Wollo University

Kombolcha Institute of Technology (KIoT)

College of Informatics

Department of Software Engineering

We here by Submitted a project in Edible and Non-Edible Fruit Recognition System in partial fulfilment of Bachelor's Degree in Software Engineering

Submitted to the Departments of Software Engineering

Group Members

No.	**Name**		**ID Number**
1.	Misganaw	Getahun	0586/10
2.	Yossef	Kassay	0636/10
3.	Wondwosen	Tedla	0623/10

Advisor's Name: **Bihonegn Abebe (MSc)** Signature**: ________________**

Submission Date: -13/11/2014 E.C

Kombolcha, Wollo, Ethiopia

Discriminations

We hereby declare that our project in titled Edible and Non-Edible Fruit Recognition System is original and not submitted/Published by any individual/ Organization.

Group members

S.N.	Name List	ID Number	Date	sign:
1.	Misganaw Getahun	0586/10	----/----/2022	__________
2.	Yossef Kassay	0636/10	----/----/2022	__________
3.	Wondwosen Tedla	0623/10	----/----/2022	__________

Advisor's Name: Bihonegn Abebe (MSc) Date: ---------/----/2022 Signature: __________

Approval of the Board of Examiners

We, the undersigned, members of the board of examiners of the final open defense by Yossef Kassay, Wondwosen Tedla and Misganaw Getahun read and evaluated their project entitled "**Edible and Non-Edible Fruit Recognition System**" and examined their candidates. This is therefore, to certify that the project has been accepted in partial fulfillment of the requirement of the degree of Bachler in software Engineering.

Bihonegn Abebe	___________	____________
(advisor)	Signature	Date
________________	___________	____________
(Chairperson)	Signature	Date
________________	___________	____________
(Examiner 1)	Signature	Date
________________	___________	____________
(Examiner 2)	Signature	Date
________________	___________	____________
(Examiner 3)	Signature	Date
________________	___________	____________
(Examiner 4)	Signature	Date

Acknowledgement

First and foremost, we would like to thanksgiving the almighty GOD for his priceless blessings throughout the process of developing this project. Secondly, we want to express our deepest gratitude to our advisor Mr. Bihonegn Abebe (MSc). Without his continuous and perpetual follow up and advising this project was not come real. Finally, we want to give deepest gratitude and glory to all people who have helped us in achieving our project.

Table of Contents

List of Tables

List of Figures

Abstract

Fruit classification using a machine learning is one of the most promising applications in personal computer vision. Profound learning-based characterizations are making it possible to recognize fruits from pictures. In this project, Edible and Non-Edible Fruit Recognition framework is proposed. The proposed strategy utilizes SVM learning methods for the grouping. The model uses the fruit shape and feature in order to recognize each fruit images. We have used the Fruits-10 dataset for the evaluation purpose. An exact and dependable image-based Edible and Non-Edible Fruit Recognition system is crucial for supporting pomologists, foreign guests, and System users.

Chapter One: Introduction

1.1 Introduction

Fruits are generally known as part of plant components harvested from fruit farms in the field of agriculture. Classifications of the plant fruit is the key to agricultural product. The studies of the plant fruit mean the studies of visually observable patterns seen on the plant fruit. It is very difficult to classify the plant fruits manually because it requires prior Knowledge. It requires tremendous amount of work, expertise in the plant fruit, and also require the excessive processing time. Hence, image processing and Machine learning techniques are used for the detection and classification of plant fruits. fruit detection and classification involve the steps like image acquisition, image pre-processing, image segmentation, feature extraction and classification. Fruits can be categorized into several types or classes, which includes apples and potato, lemon, carrot, Oliver as well as tomatoes and avocados. Most of these fruits are usually safe to consume by human or animals, but the others have some side effects for human beings. Fruit is grown as it has many vitamins and necessary nutrients for human bodies. Fruits, commonly consumed in daily diets, are a major source of anti-oxidants. They are the fleshy products of plant growth and the majority of their flesh parts are edible and some other are not edible.

1.2 Background

Knowledge about an edible and a non-edible fruit has a high direct-use value. Yet, little is known about factors shaping the distribution and transfer of knowledge of edible and non-edible fruits and there is a concern that use knowledge about edible and non-edible fruits is decreasing. knowing how to find and identify edible and non-edible fruit is an incredibly valuable skill. There are a number of several traditional ways for determining whether the fruits are edible or not. They were distinguishable through colors, shapes, tastes, reading guidebooks, Smells, asking former elder persons. But this is not accurate enough and time-consuming work. There for in order to find a remedy for this problem, it is necessary to develop an automated edible and non-edible fruit recognition system using the concepts of machine learning and image processing algorithms. Image processing in agriculture has been applied in the areas of detection and recognizing the products.

In agriculture field recognizing edible and non-edible fruits play an indispensable role to estimate the nature of fruits. Estimating edible and non-edible fruits in a farm using human power is a very tiring job and need plenty of time to complete the task. The image processing techniques can help to perform accurately this task. The automated edible and non-edible fruit recognition system can predict whether they are edible or not.

1.3 Existing System of the project

There are different types of fruit categories in our mother land Ethiopia and the world. In spite of the fact that there is no any automated system before in order to identify whether the given fruit is edible or not. However, the consumers of the fruit use different mechanisms so as to identify the edibility or non-edibility nature of fruits manually. For instance, by asking former aged persons, agricultural expert፣ by smelling, by tasting፣ inherited by culture፣by reading guidebooks.

1.4 Motivation

In the existing system there are a number of challenges that are difficult to handle easily. Some of these are Outsiders or foreigners are not easily identified with indigenous edible and non-edible fruits, immature children cannot distinguish between edible and non-edible fruits, Blind people cannot distinguish between edible and inedible fruits. Therefor in order to tackle these problems, we are eager to develop an automated system that classifies the group of fruits as edible and non-edible. The other motivation of conducting this project is to help people and provide a guidance to them who is having an issue in identifying edible and non-edible fruits through machine learning methods. Since now there was no standard reference or interpretation given by machine learning that whether a fruit is edible or non-edible, therefore this project was to build a machine learning model that can identify or evaluate the fruits nature.

1.5 Problem Statement

Plant fruits play a pivotal role for the growth of our body simultaneously there are also some parts of fruits which are toxic, serves only medical purpose and causes a connotative welfare issue to our body. The ways of classifying them as edible or non-edible manually is not accurate enough. Because the methods used by the consumers to identify the edibility and non-edibility nature of the fruit is tedious and somewhat perplexing. By eating what you do not eat, the consumers welfare is being under question. The manual ways of identifying them as edible and non-edible was time consuming and requires a great effort of labor force. Lack of knowledge is the other bottleneck in the existing system since the consumers use the above-mentioned methods in the existing system to identify them. The methods are not full of knowledge about fruits. People may interpret the non-edible fruit to a still edible one or a partially non-edible fruit to an edible fruit. There was no absolute measurement for consumers to determine the current edible states of fruits, Outsiders or foreigners are not easily identified with indigenous edible and non-edible fruits, immature children cannot distinguish between edible and non-edible fruits, blind people cannot also distinguish between edible and non-edible fruits.

1.6 Proposed System

The new system that we are going to propose is to minimize and to discard or remove as much as possible the problem that faced in existing system. Therefore, our system is easy and simple to use and it saves wastage of time, and resource. To overcome the problems that arises in the existing system, the proposed system will add here the following functions.

- Design interactive system that is integrated and accessible.
- The proposed system automatically shows the fruits category whether they are edible or non-edible.
- After training, the model is able to estimate the fruits based on the knowledge the model has acquired without the aid of human beings.

1.7 Objective of the project

1.7.1 General objective

The general objective of our project is to develop an automated edible and non-edible fruit recognition system using appropriate machine learning algorithms.

1.7.2 Specific objective

In order to achieve the general objective of our project, we have performed the following tasks.

- To gather relevant information about the existing system.
- To acquire and segment fruits by using image processing techniques.
- To develop the recognition model using a pre-trained machine learning algorithm.

- To develop a fruit recognition and classification model and contribute to the existing literature.
- To test the performance and reliability of the model.
- To identify and document edible and non-edible fruits
- To draw conclusions and forward recommendations.

1.8 Scope and limitation of the project

1.8.1 Scope of the Project

In this project we are going to design and implement robust, scalable, and interactive system. Scope of project revolves around recognition of fruits, transmitting it in an auditory voice and auditory fruit recognition.

1.8.2 Limitation of the project

The algorithm developed for this project is not able to detect the fruits which are a dark place and under the shade. The algorithm only detects the fruit with controlled illumination and direction of sun light to obtain an accurate result by automated detection. The system cannot detect whether the fruit is healthy or not. If the image exhibit information loses such as surface damage, noise level, sharpness issues and so on, the recognition may fail and the user has to do the processing again. Our system also cannot classify all groups of fruit species as edible and non-edible

1.9 Methodology for the project

1.9.1 Data collection and fact-finding techniques

1.9.1.1 Interview

We have asked informants in following manner:

- Can you show the plants you use in your region?
- Can you tell the local names of the plants you use in your region?
- Which parts of the plants do you use?
- How do you consume it?

1.9.1.2 Dataset

The below steps are followed in the process of collecting the datasets

Step 1: Define the objective of the Problem Statement

At this step, we must understand what exactly needs to be predicted. In our case, the objective is to predict the possibility of fruit by studying their edibility and non-edibility nature. It is also essential to take mental notes on what kind of data can be used to solve this problem or the type of approach we must follow to get to the solution.

At this stage, we have asking questions such as,

- What kind of data is needed to solve this problem?
- Is the data available?
- How can we get the data?

Once we know the types of data that is required, we have understood how we can derive this data. Coming back to the problem at hand, the data needed for our system includes combination of edible and non-edible fruits. Such data must be collected and stored for analysis.

Step 2: Data Preparation

The data we collected is almost never in the right format. We will encounter a lot of inconsistencies in the data set such as missing values, redundant variables, duplicate values, etc. Removing such inconsistencies is very essential because they might lead to wrongful computations and predictions. Therefore, at this stage, we scan the data set for any inconsistencies and we fix them then and there.

Step 3: Exploratory Data Analysis

This stage is all about finding all the hidden data mysteries or features. EDA or Exploratory Data Analysis is the brainstorming stage of Machine Learning. Data Exploration involves understanding the patterns and trends in the dataset. At this stage, all the useful insights are drawn and correlations or relationships between the variables are understood.

Step 4: Building a Machine Learning Model

All the insights and patterns derived during Data Exploration are used to build the Machine Learning Model. This stage always begins by splitting the data set into two parts, training data, and testing data. The training data will be used to build and analyze the model while the testing dataset is used to test the knowledge of the model. The logic of the model is based on the Machine Learning Algorithm that is being implemented in our case Support vector machine is the selected algorithm. The output will be in the form of Edible (if the fruit is edible) or non-Edible (if the fruit is non edible).

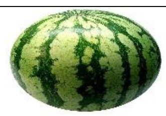
Step 5: Model Evaluation & Optimization

After building a model by using the training data set, it is finally time to put the model to a test. The testing data set is used to check the efficiency of the model and how accurately it can predict the outcome. Once the accuracy is calculated, any further improvements in the model can be implemented at this stage.

Step 6: Predictions

Once the model is evaluated and improved, it is finally used to make predictions. The final output can be a Categorical variable (edible or non-edible). For this project we have used public dataset for fruit detection and classification. The dataset consists of 10 selected classes of fruit and among those we are going to use 6 edible and 4 non edible fruits for experimentation of our algorithm (About Us: A. Google Corporation, n.d.).

Table 1: Fruit Species used in the project

No	Fruit Name	Icon for the Image	No of Images	Category
1	cactus		1000	Edible
2	Watermelon		1000	Edible
3	peach		1000	Edible
4	Date		1000	Edible

5	Lime		1000	Edible
6	Mango		1000	Edible
7	Bitterball		1000	Non-Edible
8	Arnchata		1000	Non-Edible
9	Jafra		1000	Non-Edible
10	Gumero		1000	Non-Edible

1.9.1.3 Image Pre-Processing

Data pre-processing is a process of cleaning the raw data i.e., the data is collected in the real world and is converted to a clean data set. In other words, whenever the data is gathered from different sources it is collected in a raw format and this data isn't feasible for the analysis. Therefore, certain steps are executed to convert the data into a small clean data set, this part of the process is called as data pre-processing. This involves the transformation of raw data into an understandable format suitable to train a model. The aim of image pre-processing is enhancing image features relevant for further processing and analysis task.

1.9.1.3.1 Types of Image Pre-Processing Techniques

Pixel brightness transformations/ Brightness corrections

Modify pixel brightness and the transformation depends on the properties of a pixel itself. In PBT, output pixel's value depends only on the corresponding input pixel value. Examples of such operators include brightness and contrast adjustments as well as color correction and transformations.

Geometric Transformations

The positions of pixels in an image are modified. Geometric transformation permits the elimination of geometric distortion that occurs when an image is captured.

- Scaling: Scaling is just resizing of the image
- Translation: Translation is the shifting of object's location
- Rotation: Just rotating an object with theta degrees

Image Segmentation

Image segmentation is a commonly used technique in image processing and analysis to partition an image into multiple parts or regions, often based on the characteristics of the pixels in the image. Image segmentation could involve separating foreground from background, or clustering regions of pixels based on similarities in color or shape.

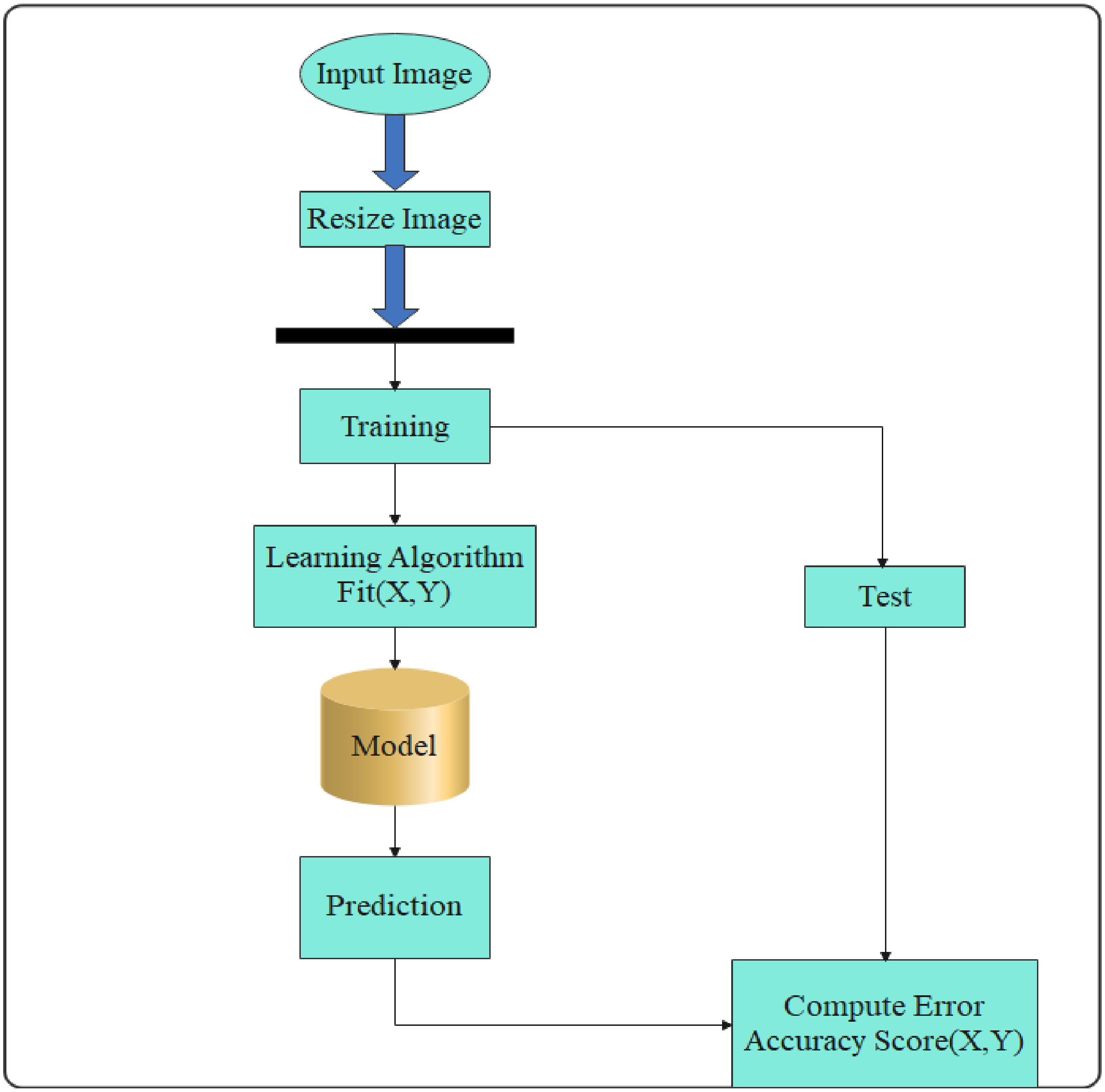


Figure 1: Steps for data preprocessing

1.9.1.4 Algorithm used in the project

Naive Bayes (NB) Classifier

Naive Bayes (NB) allows constructing simple classifiers based on Bayes' theorem. Thus, it assumes that any feature value is independent of the value of the other features**.** NB models can accomplish high levels of accuracy while estimating the class-conditional marginal densities of data. Because of the independence assumption, NB doesn't need to learn all possible correlations between the features. Thus, NB classifiers can learn easier from small training data sets due to the class independence assumption.

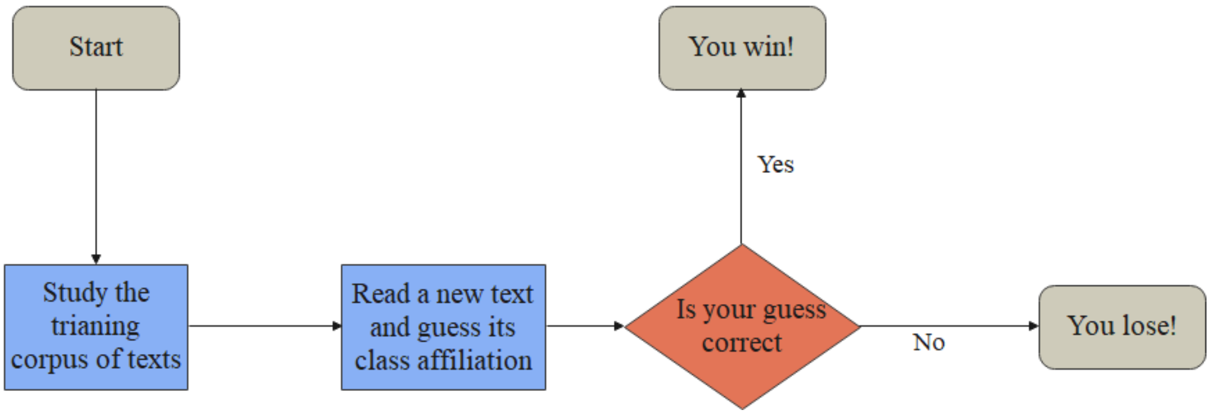


Figure 2: General flow of NB algorithm

The runtime complexity of the Naïve Bayes classifier is O(NK), where N is the number of features and K is the number of label classes.

Support Vector machine

SVM draws a decision boundary which is a hyperplane between any two classes in order to separate them or classify them.

- Support Vectors: -Support vectors are the data points, which are closest to the hyperplane. These points will define the separating line better by calculating margins. These points are more relevant to the construction of the classifier.
- Hyperplane: -A hyperplane is a decision plane that separates between a set of objects having different class memberships.
- Margin: -A margin is a gap between the two lines on the closest class points. This is calculated as the perpendicular distance from the line to support vectors or closest points. If the margin is larger in between the classes, then it is considered a good margin, a smaller margin is a bad margin.

How does SVM work?

The main objective is to segregate the given dataset in the best possible way. The distance between the both nearest points is known as the margin. The objective is to select a hyperplane with the maximum possible margin between support vectors in the given dataset. SVM searches for the maximum marginal hyperplane in the following steps:

- Generate hyperplanes that segregate the classes in the best way.
- Select the right hyperplane with the maximum segregation from the both nearest data points as shown in the figure.

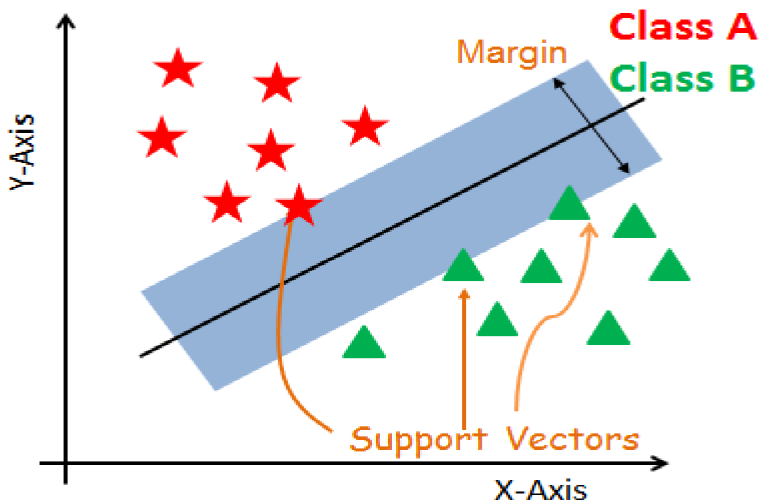


Figure 3: General flow of SVM algorithm

It can easily handle multiple continuous and categorical variables. SVM constructs a hyperplane in multidimensional space to separate different classes. SVM generates optimal hyperplane in an iterative manner, which is used to minimize an error. The core idea of SVM is to find a maximum marginal hyperplane (MMH) that best divides the dataset into classes.

How Do Both Methods Compare?

Naive Bayes (NB) is a very fast method. It depends on conditional probabilities, which are easy to implement and evaluate. Therefore, it does not require an iterative process. NB assumes that features are independent between them, but this assumption does not always hold. Even though, NB gives good results when applied to short texts. For some datasets, NB may defeat other classifiers using feature selection.

SVM is more powerful to address non-linear classification tasks**.** SVM generalizes well in high dimensional spaces like those corresponding to texts. It is effective with more dimensions than samples. It works well when classes are well separated. SVM is a binary model in its conception, although it could be applied to classifying multiple classes with very good results. SVM takes a long time while train large datasets.

Both NB and SVM allow the choice of kernel function for each and are sensitive to parameter optimization. Comparing the accuracy of SVM and NB in spam classification showed that the basic NB algorithm gave the best prediction results (97.8%). At the same time, SVM and NB algorithms obtained an accuracy well above 90% using parameter passing when required. Therefor the type of machine learning algorithm that we have selected for our project is Support Vector Machines because SVM is a very good algorithm than NB since NB assumes that any feature value is independent of the value of the other features.

1.9.2 Technology requirements

The technology requirements that we have used in our project broadly classified into two distinctive groups.

1.9.2.1 Software requirements

The software/platform involved in this project development:

- Operating System: Windows 7 and above.
- Programing language: Python 2.7 and above.
- Supporting libraries: Tensorflow, OpenCV, PIL, tkinter, os, SKlearn etc.
- WonderShare EdrawMax
- Enterprise Architect
- Mockup plus
- TLite

1.9.2.2 Hardware Development Tools

The hardware involved throughout the implementation and testing of the project:

- Processor: 2.5 gigahertz (GHz) frequency or above.
- RAM: A minimum of 4 GB of RAM.
- Hard disk: A minimum of 20 GB of available space.
- Input Device: High resolution camera
- Monitor: Minimum Resolution 1024 X 768.

1.10 Feasibility Study

Feasibility study is an important phase in the software development process of this project. Our project is feasible because it can fulfill the requirement of good software which are economical, technical, political, and operational feasible.

1.10.1 Operational Feasibility

This software system brings better achievement for the operations performed by the enterprise and for the customer by providing efficient automation and storage of work actor information, easy updating, detection and testing etc.

- Any user can operate with the system
- User friendly (the software for the proposed system is not difficult to use, manually consumers may make too many errors and avoid using it).
- The proposed system interface is understandable to the users.

1.10.2 Technical feasibility

Our project is technically feasible. Because everyone can access local or offline environment using training and testing datasets. It is also compatible with modern technology. The application Integrated with computers. The proposed application expected to be maintained and repaired easily, because the system is developed by familiar programming language (environment).

1.10.3 Economic Feasibility

In the manual system vast time and cost is required in order to now the decay nature of fruits, but the proposed system overcomes this problem by reducing the time required to gather clues about the decay nature of fruits.

1.10.4 Political Feasibility

In fact, nobody refuses our system because it deals about the fruit detection and recognition system. It is acceptable for all people and any other users

1.10.5 Schedule Feasibility

The schedule for this project will be feasible due to wealthy information exchange between the developing team, advisor and the department. The project team member expects the project to be completed on time without any delay.

Table 2: Schedule of the project

No	Task	Start date	Finishing date	Responsible member
Introduction and project selection phase				
1	Identify the problem	7/07/2014	15/07/2014	All members
2	Gathering information	15/07/2014	18/07/2014	All members
3	Requirement analysis	18/07/2014	19/07/2014	All members
4	Study of existing system and others	20/07/2014	1/08/2014	All members
System Requirement Specification				
1	System Requirements	02/08/2014	10/08/2014	All members
2	System Requirement Modeling	11/08/2014	20/08/2014	All members
Requirement Analysis modeling				
1	Constructing UML modeling of proposed system	21/08/2014	30/08/2014	All members
System Design				
1	Constructing proposed system Architecture and others	01/09/2014	30/09/2014	All members
2	Completing the final SRS document till chapter four	1/11/2014	13/11/2014	All members

1.11 Risk assessment strategy

1.11.1 Communication problem when collecting our dataset

There was a communication problem between the team members especially during the phase of requirement gathering. The team member finds a concurrent remedy for this problem by creating a safe space for communication, Communication with remote employees.

1.11.2 Schedule/Time

As we have been back to class after the country's political crisis, the educational schedule has been changed into short and short. As a result, time was the main constraint we’ve faced to do our project. Even though time has become our barriers to us we have find a solution to this challenge by working day and night as much as possible.

1.11.3 Lack of quality camera

Also, one of the risks that we faced was getting high quality camera that train the fruit layers in different direction with in high quality. By the time we have resist this problem using what we have (i.e., by using our mobile phone camera).

1.11.4 Getting quality data for model training

Data is very critical point in machine learning project. In this specific project also getting good and enough quality data can be risk due to different factors, managing this risk is worth to note.

1.12 Significance of the project

The major contribution of our project is that the developed model can recognize and classify varieties of fruits with high accuracy and reasonable computational time which will reduce the associated labor cost (the users travel a long distance in order to ask aged persons to know the edibility and non-edibility nature of fruits), time, and eliminate the possibility of human error for stakeholders. Increased productivity, clearer understanding of customers, fewer repetitive tasks for workers, and more advanced and human-like output.

1.12.1 Beneficiary of the project

- For visually impaired persons: - It plays a crucial role for those persons by generating sound for the edibility and non-edibility nature of fruits.
- Foreign guests፦ when these guests come in to our country for different purpose, they can easily use the system and easily identify which one is edible and which one is not edible.
- For System users which include customers, retail shops, supermarkets, and farmers who has no a clue concerning with the edibility and non-edibility nature of fruits.
- For developers by increasing our profit, knowledge, and by escalating our coordination and collaboration status among us.
- For pomologists (who studies about fruit nature).

1.13 Project Budget break-down and cost analysis

Table 3: Project Cost Breakdown

Project Cost Breakdown							
		Work cost			Material costs		Fixed costs
No	Task	Hours	Rate per hour	Cost per task	Items	Cost per item	
Month 1, Month 2, Month 3							
1	Requirement gathering	2 weeks	--------	200	PC	30000	
2	Analysis	1 week	-------	50	Flash	280	
					Smart Phone	9000	
3	Design	1 week	-------				
Subtotal		250			39,280		39,530
Month 4							
1	Implementation						
2	Testing						

1.14 Team composition of the project

Table 4: Team composition of the project

List of tasks	Role of group members		
	Misganaw Getahun	Yossef Kassay	Wondwosen Tedla
Requirement analysis and gathering (collecting datasets)	⬆		⬆
Design	⬆	⬆	
Writing the SRS document		⬆	⬆

Chapter Two: System Requirement Specification

2.1 Overview

we propose SRS document for the decay classification of plant fruits. We used SVM for feature extraction and for preparing the training set. Based on this training set, the model learns to detect the requirement specification. Even though this project is an experimental benchmark, we optimist that this approach may contributes to enhance the quality of SRS

2.1.1 Scope

The extent of this chapter is limited to: -

- Intended audience and suggested readings
- List of Acronyms, Abbreviations and Definitions used in the project
- Overall descriptions of software requirements
- General constraints
- Specific requirements
- External interface requirements
- System requirement modeling

2.1.2 Purpose

The overall purpose of this system is to give new knowledge to users by classifying fruits as edible and non-edible. The user uses the system and through this system the edibility and non-edibility nature of fruits is being known for them. The System helps to know different types of fruits whether they are edible or not. It helps to speed up the work. It reduced the manual work.

2.1.3 Intended Audience and Suggested Readings

The document is intended for requirements engineer, domain expert, developer, users and project manager. Before using the system, it is highly recommended to read the SRS Document to get an overview of the product.

2.1.4 List of Acronyms, Abbreviations and Definitions

- TLite➔Stands for Tensorflow lite.
- GUI ➔Stands for graphical user interface.
- SRS➔Software Requirements Specification
- UML➔Stands for Unified Modeling Language
- PC➔Stands for personal Computer
- NB➔Naive Bayes
- SVM➔Support vector machine
- EDR➔ Exploratory Data Analysis
- PVM➔Python Virtual Machine
- Edible fruit: -is a type of fruit that is edible and useful for the body.
- Non edible fruit: -is a type of fruit that is not edible and sometimes toxic to our body.

2.2 Overall Description of Software Requirements

2.2.1 Product Perspectives

The edible and non-edible fruit recognition system is a new system that is design to replace the manual system. Our system is a first version that overcomes the limitations of manual system. It is used to speed up the estimation of fruits.

2.2.2 User characteristics

Target user groups for the edible and non-edible fruit recognition system are:

Farmers

- Individual farmers
- Farmer groups
- Farmer cooperatives

Government

- Department of Agriculture & Cooperation

- Testing Labs
- Academic and Research Institutions

Private Sector and others

- Manufacturers / Wholesalers / Dealers of inputs
- Importers and exporters of agriculture products
- Traders, Buyers and Commodity Exchanges

2.3 General Constraints

2.3.1 Software constraints

- Training the model with small amounts of data or availability of data: - Data is needed in huge chunks to train model.
- Availability of the data required for the problem.
- When the model gets poor quality image or lack of quality dataset: - If we are passing low-quality data during training it would result in a bad model, which would also result in inaccurate predictions. Because of the model could encounter bad data during the prediction process. The data we have for training is a representation of the data that the model will encounter when deployed in production.
- Security of data: - Differentiating between sensitive and insensitive data is essential to implementing machine learning correctly and efficiently.

2.3.2 Hardware Constraints

- Choosing the correct machine learning hardware is a complicated process. The three core hardware options for machine learning: processing units, memory, and storage.
- Those delays caused by faulty, slow, or out of date equipment or a lack of sufficient space. or slow computers

2.4 Assumptions and Dependencies

- Get more labeled data: - It is valuable resource, as it has real and measurable value. accurate and timely data is critical to the model.
- Location of fruits: - the required fruits are located in various sparse areas of the country.

- Work with small amounts of data is one challenge in our project which leads to an incorrect model interpretation.
- The quality of camera being used to capture the image is another dependent factor for the training of fruit images.
- Machine understanding: -The lack of the computer's ability to actually understand what it sees and hears

2.5 User Documentations

New users are much more likely to successfully carried with our system if we provide them with informative user documentation. They can spend time browsing the docs and learning how the system works.

- **Use plain language**

Writing it in a way that anyone can understand. Stay away from cross reference or jargon and complex terms.

- **Prepare step-by-step instructions**

Our user documentation is being well formatted as step-by-step instructions. Meaning our content is accessible to the customers. Instead of presenting users with a long wall of text, step-by-step instructions are clearly laid out. So that customers can follow one step at a time. This keeps them engaged in the task and avoids distraction.

- **Add visual contents**

A picture is worth a thousand of words. Our documentation is more interesting for the customers by providing images and tabular formats. Documentation that is broken up with images is a lot more inviting to users than disturbing them.

- **Add table of contents**

Customers will be looking for particular sections within individual text. This is where a table of contents can come in really handy. A table of contents appears at the beginning of our document and lays out all of the sections contained within the document. Customers can click the section they want rather than having to read the whole document from beginning to end.

- **Collect feedback**

Our user documentation is never really finished. We need to collect feedback from our customers on an ongoing basis in order to find out the areas that could be improved, or content that is missing.

2.6 Specific Requirements

2.6.1 User requirements

The consumers seek an automated system that automatically identifies edible and non-edible fruits in plant species in order to keep their welfare from toxic fruits and so as to increase their level of knowledge in line with the edibility and non-edibility nature of fruits.

2.6.2 Functional Requirements

*Table 5:*Functional Requirement

FR ID	Requirement Description	Use case	Priority
FR01	Either the Developer or the Tester must able to capture the image of fruit using mobile phone camera.	Capture Image	HIGH
FR02	The model must extract features of the fruit images in order to recognize them.	Feature Extract	HIGH
FR03	The extracted features are matched with the already existing features which have been stored in model.	Detect Fruit	HIGH
FR04	The model must able to generate an audio sound for visually impaired people.	Generate Sound	HIGH
FR05	The developer should upload images.	Upload Image	MEDIUM
FR06	The Model must able to label class of fruits	Labelling Image	HIGH
FR07	The Developers must collect the dataset before the training.	Collect Dataset	HIGH

2.6.3 Non-Functional Requirements

Table 6:Non-Functional Requirement

NFR_ID	Requirement Description	Requirement Group
NFR01	Reliability of the system is very good but it also depends on the quality of fruits.	Reliability
NFR02	The system is able to successfully complete the tasks required by the user	Usability
NFR03	The model is efficient if the fruit images have a good quality.	Efficiency
NFR04	The System designed to be easily moved from one computing environment to another	Portability
NFR05	The model finds unique patterns of the given image	Performance
NFR06	The system is available 24/7 hour to the tester in the presence of electric power.	Availability
NFR07	The model must label the category of fruit images to the user.	Accuracy

2.7 External Interface requirements

External interface requirements specify hardware, and software with which our system must interface. This section provides information to ensure that the system will communicate properly with external components.

2.7.1 User Interfaces

Appropriate GUI will be provided. The system will be completely user friendly. It will be designed with minimum number of clicks. So, user should be able to access desired information, consequently performs the particular task. User interfaces such as screen formats for Input forms, output screens, structures have been explained in subsequent sections.

2.7.2 Hardware Interfaces

If we feed a machine a good amount of data, it will learn how to interpret, process and analyze this data by using SVM Algorithms.

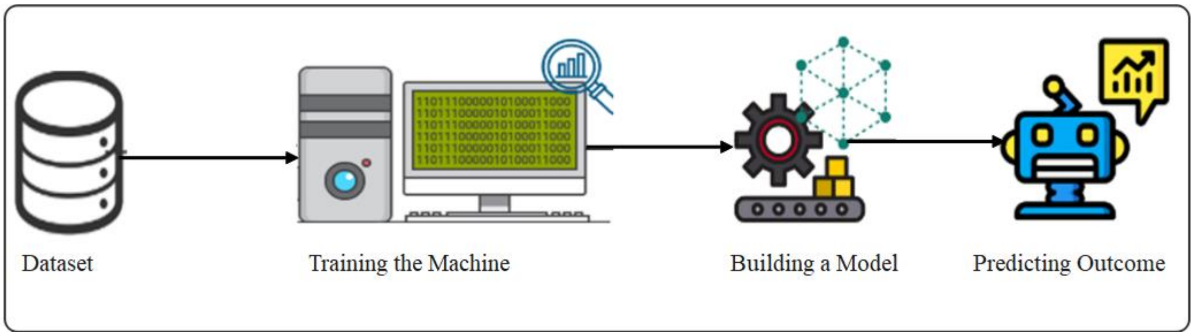


*Figure 4:*Hardware Interface

To sum it up, take a look at the above figure:

- A process begins by feeding the model lots of data.
- The model is then trained on this data, to detect hidden insights and patterns of fruit images.
- These insights are used to build a Machine Learning Model by using a SVM algorithm in order to solve a problem.

2.7.3 Software Interfaces

User will be able to access the system by initiating it to run and retrieve the image to be trained from the model.

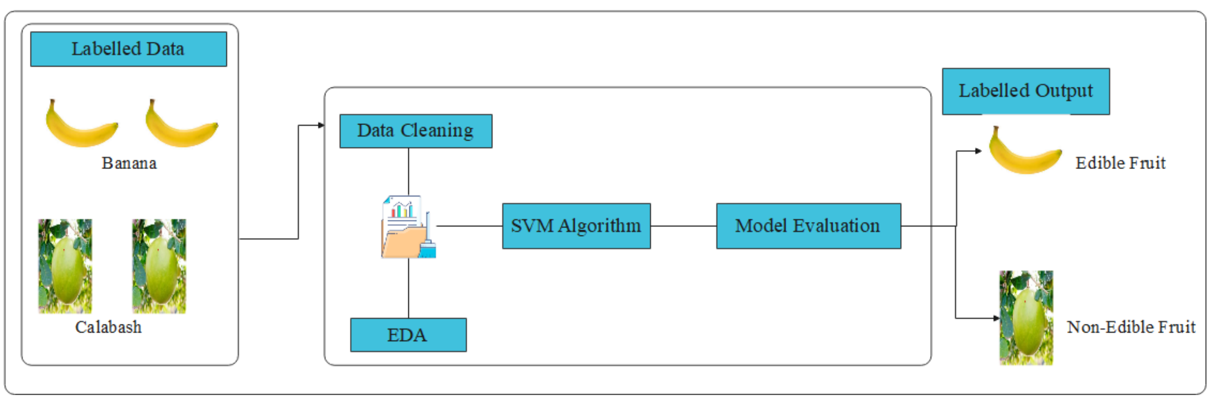


Figure 5:Software Interface

Consider the above figure. Here we're feeding the model images of edible and non-edible fruit and the goal for the model is to identify and classify the images into two groups (edible fruit images and non-edible fruit images). The training data set that is fed to the model and then labeled, as in, we're telling the machine, this is how edible fruit looks and this is non edible. By doing so we're training the model by using labeled data.

2.7.4 Communication Interfaces

To access the system first it is necessary to set up the communication between the model and the fruit images. Firstly, the images are properly imported then the imported images are cleaned or well prepared for training. Then the dataset is partitioned into two groups.i.e., the training dataset and the test dataset which is 80% for training and 20% for test dataset. Then after the partition, the model is created in order to acquire knowledge about the images. Once the model is created, it will train and gain knowledge for prediction. The model works its classification based on the knowledge it has acquired or makes prediction based on its prior knowledge. Finally, it improves the ability to evaluate the prediction.

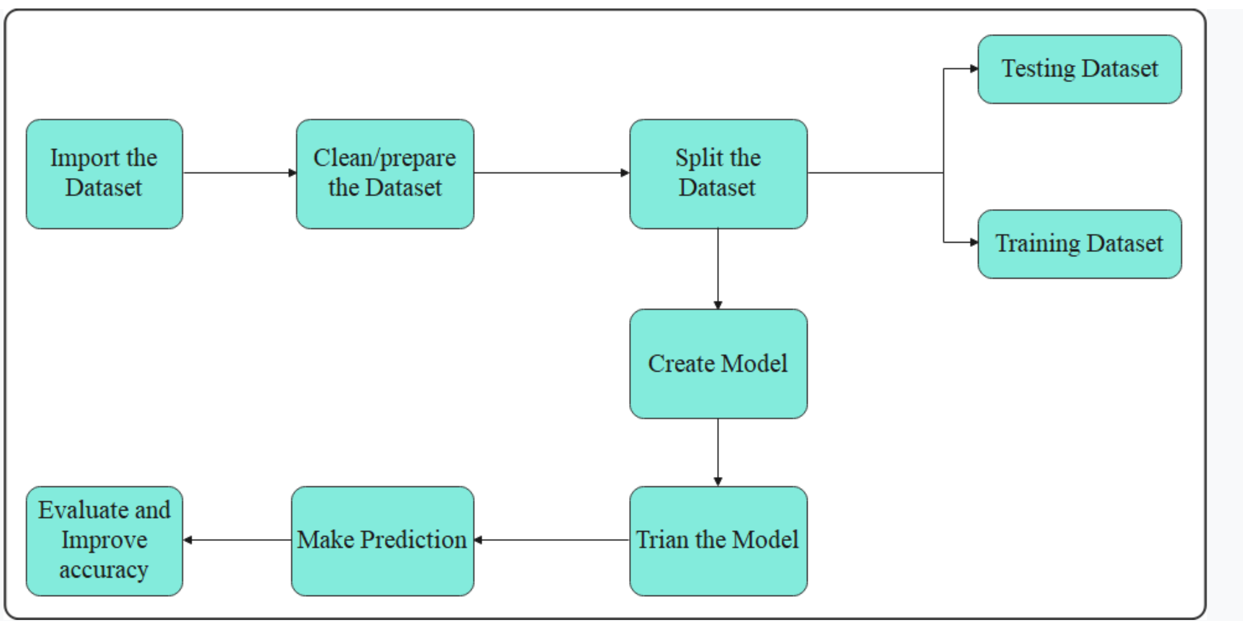


Figure 6:Communication Interface

2.8 System Requirements Modeling

Requirements modeling in our project comprises several stages, or patterns, behavioral modeling's (use case modeling, activity diagram, sequence diagram). Each of these stages or patterns examines the same problem from a different perspectives or angles.

2.9 Essential use case Diagrams

A significant advantage of essential use cases is that they enable us to stand back and ask fundamental questions like what's really going on and what do we really need to do without letting implementation decisions get in the way. These questions often lead to critical realizations that allow us to rethink, or reengineer if we prefer that term, aspects of the overall business process.

Use case name: Capture Image

ID: UC 01

Table 7:Essential use case Description

Developer Intention	System Responsibility
The Developer has to upload images to the model before using the system	If there is no image uploaded before, the system displays an alert message
Th Developer initiates the system in order to determine the edibility and non-edibility nature of fruits.	The system is being ready in order to detect fruit images
The Developer bears a fruit image to the system	The model detects the fruit image given by the Developer.
The Developer got knowledge on the edibility and non-edibility nature of fruits	The model classifies the detected fruit image as edible or non-edible

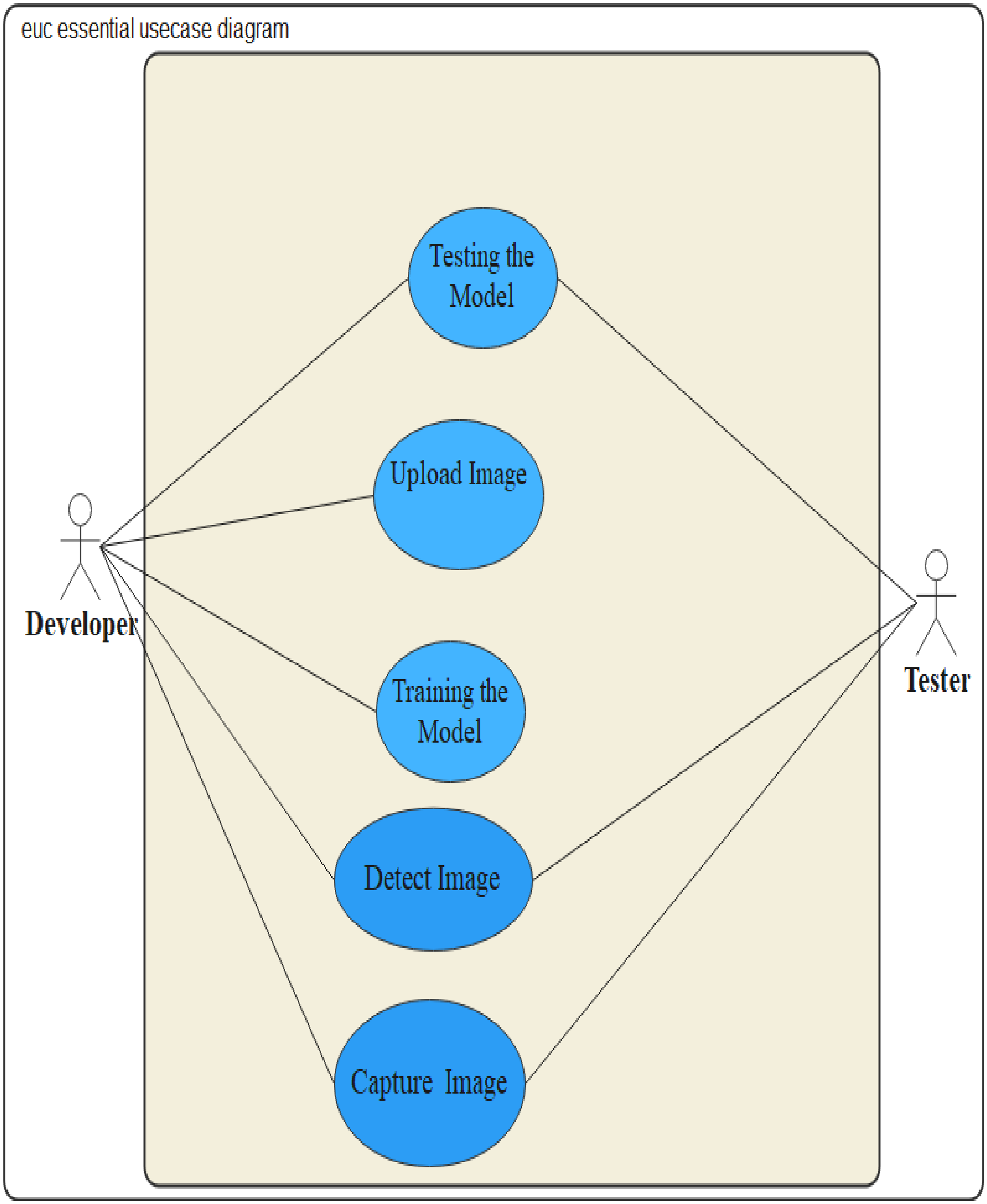


Figure 7:Essential Use case Diagram

Chapter Three: Requirement Analysis Modeling

3.1 Overview of Analysis Model

The analysis stage acts as the bridge between design and requirement and it happens after the requirement phase of our project. The developer critically explores the requirements and ensures that the system's operational characteristics are understood clearly. It helps us to understand the requirements in a better way. It encapsulates the system use case modeling, sequence diagram and activity diagram.

3.2 System use case Diagram

3.2.1 Scenario

A use case scenario is a description of a potential way our system is used. As we remind before data plays priceless role for machine learning algorithms. The one prominent connotative issue is the quality of data captured that is going to be trained. If the quality of data stored in the model is less, then the model failed to meet customers' expectation or system requirements. In order to overcome this problem, we have tried to use a quality smart phone camera.

3.2.2 Use case Model

Below is a use case diagram which we have prepared for Edible and Non-Edible Fruit Recognition System. It would help us to understand the role of various actors in our project. actors in our use case diagram are: Tester and Developer**.** The main use cases are in the system where as the diagram illustrates on how the tester or the developer interact with the use cases. For e.g. During detecting the fruit images, testers and developers need to interact with the use case and not the system whereas when it comes to categorizing fruits, only the model would be required.

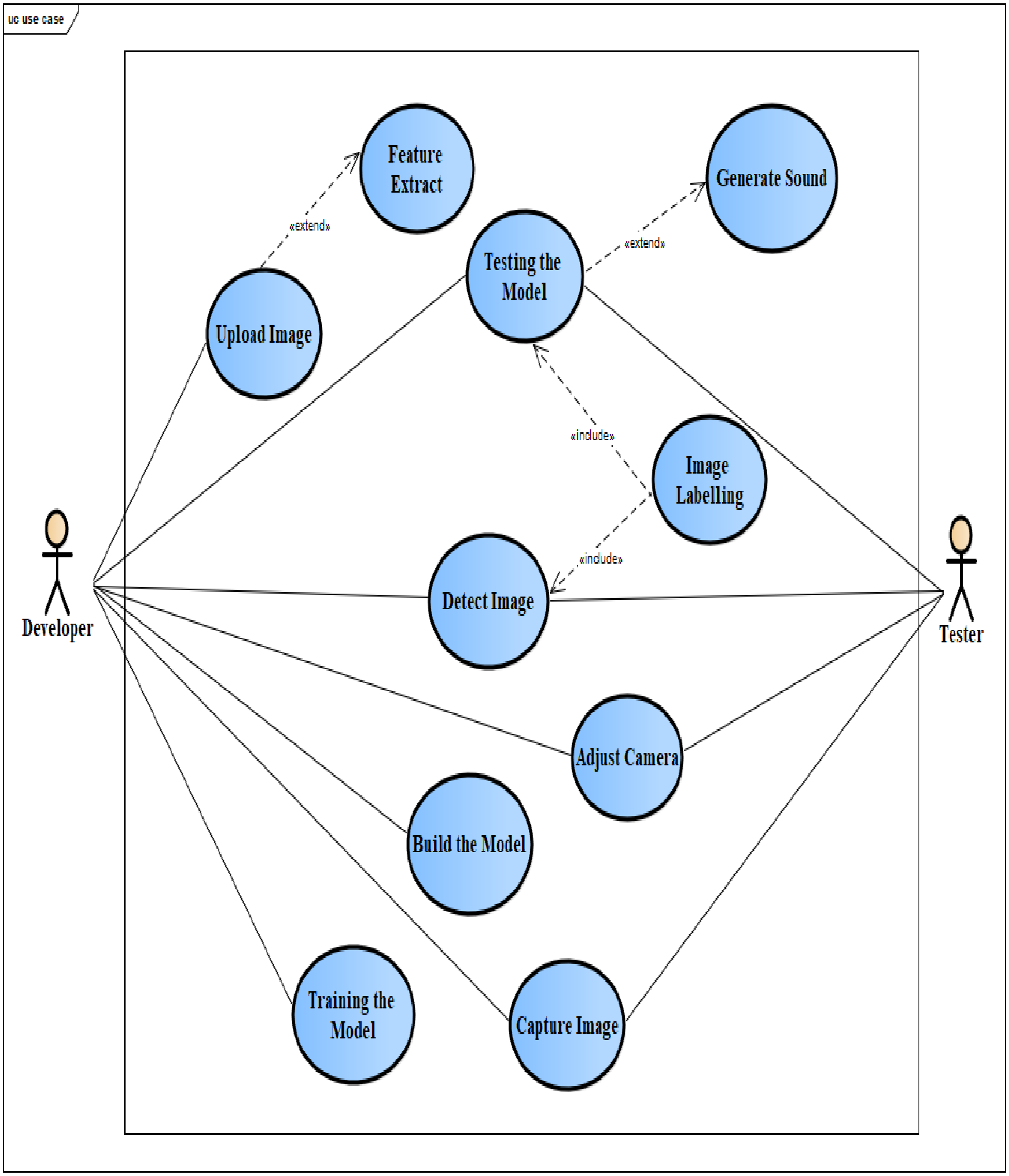


Figure 8:Use case Diagram

3.2.3 Use case Description

Table 8:Use case description for Capture Image

Use case name	Capture Image
Use case id	01
Actor	Tester and Developer
Description	The developer is obligated to capture quality image in order to train the model
Pre-condition	The Developer must have appropriate camera for capturing the image
Post condition	The captured image is feed into the model
Basic flow event	1.prepar fruit object to capture 2. Initiate the system to capture image. 3.adjust camera. 4.capture fruit image. 5.Feed to the model 6.use case end.

Table 9:Use case description for Build the Model

Use case name	Build the Model
Use case id	02
Actor	Developer
Description	Clearly define the problem that a model needs to solve and what success looks like
Pre-condition	The fruit images have to be captured and successfully uploaded
Post condition	The model is being tested in a local or offline environment using training and testing datasets
Basic flow event	1.Perform image preprocessing 2. Explore the data and choose the type of algorithm 3.Prepare and clean the dataset 4. Split the data into training dataset and test dataset. 5.Build the model. 6.use case end.

Table 10: Use case description for Train the Model

Use case name	Train the Model
Use case id	03
Actor	Developer
Description	In order to classify the fruits, first the model must be well trained.
Pre-condition	Quality image have to be captured
Post condition	The model contains knowledge about the fruit images

Table 11:Use case description for Test the Model

Use case name	Test the Model
Use case id	04
Actor	Developer, Tester
Description	Either the Developer or the Tester wants to test the given image.
Pre-condition	The model must already have sufficient knowledge about the fruit images.
Post condition	The fruit image is being classified as edible or non-edible based prior knowledge of the model
Basic Flow Event	1.Test dataset is provided to the system by the Developer or the Tester. 2.The model preprocesses the images going to be tested. 3. Then apply knowledge. 4. The model classify the image as edible or non-edible using SVM algorithm. 5. Use case end.

Table 12:Use case description for Feature Extract

Use case name	Feature Extract
Use case id	05
Description	The model extracts unique feature of the given fruit
Pre-condition	The system must be initiated by the developer first
Post condition	The image feature is being extracted based on its color and shape.
Basic Flow Event	1.The Developer initiates the system. 2.The Model preprocess the given image 4.The Model extracts the hidden layers of the given image based on its color and shape. 5.The Model contains knowledge 6.Use case end.

Table 13:Use case description for Generate Audio

Use case name	Generate sound
Use case id	06
Description	The model process available resources to give a response for the user.
Pre-condition	The model must be correctly built and trained.
Post condition	The system generates an audio signal for the user
Basic Flow Event	1.The Model detect the image. 2. Label the images 3. The model generated audio signal to the user. 4. Use case end.

3.3 Sequence Diagram

we have used sequence diagram to show sequence of operations or how the model acquire knowledge and operates its particular task in what order.

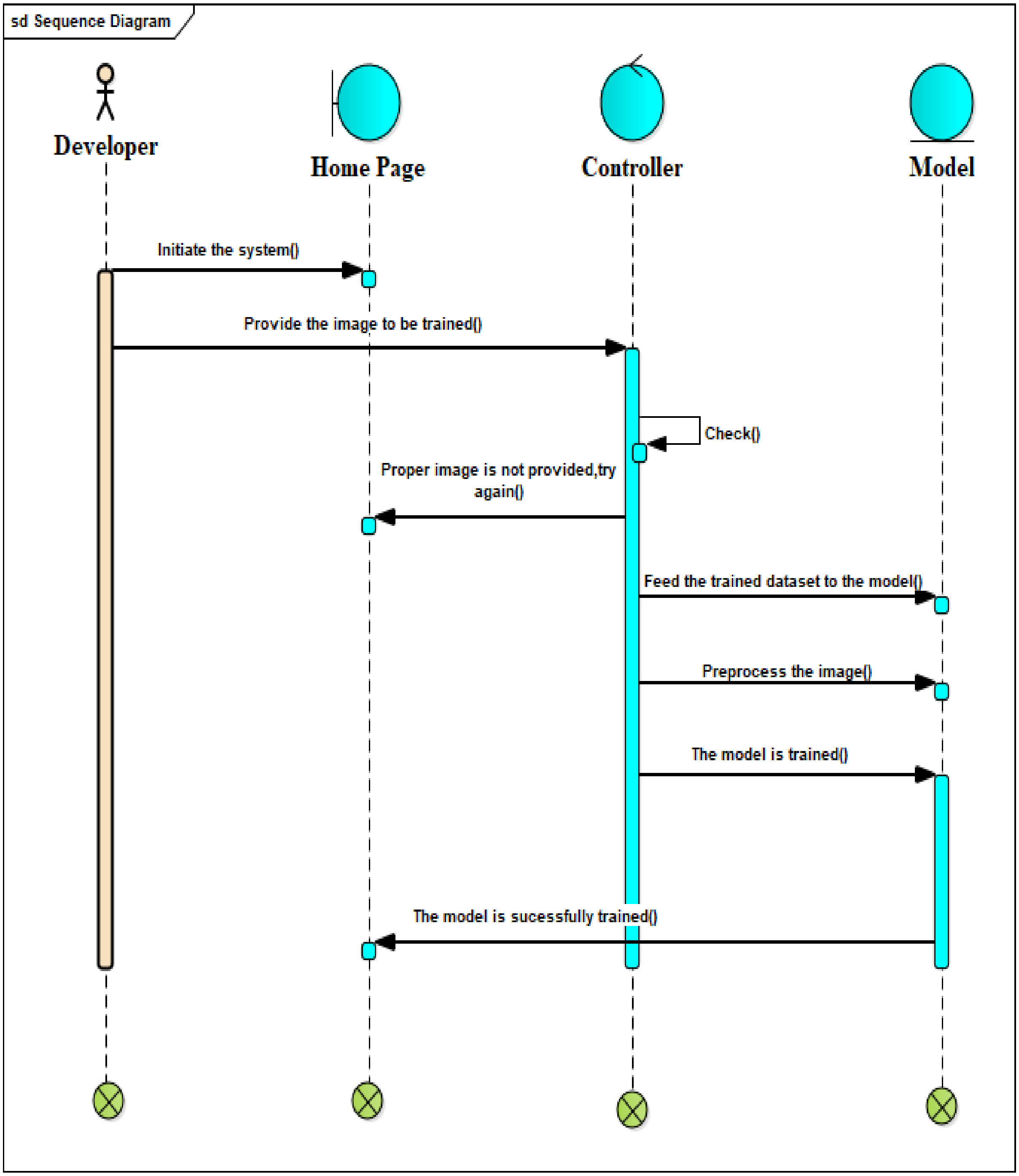


Figure 9:Sequence diagram for Train the Model

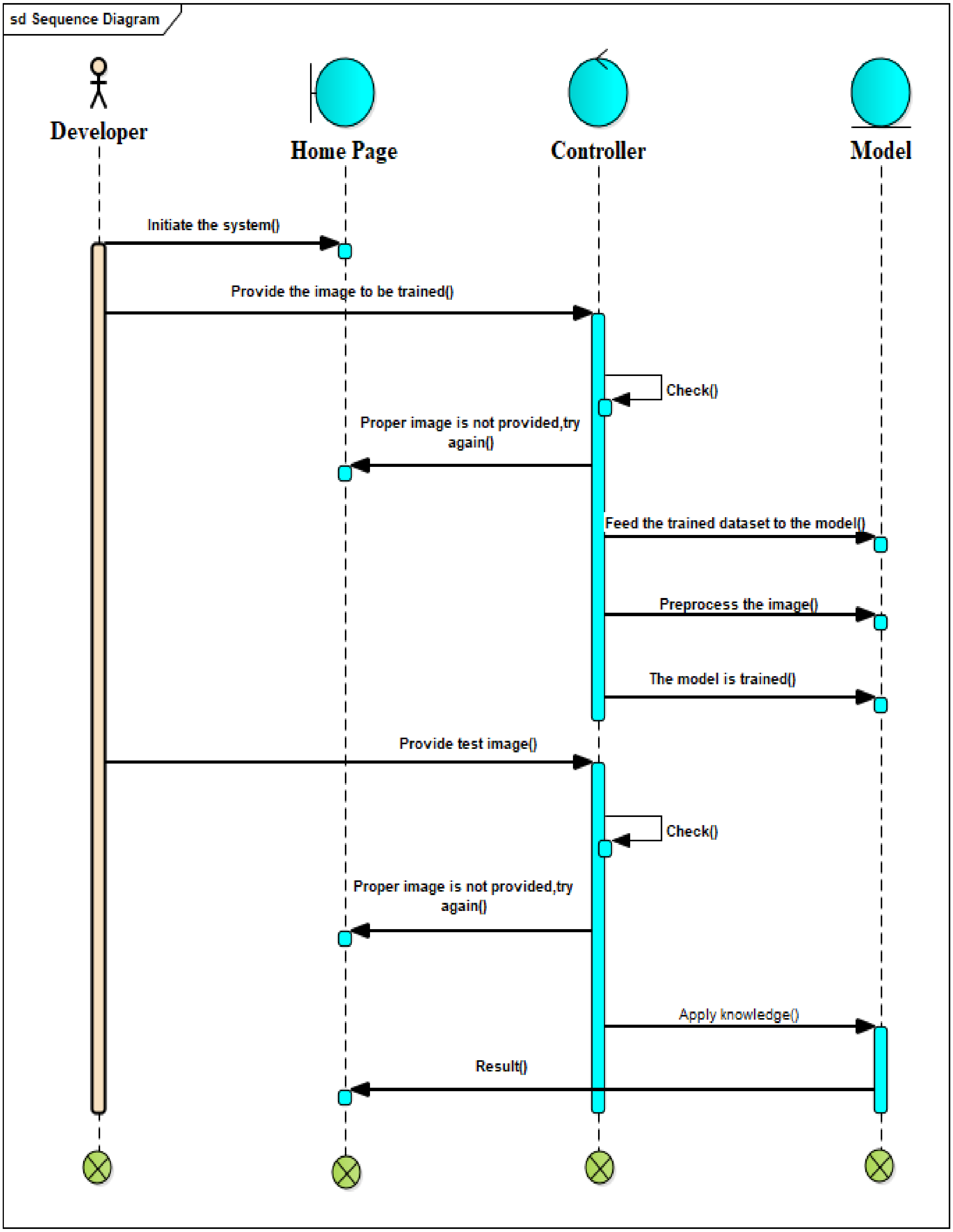


Figure 10:Sequence diagram for Test Data

3.4 Activity Diagram

Activity diagram of Edible and Non-Edible Fruit Recognition System shows the flows between activities of the project. Such as extracting the hidden layer or features of the labeled fruit images. The diagram below helps us to demonstrate how the model extract features of a particular fruit image in Edible and Non-Edible Fruit Recognition System.

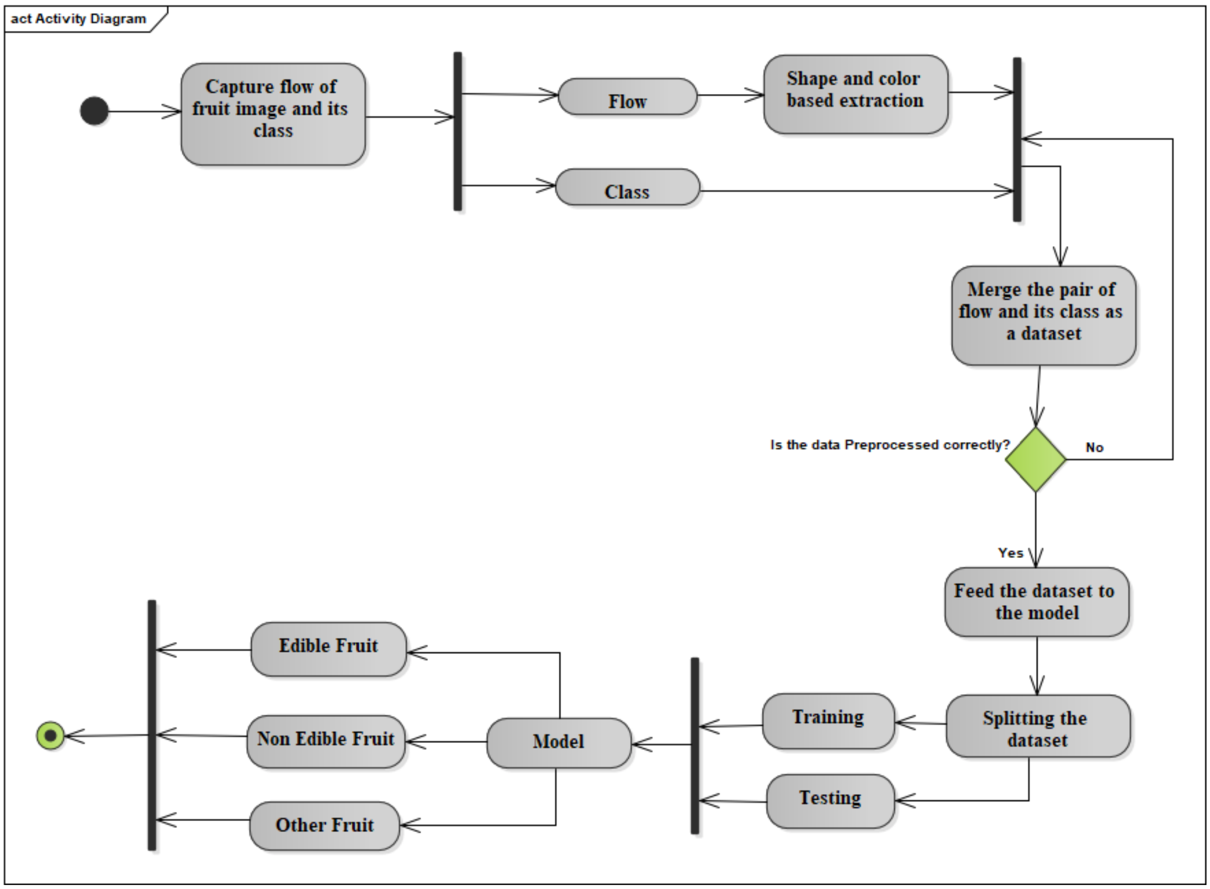


Figure 11:Activity Diagram

The above flow represents the flow from one activity to another activity, the activity starts from input fruit image through digital camera, and then input fruit is preprocessed and extract the features like color, shape, texture and so on. Now, the processed image is classified as either edible non-edible or other.

Chapter Four: System Design

4.1 Overview

Design is an interdisciplinary engineering activity that enables the realization of successful our systems. The process of systems design includes defining software and hardware architecture, components, modules, interfaces, and data to enable a system to satisfy a set of well-specified operational requirements. This portion incorporates goals of design, design consideration and purpose, proposed system architecture and low-level design model.

4.1.1 Design goals

- Adopt service-oriented architecture
- Develop and expose business functionality as services
- Support multiple access devices such as desktop computers, Mobile phones.
- Ensure confidentiality of user data
- Enable easy discovery of information
- Allow internal, external and associated agencies access to system
- Facilitate delivery of re-engineered manual processes through an online platform
- With the above objective in mind, the following section articulates the design considerations that have to be kept in mind during the design of the proposed technology our system.

4.1.2 Design considerations and Purposes

4.1.2.1 Quality of code

- Should use a labeled data set
- Should use Valid image format
- The fragment should not use unnecessary classes
- Code should be well structured.
- The system should have good performance in terms of finding patterns of an image

4.1.2.2 Degree of separation between content and presentation

- All labeled images should use valid formats, and resized pixels
- All labeled images should be feed into the model

- Any feature value is independent of the value of the other features

4.1.2.3 Accessibility for users

- The practice of our project involves taking data, examining it for patterns and developing some sort of prediction about future outcomes. By feeding the model more data over time, the developers and testers can sharpen the machine learning model's predictions
- It also involves feeding an algorithm large amounts of labeled training data and asking it to make predictions on never-before-seen data based on the correlations it learns from the labeled data.

4.1.3 Design guidance and Issues

4.1.3.1 Performance

The following are few guiding principles we should adhere in order to achieve high performance

- Use SVM algorithm so as to find undiscovered patterns of new images
- Use proper algorithms such machine learning algorithms distributed uniformly on all available fruits within a layer.
- Use proper quality camera for master data which is going to be trained.
- Perform time consuming tasks asynchronously.
- Code should be optimized using performance analysis tools.

4.1.3.2 Reliability

Guidelines for realizing Reliability

- Before the deployment of the application, it must undergo testing, verification and validation steps.
- After deployment of the application, field data can be gathered and analyzed to study the behavior of software defects.

4.1.3.3 Maintenance

In order to ensure maintainability of our system, the following points should be insured

- Modular Software Code: Software Code must be modular and well documented.
- Avoid Complex Coding: complexity is quite difficult to maintain safely. Hence it should be avoided.

- Early planning: anticipating what and how programs might be modified at a later stage.
- Modular design: defining subsets and simplifying functionality
- Uniform conventions: facilitating error detection and debugging.
- Naming conventions: providing understandable codes.
- Use of Coding standards, comments, and style enhancing readability of the program.
- Proper versioning of the software to be maintained.
- All the artifacts related to the software such as code, SRS, User Manual should be well documented and self-explanatory for any programmer to understand.
- Detailed documentation shall be available at each stage for easy comprehensions of the application system.
- All documents shall be prepared as per the defined documentation standards.

4.2 Proposed system architecture

The main goal of this project is to develop an efficient method for classifying the fruits as edible, non-edible or other. To do so, the proposed system integrates various existing technologies in order to achieve optimal performance. Specifically, the system makes use of smart phone technologies to provide quality image that can be used by the model to perform the classification of fruits.

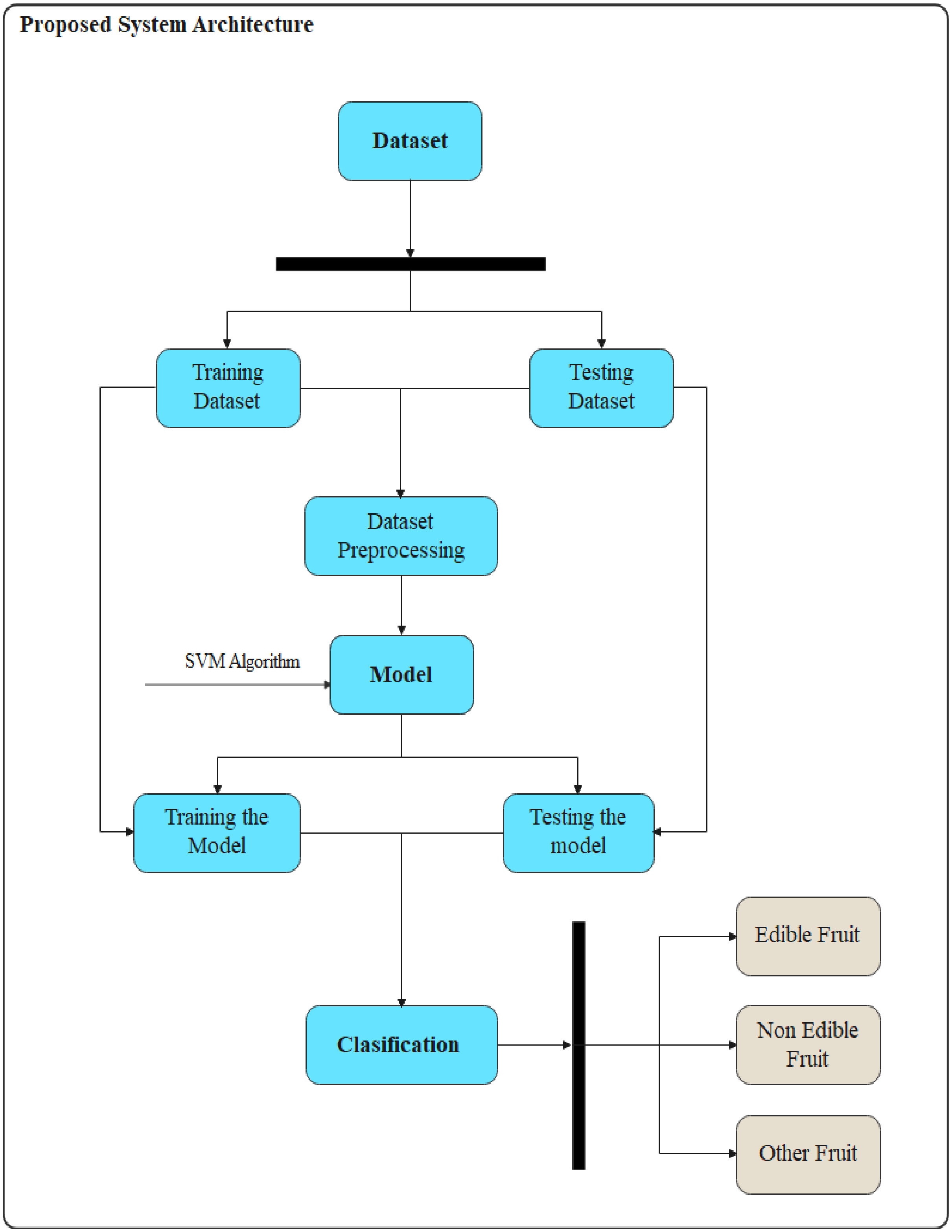


Figure 12: Proposed System Architecture

4.2.1 System process

First datasets have to be collected and captured in an appropriate way. Then the collected datasets are labeled edible and non-edible as depicted in the following figure. When the new dataset is coming the system extracts the features of the new dataset and it groups the coming dataset either edible or non-edible based on the knowledge it has acquired before.

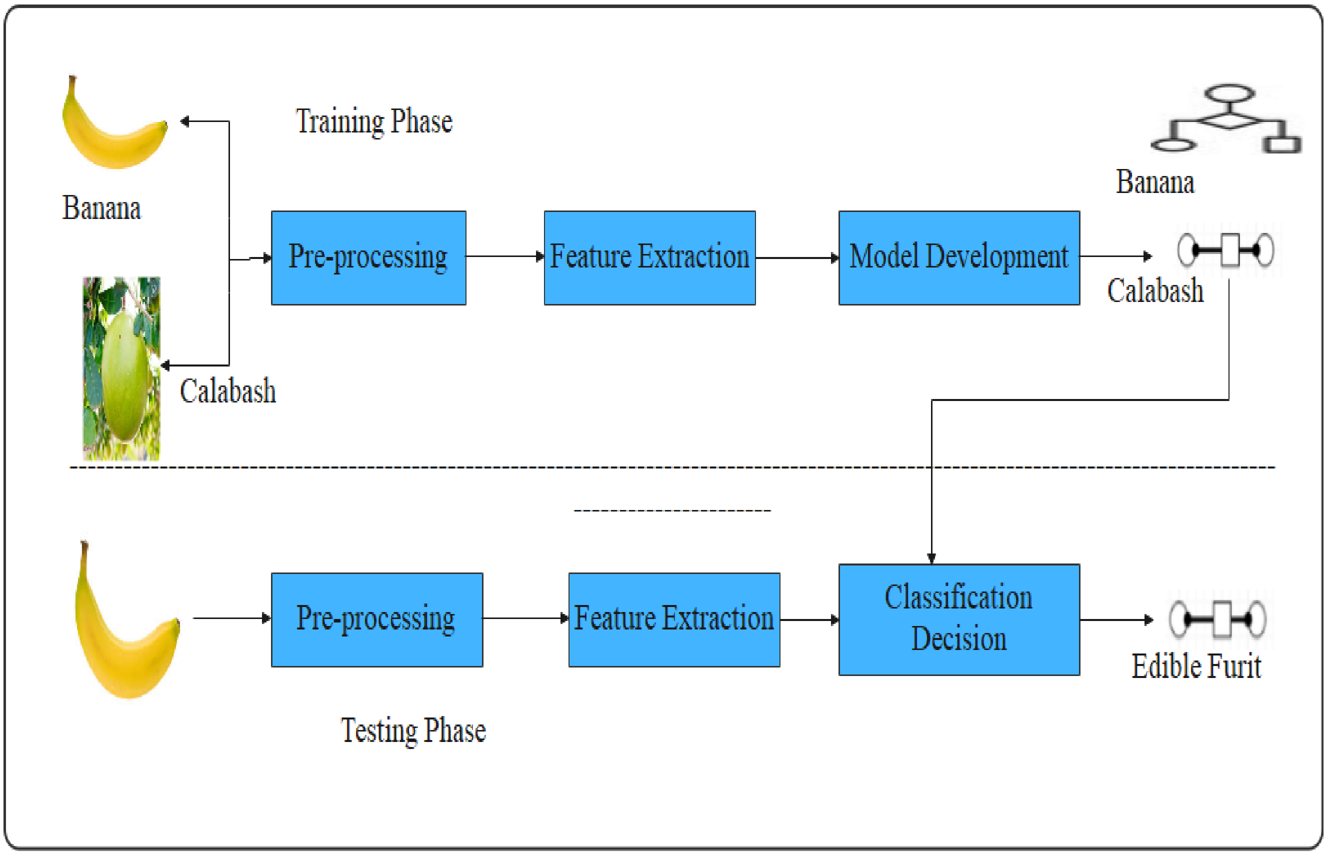


Figure 13: System Process

4.2.2 Subsystem decomposition

we can present the overall system as a hierarchy and view the relationships between the model, the training data and the test data.

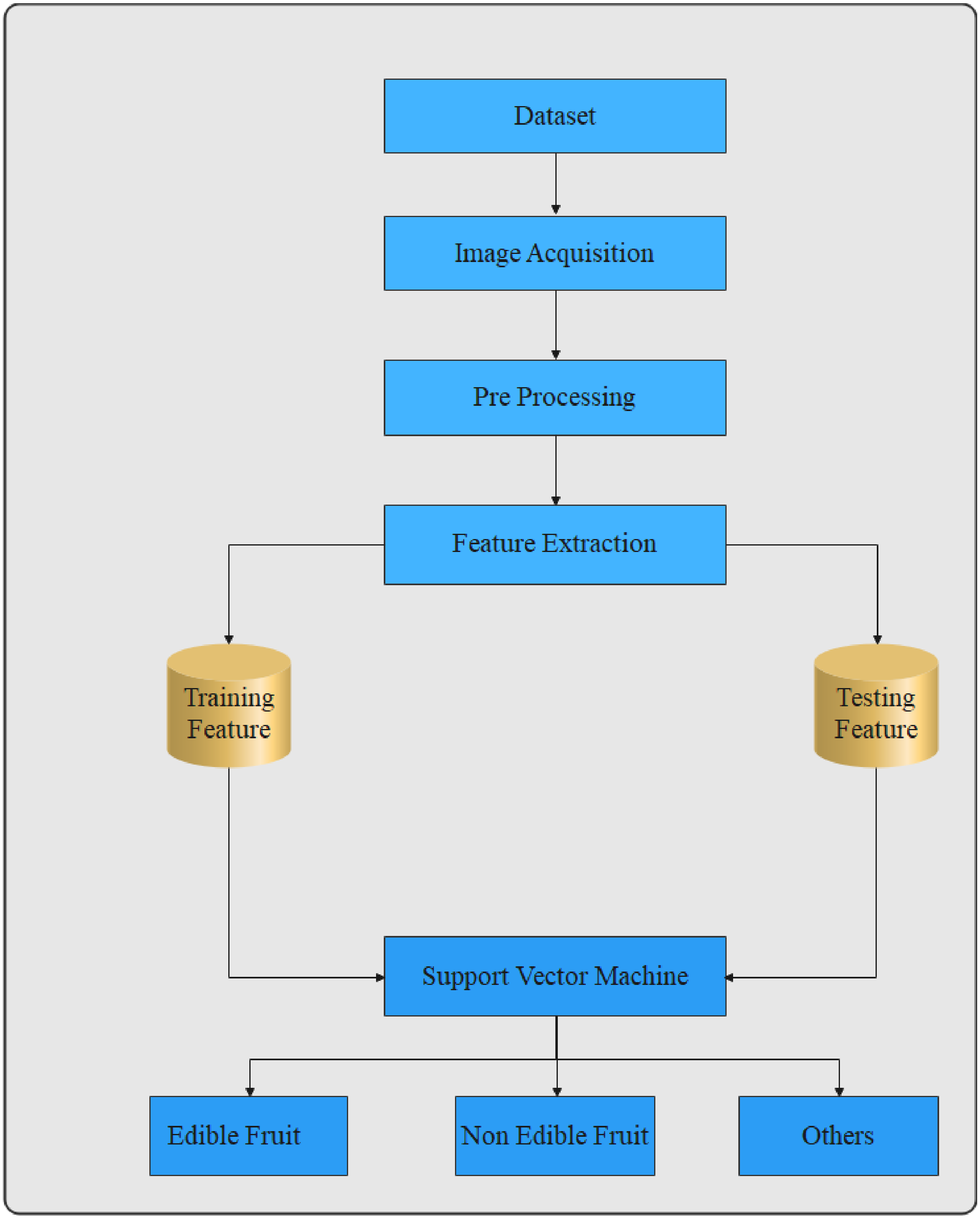


Figure 14: Sub System Decomposition

4.2.3 Hardware / software mapping

Computer can understand only machine code (binary code). Python Virtual Machine (PVM) first understands the operating system and processor in the computer and then converts it into machine understandable code. Further, these machine code instructions are executed by processor and the results are displayed.

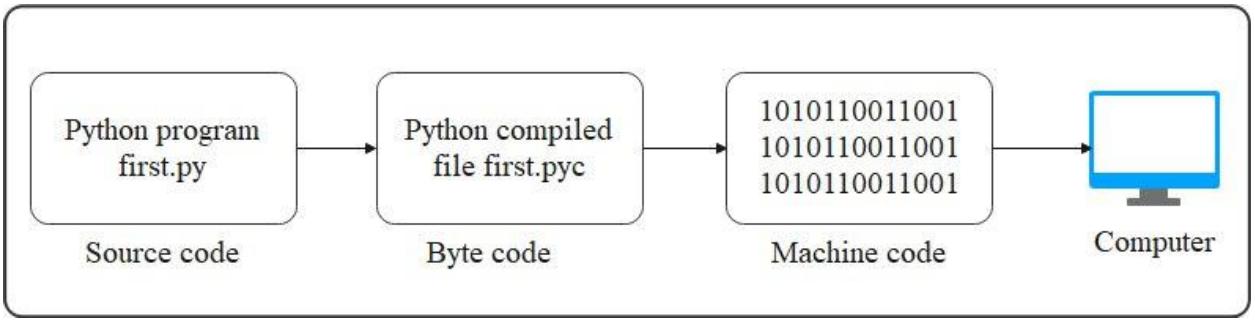


Figure 15:Compiling PVM

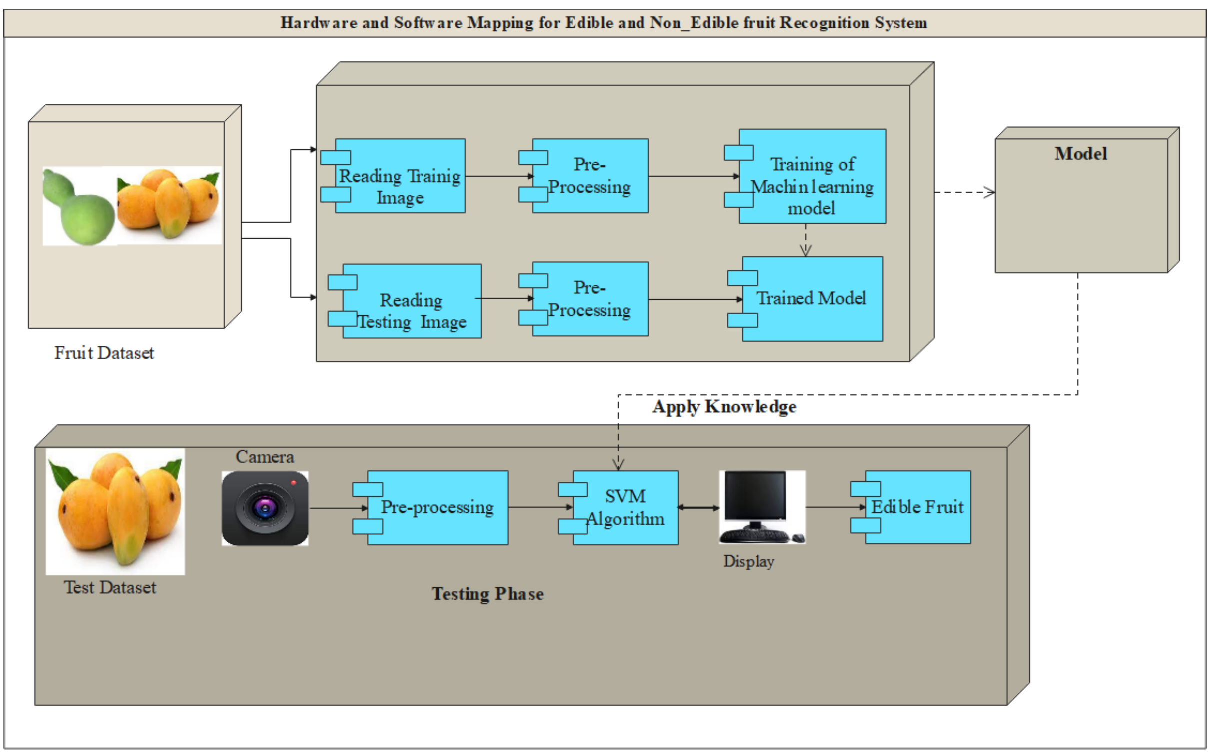


Figure 16:Hardware and Software mapping

Generally speaking, developing software requires the same steps as general-purpose software: designing, encoding, compiling, linking, packaging, deploying, debugging, and optimizing.

4.3 Low-level Design Model

4.3.1 Component diagram

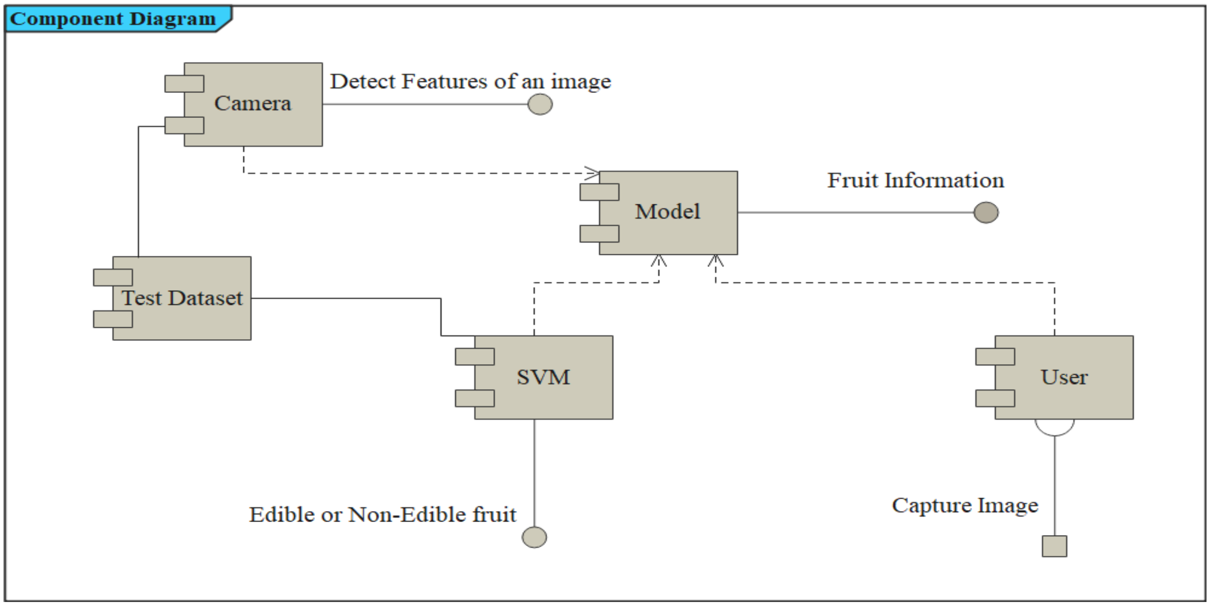


Figure 17: Component Diagram

The dataset flows into the model through image preprocessing steps and converted into a format that the model can understand. The interfaces on the user side are known as required interfaces which represents the pre conditions that the model seeks in order to carry out its task. The prepared dataset passed into the model and the model classify the images as edible and non-edible using SVM algorithm.