SHIVAJI UNIVERSITY

Detecting Defects of Fruit by Image Processing

by

Author Name

A project report submitted in partial fulfillment for the
Bachelor of Engineering

in the

Department of Computer Science and Engineering
Sharad Institute of Technology College of Engineering,Yadrav-Ichalkaranji

April 2015

SHIVAJI UNIVERSITY

Abstract

Department of Computer Science and Engineering
Sharad Institute of Technology College of Engineering,Yadrav-Ichalkaranji

Bachelor of Engineering

Nowadays, overseas commerce has increased drastically in many countries. Plenty fruits are imported from the other nations such as oranges, apples etc. Manual identification of defected fruit is very time consuming. This work presents a novel defect segmentation of fruits based on color features with K-means clustering unsupervised algorithm. Hassu algorithm used for quality analysis. We used color images of fruits for defect segmentation. Defect segmentation is carried out into two stages. At first, the pixels are clustered based on their color and spatial features, where the clustering process is accomplished. Then the clustered blocks are merged to a specific number of regions. Using this two step procedure, it is possible to increase the computational efficiency avoiding feature extraction for every pixel in the image of fruits. Although the color is not commonly used for defect segmentation, it produces a high discriminative power for different regions of image. This approach thus provides a feasible robust solution for defect segmentation of fruits. We have taken apple as a case study and evaluated the proposed approach using defected apples. The experimental results clarify the effectiveness of proposed approach to improve the defect segmentation quality in aspects of precision and computational time. In result of hassu algorithm we get the quality of fruit such as unripe, partially ripe, ripe, over ripe.

Acknowledgements

We take this opportunity to express our profound gratitude and deep regards to our guide Mr.S.Gurav for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him time to time shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to our Principal, Dr. S. A. Khot and our H.O.D., Prof. O.D.Joshi for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are obliged to all faculty and staff members of all departments of our Sharad Institute of Technology College of Engineering, Yadrav for the valuable information provided by them in their respective fields. We are grateful for their cooperation during the period of our assignment.

Lastly, we thank almighty, our parents, brothers, sisters, friends and our colleagues for their constant encouragement without which this assignment would not be possible.

Miss. Karadage Akanksha Satish

Miss. Khamkar Alka Gajanan

Miss. Malke Divya Sanjay

Miss. Bondarde Trushali Rajendra

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Introduction of project

India is an agricultural country where in about 70In this proposed system we implement quality of fruit by using two techniques. One is K-means and another is Hassu algorithm[1] [2]. We use color images of fruits for defect segmentation. K-means segmentation is carried out into two stages.[3] At first, the pixels are clustered based on their color and spatial features, where the clustering process is accomplished.[4]Then the clustered blocks are merged to a specific number of regions. Using this two step procedure, it is possible to increase the computational efficiency avoiding feature extraction for every pixel in the image of fruits. In second technique we proposed to quality analysis and detect defects of further which can be implemented for grading and sorting of a particular fruit by its visual color of surface using the non-destructive technique to automated quality verification systems for agricultural products with the help of digital images which involve visual examination and inspection of color, size, shape, defects are highlighted further for image processing. In this project we use two techniques since users can use any technique which is suitable for them.

1.2 What is clustering?

Clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

1.3 Why we need clustering?

Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem. The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results.

1.4 Image segmentation

Image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics. The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image.

Chapter 2

LITERATURE REVIEW

2.1 K-Means clustering segmentation

In this paper we studied what is clustering and what is image segmentation.[Shiv Ram Dubey, Pushkar Dixit, Nishant Singh, Jay Prakash GuptaInfected fruit Part Detection using K-Means Clustering segmentation Technique].

2.1.1 Processing K-Means clustering on image

K-means is generally used to determine the natural groupings of pixels present in an image. It is attractive in practice, because it is straightforward and it is generally very fast. It partitions the input dataset into k clusters. Each cluster is represented by an adaptively changing center (also called cluster center) starting from some initial values named seed-points. K-means clustering computes the distances between the inputs and centers and assigns inputs to the nearest center.

2.1.2 Image segmentation using K-Means algorithm

Image segmentation using k-means algorithm is quite useful for the image analysis. An important goal of image segmentation is to separate the object and background clear regardless the image has blur boundary. Defect segmentation of fruits can be seen as an instance of image segmentation in which number of segmentation is not clearly known. Figure 1 shows the framework for the fruits defect segmentation. The basic aim of the proposed approach is to segment colors automatically using the K-means clustering technique and L*a*b* color space.

2.2 Comparative analysis using image processing

Proposed updated hassu Algorithm to analysis quality of fruits. [Hassan Sardar A role of computer system for comparative analysis using image processing to promote agriculture business.].

2.2.1 Hassu algorithm

Load 50 reference images of fruit. Pass all 50 reference images through color array to calculate image value by intensity value in the continuous series with help of step deviation method. Take new image of fruit. Pass it through color array to calculate image value by intensity value in the continuous series with help of step deviation method. If image value match lie between 1 to 33 image. Then Result is un-ripe Else if Image value match lie between 34 to 40 images. Then Result is partially ripe Else if Image value match lie between 41to 45 images. Then Result is ripe Else if Image value matches with 46 and50 image. Then Result is over ripe. Repeat the Step 2 for next new image.

Chapter 3

OBJECTIVE AND SCOPE

3.1 Objective and scope

3.1.1 Objectives

Following are the objectives of the proposed system:

- Main aim of the research is to develop a new and efficient algorithm for of fruit detect the defect identify the quality of fruits.
- To compare the speed of identification for checking the quality of fruits with manual method.

3.1.2 Scope

- Acquiring fruit images.
- Noise reduction.
- To detect Region of Interest fruits.
- To quantify affected area by disease.
- To find shape of affected area of fruit.
- To determine color of affected area of fruit.
- To determine size and shape of fruit.

3.1.3 Out of scope

- Internally detecting defects of fruits by using Infra ray.
- The future work includes automatic determination of number of clusters required to segment the defects more accurately.

Chapter 4

REQUIREMENT ANALYSIS

4.1 Hardware requirements

- Smart Phone
- PC/Laptop
- Digital camera

4.2 Software requirements

- OS for server:Windows 7 and Windows 8
- OS for mobile or tablet clients: Android
- JDK
- Eclipse
- Net Beans IDE 8.0.1
- MyPublicWifi

4.2.1 Java

a) Description:

A high-level programming language developed by Sun Microsystems. Java was originally called OAK, and was designed for handheld devices and set-top boxes. Oak was unsuccessful so in 1995 Sun changed the name to Java and modified the language to take advantage of the burgeoning World Wide Web.

b) Java:
It is a computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of computer architecture.

c) Java platform:
It is the name given to the computing platform from Oracle that helps users to run and develop Java applications. The platform does not just enable a user to run and develop a Java application, but also features a wide variety of tools that can help developers work efficiently with the Java programming language.

The platform consists of two essential softwares:

1) The Java Runtime Environment (JRE), which is needed to run Java applications and applets.

2) The Java Development Kit (JDK), which is needed to develop those Java applications and applets.

d) JDK: The Java Development Kit (JDK) is an implementation of either one of the Java SE, Java EE or Java ME platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, Mac OS X or Windows. The JDK includes a private JVM and a few other resources to finish the recipe to a Java Application. Since the introduction of the Java platform, it has been by far the most widely used Software Development Kit (SDK). [citation needed] On 17 November 2006, Sun announced that it would be released under the GNU General Public License (GPL), thus making it free software. This happened in large part on 8 May 2007, when Sun contributed the source code to the Open JDK.

4.2.2 Eclipse

a) Description:

Eclipse is a Java-based open source platform that allows a software developer to create a customized development environment (IDE) from plug-in components built by Eclipse members. The original goal of Eclipse was to create and foster an open source IDE community that would complement the community that surrounds Apache. The Eclipse Platform is written in Java, it supports plug-ins that allows developers to develop and test code written in other languages.

b) Platform:

1) Plug-in development environment(PDE), provides a number of views and editors that make is easier to build plug-ins for Eclipse.

2) The Java development tools (JDT), provides the tool plug-ins for the platform that implement a Java IDE for power-users, that supports the development of any Java application, including Eclipse plug-ins.

4.2.3 Net Beans IDE 8.0.1

Net Beans IDE 8.0.1 delivers full support for the latest Java 8 technologies Java SE 8, Java SE Embedded 8, and Java ME Embedded 8. The IDE also provides a range of new enhancements for Maven and Java EE with Prime Faces; new tools for HTML5, in particular for Angular and improvements to PHP and C/C++ support. Net Beans IDE is the official IDE for Java 8. With its editors, code analyzers, and converters, you can quickly and smoothly upgrade your applications to use new Java 8 language constructs, such as lambdas, functional operations, and method references. Batch analyzers and converters are provided to search through multiple applications at the same time, matching patterns for conversion to new Java 8 language constructs. With its constantly improving Java Editor, many rich features and an extensive range of tools, templates and samples, Net Beans IDE sets the standard for developing with cutting edge technologies out of the box.

a) Fast and smart code editing: An IDE is much more than a text editor. The Net Beans Editor indents lines, matches words and brackets, and highlights source code syntactically and semantically. It also provides code templates, coding tips, and refactoring tools. The editor supports many languages from Java, C/C++, XML and HTML, to PHP, Groovy, Javadoc, JavaScript and JSP. Because the editor is extensible, you can plug in support for many other languages.

b)Easy and efficient project management: Keeping a clear overview of large applications, with thousands of folders and files, and millions of lines of code, is a daunting task. Net Beans IDE provides different views of your data, from multiple project windows to helpful tools for setting up your applications and managing them efficiently, letting you drill down into your data quickly and easily, while giving you versioning tools via Subversion, Mercurial and Get integration out of the box. When new developers join your project, they can understand the structure of your application because your code is well-organized.

4.2.4 MyPublicWifi

MyPublicWiFi is an easy-to-use software that turns your laptop/PC into a Wi-Fi wireless access point. Anyone nearby can surf the Internet through your sharing. This is also an ideal solution for setting up a temporary Access Point in a hotel room,meeting room,at home or the like. The MyPblicWiFi-Firewall can be used to restrict user access to specific servers. You can also prevent the use of certain Internet services (e.g. file sharing programs). MyPublicWiFi allows you to record and track all visited url pages on your virtual WIFI-Hotspot.

4.2.5 Android operating system

Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google OHA. With a user interface based on direct manipulation, Android is designed primarily for touch screen mobile devices such as Smartphone and tablet computers, with specialized user interfaces for televisions (Android TV), cars (Android Auto), and wrist watches (Android Wear). The OS uses touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a virtual keyboard. Despite being primarily designed for touch screen input, it also has been used in game consoles, digital cameras, and other electronics. Android is the most popular mobile OS. As of 2013, Android devices sell more than Windows OS and Mac OS devices combined, with sales in 2012, 2013 and 2014 close to the installed base of all PCs. As of July 2013 the Google Play store has had over 1 million Android apps published, and over 50 billion apps downloaded. A developer survey conducted in AprilMay 2013 found that 71Android's source code is released by Google under open source licenses, although most Android devices ultimately ship with a combination of open source and proprietary software. Initially developed by Android which Google backed financially and later bought in 2005, Android was unveiled in 2007 along with the founding of the Open Handset Alliance a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices.

Chapter 5

SYSTEM DESIGN

5.1 An overview of UML

UML is a language for

- Visualizing
- Specifying
- Constructing
- Documenting

UML LANGUAGE

A language provides a vocabulary and the rules for combing words in that vocabulary for the purpose of the communication. A modeling language is a language whose vocabulary and rules focus on conceptual and physical representation of a system. A modeling language such as UML is thus a standard language for software blueprints. In this context, specifying means building models that are precise, unambiguous, and complete. In particular, UML addresses the specification of all the important analysis, design and implementation decision that must be made in developing and deploying a software intensive system. UML is not a visual programming language, but its model can be directly connected to a verity of programming languages. This means that its possible to map from a model in UML to a programming language such as java, cpp, or visual basic or even to tables in a relational database. Things that are best expressed graphically are done so graphically in UML, whereas things that best expressed textually are done so in the programming language. A healthy software organization produces

all sorts of artifacts in addition to raw executable code. These artifacts include requirements, architecture, design, source code, project plans, tests, prototypes, releases. UML addresses the documentation of a systems architectures and all of its details. UML also provides for expressing requirements and for tests. Finally, UML provides a language for modeling the activities of project planning and release management.

5.2 Goals of UML

The primary goals in the design of UML were:

- Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models. Provide extensibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development processes.Provide a formal basis for understanding the modeling language
- Encourage the growth of the OO tools market.
- Support higher-level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices

5.3 A conceptual model of UML

To understand UML, you need to form a conceptual model of the language, and this requires learning three major elements: UMLs basic building blocks, the rules that dictate how those building blocks may be put together, and some mechanisms that apply throughout UML. Once you have grasped these ideas, you will be able to read UML models and create some basic ones. As you gain more experience in applying UML, you can build on this conceptual model, using more advanced features of the language.

5.3.1 Building blocks of UML

The vocabulary of UML encompasses three kinds of building blocks:

- Things

- Relationships
- Diagrams

These are the abstractions that are first-class citizens in a model; relationships tie these things together; diagrams groups interesting collections of things.

5.4 Diagrams in UML

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices (things) and arcs (relationships). You draw diagrams to visualizing a system from different perspectives, so a diagram is a projection into a system. For all but the most trivial systems, a diagram represents an elided view of the elements that make up a system. The same element may appear in all diagrams. In theory, a diagram may contain any combination of things and relationships. The views that comprise the architecture of software intensive system. For this reason, UML includes following diagrams:

- Use case diagram
- Class diagram
- Sequence diagram
- Deployment diagram

5.5 Use case diagram

A use case diagram is a diagram that shows a set of use cases and actors and their relationships. A use case diagram is a just special kind of diagram and shares the same common properties as do all other diagram-a name and graphical contents.

5.5.1 Contents

Use case diagrams commonly contain

- **Use Case**

  Use case is a description of a set of sequence of actions that a system performs that yields an observable result of value to a particular actor. A use case is rendered

as an ellipse with solid lines usually including its name. A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well. A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements

- **Actors**

  An actor represents a role that an outsider takes on when interacting with the business system. For instance, an actor can be a customer, a business partner, a supplier, or another business system and every actor has a name.

- **Dependency, generalization, and association relationships.**

  A dependency is a semantic relationship between two things in which a change to one thing may affect the semantics of the other thing.

FIGURE 5.1: Dependencies

A *generalization* is a relationship in which objects of specialized elements (the child) are substitutable for objects of the generalized element.

An *association* is a structural relationship that describes a set of links, a link being connection among objects

Like all other diagrams, use case diagram may contain notes and constraints.

5.5.2 Common uses

Use case diagram typically contain in one of two ways.

- To model the context of the system

  Hear system involves drawing line around the whole system and actors outside of the system and interact with it.

- To model the requirement of a system

  Hear specifies what the system should do, independent of how that system should do.

5.5.3 Use-case diagram

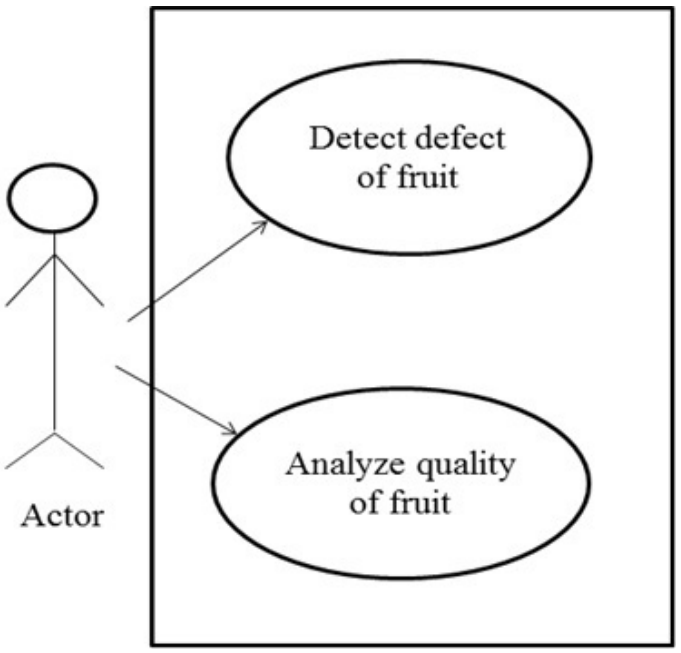

FIGURE 5.2: Use Cases

In this use case diagram user interaction with the two system one is the detect defect of fruit and another is the Analyze quality of fruit. In this use case diagram user well get the which is methodology used in system analysis to identify, clarify, and organize system requirements.

TABLE 5.1: Use case scenario table

USE CASE	**USE CASE SCENARIO**
Detect the defects of fruit	User is the main Actor in this phase. The main flow of events: 1) User captures image of fruit with the fruitdetector.apk on the smart phone. 2) fruitdetector.apk Send that fruit image to detect defect.java. 3) At transformer it transforms the image from RGB (i.e. Red, Green, Blue) color to L*a*b* color space (i.e. luminosity layer in L* and chromaticity layer in a* and b*). 4) The L*a*b* image from transformer send to the cluster. 5) In cluster it classify colors k-means clustering by a* and b* color space. 6) Determining the Euclidean matrix and measuring the difference between colors by Euclidean matrix.. 7) Label each pixel by taking each cluster index. 8) Collecting all cluster indexes and send to image generator. 9) Image generator generates image and result will be send to fruitdetector.apk. 10) Display the result. 11) Exit.
Analyze quality of fruit	User is the main Actor in this phase .The main flow of events: 1) User captures image of fruit with the fruitdetector.apk on the smart phone. 2) Image will be send to analyzer and Pass image through color array to calculate image value by intensity value in the continuous series. 3)At the Deviation Step deviation method is applied to intensity value. 4) In deviation Compare value with the 50 already loaded images . 5 If image value match lie between 1 to 33 image then result is un-ripe. 6) If image value match lie between 33 to 40 image then result is partially-ripe. 7) If image value match lie between 41 to 45 image then result is ripe. 8)If image value match lie between 46 to 50 image then result is over-ripe. 9)Determine the result and display the result. 10) Exit.

5.6 Sequence diagram

5.6.1 Contents

Sequence diagram commonly contains

- Objects
- Links
- Messages

5.6.2 Definition and overview

A *sequence* diagram is an interaction diagram that emphasizes the time ordering of messages. A sequence diagram shows a set of objects and the messages sent and received by those objects. The objects are typically named or anonymous instances of classes, but may also represent instances of other things, such as collaborations, components, and nodes. You use sequence diagrams to illustrate the dynamic view of a system. An Actor models a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data), but which is external to the subject (i.e., in the sense that an instance of an actor is not a part of the instance of its corresponding subject). Actors may represent roles played by human users, external hardware, or other subjects. Note that an actor does not necessarily represent a specific physical entity but merely a particular facet (i.e., "role") of some entity that is relevant to the specification of its associated use cases.

Sequence diagram have two features that distinguish them from collaboration diagrams.

- First, there is the object lifeline. An object lifeline is the vertical dashed line that represents the existence of an object over a period of time. So these objects are at the top of the diagram. With their lifelines drawn from the top of the diagram to the bottom.
- Second, there is the focus of control. The focus of control is a tall, thin rectangle that shows the period of time during which an object is performing an action, either directly or through a subordinating procedure. The top of the rectangle is aligned with the start of the action; the bottom is aligned with its completion and also it can be marked by replay message.

5.6.3 Sequence diagrams

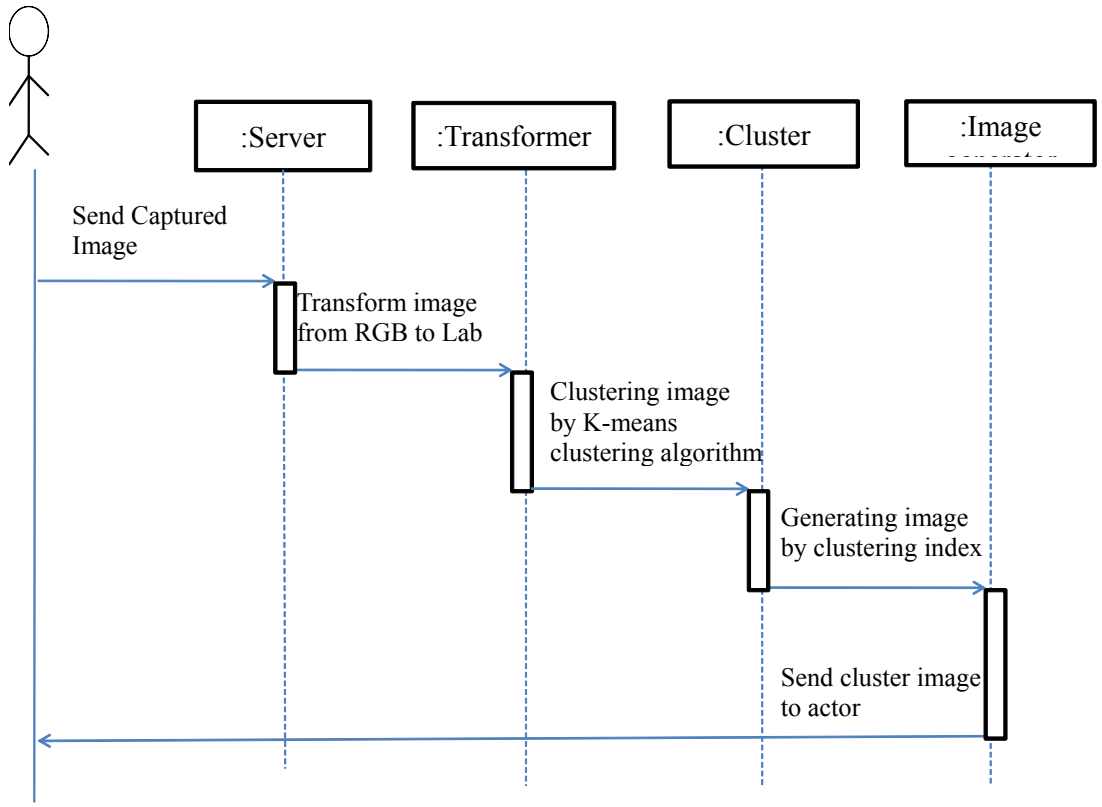

FIGURE 5.3: Sequence diagram for detect the defects of fruit.

5.6.3.1 Figure description

Capture image of fruit at client side. Send that fruit image to server side. At server side it transforms the image from RGB (i.e. Red, Green, Blue) color to L*a*b* color space (i.e. luminosity layer in L* and chromaticity layer in a* and b*). Classify colors k-means clustering by a* and b* color space. Determining the Euclidean matrix and measuring the difference between colors by Euclidean matrix. Label each pixel by taking each cluster index. Collecting all cluster indexes and generating image. Determine the defected part of image by cluster. Send result from server to client. Display the result.

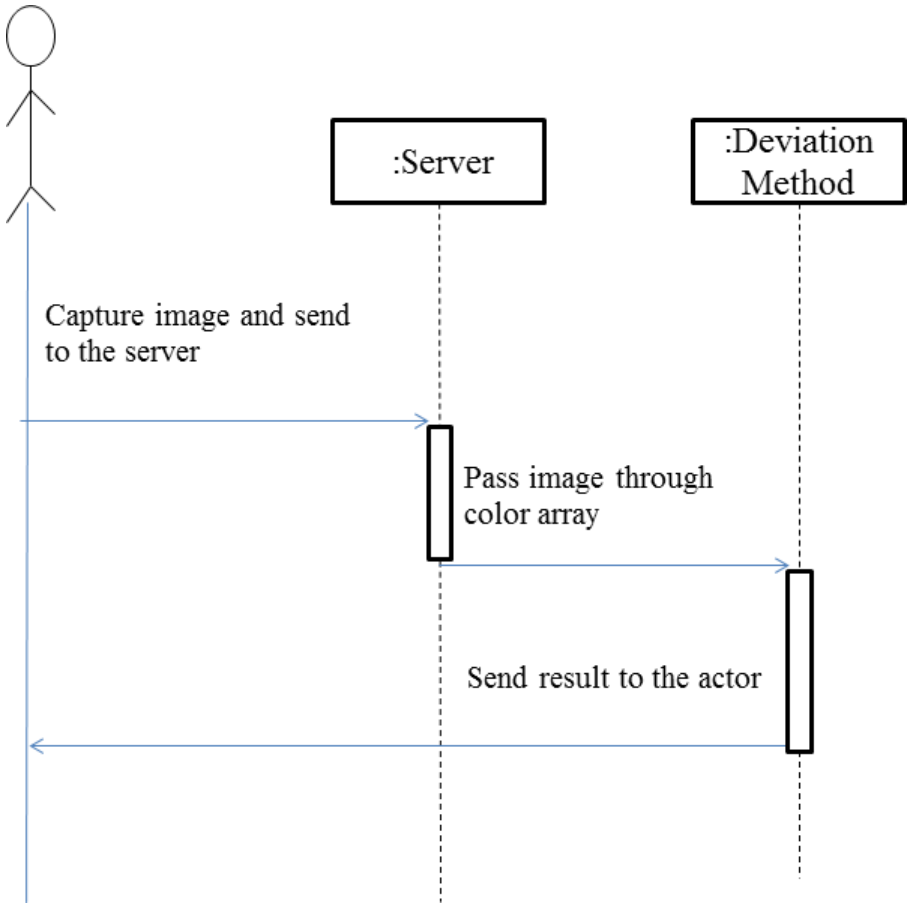

FIGURE 5.4: Sequence diagram for Analyze quality of fruit..

5.6.3.2 Figure description

Capture image at client side and send it to the server side. Pass image through color array to calculate image value by intensity value in the continuous series with help of step deviation method. Compare it with the 50 already loaded images. If image value match lie between 1 to 33 image then result is un-ripe. If image value match lie between 33 to 40 image then result is partially-ripe. If image value match lie between 41 to 45 image then result is ripe. If image value match lie between 46 to 50 image then result is over-ripe.

5.7 Class diagram

5.7.1 Contents

Class diagram commonly contain the following things:

- Classes
- Interfaces
- Collaborations
- Dependency, generalization, and association relationships.

5.7.2 Definition and common uses

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity. In the diagram, classes are represented with boxes which contain three parts:

- The top part contains the name of the class. It is printed in Bold, centered and the first letter capitalized.
- The middle part contains the attributes of the class. They are left aligned and the first letter is lower case.
- The bottom part gives the methods or operations the class can take or undertake. They are also left aligned and the first letter is lower case.

5.7.3 Class diagram

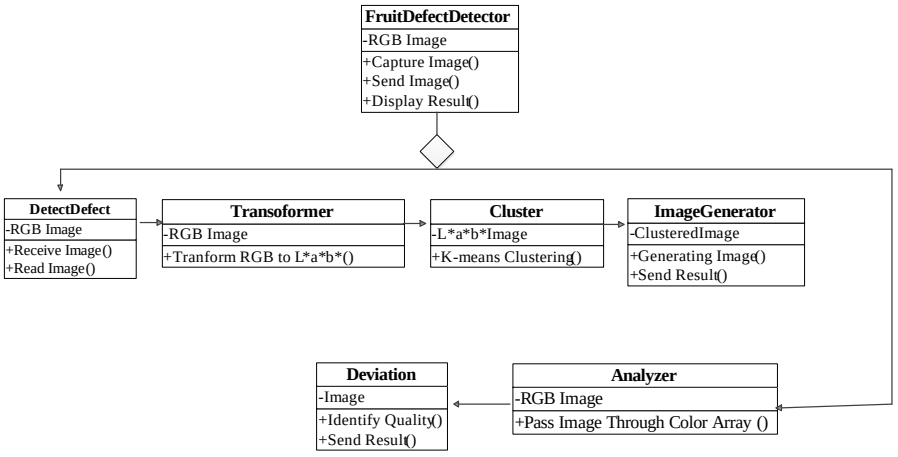

FIGURE 5.5: Class diagram.

There are seven classes in our system. FruitDefectDetector, DetectDefect, Transformer, Cluster, ImageGenerator, Deviation and Analyzer. In our System FruitDefectDetector is the main class.It contains RGB Image as Attribute.There are three operations of this class Capture Image, Send Image and Display Result.This FruitDefectDetector contain two other class DetectDefect and Analyzer. In class DetectDefect ,first image is capture then send to transformer.Transformer convert the RGB image to L*a*b* image.L*a*b* image is then cluster by using cluster index.And ImageGenerator generate the result. In Analyzer, RGB image is the attribute.It pass to Deviation class.Deviation class contains two operation identify quality and send result.

5.8 Deployment diagram

5.8.1 Definition

A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components exist (e.g., a web server, an application server, and a database server), what software components run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes.

5.8.2 Deployment diagram

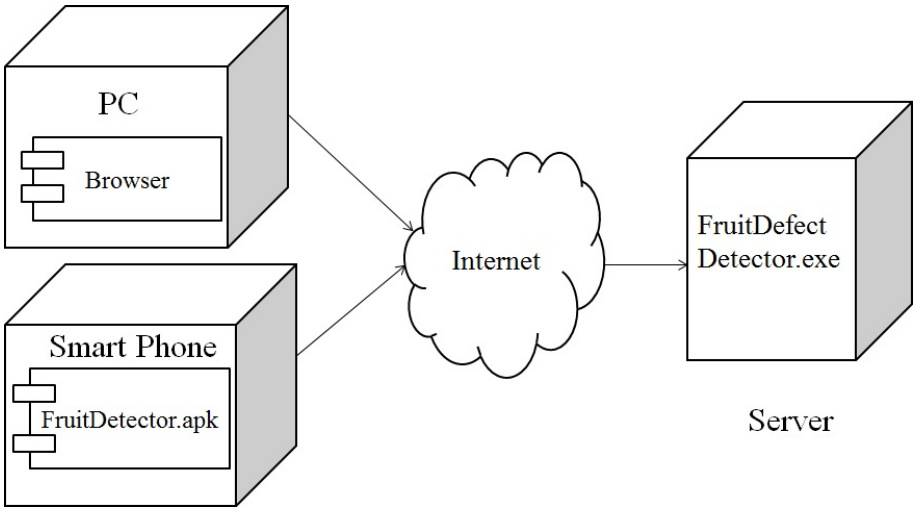

FIGURE 5.6: Deployment diagram

In our system their are two clients one is PC and another is Smart Phone.Their is one server machine.Clients send image to server. On the server machine FruitDefectDetector.exe file is present.If client send request to server then it will be execute on the server machine and result will display on client side.The client and server are connected by Internet.

Chapter 6

CODING

6.1 Introduction of tools and installation

6.1.1 Android

There's no other software quite like Android. Google engineered Android, and Google own apps run better on it. And with millions of apps, games, songs, and videos on Google Play, Android is great for fun, and for getting things done.

Android devices come in all kinds of sizes, with all sorts of features, and in all sorts of prices. Each version of Android is named after a dessert, and the most recent version of Android is lolipop. With Android, youre in control of your mobile experience.

The world is contracting with the growth of mobile phone technology. As the number of users is increasing day by day, facilities are also increasing. Starting with simple regular handsets which were used just for making phone calls, mobiles have changed our lives and have become part of it. Now they are not used just for making calls but they have innumerable uses and can be used as a Camera , Music player, Tablet PC, T.V. , Web browser etc. . And with the new technologies, new software and operating systems are required.

- What is android

Operating Systems have developed a lot in last 15 years. Starting from black and white phones to recent smart phones or mini computers, mobile OS has come far away. Especially for smart phones, Mobile OS has greatly evolved from Palm OS in 1996 to Windows pocket PC in 2000 then to Blackberry OS and Android.

- ADT Bundle

The Android SDK is a software development kit which provides API libraries and necessary developer tools necessary for building Android applications. Android SDK is officially provided by android developers.

steps for the installation and set-up of Android development environment:

1. Download Eclipse
2. Download JDK and install it, set the environment path.
3. Download ADT plugin inside Eclipse.
4. Set the Preference with Android-SDK path.
5. Download the latest platform-tools and everything.

The ADT Bundle includes everything you need to begin developing apps:

1. Eclipse + ADT plugin
2. Android SDK Tools
3. Android Platform-tools
4. The latest Android platform
5. The latest Android system image for the emulator

Yes there are also possible ways if you want to use existing version of Eclipse or any other IDE.

- **Setting Up the ADT Bundle:**

As you have downloaded ADT bundle, follow below steps to setup it:

1. Unpack the ZIP file named “adt bundle osplatform.zip ” and save it to an appropriate location such as a “Development” directory in your home directory.
2. Open the adt bundle osplatform goto eclipse and next directory and launch eclipse.

6.2 Code for the k-means clustring

```
public BufferedImage calculate(BufferedImage image,
                                        int k, int mode)
       {
             long start = System.currentTimeMillis();
             int w = image.getWidth();
                  int h = image.getHeight();
             clusters = createClusters(image,k);
             int[] lut = new int[w*h];
             Arrays.fill(lut, -1);
             boolean pixelChangedCluster = true;
             int loops = 0;
             while (pixelChangedCluster)
            {
                pixelChangedCluster = false;
                loops++;
                for (int y=0;y<h;y++)
               {
                    for (int x=0;x<w;x++)
                   {
                        int pixel = image.getRGB(x, y);
                        Cluster cluster = findMinimalCluster(pixel);
                        if (lut[w*y+x]!=cluster.getId())
                       {
                            if (mode==MODE_CONTINUOUS)
                            {
                                  if (lut[w*y+x]!=-1)
                                  {
                                         // remove from possible previous
                                        // cluster
                                       clusters[lut[w*y+x]].removePixel( pixel);
                                  }
                                 // add pixel to cluster
                                 cluster.addPixel(x,y,pixel);
                             }
                             // continue looping
                             pixelChangedCluster = true;

                             // update lut
                              lut[w*y+x] = cluster.getId();
                         }
                     }
                }
                if (mode==MODE_ITERATIVE)
                {
                    // update clusters
                   for (int i=0;i<clusters.length;i++)
                   {
                       clusters[i].clear();
                   }
                   for (int y=0;y<h;y++)
                  {
                     for (int x=0;x<w;x++)
```

```
                {
                      int clusterId = lut[w*y+x];
                      // add pixels to cluster
                      clusters[clusterId].addPixel(x,y, image.getRGB(x, y));
                 }
             }
         }
     }
     // create result image
     BufferedImage result = new BufferedImage(w, h,
                                BufferedImage.TYPE_INT_RGB);
     for (int y=0;y<h;y++)
     {
         for (int x=0;x<w;x++)
         {
             int clusterId = lut[w*y+x];
             result.setRGB(x, y, clusters[clusterId].getRGB());
         }
     }
     long end = System.currentTimeMillis();
     System.out.println("Clustered to "+k
                        + " clusters in "+loops
                        +" loops in "+(end-start)+" ms.");
     return result;
 }
   public Cluster[] createClusters(BufferedImage image, int k)
{
     Cluster[] result = new Cluster[k];
     int x = 0; int y = 0;
     int dx = image.getWidth()/k;
     int dy = image.getHeight()/k;
     for (int i=0;i<k;i++)
     {
         result[i] = new Cluster(i,image.getRGB(x, y));
         x+=dx; y+=dy;
     }
     return result;
  }

 public Cluster findMinimalCluster(int rgb)
{
     Cluster cluster = null;
     int min = Integer.MAX_VALUE;
     for (int i=0;i<clusters.length;i++)
     {
         int distance = clusters[i].distance(rgb);
         if (distance<min)
        {
             min = distance;
             cluster = clusters[i];
        }
   }
     return cluster;
 }
```

Description:

In this algorithm user first capture the fruit image either by camera or by the gallary . And pick the color pixel of defected area. Then click on the clustering result button. Then we got the result of defected area in percentage.

6.3 Code for hassu algorithm

```
public class HassuAdmin
{

      public static BufferedImage resizeImage(BufferedImage originalImage, int type)
      {
         int IMG_WIDTH=250,IMG_HEIGHT=250;
        BufferedImage resizedImage = new BufferedImage(IMG_WIDTH, IMG_HEIGHT, type);
        Graphics2D g = resizedImage.createGraphics();
        g.drawImage(originalImage, 0, 0, IMG_WIDTH, IMG_HEIGHT, null);
        g.dispose();
        return resizedImage;
      }
      public static double calculate(BufferedImage image , int cl)
     {
           int  width = image.getWidth();
           int height = image.getHeight();
           int count = 0;
           int redmatrix[][] = new int[height][width];
           int greenmatrix[][] = new int[height][width];
           int bluematrix[][] = new int[height][width];
           for(int i=0; i<height; i++)
           {
               for(int j=0; j<width; j++)
               {
                   count++;
                   Color c = new Color(image.getRGB(j, i));
                   redmatrix[i][j] = c.getRed();
                   greenmatrix[i][j] = c.getGreen();
                   bluematrix[i][j] = c.getBlue();
               }
           }
           SD s = new SD();
           ArrayList<Range> t = s.getRange(0,255);
           double redsum = s.getStepDeviation(t,redmatrix,cl);
           double greensum = s.getStepDeviation(t,greenmatrix,cl);
           double bluesum = s.getStepDeviation(t,bluematrix,cl);
           return redsum+greensum+bluesum;
    }
   public static void main(String a[])
   {
       try
       {
               BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
```

```
                System.out.println("Enter Dataset Path");
                String path = br.readLine();
                System.out.println("Enter No Of Class");
                int cl = Integer.parseInt(br.readLine());
                File dFolder = new File(path);
                File []files = dFolder.listFiles();
                for ( File f : files)
                {
                    BufferedImage src = ImageIO.read(f);
                    BufferedImage dst = resizeImage(src, BufferedImage.TYPE_INT_RGB);
                    System.out.println(calculate(dst,cl));
                }
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

Description:

Capture the fruit image and then click on the Hassu Algorithm button. Then we got result in quailty like Un-ripe, Partially-ripe, Ripe, Over-ripe.

Chapter 7

TESTING

7.1 What is software testing

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test.[1] Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs (errors or other defects). It involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

7.1.1 Black box testing

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well.

7.1.2 White box testing

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming

skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT).

White-box testing can be applied at the unit, integration and system levels of the software testing process. Although traditional testers tended to think of white-box testing as being done at the unit level, it is used for integration and system testing more frequently today. It can test paths within a unit, paths between units during integration, and between subsystems during a systemlevel test. Though this method of test design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specification or missing requirements.

7.1.3 Unit testing

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. It is also known as component testing. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

7.1.4 Test cases

Test cases are built around specifications and requirements, i.e., what the application is supposed to do. Test cases are generally derived from external descriptions of the software, including specifications, requirements and design parameters. Although the tests used are primarily functional in nature, non-functional tests may also be used. The test designer selects both valid and invalid inputs and determines the correct output without any knowledge of the test object's internal structure.

TABLE 7.1: Test cases.

TC_ID	**TestCases**	**Expected Result**	**Actual Result**	**Remark**
1	Noise Reduction	Application reduce the noise present in the image	Image is free from noise	Better
2	Detect Region Of Interest	Application should compute ROI	Detected ROI	Better
3	Quantify defected area	Application identify the defect of fruit.	Detected defect on each image	Better
4	Find the shape of defect	Application find out the shape of defect.	Shape is computed like	Better
5	Determine the color of defect	Application determine the color of the defect.	Determine the defect.	Better
6	Give 5 Un-ripe Fruits	Application give result of 5 Un-ripe images.	Result is 4 un-ripe images.	Good
7	Give 5 Ripe Fruits	Application give result of 5 Ripe images.	Result is 4 Ripe images.	Good
8	Give 5 Partially-ripe Fruits	Application give result of 5 Partially-ripe images.	Result is 3 Partially-ripe images.	Good

Continued on next page

Table 7.1 — continued from previous page

TC_ID	TestCases	Expected Result	Actual Result	Remark
9	Give 5 Over-ripe Fruits	Application give result of 5 Over-ripe images.	Result is 2 over-ripe images.	Good
10	Hassu Algorithm	Performance is 100%	Performance is 80%	Good

Chapter 8

SNAPSHOTS

8.1 Windows application

8.1.1 Server running

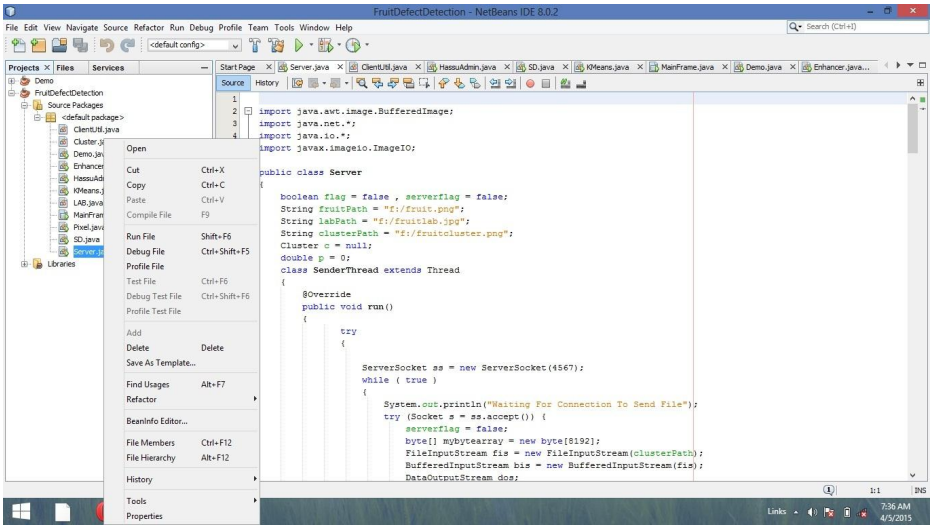

FIGURE 8.1: Server Running.

At this we have to run the server program.

8.1.2 Frame

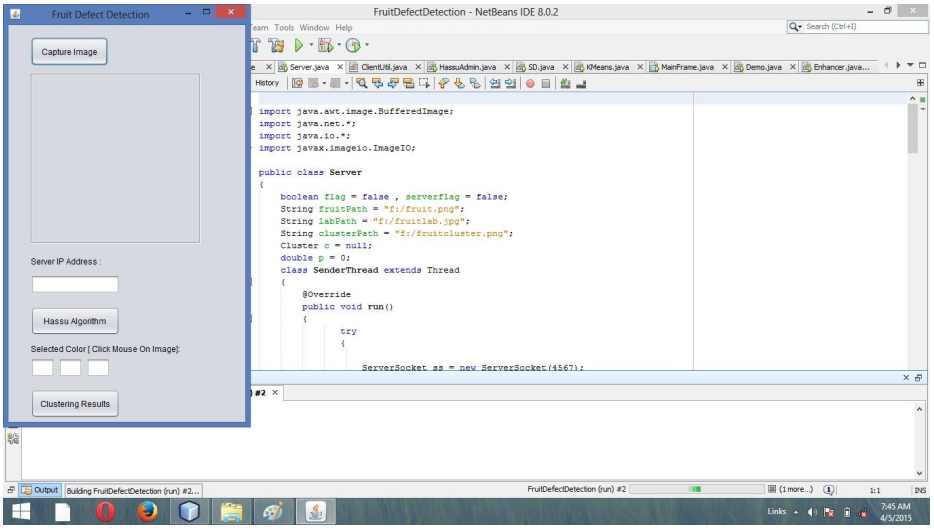

FIGURE 8.2: MainFrame

First run the server program then run the mainframe program after running this two program we get the frame.

8.1.3 Capture image

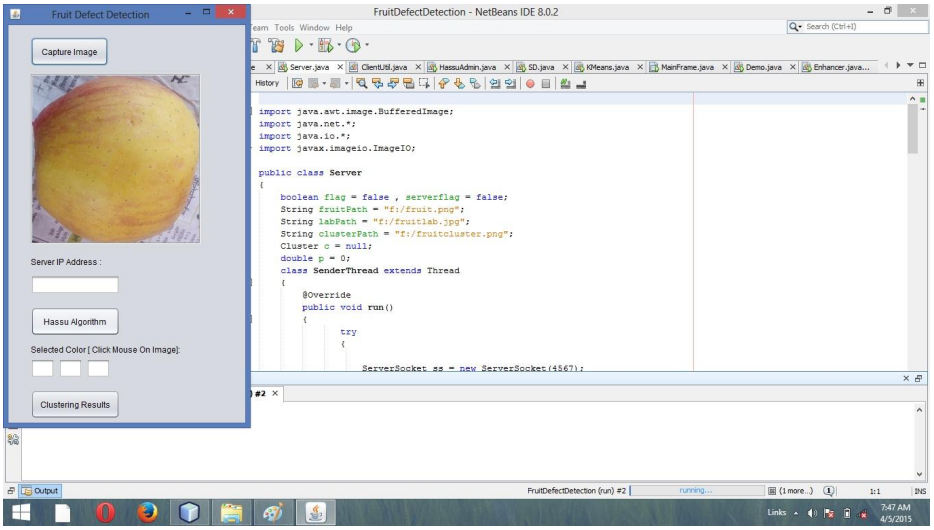

FIGURE 8.3: Capture Image

After displying mainframe we have to click on the button capture image.

8.1.4 Unriped result image

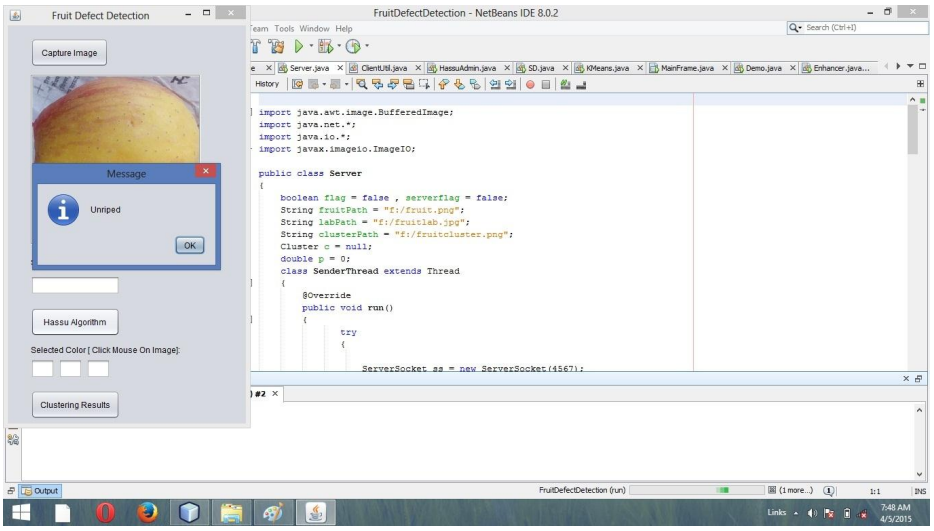

FIGURE 8.4: Unriped Image

After getting image we have to click on button hassu algorithm then we will get result unripe.

8.1.5 Partialy riped result image

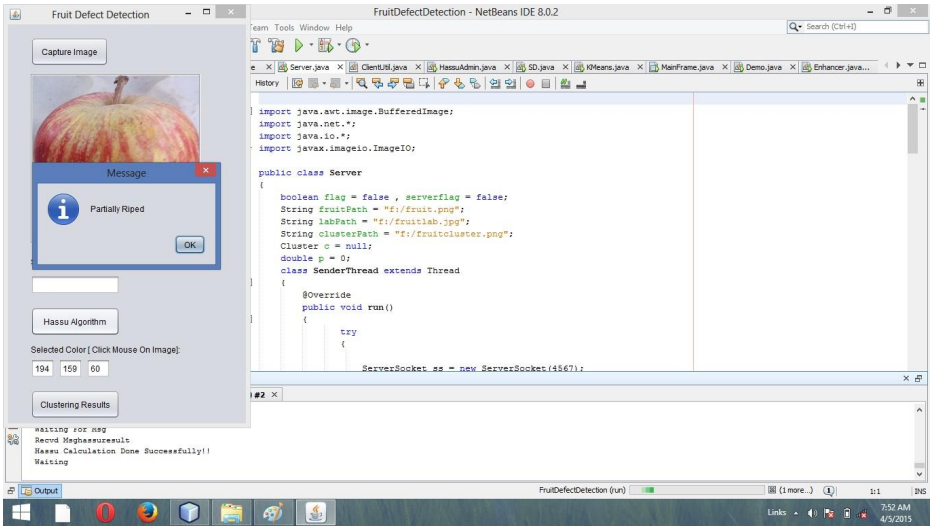

FIGURE 8.5: Partialy Riped Image

After getting image we have to click on button hassu algorithm then we will get result partialy ripe.

8.1.6 Riped result image

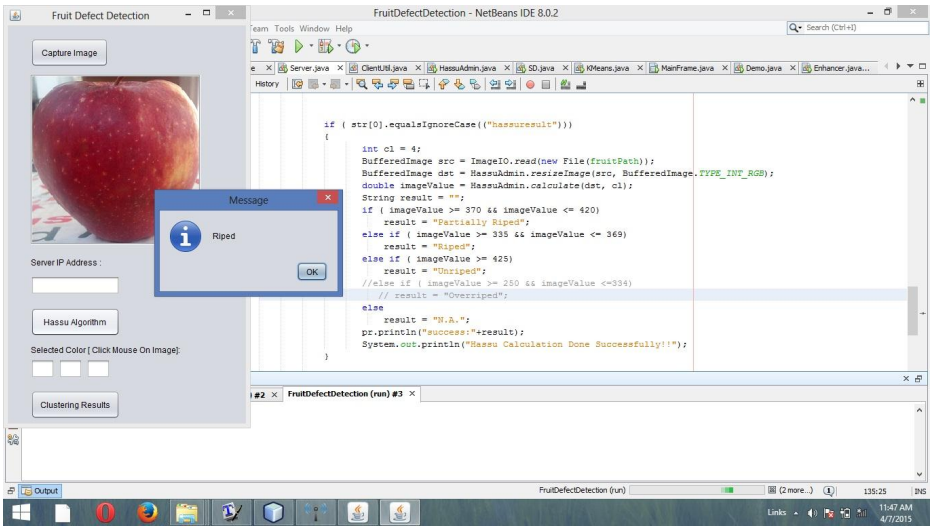

FIGURE 8.6: Riped Image

After getting image we have to click on button hassu algorithm then we will get result ripe.

8.1.7 Over riped result image

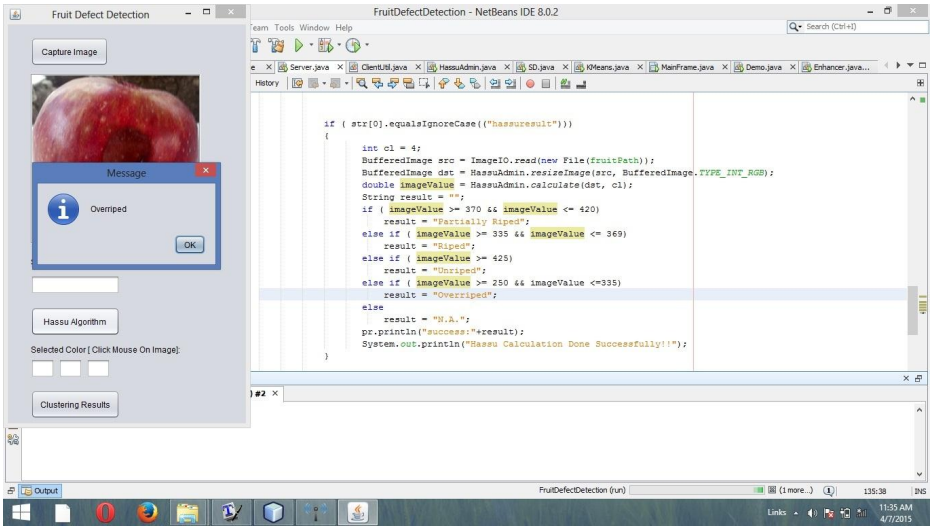

FIGURE 8.7: Over Riped Image

After getting image we have to click on button hassu algorithm then we will get result Over ripe.

8.1.8 K-means clustering

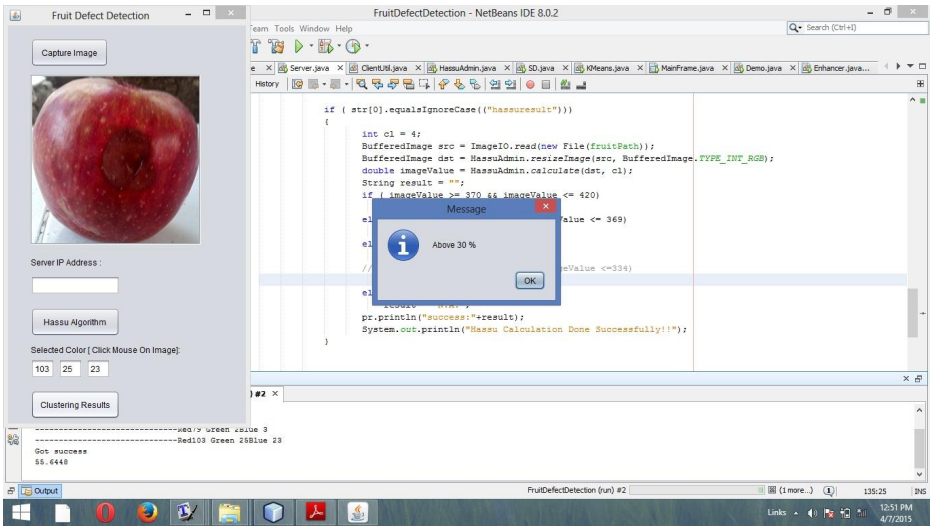

FIGURE 8.8: Result Image

Another option is clustering algorithm as per choice we can use. we have to click on clustering algorithm then we will get clustered result.

8.1.9 Clustered image

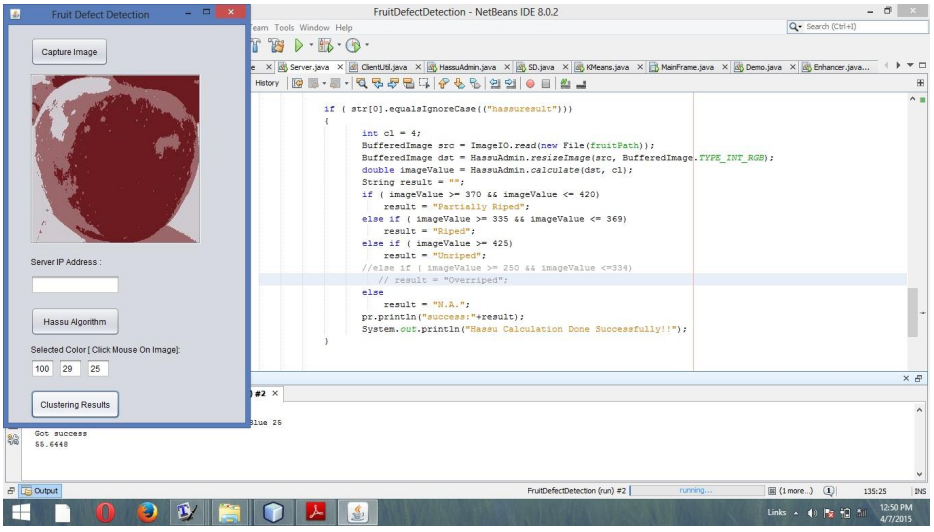

FIGURE 8.9: Clustered Image.

8.2 Android application

8.2.1 Frame

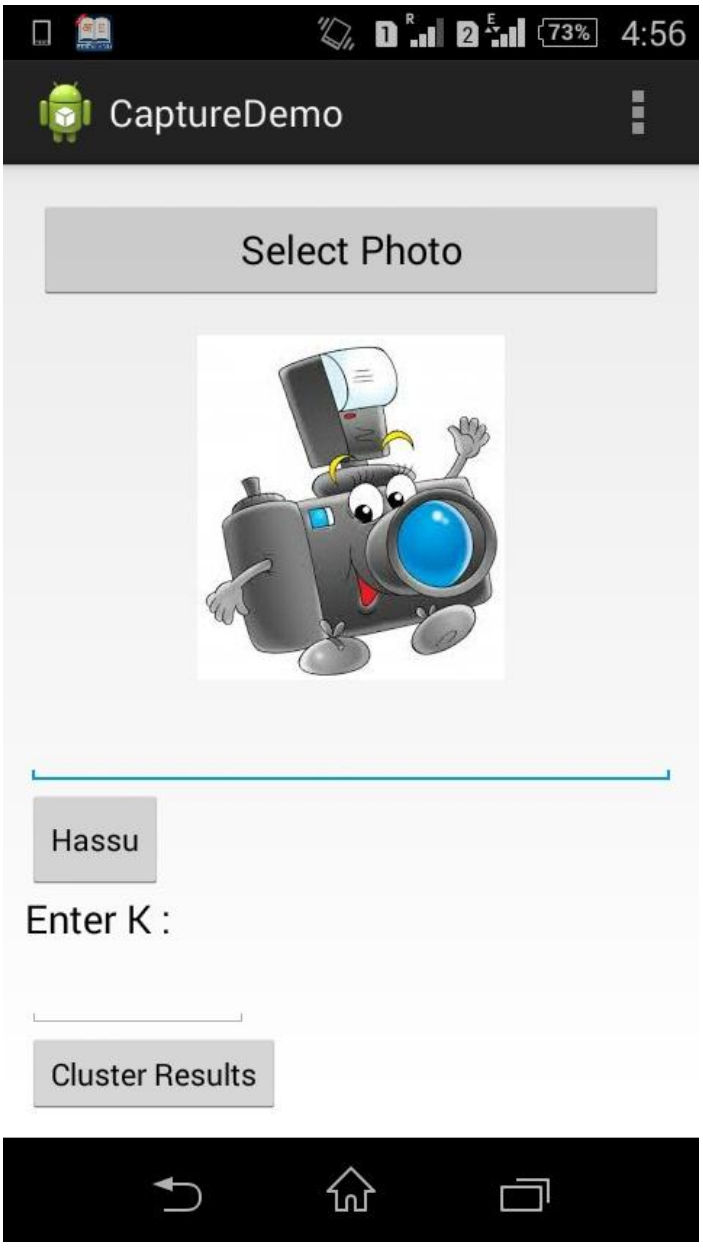

FIGURE 8.10: Result image

This is the frame of android application.

8.2.2 Select photo frame

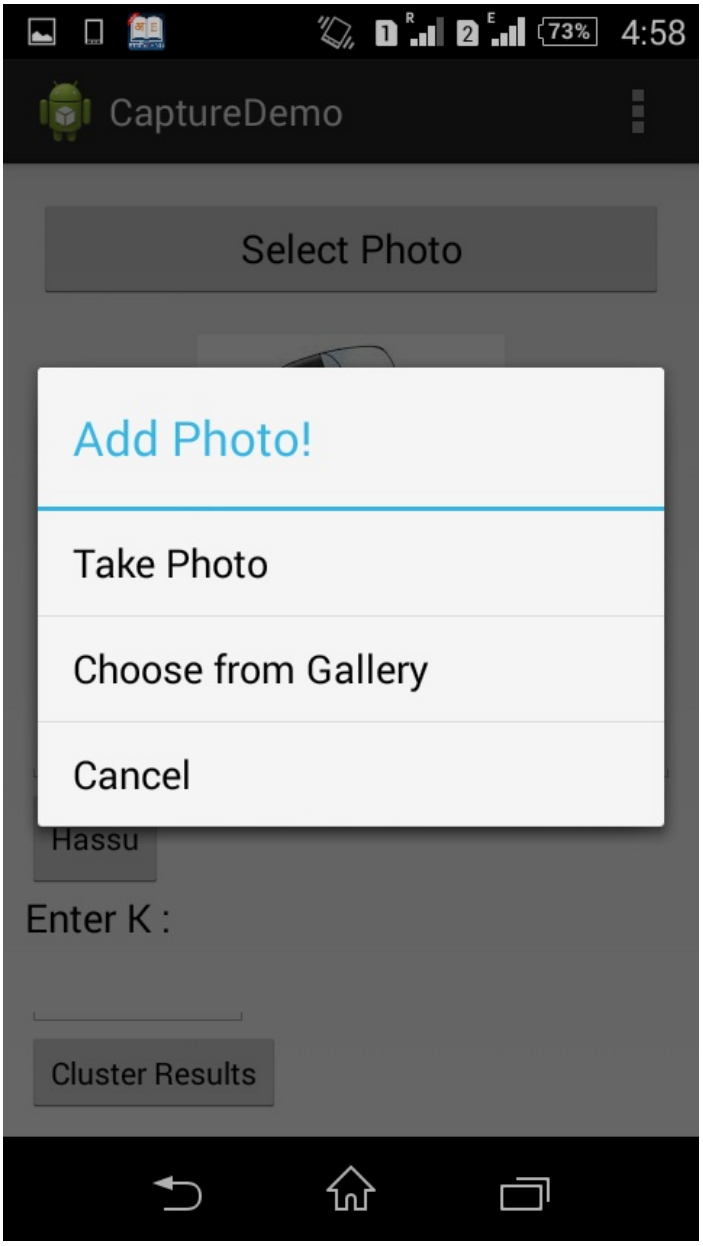

FIGURE 8.11: Capture image frame

After click on select photo We have to select photo either by camera or by taking from galary.

8.2.3 Hassu result image

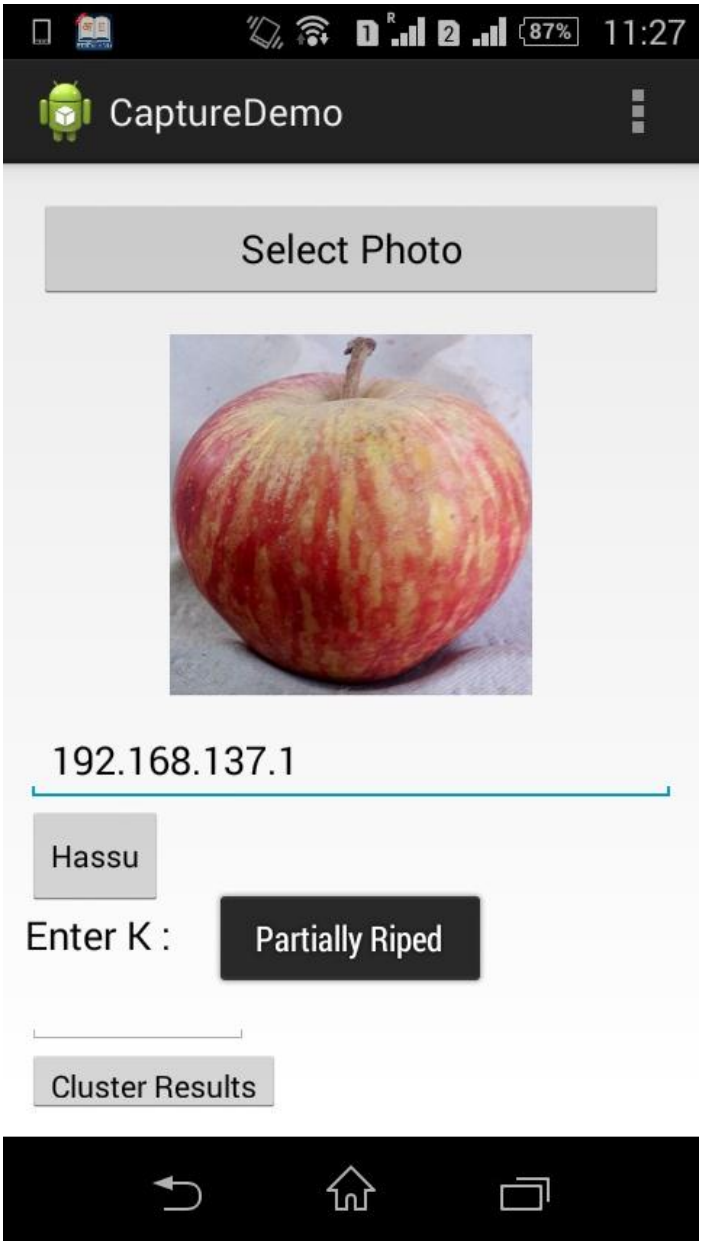

FIGURE 8.12: Result image

After capturing image give servses IP address and click on the button Hassu.We will get result.

8.2.4 Clustering image

FIGURE 8.13: Clustering image

Another option is Cluster result We have to enter no of k.

8.2.5 Cluster result image

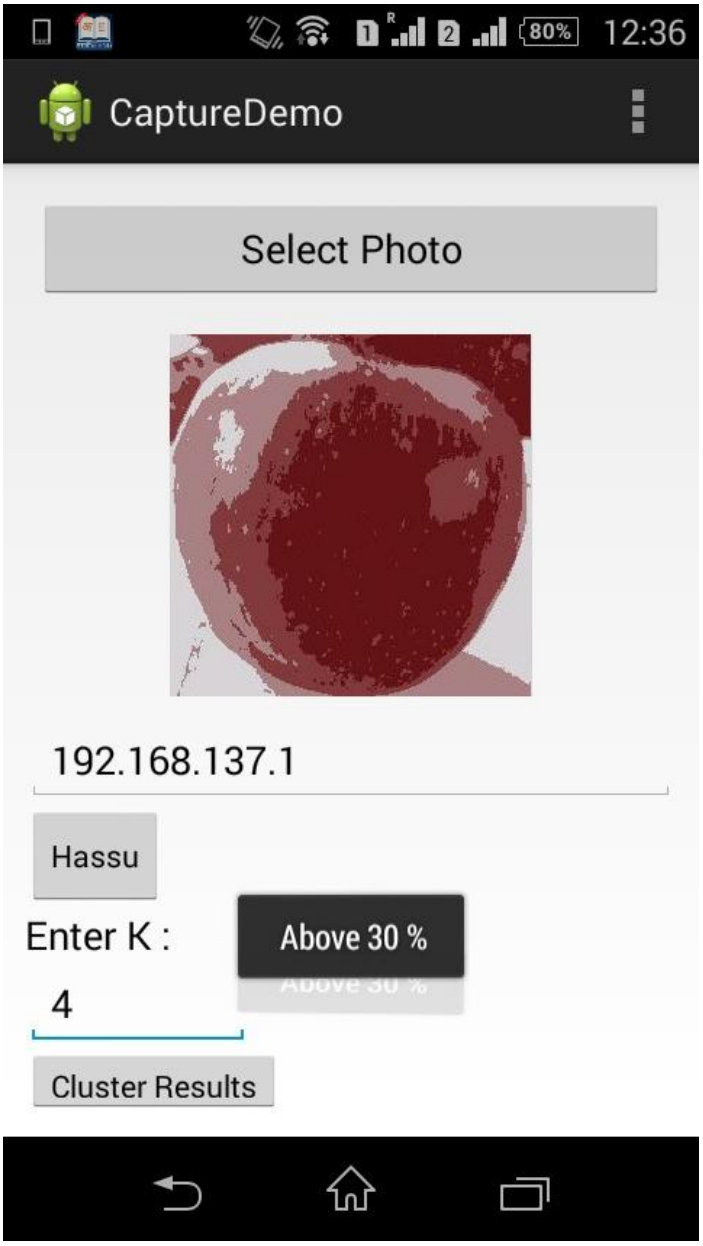

FIGURE 8.14: Result image

Then click on cluster image We will get clustered image and result.

Chapter 9

CONCLUSION

9.1 Conclusion

The proposed system in this paper ,clasifies the fruits as ripe,unripe,partially ripe,overripe.The system uses the K-means clustering tecnique for segmenting defect with three or more cluster. Here,we have used apples as a case study.Experimental result suggest that the the proposed approach is able to accurately segment the defected area of fruits present in the image. The approach given 80 persent of accuracy in classifying the image as ripe, unripe, partially ripe and overripe.

Bibliography

[1] 2. Hassan Sardar. A role of computer system for comparative analysis using image processing to promote agriculture business.

[2] Nishant Singh Jay Prakash Gupta 1.Shiv Ram Dubey, Pushkar Dixit. Infected fruit part detection using k-means clustering segmentation technique.

[3] 3. N. R. Pal and S. K. Pal. A review on image segmentation techniques.

[4] Clustering by competitive agglomeration.