public boolean onUnbind(Intent intent) {
        // TODO Auto-generated method stub
        return super.onUnbind(intent);
    }
}

```

II. CODE FOR CAPTURING PHOTO:

```

package com.agribot.automaticcapture;

import android.app.Activity;
import android.hardware.Camera;
import android.os.Bundle;
import android.util.Log;
import android.widget.Button;
import android.widget.FrameLayout;

import com.agribot.dashboard.R;

public class CapturePhoto extends Activity{
    private static final String TAG = "CameraDemo";
    Camera camera;
    Preview preview;
    Button buttonClick;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.camerapreview);

        preview = new Preview(this);
        ((FrameLayout) findViewById(R.id.preview)).addView(preview);
        Log.d(TAG, "onCreate'd");
    }
}

```

III. CODE FOR STORING IMAGE ON THE SD CARD:

```

package com.agribot.automaticcapture;

import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.ImageView;

```

```

import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.agribot.dashboard.R;
import com.agrobot.capturephoto.AgribotCapturePhoto;

public class AfterCapture extends Activity {

    ImageView imageView;
    ProgressBar progressBar;
    TextView textView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.afterautomaticcapture);
        imageView=(ImageView) findViewById(R.id.afterimagecapture);
        progressBar=(ProgressBar) findViewById(R.id.progressBar1);
        textView=(TextView) findViewById(R.id.afterimagecapturetext);
        imageView.setVisibility(View.INVISIBLE);
        textView.setVisibility(View.INVISIBLE);
        progressBar.setVisibility(View.VISIBLE);

        System.out.println(Preview.phototakenname);

        Bitmap bitmap =
        BitmapFactory.decodeFile("/sdcard/"+Preview.phototakenname+".jpg");
        Bitmap newphoto = AgribotCapturePhoto.doGreyscale(bitmap);
        Bitmap newphoto2 = AgribotCapturePhoto.binarize(newphoto,
        bitmap);

        String a = "";
        a = AgribotCapturePhoto.findcolor(newphoto2);

        /*imageView.setLayoutParams(new LinearLayout.LayoutParams(
            LinearLayout.LayoutParams.WRAP_CONTENT,
            LinearLayout.LayoutParams.WRAP_CONTENT));

        imageView.getLayoutParams().height = 300;
        imageView.getLayoutParams().width = 300;
        imageView.setImageBitmap(newphoto2);*/
        Log.d("", ""+a);
        Toast.makeText(this, a, Toast.LENGTH_LONG).show();
        textView.setText(a);
        imageView.setImageBitmap(newphoto2);
        progressBar.setVisibility(View.GONE);
        imageView.setVisibility(View.VISIBLE);
        textView.setVisibility(View.VISIBLE);

    }

}

```

IV. CODE FOR PROCESSING OF IMAGE AND SENDING SMS TO USER:

```

package com.agrobot.capturephoto;

```

```

import android.app.Activity;
import android.app.PendingIntent;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.Color;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;

import com.agribot.dashboard.R;

public class AgribotCapturePhoto extends Activity {

    Button button;
    ImageView imageView;
    int requestcode = 114;
    Bitmap photo, newphoto, newphoto2, newphoto3;
    TextView textView;

    public static Bitmap doGreyscale(Bitmap src) {
        // constant factors
        final double GS_RED = 0.299;
        final double GS_GREEN = 0.587;
        final double GS_BLUE = 0.114;
        // create output bitmap
        Bitmap bmOut = Bitmap.createBitmap(src.getWidth(),
src.getHeight(),
src.getConfig());
        // pixel information
        int A, R, G, B;
        int pixel;

        // get image size
        int width = src.getWidth();
        int height = src.getHeight();

        // scan through every single pixel
        for (int x = 0; x < width; ++x) {
            for (int y = 0; y < height; ++y) {
                // get one pixel color
                pixel = src.getPixel(x, y);
                // retrieve color of all channels
                A = Color.alpha(pixel);
                R = Color.red(pixel);
                G = Color.green(pixel);
                B = Color.blue(pixel);
                // take conversion up to one single value
                R = G = B = (int) (GS_RED * R + GS_GREEN * G +
GS_BLUE * B);

                // set new pixel color to output bitmap
                bmOut.setPixel(x, y, Color.argb(A, R, G, B));
            }
        }

        // return final image
    }

```



```

        return bmOut;
    }

    public static Bitmap binarize(Bitmap original, Bitmap original2) {

        int red;
        int newPixel, pixel;

        int threshold = otsuTreshold(original);
        System.out.println(">>>>>>>>>>>>" + threshold);
        Bitmap binarized = Bitmap.createBitmap(original.getWidth(),
            original.getHeight(), original.getConfig());

        int[] pix = new int[original.getWidth() *
original.getHeight()];

        for (int i = 0; i < original.getWidth(); i++) {
            for (int j = 0; j < original.getHeight(); j++) {

                float[] hsv = new float[3];

                pixel = original.getPixel(i, j); // Get pixels
                int index = i * original.getWidth() + j;
                int A = Color.alpha(pixel);
                int R = Color.red(pixel);
                Color.colorToHSV(original2.getPixel(i, j), hsv);

                float H = hsv[0];
                float S = hsv[1];
                float V = hsv[2];

                if (H > threshold) {
                    newPixel = 255;
                    newPixel = colorToRGB(0xff, newPixel,
newPixel, newPixel);

                    binarized.setPixel(i, j, newPixel);
                } else {

                    binarized.setPixel(i, j,
Color.HSVToColor(hsv));

                }
                // newPixel = colorToRGB(0xff, newPixel, newPixel,
newPixel);

            }
        }

        return binarized;
    }

    public static int colorToRGB(int alpha, int red, int green, int
blue) {

        int newPixel = 0;
        newPixel += alpha;
        newPixel = newPixel << 8;
        newPixel += red;
        newPixel = newPixel << 8;
        newPixel += green;
        newPixel = newPixel << 8;
        newPixel += blue;

```

```

        return newPixel;
    }

    public static int[] imageHistogram(Bitmap input) {

        int[] histogram = new int[256];
        int[] pix = new int[input.getWidth() * input.getHeight()];
        int pixel;
        for (int i = 0; i < histogram.length; i++)
            histogram[i] = 0;

        for (int i = 0; i < input.getWidth(); i++) {
            for (int j = 0; j < input.getHeight(); j++) {

                pixel = input.getPixel(i, j); // Get pixels

                int index = i * input.getWidth() + j;
                int R = Color.red(pixel);
                // bitwise shifting

                histogram[R]++;
            }
        }

        return histogram;
    }

    public static int otsuTreshold(Bitmap original) {

        int[] histogram = imageHistogram(original);
        int total = original.getHeight() * original.getWidth();

        float sum = 0;
        for (int i = 0; i < 256; i++)
            sum += i * histogram[i];

        float sumB = 0;
        int wB = 0;
        int wF = 0;

        float varMax = 0;
        int threshold = 0;

        for (int i = 0; i < 256; i++) {
            wB += histogram[i];
            if (wB == 0)
                continue;
            wF = total - wB;

            if (wF == 0)
                break;

            sumB += (float) (i * histogram[i]);
            float mB = sumB / wB;
            float mF = (sum - sumB) / wF;

            float varBetween = (float) wB * (float) wF * (mB - mF) *
(mB - mF);

```

```

        if (varBetween > varMax) {
            varMax = varBetween;
            threshold = i;
        }
    }

    return threshold;
}

public static String findcolor(Bitmap newphoto4) {
    //
    Toast.makeText(this, ""+newphoto4.getHeight()*newphoto4.getWidth(), Toast.LENGTH_SHORT);

    int[] pix = new int[newphoto4.getWidth() *
newphoto4.getHeight()];
    int pixel, unripe = 0, medripe = 0, ripe = 0, total = 0;
    for (int i = 0; i < newphoto4.getWidth(); i++) {
        for (int j = 0; j < newphoto4.getHeight(); j++) {

            float[] hsv = new float[3];

            pixel = newphoto4.getPixel(i, j); // Get pixels
            Color.colorToHSV(newphoto4.getPixel(i, j), hsv);

            float H = hsv[0];
            float S = hsv[1];
            float V = hsv[2];
            int R = Color.red(pixel);
            int G = Color.green(pixel);
            int B = Color.blue(pixel);

            //System.out.println("  H -->" + H + "    S -->" + S
+ "    V -->" + V);

            if (((H >= 2.0 && H <= 6.0) || (H >= 350.0 && H <=
360.0)
                && S >= 0.83 && S <= 0.85 && V >= 0.45
                || ((R >= 150) && (R <= 255) && (G >=
0) && (G <= 75)
                    && (B >= 0) && (B < 75)))) {
                //System.out.println("RED");
                ripe++;
            }

            if (((H >= 20.0 && H <= 50.0) && (S >= 0.65 && S <=
1.00) && (V >= 0.65 && V <= 1.00))
                || ((R >= 233 && H <= 255) && (R >= 69
&& R <= 160) && ((G >= 0 && G <= 122)))) {
                //System.out.println("ORANGE");
                medripe++;
            }
            //
            if (R == 178 && G == 236 && B == 93)
                unripe++;
            if (R == 124 && G == 252 && B == 0)
                unripe++;

```

```

        if (R == 102 && G == 255 && B == 0)
            unripe++;
        if (R == 172 && G == 225 && B == 175)
            unripe++;
        if (R == 178 && G == 190 && B == 181)
            unripe++;
        if (R == 163 && G == 193 && B == 173)
            unripe++;
        if (R == 147 && G == 197 && B == 114)
            unripe++;
        if (R == 133 && G == 187 && B == 101)
            unripe++;
        if (R == 3 && G == 192 && B == 60)
            unripe++;
        if (R == 120 && G == 134 && B == 107)
            unripe++;
        if (R == 115 && G == 134 && B == 120)
            unripe++;
        if (R == 0 && G == 128 && B == 0)
            unripe++;
        if (R == 19 && G == 136 && B == 8)
            unripe++;

        if ((R >= 175) && (R <= 250) && (G >= 100) && (G <=
255)
            && (B >= 0) && (B < 10)) {
            unripe++;

            //System.out.println(" Unripe ");
        }

        // if (!(H == 60.0) && (S == 0.0) && (V == 1.0))
|| ((H == 0.0)
        // && (S == 0.0)&& ((V >= 0.30) && (V <= 0.40)))) {
        // total++;
        //
        //
        // }
        if (!(R == 255 && G == 255 && B == 255))
            || ((H == 60.0) && (S == 0.0) && (V ==
1.0)) || ((H == 0.0)
            && (S == 0.0) && ((V >=
0.30) && (V <= 0.40)))) {
            total++;
        }
    }

    }

    int totalpercent = unripe + medripe + ripe;

    String a = "UNRIPE: " + ((unripe * 100) / totalpercent) + "%
MEDRIPE: "
        + ((medripe * 100) / totalpercent) + "% RIPE: "
        + ((ripe * 100) / totalpercent) + " % ";
        //+ " Threshold : " + otsuTreshold(newphoto4);
    // TODO Auto-generated method stub
    return a;
}

@Override

```

```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.agribotrealtime);

    button = (Button) findViewById(R.id.button1);
    imageView = (ImageView) findViewById(R.id.imageview);
    textView = (TextView) findViewById(R.id.state);
    // imageView.getLayoutParams().height = 50;

    button.setOnClickListener(new OnClickListener() {

        public void onClick(View v) {
            // TODO Auto-generated method stub
            Intent i = new Intent(
                android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
            startActivityForResult(i, requestcode);
        }
    });

}

@Override
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    if (requestCode == requestcode) {
        photo = (Bitmap) data.getExtras().get("data");

        newphoto = doGreyscale(photo);
        newphoto2 = binarize(newphoto, photo);
        String a = "";
        a = findcolor(newphoto2);

        imageView.setLayoutParams(new LinearLayout.LayoutParams(
            LinearLayout.LayoutParams.WRAP_CONTENT,
            LinearLayout.LayoutParams.WRAP_CONTENT));

        imageView.getLayoutParams().height = 300;
        imageView.getLayoutParams().width = 300;
        imageView.setImageBitmap(newphoto2);
        textView.setText(a);

        //Sending sms
        PendingIntent pi = PendingIntent.getActivity(this, 0, data,
0);

        SmsManager sms = SmsManager.getDefault();
        sms.sendTextMessage("8097511597", null, a, pi, null);

    }

}

}

```

5. TEST CASES:

Test Case No:	Test Case	Result Expected	Actual Result
1	Alarm Service	It must be able to set time	Time is set
2	Capture photo	Photo must be captured at specified time	Photo is captured At specified time
3	Saving photo to sd card	Photo must be saved automatically to sd card	Photo is saved to sd card
4	Processing of photo	Photo must be processed in the phone itself	Photo is processed in the phone
5	Extracting RGB values	Extract RGB value of each pixel	RGB value of each pixel is extracted
6	Converting RGB values to HSV	RGB values must be converted to HSV model	RGB values converted to HSV model
7	Calculating Ripe, Unripe, Medium ripe and Total no of pixels	Ripe, Unripe, Medium ripe and Total no of pixels must be calculated	Ripe, Unripe, Medium ripe and Total no of pixels are calculated
8	Finding percentage of ripe, medium ripe and ripe part	Find percentage of ripe, medium ripe and ripe part	Percentage of ripe, medium ripe and ripe part is calculated
9	Sending SMS	Send SMS to the user	SMS is sent

Test Case No.	Test Case	Result Expected	SMS Received
1	RIPED TOMATO	UNRIPE:0% MEDRIPE:0% RIPE:100%	UNRIPE: 0% MEDRIPE:0% RIPE:100%
2	UNRIPE TOMATO	UNRIPE:80% MEDRIPE:20% RIPE:0%	UNRIPE:70 % MEDRIPE:30% RIPE:0%
3	MEDIUM RIPE TOMATO	UNRIPE:0% MEDRIPE:100% RIPE:0%	UNRIPE:0 % MEDRIPE:99% RIPE:0%
4	RIPED AND UNRIPED TOMATO	UNRIPE:20% MEDRIPE:10% RIPE:70%	UNRIPE: 20% MEDRIPE:10% RIPE:70%
5	RIPED AND MEDIUM RIPE TOMATO	UNRIPE:0% MEDRIPE:40% RIPE:70%	UNRIPE: 0% MEDRIPE:33% RIPE:67%
6	MEDIUM RIPE AND UNRIPED TOMATO	UNRIPE:35% MEDRIPE:50% RIPE:15%	UNRIPE: 33% MEDRIPE:54% RIPE:13%

7	RIPED, MEDIUM RIPED AND UNRIPED TOMATO	UNRIPE:20% MEDRIPE:60% RIPE:20%	UNRIPE: 13% MEDRIPE:67% RIPE:18%
---	--	------------------------------------	-------------------------------------

Note: Above results are varied for lighting conditions and also tomato color variations.

7. FUTURE SCOPE:

The Project can be modified for various applications including the following:

- Robotic fruit harvesting
- Image based fruit disease detection.
- Other applications which requires vision as a sensor.

8. CONCLUSION:

Thus we have successfully done with our project. We have implemented the application software for android based phone which can detect ripeness degree of either single tomato fruit or combined ripeness degree of multiple tomatoes.

We have also implemented automation in the project with the help of alarm clock service of android phone.

Thus, this application is able to automatically capture photos of tomatoes and save it in sd card for processing it.

We have also implemented SMS service which delivers SMS regarding the ripeness degree of fruit in terms of percentage.

We have also implemented real time fruit ripeness detection module which can give status of ripeness for few pixels of fruit which are in the range of camera.

The project we have built will be useful for greenhouse owners after some modifications and enhancement in coding.

9. REFERENCES:

- [1] Izzatdin Abdul Aziz, Mohd Hilmi Hasan, Mohd Jimmy Smail, Mazlina Mehat, Nazleen Samiha Haronr , “Remote Monitoring in Agricultural Greenhouse Using Wireless Sensor and Short Message Service (SMS)”, International Journal of Engineering & Technology IJET Vol: 9 No: 9
- [2] http://www.vogella.com/articles/Android/article.html#overview_adt
- [3]<http://www.stanford.edu/class/ee368/Android/Tutorial-1-Basic-Android-Setup-Windows.pdf>
- [4] <http://developer.android.com/guide/developing/device.html>