An algorithmic approach for automating the sorting and grading harvested TJC mangoes

Patabendige S.S.J   16001011
A.V.P. Sewwandi   16001354
D.D.Tharaka   16001443

Supervisor: Dr. Thilina Halloluwa
Co-Supervisors: Dr. M.I.E Wickramasinghe
Dr. Kasun Karunanayaka

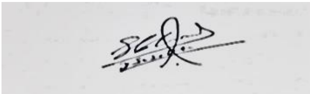
March 2021

Submitted in partial fulfillment of the requirements of the B.Sc.(Hons) in Software Engineering 4th Year Project (SCS4123)

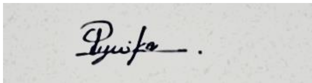
Declaration

I certify that this dissertation does not incorporate, without acknowledgment, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

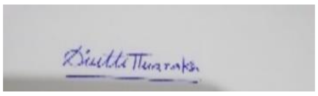
Candidate Name: Mr. Patabendige S.S.J

Signature of Candidate:  Date:......26/03/2021................

Candidate Name: Ms. A.V.P. Sewwandi

Signature of Candidate: .  Date:......26/03/2021..............

Candidate Name: Ms. D.D.Tharaka

Signature of Candidate:  Date:......26/03/2021.............

This is to certify that this dissertation is based on the work of Mr. Patabendige S.S.J, Ms. A.V.P. Sewwandi, Ms. D.D.Tharaka under my supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

Supervisor's Name: Dr.Thilina Halloluwa

…………………………   ……………..
Signature of Supervisor   Date:

Co- Supervisor's Name:

Dr. M.I.E. Wickramasinghe   Dr. Kasun Karunanayaka

………………………   ……………..   ……………………...   ..…..………
Signature of Supervisor   Date: Signature of Supervisor   Date:

Abstract

Mango grading is a quality assessment method carried out by mango exporting industries. Precisely graded mangoes support to uplift the demand of the industry as well as the revenue.

Currently, both manual and machinery-based approaches are being used in the global market. Sri Lanka follows a manual grading approach to grade TJC mangoes which is the predominant exporting mango variety in Sri Lanka with a beautiful golden orange and unblemished skin. However, manual grading of mangoes using visual perception is laborious, inaccurate, and inconsistent. Therefore, this project aims to overcome these issues by introducing a machine learning and computer-vision based solution to grade harvested TJC mangoes in terms of their surface spots, size, and weight. This research has focused on the six perspectives of each mango to get an accurate final result from the machine learning model for the mango grade.

This work is carried out in four phases: The first phase is the acquisition of images for training and testing purposes. Images were acquired by capturing six images per mango from six perspectives. The generated dataset consists of 1500 color images from five different quality classes. For experimentation purposes, the dataset was split into two groups containing 1350 and 150 images respectively. The larger group was used to train the classifier whilst the smaller group was used as the test dataset. Several image pre-processing techniques were followed in the second phase to enhance the images and extracted the essential features for the classification processes to identify the color scale, thresholding, noise removal, and contour detection techniques. Morphological operations and histogram of gradient (HOG) and local binary pattern (LBP) were used in the feature extraction phase. K-means clustering and K-medoid was used to detect the surface around the mango stalk. Sequential forward selection algorithm was applied to identify the best feature subset from the extracted superset of features in the feature extraction phase. The final grade classifier considered all the six sides of images in each mango. In the fourth phase, a model consisting of two main classifiers for analyzing the size and surface spots was implemented.

The classifier built to classify mangoes based on the surface spots, has used different algorithms to classify top-side, bottom-side, and remaining four sides of images of each mango. This research was done by considering four classification algorithms FFNN, SVM, KNN, and CNN to classify six sides of each mango. The classifier of top-side images was implemented with an accuracy of 77.77%, 44.44%, 90%, and 40% for the above algorithms respectively. The bottom side images classifier was implemented for the above algorithms with an accuracy of 90%, 55.55%, 66.66%, 45% respectively. For the other four sides 93%, 64%, 91% and 72% accuracies were given by FFNN, SVM, KNN, and CNN algorithms respectively. The mango size analyzing classifier was implemented with three classification algorithms FFNN, SVM, KNN with the accuracy of 100%, 86%, 100% respectively. In addition to the main classifiers, a rule-based classifier combination algorithm was implemented to determine the final result. The highest accuracy for the side images, top images, bottom images were given by FFNN, KNN and FFNN respectively by considering surface spots. The highest accuracy for the size analyzer was given by FFNN. For the final grade classifier which gave the highest accuracy of 93% by the rule-based system.

Acknowledgment

The authors of the thesis take this opportunity to gratefully acknowledge the assistance and contribution of supervisors and advisors who have been helping throughout the project.

First and foremost, we would like to express gratitude to the Supervisor Dr. Thilina Halloluwa and the Co-supervisors Dr. M. I. E. Wickramasinghe and Dr. Kasun Karunanayaka for their guidance, encouragement and immense knowledge to do a better quality project.

We are grateful to Mr. Udeshika Dilan(Agricultural Instructor) for giving this idea of this research project and for providing us with basic knowledge related to the research area. Finally, we would like to thank Mr. Upul Jayawardena(Sales manager) Nelna agri mango outlet, Malabe for his valuable commitment for supporting us with necessary resources and allowing us to collect the dataset.

Table of Contents

List of Figures

List of Tables

List of Acronyms

CVS	-	Computer Vision System
CV	-	Computer Vision
CNN	-	Convolutional Neural Network
MLP	-	Multilayer Perceptron
GLCM	-	Gray Level Co-occurrence Matrices
SFS	-	Sequential Forward Selection
FFNN	-	Feed Forward Neural Network
GSD	-	Stochastic Gradient Descent
GUI	-	Graphical User Interface
ANN	-	Artificial Neural Network
SVM	-	Support Vector Machine
SVR	-	Support Vector Regression
MADM	-	Multi Attribute Decision Making
ML	-	Machine Learning
NN	-	Neural Network
AI	-	Artificial Intelligence
SGD	-	Stochastic gradient descent
IT	-	Information Technology
WS	-	Web Socket

Chapter 1 - Introduction

1.1 Domain Description

The Sri Lankan agriculture industry represents a significant portion of the Sri Lankan economy including the export market [1]. The agriculture sector showed a negative growth rate of 4.2% in 2015-2017[12]. Low levels of mechanization and lack of marketing were identified as the root causes of this drastic decline [13]. It has been identified that the majority of Sri Lankan farmers rely on dated cultivation techniques and methods (i.e., Lowland paddy farming, Upland organic paddy cultivation which can lead to low market value for their products [14]. Consequently, various local [2] and international [34] initiatives have introduced export quality procedures aiming to uplift the farming standards. The product (i.e., fruit or vegetable) and the destination (countries) to which the product ships to is the varying factor for these standards [34]. Therefore, grading and sorting techniques have to be done after harvesting as a method of value addition [Fig. 1]. Grading can be done according to size, variety, color, maturity, and diseases. After grading, the different grades of fruits and vegetables could be sold at different rates.

As a fresh fruit, mango has a demand in local markets as well as in the global market [4]. Maximum profits out of fresh mango produce are tightly coupled with the quality grading received at the evaluation conducted at the buyer's premises as well as the physical appearance [4]. The export market and quality evaluation are affected by the assortment of fruits and vegetables.

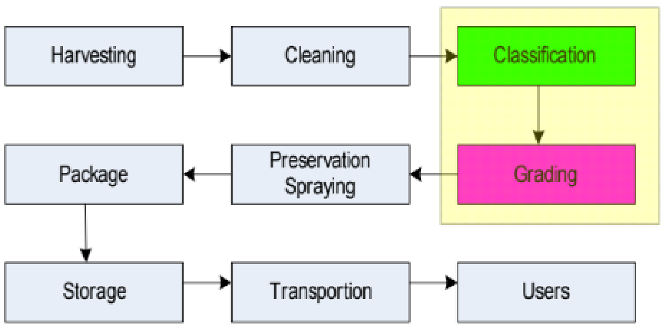


Figure 1: Post-harvest processing of mangoes[7]

1.2 Problem Definition

To calculate the quality of the mango for grading, there are several factors that are taken into consideration by farmers. These factors can be classified into two categories. The external quality factors and the internal quality factors. External quality factors are derived from visual appearances such as area size, shape, color, gloss, surface defects, and texture (fruit surface patterns) [23]. The fruit smell, such as aroma, flavor, sweetness, and sourness, etc. describe the internal quality factors. Other factors including firmness, crispness, and toughness that can be defined by touching the fruit and may be considered external or internal factors. For marketing purposes, fruits are generally graded on the basis of their external quality features also known as morphological features that can be judged by visually inspecting, touching, and occasionally smelling the fruit. In most countries including Sri Lanka, grading is done manually [Fig. 2] through visual inspection which is time-consuming and inconsistent since decisions can vary among investigators [4,33].

Figure 2: Manual sorting of mangoes

Nowadays, mango grading machines [Fig. 3, Fig. 4] which are based on several machinery techniques are used in other countries to grade mangoes. However, the main limitation of these machines is that they primarily consider only the weight while excluding the other factors such as size, surface defects [8]. Even though researchers have developed certain mechanical, ML, CV based prototypes to help and automate the fruit and vegetable grading systems, there is a lack of an appropriate system to grade harvested mangoes by considering its external quality factors (size, surface defects, weight).

This project is a combination of software and hardware which semi automates the manual grading process of harvested TJC mangoes. Hardware prototype will be developed to capture the images and a software system will be implemented to calculate the quality of the mango by grading, to do the analytical functions, and for data visualization.

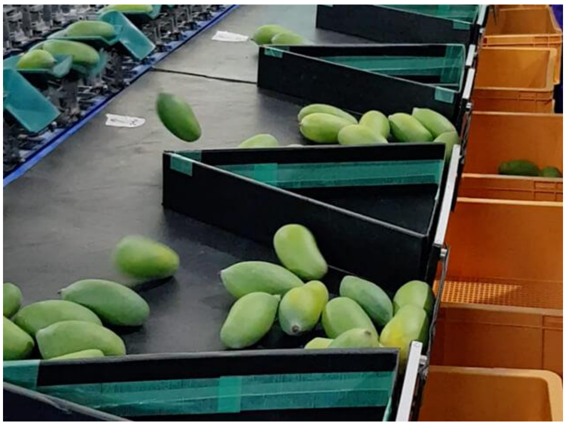


Figure 3: Machine based sorting [36]

Figure 4: Machine based sorting [6]

The following [Tab. 1] shows the grading criteria taken into consideration in the process of manual grading.

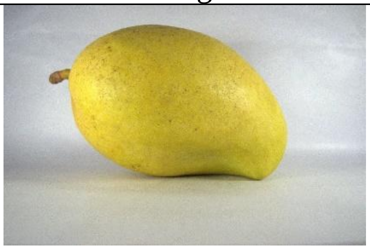
Grade	Image	Attributes for gradation
G1 (Very Good)		Mangoes in this class must be matured, superior quality, and this type used for export. They must be big in size, weight is >400g, and free of dark spots in the skin.
G2 Large (Good)		Mangoes in this class must be matured and of good quality. They are big in size, weight is >400g, slight dark spots in the skin.
G2 Small (Very Good / Good)		Mangoes in this class must be matured, superior quality, or good quality. They are small in size, weight is <400g, and free of dark spots in the skin or can have slight dark spots.
G3 (Medium)		Mangoes in this class must be matured, medium quality. They can be big or small in size. They have dark spots in the skin than G1 and G2
G4 (Poor)		Mangoes in this class must be matured, poor in quality. They can be big or small in size. They have large dark spots on their skin. Generally, such mangoes are not supplied to the customer but may be used for industrial processing.

Table 1: Grading criteria

1.3 Vision

Uplift the TJC mango export industry by introducing a replacement with an advanced software-intensive solution for laborers to optimize and make efficient the mango grading process and sorting of mangoes and increase the contribution to the economy.

1.4 Goal

In order to reduce inconsistency in manual grading process, our goal is to implement a relatively accurate classifier to grade TJC mangoes, build a prototype to capture images and a software to display the results with analytical information obtained from the classifier using ML, CV, and software engineering techniques to semi automate the manual grading process.

1.5 Objectives

As explained in the description, mango grading in Sri Lanka is currently done by human expertise with their practice and experience. Since this approach is inconsistent, labor intensive, time consuming and less accurate, this could lead the mango export market unprofitable and can be degraded due to exporting less quality graded mangoes. Therefore, the objective of the proposed system is to reduce inconsistency and inaccuracy by semi automating the grading process. The developed system focuses on giving the final grade of harvested TJC mangoes by considering main factors such as weight, surface spots, and size. This study is a proof of concept to prove that grading fruits and vegetables using CV and ML techniques has a practical potential. This research and development project mainly focuses on implementing an accurate algorithmic model by identifying most suitable feature extraction, feature selection methods, and classification algorithms.

1.6 Delimitation of Scope

The ultimate goal of this project is to build an advanced software-intensive system that semi automates the manual mango grading process done by human visual inspection. However, this system only focuses on the TJC mango variety since it is predominant exporting mango variety in Sri Lanka. Even though researches have been done for maturity prediction the proposed system can be used to grade only the harvested mangoes which are pre-checked for their maturity when they are harvesting. That means maturity prediction will not be covered by this system since it is not a requirement of the client company (Nelna Mango Store).

This system classifies TJC mangoes according to the size, surface defects and spots, and weight features which are extracted from the six input images from each mango. After the classification, the results are displayed on the dashboard with reports and analytical details. Users will be able to monitor the process details of graded mangoes and analyze the data using details shown by charts in the dashboard.

After the grading process, separating mangoes into relevant bins according to their grades will not be covered in this project. However, that can be done as a future work using machinery or robotic technologies.

This system will not process mangoes in batches as in machinery solutions. However, this solution can be improved into a more generalized version using more machinery resources, AI, and robotic technologies as mentioned in the suggested system design under the Suggested System Design section. Maturity prediction also can be done as an additional function for future work.

1.7 Methodology

The ultimate outcome of this system is, generating the correct quality grade of a given harvested TJC mango with a good performance (within around 5 seconds). In this research-based project first we identified the main factors considered by human expertise when doing manual grading (weight, size, surface spots). Then the images were captured from 6 sides from each mango to create the data set and preprocessed those collected data to make those images suitable for the model to train correctly. Then the most important features were extracted and selected using suitable feature extraction and selection algorithms. Then some classifiers were implemented separately for top-side, bottom-side, and remaining 4 sides images to classify mangoes into 4 main grades namely grade1, grade2, grade3, and grade4 based on surface spots according to the scientific background we gained from the background study. Next, we implemented some classifiers to classify grade1 and grade2 TJC mangoes based on their size. Out of those classifiers we implemented to determine size and surface spots of a TJC mango, we selected most suitable classifiers to analyze surface spots and size by considering their accuracy and performance. Then, the outcome of all those classifiers were combined using a rule-based algorithm. Finally, the final grade of TJC mango is sent as an input to the front-end application to display it to the user. The below images[*Fig. 5, 6, 7*] explain the 6 sides we captured from a mango.

Figure 6: Top side of a mango

Figure 5: Bottom side of a mango

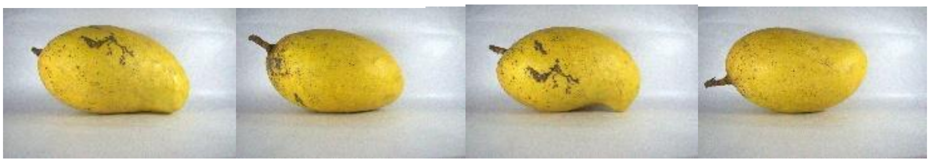


Figure 7: Remaining 4 sides of a mango

1.8 Justification for the Project

Software engineering projects focus on the development of software products with the main intention of doing an innovation rather than an invention. From the facts that were gathered from the expertise and from the background study, there is no AI system to semi-automate the mango grading and sorting. Hence this will be an innovative idea.

The concept mentioned above has a research component and a development component as well. Image analysis and feature extraction from those images is a complex part that was addressed. A research was carried out to identify suitable ML techniques to extract and select features which are related to this study. Based on those features the best classification methods for classifying the images were found out as a part of the research work. Then using selected optimal classification algorithms a model, and to depict the results of those research components, a desktop application which takes images as inputs and outputs the relevant grade for a particular mango is also implemented as a part of the development component. Hence the system can be named as a research-based software engineering project.

Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software (IEEE Standard Glossary). After considering the overall characteristics of the project, capabilities of the members and the time constraints, iterative and incremental methodology was adopted as the software engineering process model. Version controlling and code reviews (peer reviews) was carried throughout the project to maintain best coding structures and increase the quality of coding. This project did a feasibility study and background research on similar products before starting the development.

To maintain the quality of this application quality assurance principles are applied for the process from the beginning of the project. Hence, this study could reach its goal with a quantifiable approach. Since a particular software product to a real-world application is developed, this project falls into the product-based software engineering project category. As the study follows the above software engineering techniques and principles, this is a software engineering project and to be more specific; this is a product-based software engineering project.

1.8.1 Software engineering best practices used in the project

Design/Architectural patterns

- Factory design pattern - Factory design pattern was applied to implement the final grade evaluation component. A factory method pattern defines an interface or abstract class for creating an object but lets the subclasses decide which class to instantiate. In other words, subclasses are responsible for creating the instance of the class [38].

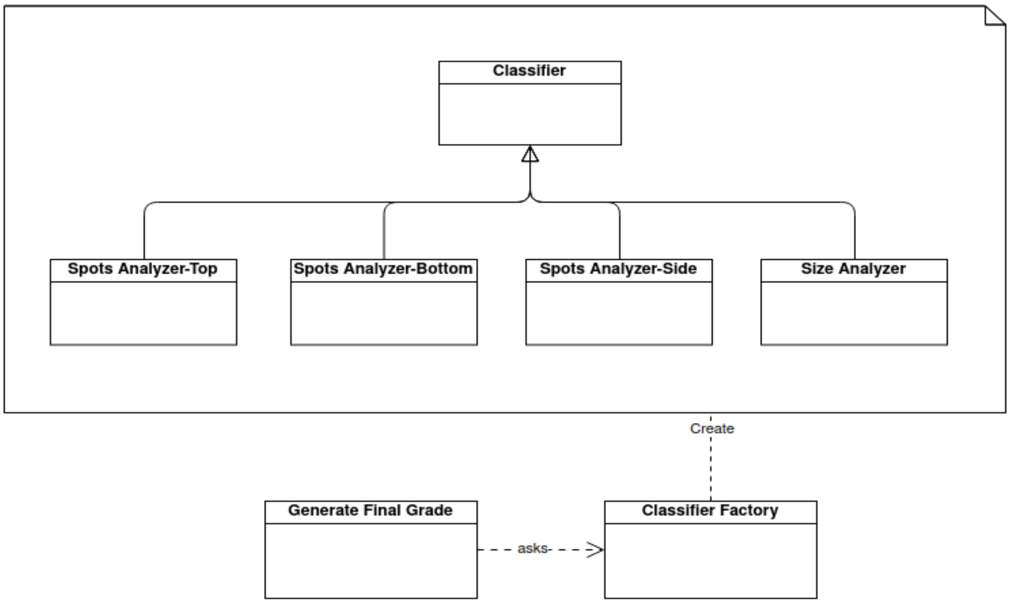


Figure 8: UML diagram for factory design pattern

Spots analyzer-top, spots analyzer-bottom, spots analyzer-side and size analyzer have responsibility for creating instances of each function when the classifier factory needs.

- Singleton design pattern - We used singleton design pattern when making the database connection instance in python flask APIs to make the system memory efficient and to enhance the system performance. Here the system allows to create only a single instance of database connection to process every mango and to interact with the database by saving and accessing data. So that we can ensure the database configuration class has only a single instance and we can gain a global access point to that database connection. The singleton object is initialized only when it's requested for the first time by APIs [39].

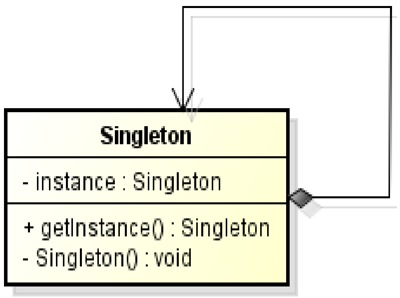


Figure 9: UML diagram for singleton design pattern

- MVC architectural pattern - We used MVC architecture in the front-end application so the modules are having less dependencies hence we were able to do changes easily. This isolates the application logic from the user interface layer hence it improves the maintainability and readability of the code and support for asynchronous techniques [40].

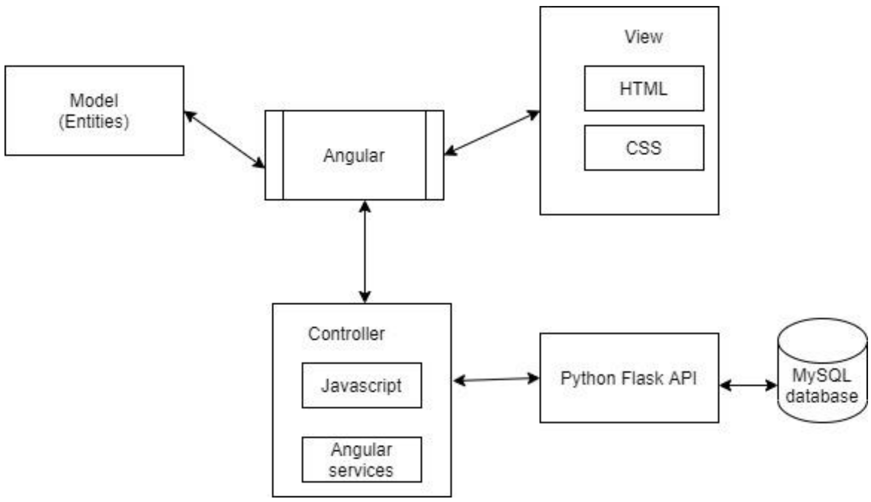


Figure 10: MVC architecture used in the system

- Observer design pattern - Observer design pattern is used in Angular services and it is a technique for event handling, asynchronous programming, and handling multiple values. In this pattern an object, called the subject, maintains a list of its dependents, called observers, and notifies them automatically of state changes [41].
- Concurrent execution - Since each mango has 6 images in its 6 perspectives, they are input to the surface spot analyzing model and evaluate them parallelly using the multi-threading concept (since we have dual core computers).

```
127.0.0.1 - - [19/Mar/2021 07:10:29] "←[37mGET /resultserial HTTP/1.1←[0m" 200 -
calling spots_Analyzer => image name =>
['1.jpg']
S1 > Grade 1
calling spots_Analyzer => image name =>
['2.jpg']
S2 > Grade 1
calling spots_Analyzer => image name =>
['3.jpg']
S3 > Grade 1
calling spots_Analyzer => image name =>
['4.jpg']
S4 > Grade 1
Apply Size Analyzer
Final Grade => Grade 1
127.0.0.1 - - [19/Mar/2021 07:10:42] "←[37mGET /resultserial HTTP/1.1←[0m" 200 -
```

Figure 11: System log for serial execution

```
127.0.0.1 - - [19/Mar/2021 07:08:41] "←[37mGET /result HTTP/1.1←[0m" 200 -
calling spots_Analyzer => image name =>
calling spots_Analyzer => image name =>
['1.jpg']
['2.jpg']
calling spots_Analyzer => image name =>
['3.jpg']
calling spots_Analyzer => image name =>
['4.jpg']
S2 > Grade 1
S1 > Grade 1
S3 > Grade 1
S4 > Grade 1
Apply Size Analyzer
Final Grade => Grade 1
127.0.0.1 - - [19/Mar/2021 07:09:01] "←[37mGET /result HTTP/1.1←[0m" 200 -
```

Figure 12: System Log for concurrent execution

- Web sockets - Two-way interactive communication session between the user's browser and the server, has been opened to give a real time experience to the user without any interference of network traffic [42]. When we are considering the problem domain, the system should be able to integrate with multiple instances of grading components since there are several grading lanes in the manual grading process. Hence implementing a WebSocket between backend and frontend was an efficient method, to send server-side events by aggregating the results of the all grading instances of each grading lane.

- Asynchronous requests - Angular front end communicate with the python back end using asynchronous requests through observables and subscribe methods. The system sends requests to the server asynchronously rather than wait until the server is ready. And also, we have used promises in Node script to send and receive requests asynchronously. Therefore, we could achieve a high performance.

1.9 Outline of the Dissertation

The dissertation contains seven main chapters each dedicated to an important aspect of the system. Chapter 1 provides an overview of the system from problem definition to the conclusion. Chapter 2 discusses the related work carried out regarding the mango sorting and grading. Chapter 3 gives a detailed description of problem analysis and system design along with UML diagrams. Chapter 4 describes the approach taken, issues identified in each approach and selected approach in detail and the tools and technologies used in the implementation process. Chapter 5 presents the results and analysis of the obtained results of the project work. Chapter 6 explains how the testing was carried out and how the system was evaluated. Chapter 7 provides the conclusion as the final chapter of the body of the dissertation.

Chapter 2 - Literature Review

2.1 Related Works

In recent years, a variety of automated real-time defect/disease identification systems for different harvested fruits and vegetables have been suggested. Although attempts are being made to create general fruit sorting and classification systems [15,16], most systems are dedicated systems that can sort tomatoes [17,18], apples [19], citrus fruit [20], pepper berries [21], eggplant [22], banana [24], mango [25], date palm [23], etc. Computer-vision based approaches to assess the fruit quality differ from one another on the basis of the quality factors and the classification methods that are used in their design. TJC mango is manually classified into grades by considering its external quality factors such as size, amount of surface defects, and its weight [26].

2.1.1 Image acquisition

Ankur M Vyas, Bijal Talati, and Sapan Naik [27] have proposed a method for quality inspection and classification of mangoes using color and size features. Image acquisition was done using a Nikon DSLR camera by keeping the collection of samples inside an image acquisition chamber. A lamp was placed at the top of the chamber in order to provide ambient lightning conditions and to correctly capture the surface of the mango.

In the method proposed by M.A. Momin, et al. [23], the images have been acquired using an XGA format 1/200 Sony CCD ICX205AK color camera of 8-bit gray levels and fitted with a C-mount lens of 6 mm focal length with polarized light (PL) filter. Image acquisition occurred in a controlled environment, where dust and stray light were avoided. The downward-looking camera was placed 220 mm above the samples, which provided a field of view of 150 mm. The distance between the center of a fruit object and the lighting panel was maintained at about 200 mm.

In machine vision techniques for grading of harvested mangoes based on maturity and quality by Chandra Sekhar Nandi, Bipan Tudu, and Chiranjib Koley [31], extracted still frame from the video image at the frame rate of 30fps with a resolution of 640x480 in RGB mode. To minimize the motion blur shutter speed was controlled and fixed to the value of 1/200s. The light intensity inside the image capturing chamber is measured with the help of Lux meter (Instek-GLS-301) and consequently controlled by a light intensity controller to the desired value of 120 Lux.

A comparative study of feature extraction methods in defect classification of mangoes using NN done by VaniAshok and D.S.Vinod [32], images were acquired using a Sony Cyber-shot DSC-WX7 digital camera. Color images were acquired with a resolution of 640 X 480 pixels and 60.5KB size. The image acquisition was done between 12 PM to 3 PM under natural light. The camera was positioned so as to avoid the formation of fruit shadow and white background was used. The images were stored in JPG format.

2.1.2 Preprocessing

The paper titled with the quality inspection and classification method of mangoes using Color and size features [27] has input original image, converted that to grayscale, and then has applied 3×3 median filtering to remove noise while preserving edge information. Then Otsu thresholding was used to segment the mango from its background and applied morphological operations dilation followed by erosion in order to smooth the curvature of the mango and then used the resultant image as a mask in the original image for background subtraction.

In defects classification of mangoes using neural network proposed by Vani Ashok and D.S.Vinod [32] has cropped the acquired images and resized to a dimension of 200 X 150 pixels, 5.58 KB size. Image segmentation was performed to analyze the information necessary for fruit defect assessment. YCbCr color space was chosen to compute global threshold segmentation. Performed morphological operations on the segmented image to remove isolated pixels in the background and to fill holes in the foreground.

Filtering, edge detection, background elimination, alignment of mango images is performed before the extraction of features using different image processing methods in the method proposed by Chandra Sekhar Nandi, Bipan Tudu, and Chiranjib Koley [31].

Payman Moallem, Alireza Serajoddin, and Hossein Pourghassem [28] have applied four steps of preprocessing technique for the data set of golden delicious apple images. As a first step background was removed, A heuristic thresholding method has been used to segment the background, and then morphological filling operations have been applied to fill identified holes inside the segmented apple images. As the next step stem end detection morphological opening operation has been applied on the binary apple image in order to detect the stem end outside of the apple image and has used a disk structural probe with a diameter of 20 pixels. In the third step, the stalk region was detected by applying the K-means clustering(K=2) on the Cb component in the YCbCr color space. The size of each cluster has been considered to correctly identify the stalk region. The last step of preprocessing was the refinement of the defect region. An MLP had been trained to identify the defected and healthy pixels in the apple image by considering the intensity values of each category of pixels. The Levenberg-Marquardt algorithm was used as an optimization algorithm in the MLP training process.

Geometry-based mass grading of mango fruits method [7] proposed an image processing-based algorithm by evaluating different color spaces (RGB, HSI, CMYK). This study has found the HSI color model is superior over RGB and CMYK color spaces. In this study first, RGB values of the original images are determined and have applied HSI transformation. Then the images are binarized using the threshold value taken from the HSI histogram. Then smoothing and dilation operations are applied to the resulting image. Then the image flood filling is applied for size filtering and mangoes are graded into small, medium, large grades using a cutoff value. The projected areas of different grades of mangos in terms of pixel counts and physical units along with other features (perimeter, Feret diameter, and roundness) are considered for grading. Even though this study grade mangoes 96.6% accurately using the area and Feret diameter, it grades mangoes with a 36.6% accuracy using the roundness. Some of the limitations in this study are, the images were manually rectified to avoid shadows and possible objects touching one another, and the effect of fruit orientation was not considered. A more robust approach is needed to implement a formal feature distribution-based classifier technique such as ANN or SVM to improve the grading accuracy.

Ebenezer et al. [29] have implemented an accurate classification model to categorize bananas into two quality groups (defective & healthy) considering the surface spots. In the preprocessing part, a weighted method was used for RGB to grayscale conversion. The grayscale images have been convoluted with a 5×5 median filter for denoising the process. A binary threshold has segmented the image and 3 ×3 Sobel kernel was used to detect edges in the input image. The dilation morphological operation has been applied for the entire original binary image to add the pixels to the image at both inner and outer boundaries and the outer boundaries of the regions.

2.1.3 Feature extraction

Payman Moallem, Alireza Serajoddin, and Hossein Pourghassem have introduced three feature extraction approaches in their apple grading research paper [28] such as statistical features, texture features, and geometric features. In this study, the mean and standard deviation of red(R), green(G), blue(B), and hue(h) components of the image has been derived as statistical features. In this approach, Haralick textural features have been calculated which are computed from Gray Level Co-occurrence Matrices (GLCM). Defect ratio, defect perimeter, defect medial axes length has been extracted as geometric features.

In the study [27] spot pixels are extracted from the segmented image by taking brown and black pixels inside the area of the mango as spot pixels and then the ratio of the spot pixels to the total number of pixels of the mango is taken as a feature. After extracting spot pixels, major and minor axis lengths are extracted and they are converted into inches. This study has focused on Totapuri and Badami mango breeds and it has mentioned Totapuri mangoes are having the oblong shape i.e., the length is more as compared to the width. Therefore, the ratio between the minor axis and major axis has given a clue to classify two mango breeds, and using the major axis length mangoes are classified into different ranges of size (small, medium, large). The color feature of the mango is extracted by the dominant intensity of the ‘a’ channel of the Lab color space. Even though this study has used the ratio between major and minor axes to classify Totapuri and Badami mango breeds, it is not a general solution to classify different mango breeds.

The study [32] obtained four intensity images from each color image. The extracted features are standard intensity features (mean, kurtosis, standard deviation, skewness, mean gradient and mean Laplacian), local binary patterns (LBP), Discrete Fourier Transforms (DFT) and Discrete Cosine Transform (DCT), Hu with intensity, and Gabor filters. Linear binary patterns which are influential features extracted for texture classification. Intensity Gabor textural features, extracted from the magnitude filtered images with different scales and orientations.

Authors [31] estimated the size by calculating the length of the maximum major axis (longitudinal axis) and the maximum minor axis (transverse axis). Surface defects are calculated by getting the difference of average R, B (i.e., R-B) value and G, B (i.e., G-B) value of vertical slice through the longitudinal axis from apex to stalk. The shape information is retrieved using Fourier descriptors.

2.1.4 Feature selection

The paper titled "Computer vision-based apple grading for golden delicious apples based on surface features" [28] has used the SFS algorithm to select the best feature subset from the extracted supper set of features. In this study, sixteen features have been selected as the best feature subset.

Sequential Forward Selection strategy with the objective function of Fisher Discriminant (SFS-Fisher) used as the method of feature selection [32].

2.1.5 Classification method, performance and accuracy

The study [28], In order to categorize 120 mango images (360 x 360 pixels) into three groups such as first rank, second rank, and rejected, extracted best feature subset with sixteen features have been trained under support vector machine (SVM) with 89.2% accuracy, multilayer perceptron algorithm (MLP) with 86.6% accuracy and K-nearest neighbor (KNN) with 85.8% accuracy.

The method that Ebenezer et al. [29] has suggested for banana grading, was implementing an FFNN with SGD optimization algorithm. Grading accuracy was 97% with 3000 epochs.

The study [27] has not used any machine learning-based training model to grade mangoes but it has used image processing techniques-based algorithms, Lab color-model-based histograms to extract color features. The dataset of the mangoes is collected in Unripe, Semi Ripe, and Ripe phases, and based on the extracted parameters (size, color, spots) with grading rules, mango is classified into grade1, grade2, grade3 or rejected. Grading accuracy of 94.97% was observed and the proposed method does the grading of mango in less than a second.

Rashmi Pandey, Prof. Nikunj Gamit, and Prof. Sapan Naik have developed A Novel Non-Destructive Grading method for Mango [30] and it has classified mangoes into Poor (Ql), Medium (Q2), Good (Q3), and Excellent (Q4) Quality categories based on length of the major axis and minor axis. As the classification approach, a fuzzy expert system has been suggested with an overall accuracy of 96.58%.

The study [32] has developed a NN using a generalized linear model. The performance of the classifier was evaluated considering different output activation functions like linear, logistic, and SoftMax.

Authors [31] predict the maturity level in terms of actual-days-to-rot using SVR and evaluation of quality using MADM is done separately. A fuzzy incremental learning algorithm has been applied to classify the mango into four different grades.

2.2 Comparison of Related Works

Title of the paper	**Author**	**Comparison**
Quality Inspection and Classification of Mangoes using Color and Size Features [27]	Ankur M Vyas, Bijal Talati, Sapan Naik	● The main feature parameters are size and color. ● Only one side of the apple image has been considered. ● Applying thresholding directly to binarize the image doesn't work due to the defects and shadow of mango. ● Cannot take an idea about the pattern of spots (the above method cannot be directly applied to classify grade3 and grade 4 in our scenario) since this study only takes a ratio between the number of black pixels and the total number of mango pixels. ● The ratio between the major and minor axis doesn't give any clue about the size however we can use major axis length to decide the size of a mango.
Computer vision-based apple grading for golden delicious apples based on surface features [28]	Payman Moallem et al.	● Only one side of the apple image has been considered. ● Binary thresholding and morphological operation (erosion)have been used to segment the image in this research, however in a scenario where the size is important, applying erosion will be a case for reducing the size of the image. ● Segmenting the defect region using the MLP algorithm is not successful since the

		defect regions and black spots have similar intensity values. (In our scenario grade 4 TJC mangoes which have defects and 2,3 TJC mangoes which have spots cannot identify correctly using this method).
Geometry-based mass grading of mango fruits using image processing	M.A. Momin et al	● Only one side of the apple image has been considered. ● Proposed techniques are simple compared to current CV based systems. However, the proposed method can be improved by using a formal feature distribution-based classifier such as ANN, SVM. ● Observations of issues have been made with regard to the unsmooth periphery due to erroneous segmentation in some binary images. The grading algorithm classified mangoes into 3 standard mass grades and the approach was based on human perception of overall size during manual grading.
Intelligent grading system for banana fruit using neural network arbitration [29]	Ebenezer obaloluwa olaniyi et al	● The flatted image has been applied for the feed-forward neural network (FFNN) as the input feature vector, hence viewpoint variation of spots was not considered by the neural network in the training process.
A machine vision technique for grading of harvested mangoes based on maturity and quality [31]	Chandra Sekhar Nandi, Bipan Tudu, and Chiranjib Koley	● Consider only one side of mango. ● The main feature parameters are size, shape, and surface defects.
A comparative study of feature extraction methods in defect classification of mangoes using neural network [32]	VaniAshok, D.S.Vinod	● Consider only one side of mango. ● Considered only textural features. ● Have used global thresholding and morphological operations for segmentation.

Table 2: Comparison of related works

2.3 Summary of Related Works

In the study "Quality Inspection and Classification of Mangoes using Color and Size Feature's [27] the main feature parameters they have considered are size, color and surface spots. They considered only one side of the apple image. Final outputs are categorized as Unripe, semi-ripe, ripe and grade 1, grade 2, grade 3, rejected. The overall accuracy is 94.97%. In the study "Computer vision-based apple grading for golden delicious apples based on surface features" [28] they have considered statistical (mean, standard deviation of red, green, blue and hue components of the image), textural (Haralick textural features have been calculated which are computed from GLCM), geometrical features (defect ratio, defect perimeter, defect medial axes length). They have considered only one side of the apple image. Final outputs given by this research are first rank, second rank, rejected and the accuracy for SVM with 89.2% accuracy, MLP with 86.6% accuracy and KNN with 85.8% accuracy has been given by the system. In the study, "An intelligent grading system for banana fruit using neural network arbitration" [29] used only one image for grading. Main feature parameter they have input to the system is the number of surface spots. Final outputs of this system were defective and healthy. In the study "A machine vision technique for grading of harvested mangoes based on maturity and quality" [31] they have considered only one side of mango. The main feature parameters are size, shape, and surface defects. A comparative study of feature extraction methods in defect classification of mangoes using neural network [32] has only considered only one side of mango. It has only considered only textural features. In our research work we have considered feature parameters such as size, surface spots and weight to classify mango into 5 grades. They are namely grade 1, grade 2-small, grade 2-large, grade 3 and grade 4. We took images for each mango from six perspectives. We achieved 93% of average accuracy.

Chapter 3 - Analysis and Design

This chapter provides a brief description of problem analysis and overall description of functional requirements, non-functional requirements, software engineering principles in detail. Further, it discusses architectural design along with any design assumptions and model diagrams

3.1 Problem Analysis

Although the ultimate goal of this research concept is to automate the TJC mango grading process fully, this study only focuses on classifying mangoes into correct grade using an accurate algorithmic model. The hardware solution is not expected to be implemented in our research. Since the manual process of mango grading is mainly based on the experience of human expertise, as a modern IT solution ML model was able to be used to automate this process. As a result of requirement analysis size, weight, and number of surface spots were identified as the main factors that affect the human expertise to make the decision.

3.2 Functional Requirements

Following are the main two functionalities of this system.

- Analyze images and find the correct quality grade of input TJC mango image
- Display the relevant grade of the mango on the dashboard
- Display analytical details of graded mangoes on the dashboard

This product is an independent application which performs its operations without being integrated with other software systems. It is a self-contained python application which only has GUI to the user. Following is the flow of actions in the system which are applicable to the process of a particular mango.

1. Run the application.
2. Connect the 3 cameras to the computer.
3. Cameras will capture 6 images
4. The captured 6 images will display in the user interface
5. The relevant grade will be displayed

3.3 Quality Attributes

3.3.1 Accuracy

Since the TJC mango variety is exported from Sri Lanka the quality must be ensured. Our client company is exporting only Grade 1 mangoes. Therefore, the system should accurately sort mangoes. If the system assigns lower grade mangoes to higher grade mangoes that won't be aligned with the grading standards. As a result, the company reputation will be harmed as well as the number of sales will be decreased resulting in a financial loss.

3.3.2 Performance

Performance is related to the response time of a system. Time taken by a manual grader to grade a single mango is around 5 seconds therefore, the process of assigning the relevant grade to each mango should happen in less than 5 seconds.

3.3.3 Usability

The main user of this system is a non-technical or novice person. Hence the system should be user-friendly for them to easily learn the system.

3.4 System architectural design

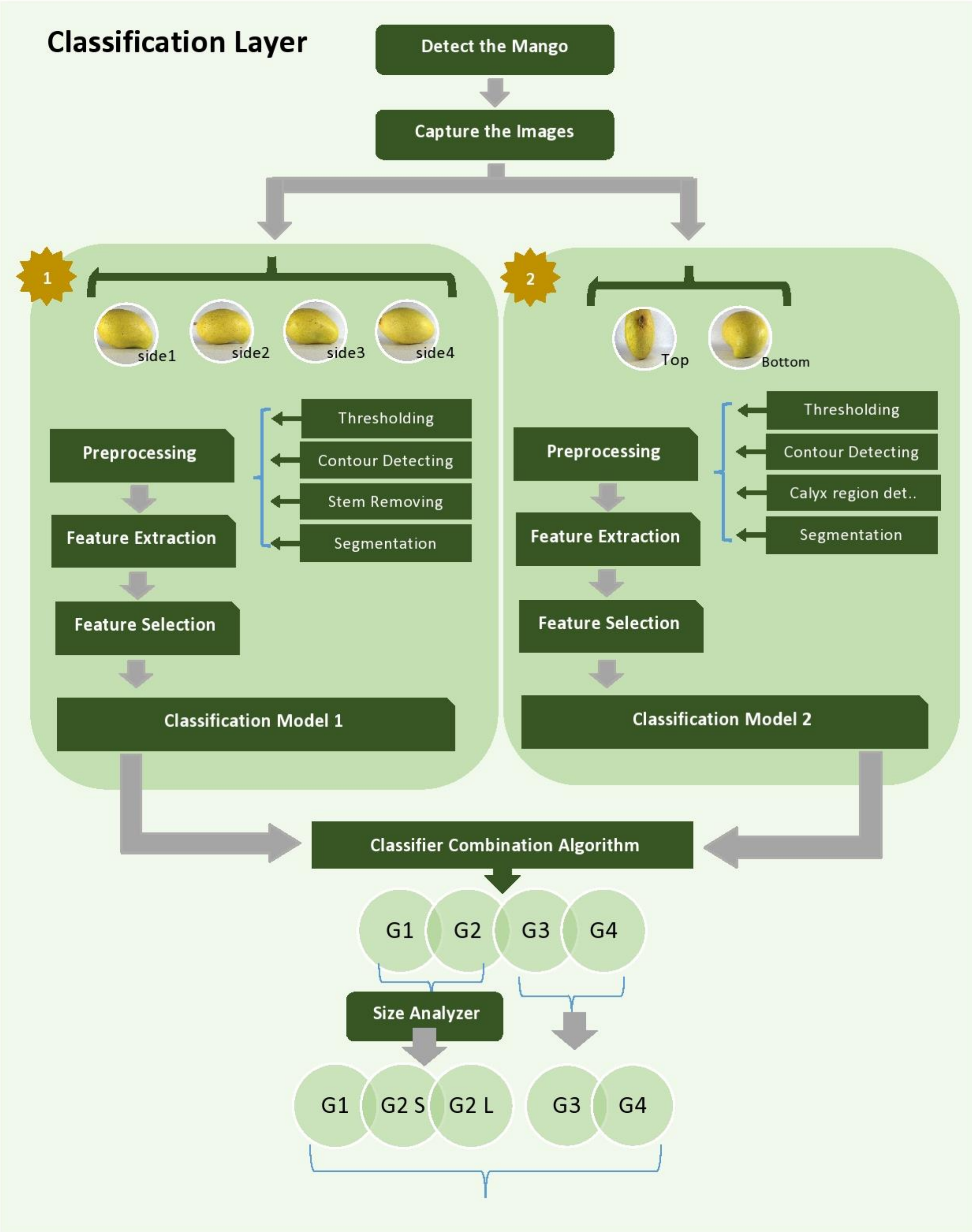


Figure 13: System architectural design(part I)

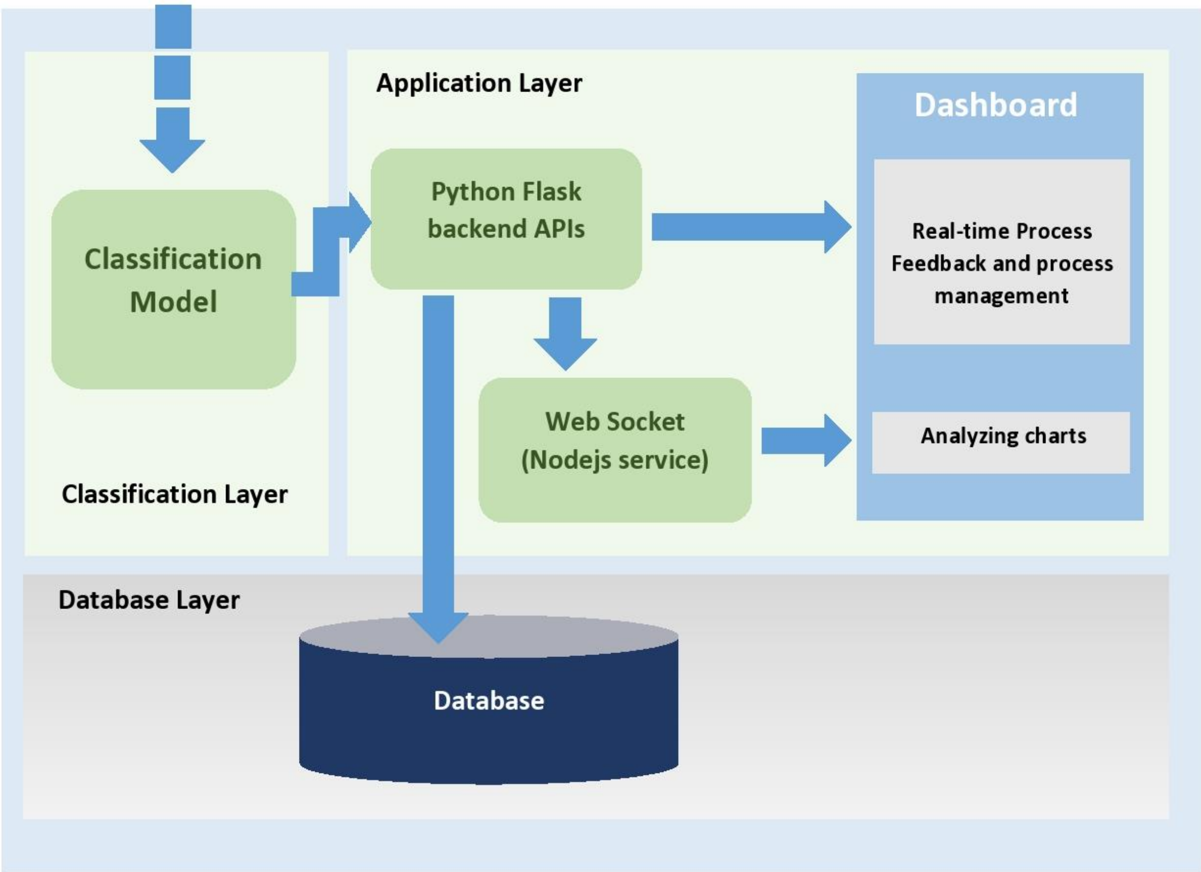


Figure 14: System architectural design(part II)

3.5 Architectural Overview

The classifier architectural design is depicted in [Fig. 9]. It provides directions to guide and automate the mango grading process based on layered architecture, with each layer providing a clear set of functions to the layers above. The four layers from the top to bottom are:

- Classification Layer
- Application Layer
- Database Layer

The objective of the classification layer is to evaluate the final grade of a given TJC mango and pass it to the Applications and Database layers. Firstly, the system detects a mango then the camera which is connected to the system captures six sides of images from each mango. According to the architectural design, the captured images are separated into two groups considering sides of the images by the system, and two classification components (component 1 and component 2) are executed parallelly for each set of images that were separated. Each classification component follows below steps.

- ❖ Preprocessing
- ❖ Feature Extraction
- ❖ Feature Selection
- ❖ Classification Model

In the preprocessing stage, a four-step image enhancement pipeline is applied to enhance the key features. According to the architectural design, the preprocessing part of component 1 consists of image thresholding, contour detection, stem end reduction, and segmentation . The preprocessing part of component 2 consists of image thresholding, contour detection, stalk region detection, and segmentation. Separate classification algorithms are built for each component and they are trained by relevant data sets. SFS approach is used to derive the best feature subset from the extracted features to increase the accuracy of the trained models.

In order to combine the result of above mentioned two classifiers, a classifier combination algorithm is implemented considering fuzzy rule base concepts. The predicted result set of this component contains grade1 (G1), grade2 (G2), grade3 (G3), and grade4 (G4) labels. The mangoes which are labeled as G1 or G2 are fed into the size analyzing component for further classification. The size analyzing component is responsible for categorizing them into three groups such as grade1 (G1), grade2 large (G2 L), and grade2 small (G2 S).

The Python Flask APIs are used to communicate between the classification layer and the application components. The web socket protocol is used to give a real-time response to the application layer without any network traffic. The dashboard basically consists of functionalities such as generating charts, real-time process feedback generation and process generation.

3.6 System Modeling

3.6.1 Use case diagram

The main users of this system are system users and the system itself. The system user can change parameters which are taken to classify mangoes into correct grades. Those parameters will be the size and the weight of mangoes.

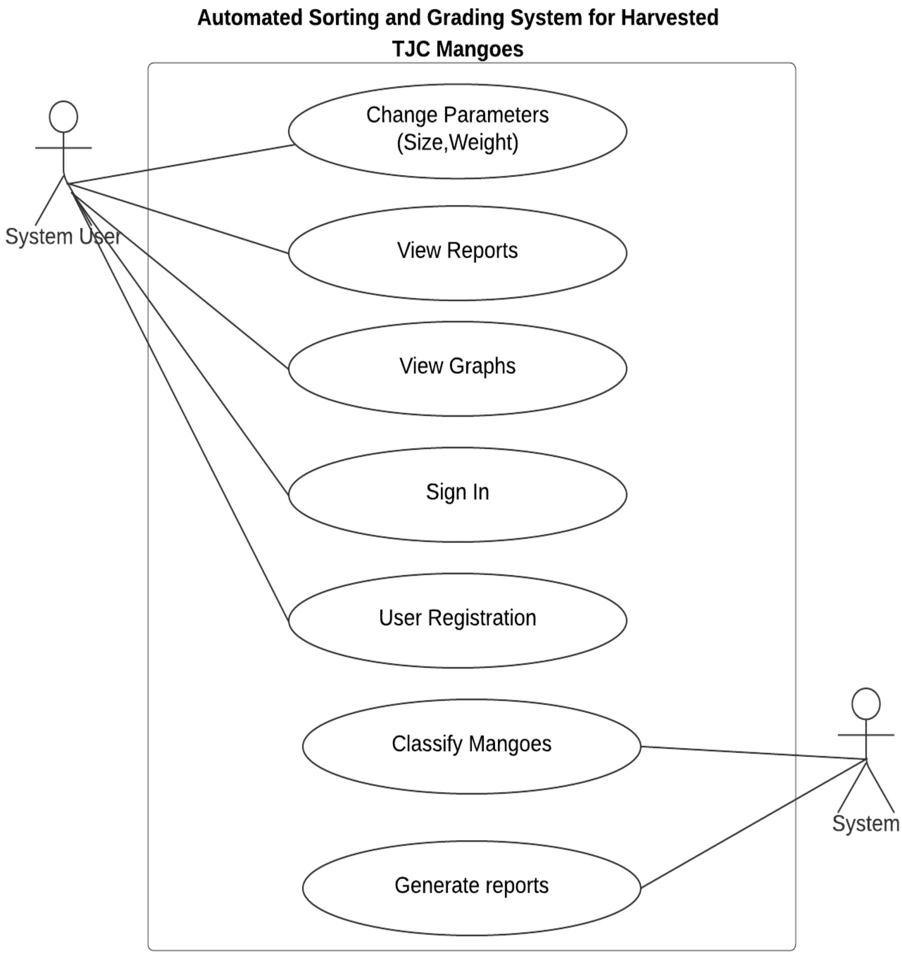


Figure 15: Use case diagram

3.6.2 Class diagram

The class diagram shows the relationships between classes namely application GUI, system user, training, report, prediction, preprocessing, classifier model and the databean along with their attributes and operations.

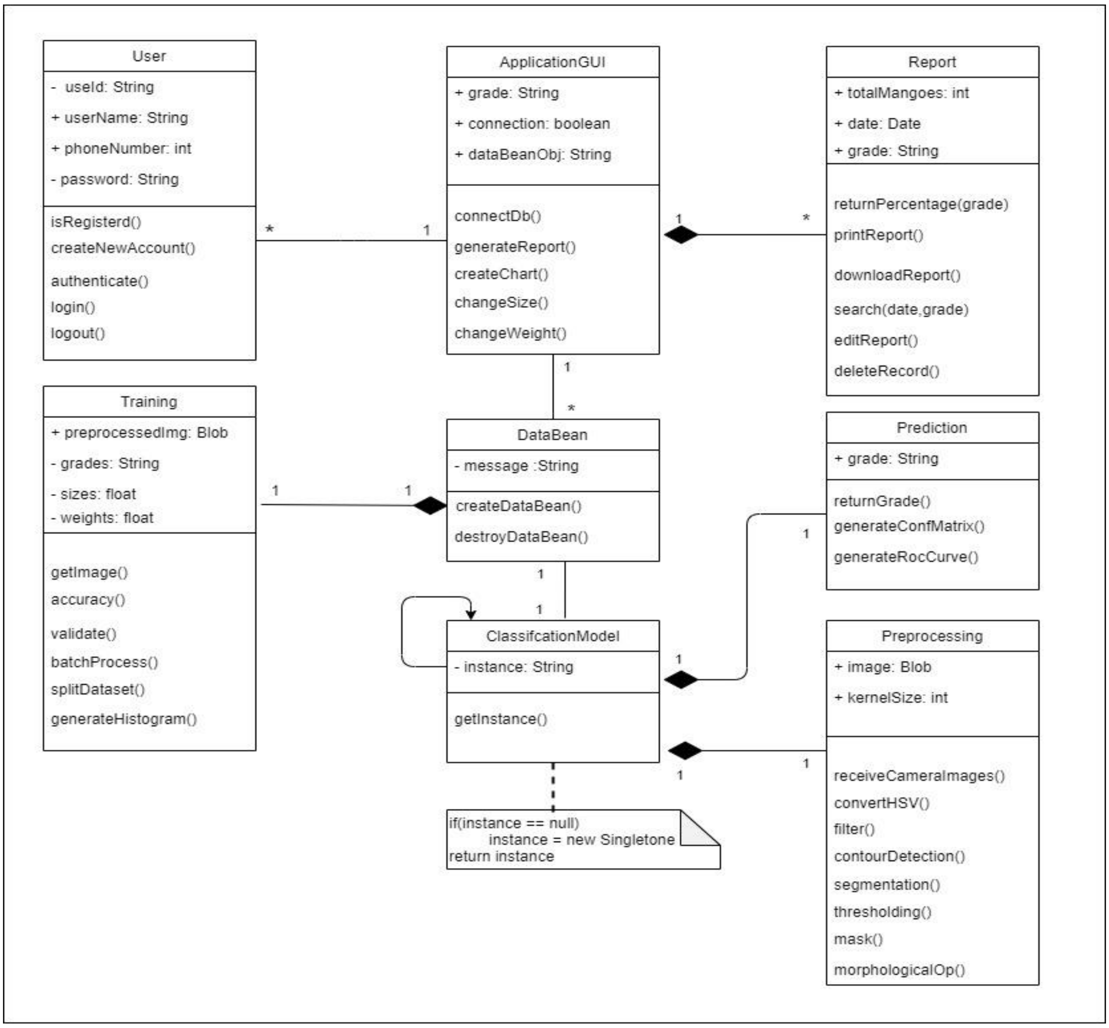


Figure 16: Class diagram

Chapter 4 - Implementation

4.1 Software Development Process Model

The iterative and incremental development process was followed at the implementation stage of the project. At first, mango images were captured and pre-processed. Then a CNN model was trained with those images. Since it didn't give a good result, some other classifiers were also built using more optimal algorithms. When the new images are obtained, the model has been trained over and over again. That means the process is iterative. As the other requirements were implemented, the system is incremented by one functionality. Main increments were the data preprocessing, feature extraction, feature selection, model building, training of the model, generating the final result, displaying the result in a dashboard and user interfaces integration. Hence the process is incremental.

4.2 Tools and Technologies

- Operating system: Windows 10 or above
- Database/Data file:
    - MySQL- (MySQL Workbench)
    - Preprocessed data set was stored in "**.npy**" file format hence it reduces the accessing time of the data.

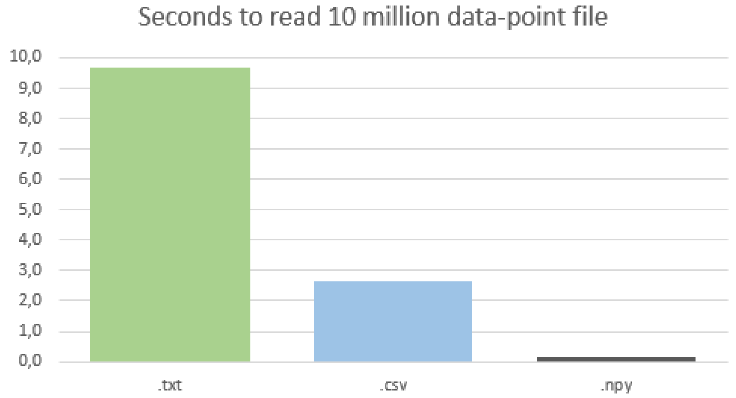


Figure 17: Seconds to read 10 million data-points[45]

    - Hierarchical Data Format 5 File (“**.H5**”) file format was used to store the trained knowledge.
- Development IDE: Jupyter, Google Colab, Spyder
- Development language of algorithmic part: Python 3.6
- Development language of Backend APIs: Python 3.6
- Development language of communication protocol: Node js
- Frontend development: Angular 8
- Version Controlling: GitHub
- Application Containerization: Docker
- Project management tool: Trello board

4.3 Justification for Tools and Technologies

Technology	**Purpose**	**Reason**	**Version**
Python	For implementing classifier models and to do image processing.	• Many libraries for image processing and neural networks exist for python which will facilitate the development process.	python 3.6
Jupyter & Spider	IDE for developing python applications.	• Ability to execute the code segment wise	V 6.2
Visual Studio Code	IDE for developing javascript applications.	• Ability to integrate with GitHub • Ability to integrate with the testing framework.	V 1.50.0

MySQL workbench & xampp server	Software for relational database management	● Ability to execute the DB query operations	V 8.0.23 V 8.0.1
TensorFlow	Framework for developing and using Neural networks.	● Availability of existing neural network models. ● Availability of functions necessary for operation in neural networks development ● Free and open source	TensorFlow 1.10.0
Python Flask	Framework for developing backend APIs.	● Framework for developing backend APIs.	V 1.1.2
OpenCV	For image processing purposes.	● Possess a variety of image processing techniques. ● Have much fast image processing techniques. ● Free and open source	V 3.4.0
Angular	Application framework for developing the desktop GUI	● single page application development component wise. ● responsive	Angular 9.1
MySQL	For storing the outputs of classification process	● Ability of managing the relational databases.	V 8.0

Nodejs	For implementing the WebSocket between backend and frontend	• It is easy to implement the WS with Nodejs, which has provided sock.io library	V 12.13.0
Chart js	For implementing the chart generating functionalities	• For implementing the chart generating functionalities	V 2.9.4
Docker	For solving the version errors when system is executing the different environment	• Whole system can be containerized with docker	V 3.0.0
NumPy Pandas Matplotlib Scikit-learn,etc...	For data manipulation tasks	• For data manipulation tasks	
GitHub	For version controlling the application	• Easy and familiar version controlling software • Easy to integrate with Vs code	Git 2.6.3

Table 3: Justification for tools and technologies

4.4 Graphical User Interface (GUI)

- Creating a new process with new process id and time stamp

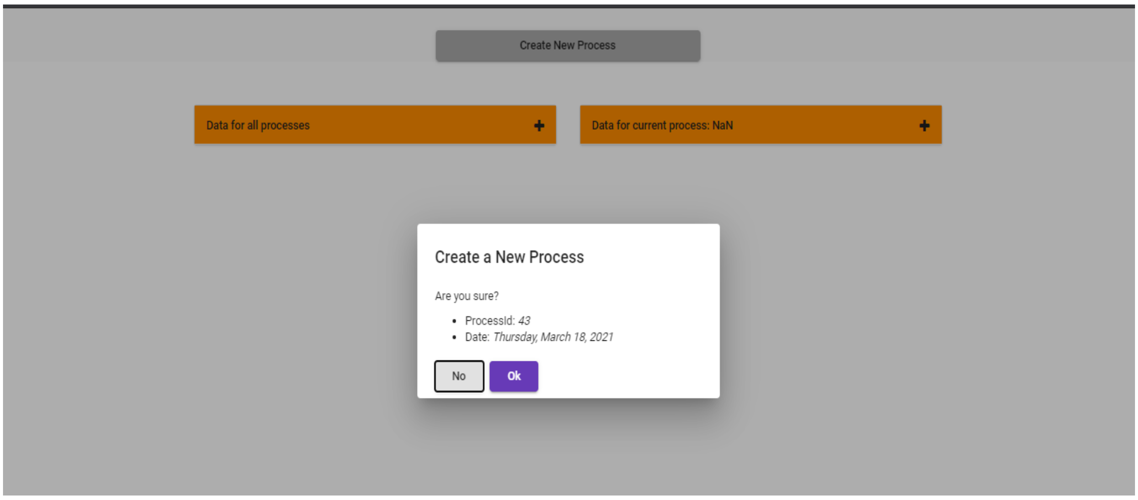


Figure 18: Creating a new process(Part I)

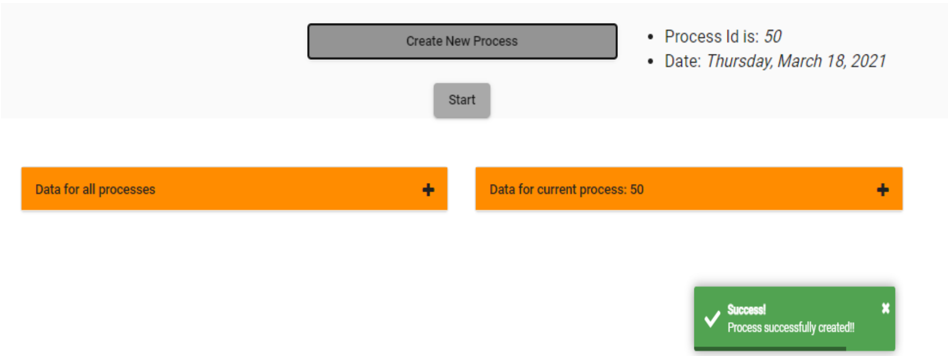


Figure 19: Creating a new process(Part II)

- After successfully creating a new process, the user can start the grading process. Grading results will be shown on the dashboard.

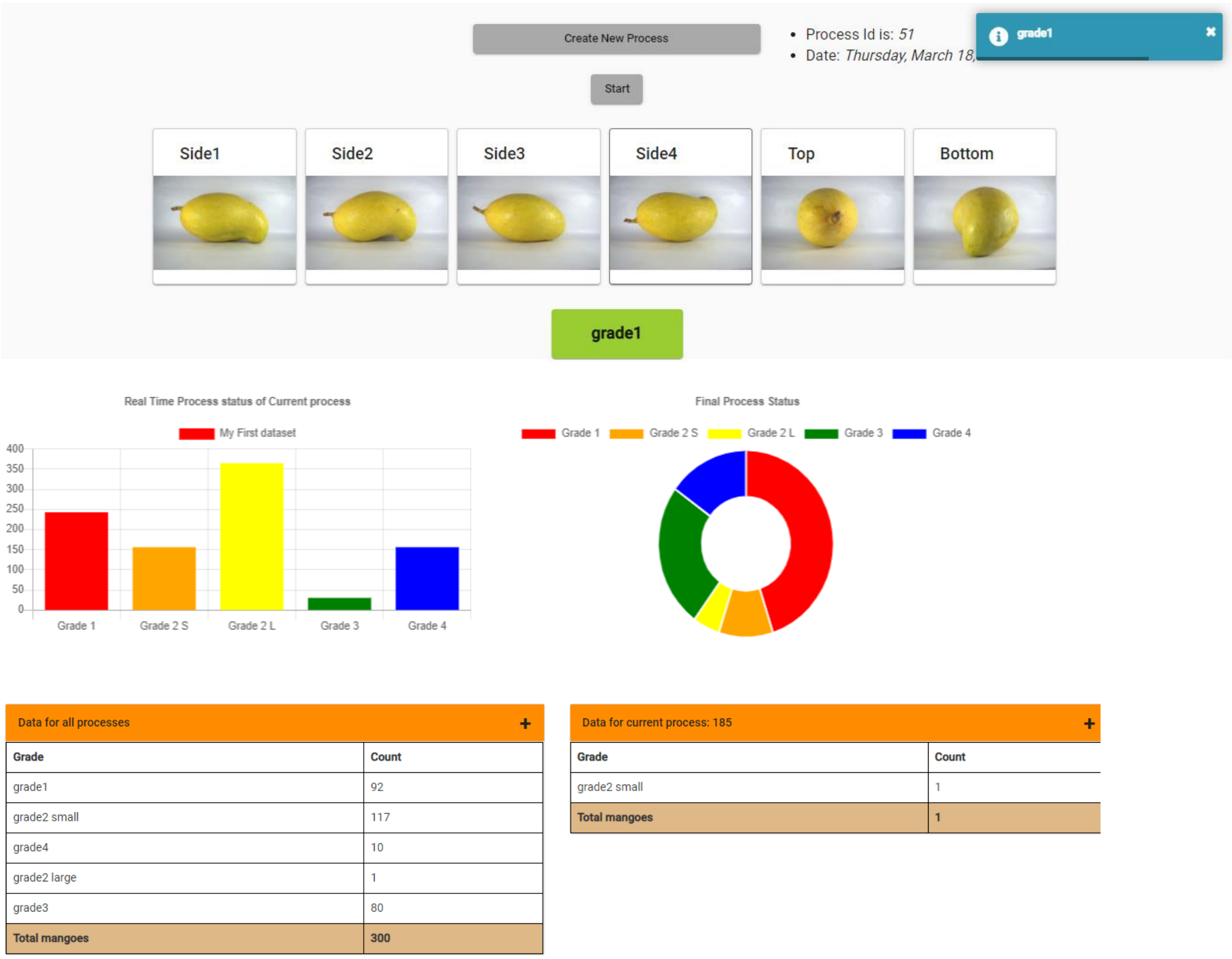


Figure 20: Showing the final result

- Our system provides two analytical charts to the user to get information about the current status of the grading process in real time.
- Bar chart → To visualize the processed mango count of the current process.
- Pie chart → To visualize the processed mango count of all processes in each grade

Figure 21: Error handling (part1)

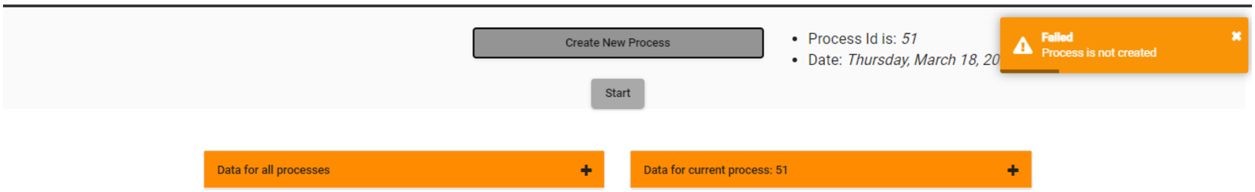


Figure 22: Error handling (part2)

4.5 Methodology

4.5.1 Data Gathering

Figure 23: Data gathering *Figure 24: Sorted mango*

Data collection is a systematic process of gathering observations or measurements. For this study, we had to make a dataset from scratch since we couldn't find any publicly available dataset that suits our research. In order to collect images, we did a field visit to "Nelna Agri Mango Outlet" which is in Malabe [Fig. 23]. It's a distribution center of harvested TJC mangoes which are already sorted manually [Fig. 24].

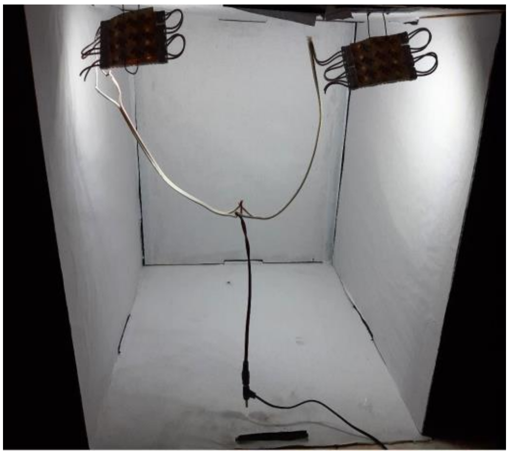


Figure 26: Image capturing chamber

Figure 25: Image capturing

We made an image capturing chamber [Fig. 25] to keep the mango and used a white background to ease the segmentation process. Uniform lighting LED sources used to ensure no shadow in the image and to get a clear image. The camera is placed in front of the mango [Fig. 26] and the distance between the object and the camera is 25cm. Images were acquired using a NIKON D5500 DSLR camera with ISO 2500, shutter speed 1/100 sec, and the focal length was 24mm. The images were stored in the NEF format.

From each mango, 6 sides of images were taken. From each grade 40 mangoes are considered. Therefore, the total number of images in our dataset is 1200. [Table 4] shows 6 sides of mango from each grade.

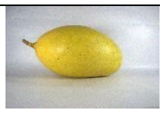
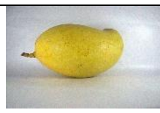
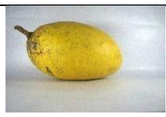
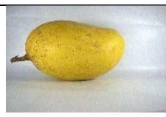
Grade	Side 1	Side 2	Side 3	Side 4	Side 5	Side 6
G1						
G2 Large						
G2 Small						
G3						
G4						

Table 4: Sample data set

Mango size is an important quality attribute when grading. The size is determined by measuring the height and width of the mango using a tape measure. The accuracy of the tape measure was 1mm. Skin dark spots and weight are other main factors they consider when grading. We measured the weight using a digital scale which has an accuracy of 0.1g.

4.5.2 Preprocessing

The aim of the pre-processing is to improve the image data by suppressing unwilling distortions and enhancing some image features important for further processing. The following sections describe the applied preprocessing pipeline.

4.5.2.1 Preprocessing for 4 sides

As for the first step, the original image was converted from RGB to HSV color scale. That color conversion enhanced the color separation of the input image [Fig. 27]. The main reason to choose HSV color space is that it separates color information from intensity or lighting. Therefore, we can construct a histogram or thresholding rules using only saturation and hue. Regardless of lighting changes in the value channel, it works finely with a better improvement. Even by singling out only the hue, we have a very meaningful representation of the base color that will likely work much better than RGB. The end result is a more robust color thresholding over simpler parameters.

Figure 27: Converted to HSV

Then we applied hysteresis binary thresholding [Fig.28] operation to segment the image from its background [Fig. 29].

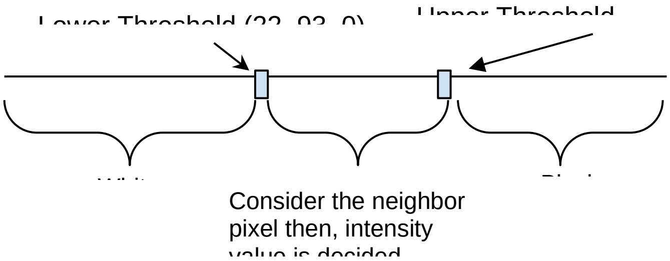


Figure 28: Hysteresis binary thresholding

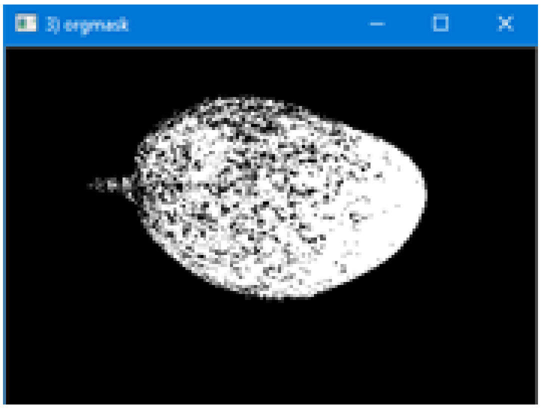


Figure 29: Thresholded image

Next, we applied the Gaussian filter [Fig. 31] to blur the image. Then we detected the contours of the image by joining all the continuous points (along the boundary) having the same color or intensity. Since there are black spots in the image, many contours [Fig. 30] were able to detect.

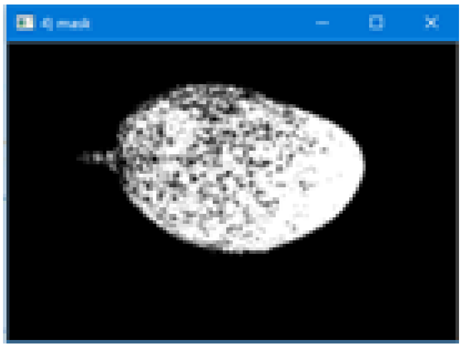


Figure 31: Gaussian blurred image

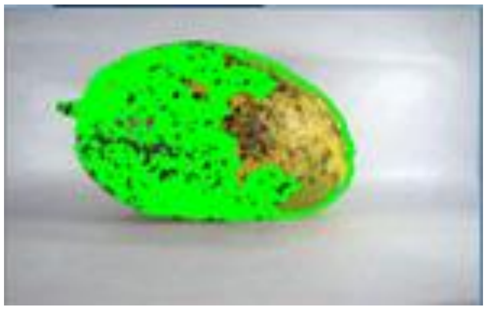


Figure 30: Contour detected

Figure 33: Detected boundary contour

Figure 32: Detected bounding

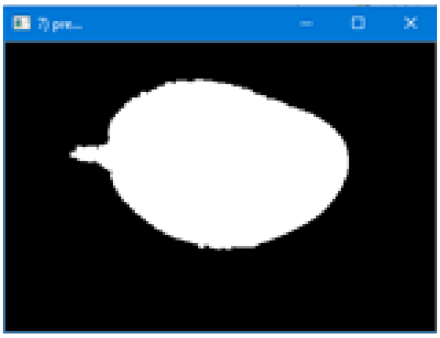
When considering the size analyzer, it is essential to detect the bounding rectangle without a stem when extracting features.

The morphological opening operation which erodes an image and then dilates the eroded image. Both the erosion and dilation operations use the same structuring element. Morphological opening operation is useful when removing the small objects from an image while preserving the shape and size of larger objects in the image and also it doesn't reduce the size of an object by a considerable amount. Hence the morphological opening operation was applied to remove the stem end from the mango image. In order to apply morphological operation, the image should be a pure binary one [Fig. 35]. To make a binary image from the color image, applying thresholding technique was not sufficient, hence the mathematical equation of the selected contour was calculated. And then by substituting the coordinates of each pixel to the derived mathematical equation the pixel is classified into an internal pixel or an external pixel.

The intensity value of the internal pixels was (255,255,255) and the external pixels were (0,0,0).

Algorithm that was used to classify pixels into internal and external.

1. $g(x, y) = calculate\_equation\_of\_contour(\ selected\ contour\ );$
2. $for\ \ x`, y` \leftarrow image:$ $\#\ get\ each\ pixel\ one\ by\ one$
3. $value \leftarrow g(x`, y`);$
4. $if\ \ value >= 0:$
5. $image[x`, y`] = (255{,}255{,}255);$ $\#\ changing\ the\ internal\ intensity\ valueS$
6. $else:$
7. $image[x`, y`] = (0{,}0{,}0);$ $\#\ changing\ the\ external\ intensity\ values$



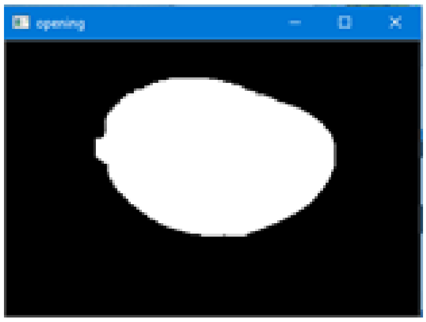


Figure 35: Pure binary image

Figure 34: Stem removed

The structuring element used for opening operation has been mentioned below.

$$\begin{bmatrix} a_{11} & a_{12} & ----a_{1n} \\ a_{21} & a_{22} & ----a_{2n} \\ --- & ---- & ------ \\ --- & ---- & ------ \\ a_{m1} & a_{m2} & ----a_{mn} \end{bmatrix}$$

from a_{11} *to* $a_{mn} = 1$

$n = m = 17$

From the stem removed binary image [Fig. 34], a new contour and a bounding rectangle was detected correctly as in [Fig. 38]. Using the derived mathematical formula of the new contour [Fig. 37], background was removed successfully [Fig. 36].

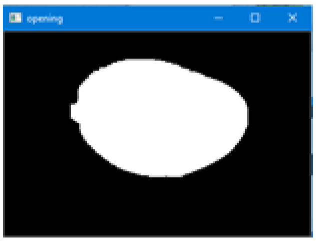


Figure 39: Stem removed

Figure 36: Detected contour

Figure 38: Bounded rectangle

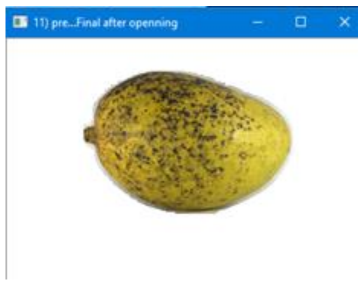


Figure 37: Background removed image

The objectives of detecting contours

- For size analyzing components, it is necessary to extract measurement of the mangoes.
- In order to remove the stem, morphological operation was applied. Therefore, it needs a binary image. Thresholding technique was applied to segment the mango image from its background. However, because the images had internal black spots[Fig. 29], it was not successful. Therefore, the morphological erosion operation was applied to erode the internal black spots from the thresholded image. However, that also was not an effective solution since it reduced the size of the mango. So, by using detected contour [Fig. 33], we were able to create a purely binary segmented image [Fig. 35] from the background as mentioned above.
- Detecting contours is important to identify the foreground pixels from the background correctly. It is useful in our feature extraction process.

4.5.2.2 Preprocessing for the top side image

The preprocessing pipeline for the top images is the same as preprocessing we did for 4 sides, however, there was no stem reduction functionality. The region around the stalk may affect the training process. Hence the K-Means clustering algorithm and K-Medoid clustering algorithm were used with K = 3 to identify the stalk region. The red circle in the [Fig.40] shows the region around the stalk.

Figure 40: Region around the stalk

When considering the shape of the stalk region, K- Means and K- Medoid algorithms are used to identify globular clusters.

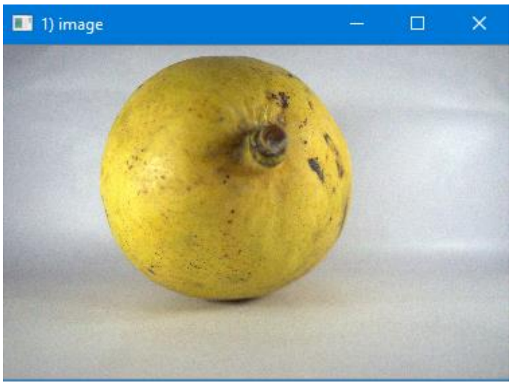


Figure 42: Original image

Figure 41: HSV converted image

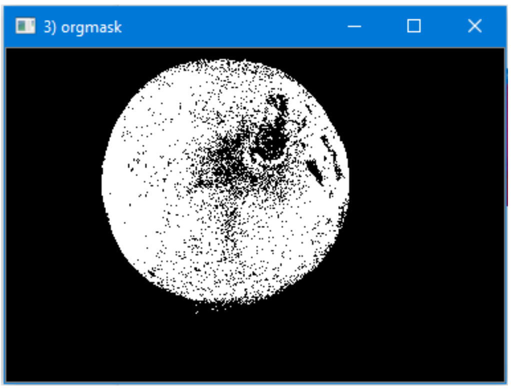


Figure 44: Hysteresis binary thresholding

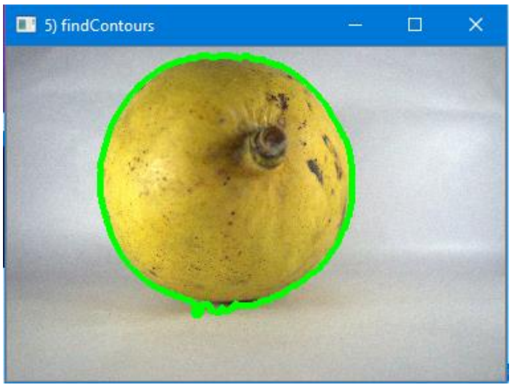


Figure 43: Final result of the contour detection process

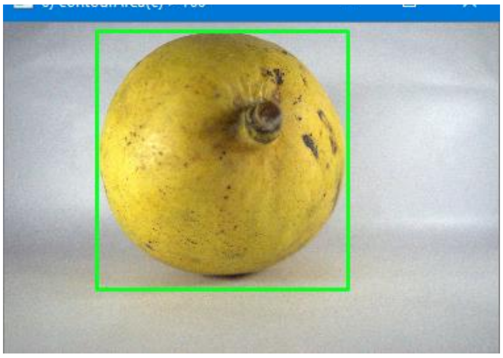


Figure 45: Detected bounding rectangle

Figure 46: Background removed image

Note: The implemented K-mean clustering and K- medoid clustering algorithms have not given 100% acceptable results.

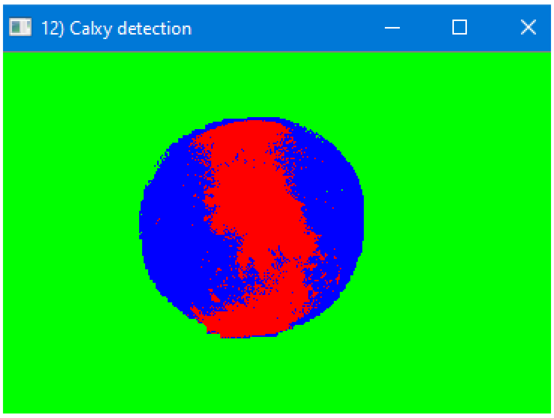


Figure 47: Stalk region detected image using the K-Means algorithm

4.5.2.3 Preprocessing for the bottom side image

The preprocessing pipeline for the bottom images is the same as preprocessing we did for 4 sides, however, there was no stem reduction functionality

4.5.3 Implementing the Classifier

Two main classifiers were implemented to classify mangoes based on surface spots and the mango size. We implemented three surface spots analyzing classifiers separately for side images, top image and bottom image. Under deep learning algorithms, Feed forward neural network (FFNN) and convolutional neural network (CNN) as well as under the machine learning classification algorithms, k- nearest neighbor (KNN) and support vector machine (SVM) used to implement each classifier. We implemented a classifier to classify mangoes based on its size using 4 sides images. We implemented FFNN, KNN and SVM based algorithms for that.

4.5.3.1 Feed Forward neural network(FFNN) Based Classifier Models

A feed forward neural network is a biologically inspired classification algorithm. It consists of a (possibly large) number of simple neuron-like processing units, organized in layers. Every unit in a layer is connected with all the units in the previous layer. These connections are not all equal due to each connection may have a different strength or weight. The weights on these connections encode the knowledge of a network. the data enters at the inputs and passes through the network, layer by layer, until it arrives at the outputs. A feed forward neural network is commonly seen in its simplest form as a single layer perceptron. In this model, a series of inputs enter the layer and are multiplied by weights. Each value is then added together to get a sum of the weighted input values. By considering the activation function and the sum of the weighted value, the final return value will be derived. In this research FFNN based four classification models were implemented. They are;

1. Classifier based on surface spots - for 4 sides images
2. Classifier based on surface spots - for the top side image
3. Classifier based on surface spots - for the bottom side image
4. Classifier based on mango size - for 4 sides images

4.5.3.2 Convolution Neural Network(CNN) Based Classification Models

Convolutional neural network consists of multiple building blocks, such as convolution layers, pooling layers, and fully connected layers. It is designed to automatically and adaptively learn spatial hierarchies of features through a backpropagation algorithm [35]. CNN's make use of filters (also known as kernels), to detect what features, such as edges, are present throughout an image. A filter is just a matrix of values, called weights. They are trained to detect specific features. The filter moves over each part of the image to check if the feature it is meant to detect is present. When the feature is present in part of an image, the convolution operation between the filter and that part of the image results in a real number with a high value. If the feature is not present, the resulting value is low. Additionally, a filter can be slid over the input image at varying intervals, using a stride value. The stride value dictates by how much the filter should move at each step. To speed up the training process and reduce the amount of memory consumed by the network, we try to reduce the redundancy patterns present in the input feature. There are a couple of ways we can down sample an image, the most common one is max pooling. In max pooling, a window passes over an image according to a set stride (how many units to move on each pass). At each step, the maximum value within the window is pooled into an output matrix, hence the name max pooling. After multiple convolutional layers and down sampling operations, the 3D image representation is converted into a feature vector that is passed into a Multi-Layer Perceptron, which merely is a neural network with at least three layers. This is referred to as a Fully-Connected Layer. In the fully-connected operation of a neural network, the input representation is flattened into a feature vector and passed through a network of neurons to predict the output probabilities. In this research CNN based classification models were implemented for three classifiers such as surface spots analyzer-side images, surface spots analyzer-top images, surface spots analyzer-bottom images.

4.5.3.3 K-Nearest Neighbor(KNN) Based Classification Models

KNN is a powerful classification algorithm used in pattern recognition. An object (a new instance) is classified by a majority vote for its neighbor classes. The object is assigned to the most common class among its K nearest neighbors.(measured by a distant function)

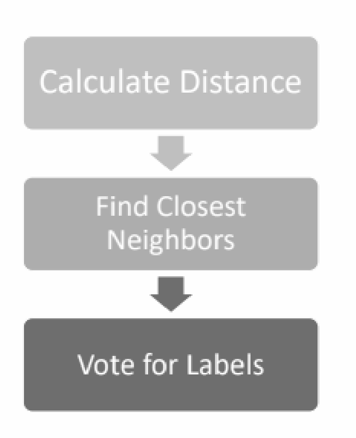


Figure 48: Classifier work flow

In this research, KNN based four classification models were implemented. They are;

1. Classifier based on surface spots - for 4 sides images
2. Classifier based on surface spots - for the top side image
3. Classifier based on surface spots - for the bottom side image
4. Classifier based on mango size - for 4 sides images

4.5.3.4 Support Vector Machine (SVM) Based Classification Models

The SVM classifier separates data points using a hyperplane with the largest amount of margin. Also known as a discriminative classifier. SVM finds an optimal hyperplane which helps in classifying new data points. It can easily handle multiple continuous and categorical variables. SVM constructs a hyperplane in multidimensional space to separate different classes. SVM generates optimal hyperplanes in an iterative manner, which is used to minimize an error. 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. In this research SVM based four classification models were implemented. They are;

1. Classifier based on surface spots - for 4 sides images
2. Classifier based on surface spots - for the top side image
3. Classifier based on surface spots - for the bottom side image
4. Classifier based on mango size - for 4 sides images

4.5.3.5 Surface Spots Analyzer

4.5.3.5.1 Feature Extraction and Selection

Considering the previous studies done on feature selection methods, the SFS method was decided to apply in our scenario to identify the best feature subset that gives a high accuracy. Statistical details of color and textural type features were extracted from the preprocessed image [Fig. 37]. 12 statistical features and 2 textural features were extracted. Those are mentioned in below [Tab. 5].

Feature category	Feature
Statistical parameters of color components of different color scales	RGB - red channel - Mean value of intensity distribution of red
	RGB - red channel - Standard deviation value of intensity distribution of red
	RGB - green channel - Mean value of intensity distribution of red
	RGB - green channel - Standard deviation value of intensity distribution of red
	RGB - blue channel - Mean value of intensity distribution of red
	RGB - blue channel - Standard deviation value of intensity distribution of red
	YCbCr - Y channel Mean value of intensity distribution of red
	YCbCr - Y channel Standard deviation value of intensity distribution of red
	YCbCr - Cr channel Mean value of intensity distribution of red
	YCbCr - Cr channel Standard deviation value of intensity distribution of red
	YCbCr - Cb channel Mean value of intensity distribution of red

	YCbCr - Cb channel Standard deviation value of intensity distribution of red
Statistical parameters of textural features	Mean value of HOG distribution
	Standard deviation of HOG distribution

Table 5: Extracted features

The best feature subset for spots analyzer has been highlighted in the above table [Tab. 5]

Intensity histogram of G1 [Fig. 50], G2 [Fig. 49], G3 [Fig. 51], and G4 [Fig. 52] mangoes have been included below.

RGB - R	RGB - G	RGB - B	YCbCr - Cb

Table 6: Color codes for histogram

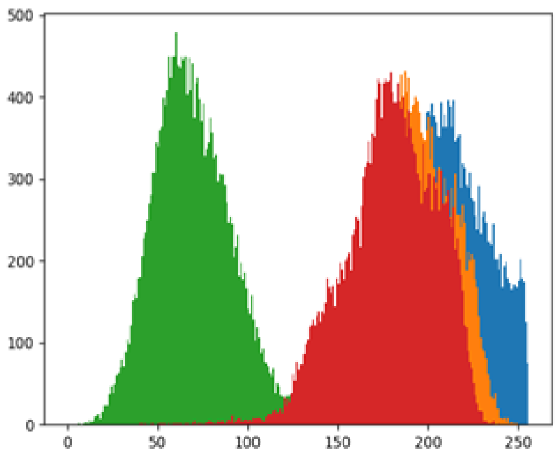


Figure 50: Intensity Histogram for Grade 1

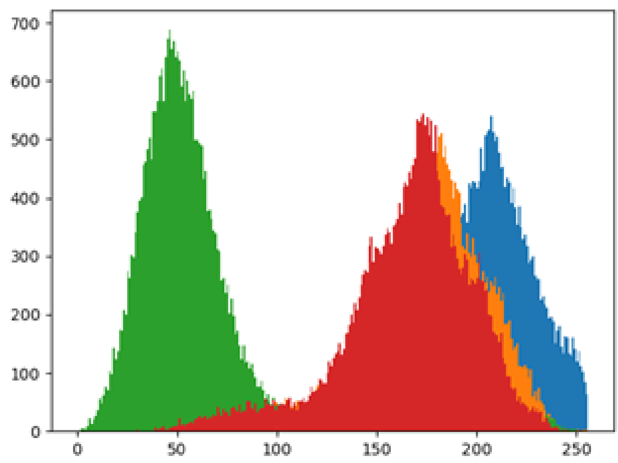


Figure 49: Intensity histogram for Grade 2

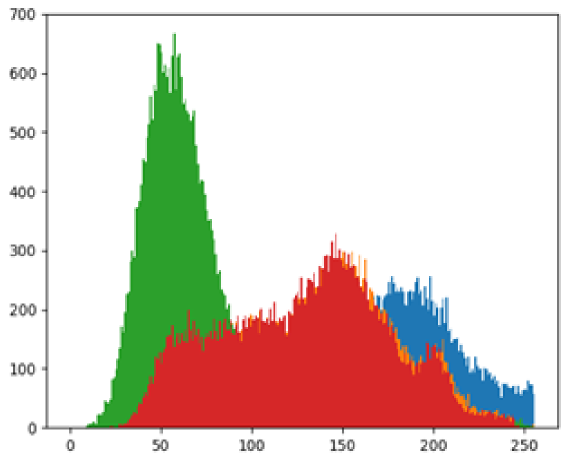


Figure 51: Intensity histogram for Grade 3

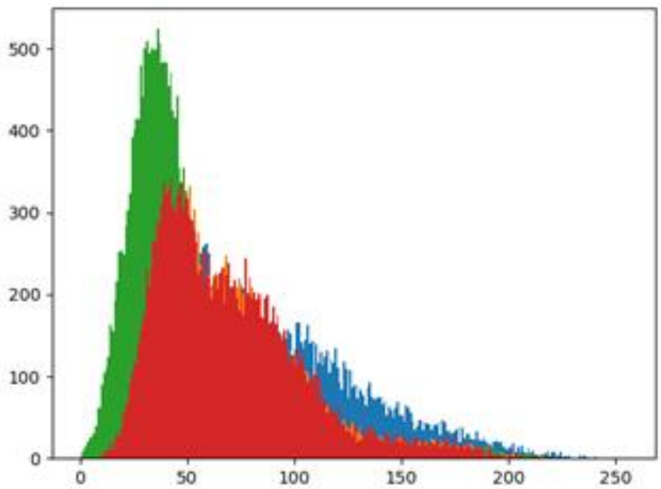


Figure 52: Intensity histogram Grade 4

Followings are the important aspects of HOG:

- HOG focuses on the structure of the object. It extracts the information of the edge's magnitude as well as the orientation of the edges.
- It uses a detection window of (64x128) pixels, so the image is first converted into (64, 128) pixels size.
- Further the image was divided into smaller parts, and then the gradient and orientation of each part was calculated. It was divided into 8x16 cells into blocks with 50% overlap, so there are going to be 7x15 = 105 blocks in total,

and each block consists of 2x2 cells with 8x8 pixels.

- We took the 64 gradient vectors of each block (8x8 pixel cell) and put them into a 9-bin histogram by considering the rangers.

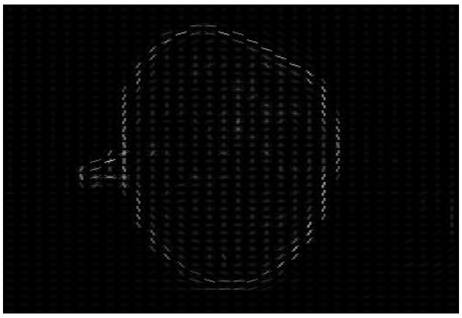
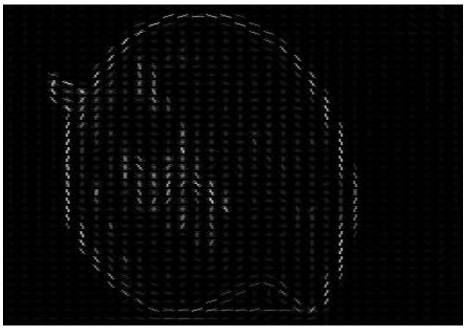
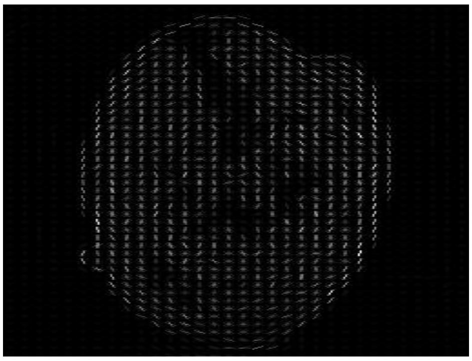
Grade	**Representation of the gradient and magnitude of the each 8x8 cell**
Grade 1	
Grade 2	
Grade 3	
Grade 4	

Table 7: HOG features

4.5.3.5.2 Surface spots analyzer for side images

- **Feed forward neural network-based implementation**

The eight statistical features which were extracted in the feature extraction part, were used as the input feature vector [Tab. 4] to the FFNN and labels were the G1,G2,G3 & G4. Relu activation function and SoftMax activation function were used inside the implemented deep learning architecture as the activation function for hidden and final layers. “Categorical cross entropy” function was used as the loss function and “adam” optimizer used as the optimization function. No dropout layers were used and all were dense layers.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 16)                144
_________________________________________________________________
dense_2 (Dense)              (None, 8)                 136
_________________________________________________________________
dense_3 (Dense)              (None, 4)                 36
=================================================================
Total params: 316
Trainable params: 316
Non-trainable params: 0
_________________________________________________________________
```

Figure 53: FFNN architecture for spots analyzing - side images

Classifier training part was done with 2000 epochs. Accuracy variation charts are mentioned below.

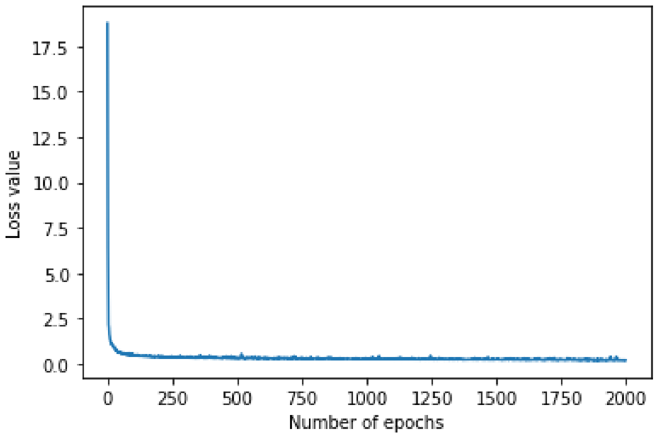


Figure 55: Trained accuracy variation

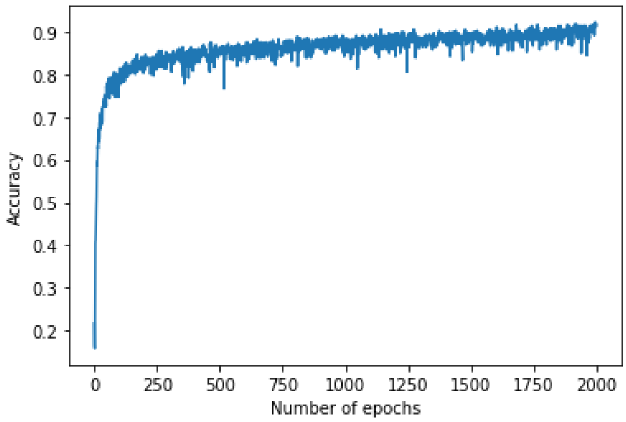


Figure 54: Loss function variation

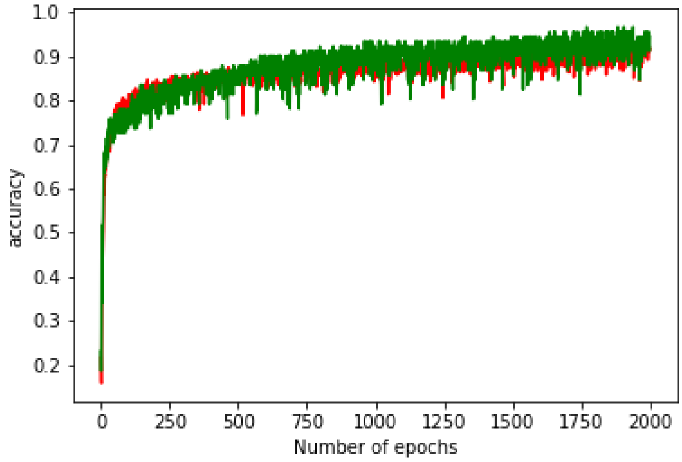


Figure 56: Validation vs Accuracy

Testing data contained 51 mangoes. Resulting accuracy was **92.15%.** Confusion matrix is shown in [Tab. 8]

		True data			
Predicted data		**G1**	**G2**	**G3**	**G4**
	G1	14	1	0	0
	G2	0	12	1	0
	G3	1	1	13	0
	G4	0	0	0	6

Table 8: Confusion matrix for FFNN

- **Convolutional neural network-based implementation**

The pre-processed 376 x 251 pixel size of RGB images were used in the training dataset and labels were G1,G2,G3 & G4. Relu activation function and softmax activation function applied in the CNN architecture. Max pooling kernel size was 2x2 pixel. Two dropout layers were used to avoid overfitting while the deep neural network was being trained.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 374, 249, 256)     7168
_________________________________________________________________
activation (Activation)      (None, 374, 249, 256)     0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 187, 124, 256)     0
_________________________________________________________________
dropout (Dropout)            (None, 187, 124, 256)     0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 185, 122, 128)     295040
_________________________________________________________________
activation_1 (Activation)    (None, 185, 122, 128)     0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 92, 61, 128)       0
_________________________________________________________________
dropout_1 (Dropout)          (None, 92, 61, 128)       0
_________________________________________________________________
flatten (Flatten)            (None, 718336)            0
_________________________________________________________________
dense (Dense)                (None, 64)                45973568
_________________________________________________________________
dense_1 (Dense)              (None, 4)                 260
=================================================================
Total params: 46,276,036
Trainable params: 46,276,036
Non-trainable params: 0
_________________________________________________________________
```

Figure 57: Confusion matrix for FFNN

Classifier training part was done with 2000 epochs. Accuracy variation charts and confusion matrix has been mentioned below.

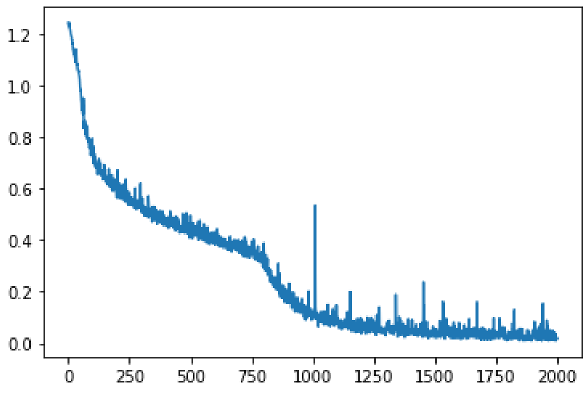


Figure 59: : Loss function variation

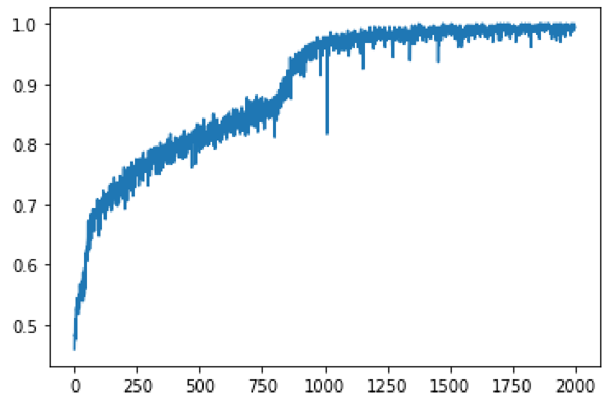


Figure 58: Trained accuracy variation

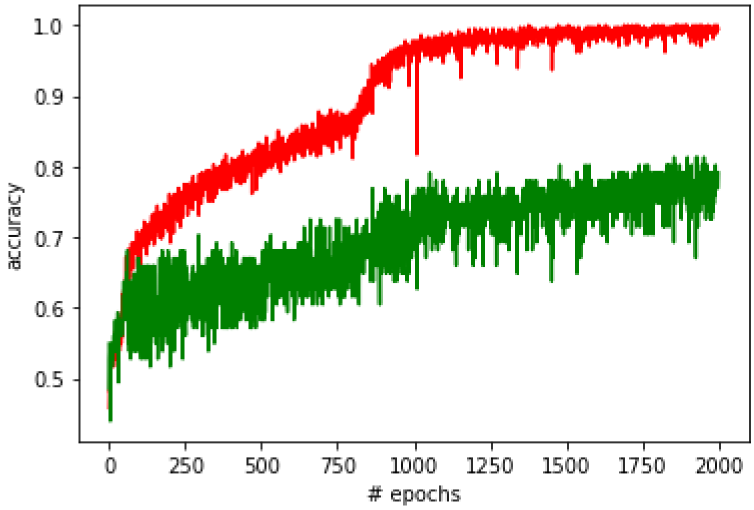


Figure 60: Validation vs accuracy

Testing was done with 51 mangoes with the **72.55%** accuracy and confusion matrix has been mentioned below [Tab. 9].

		True data			
		G1	**G2**	**G3**	**G4**
Predicted data	**G1**	12	1	0	0
	G2	2	10	2	0
	G3	2	3	10	1
	G4	0	0	3	5

Table 9: Confusion matrix for CNN

- **Support Vector Machine Based Implementation**

The eight statistical features which were extracted in the feature extraction part, were used as the input feature vector [Tab. 5] to the SVM and labels were the G1,G2,G3 & G4. Testing was done with 51 mangoes with the **60.78%** accuracy and confusion matrix has been mentioned below [Tab. 10].

	True data				
Predicted data		**G1**	**G2**	**G3**	**G4**
	G1	10	0	7	0
	G2	0	3	12	0
	G3	0	1	12	0
	G4	0	0	0	6

Table 10: Confusion matrix for SVM

- **K-Nearest Neighbor Based Implementation**

The eight statistical features which were extracted in the feature extraction part, were used as the input feature vector [Tab. 5] to the KNN and labels were the G1,G2,G3 & G4. Testing was done with 51 mangoes with the **90.19%** accuracy and confusion matrix has been mentioned below [Tab. 11].

	True data				
Predicted data		**G1**	**G2**	**G3**	**G4**
	G1	15	1	0	0
	G2	1	10	0	0
	G3	0	1	12	2
	G4	0	0	0	9

Table 11: Confusion matrix for KNN

4.5.3.5.3 Surface Spots Analyzer for Top Side of Images

- **Feed Forward Neural Network Based Implementation**

The eight statistical features which were extracted in the feature extraction part, were used as the input feature vector [Tab. 5] to the FFNN and labels were the G1,G2,G3 & G4. Relu activation function and softmax activation function were used inside the implemented deep learning architecture as the activation function for hidden and final layers. "Categorical cross entropy" function was used as the loss function and "adam" optimizer used as the optimization function. No dropout layers were used and all were dense layers.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 16)                144
_________________________________________________________________
dense_2 (Dense)              (None, 8)                 136
_________________________________________________________________
dense_3 (Dense)              (None, 4)                 36
=================================================================
Total params: 316
Trainable params: 316
Non-trainable params: 0
_________________________________________________________________
```

Figure 61: FFNN architecture for spots analyzing - Top side

Classifier training part was done with 500 epochs. Accuracy variation charts are mentioned below.

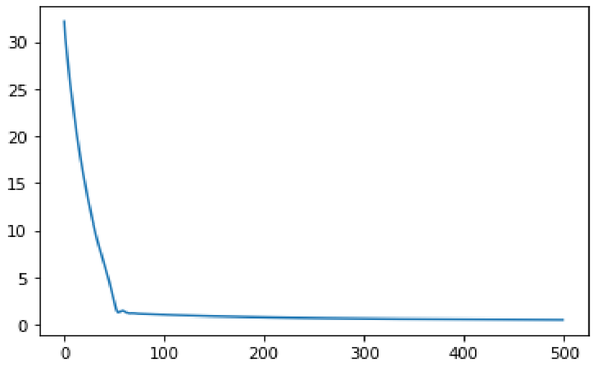


Figure 62: Loss function variation

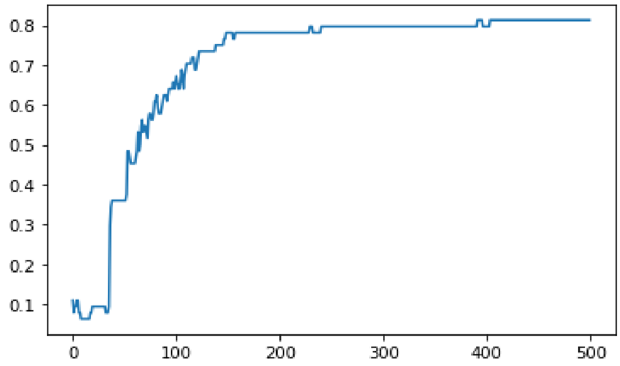


Figure 63: Trained accuracy variation

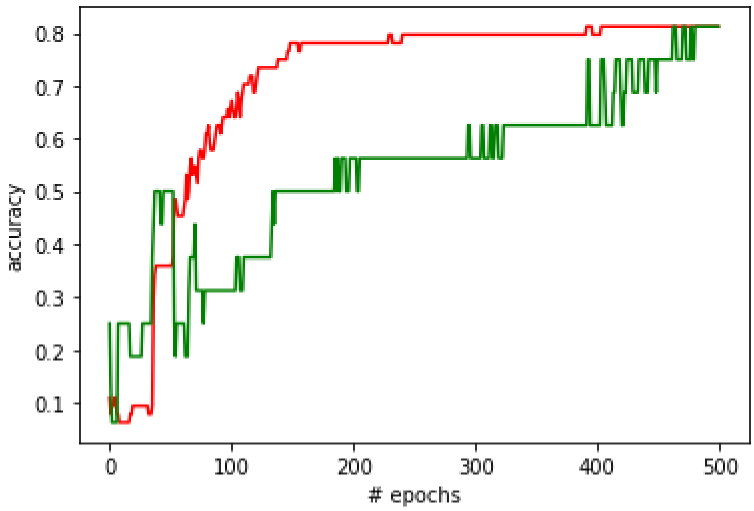


Figure 64: Validation vs accuracy

Testing was done with 9 mangoes with the **77.77%** accuracy and confusion matrix has been mentioned below [Tab. 12].

		True data			
Predicted data		**G1**	**G2**	**G3**	**G4**
	G1	3	0	0	0
	G2	1	1	0	0
	G3	0	0	3	0
	G4	0	0	1	0

Table 12: Confusion matrix for FFNN(for top images)

- **Support Vector Machine Based Implementation**

The eight statistical features which were extracted in the feature extraction part, were used as the input feature vector [Tab. 5] to the SVM and labels were the G1,G2,G3 & G4. Testing was done with 9 mangoes with the **44.44%** accuracy and confusion matrix has been mentioned below [Tab. 13].

	True data				
Predicted data		**G1**	**G2**	**G3**	**G4**
	G1	3	0	0	0
	G2	2	0	0	0
	G3	2	0	1	0
	G4	1	0	0	0

Table 13: Confusion matrix for SVM (for top images)

- **K-Nearest Neighbor Based Implementation**

The eight statistical features which were extracted in the feature extraction part, were used as the input feature vector [Tab. 5] to the KNN and labels were the G1,G2,G3 & G4. Testing was done with 10 mangoes with the **90%** accuracy and confusion matrix has been mentioned below [Tab. 14].

	True data				
Predicted data		G1	G2	G3	G4
	G1	4	1	0	0
	G2	0	1	0	0
	G3	0	0	2	0
	G4	0	0	0	2

Table 14: Confusion matrix for KNN (for top images)

4.5.3.5.4 Surface Spots Analyzer for Bottom Side of Images

- **Feed Forward Neural Network Based Implementation**

The eight statistical features which were extracted in the feature extraction part, were used as the input feature vector [Tab. 5] to the FFNN and labels were the G1,G2,G3 & G4. Relu activation function and softmax activation function were used inside the implemented deep learning architecture as the activation function for hidden and final layers. "Categorical cross entropy" function was used as the loss function and "adam" optimizer used as the optimization function. No dropout layers were used and all were dense layers.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 16)                144
_________________________________________________________________
dense_2 (Dense)              (None, 8)                 136
_________________________________________________________________
dense_3 (Dense)              (None, 4)                 36
=================================================================
Total params: 316
Trainable params: 316
Non-trainable params: 0
_________________________________________________________________
```

Figure 65: FFNN architecture for spots analyzing - Bottom side

Classifier training part was done with 1000 epochs. Accuracy variation charts are mentioned below.

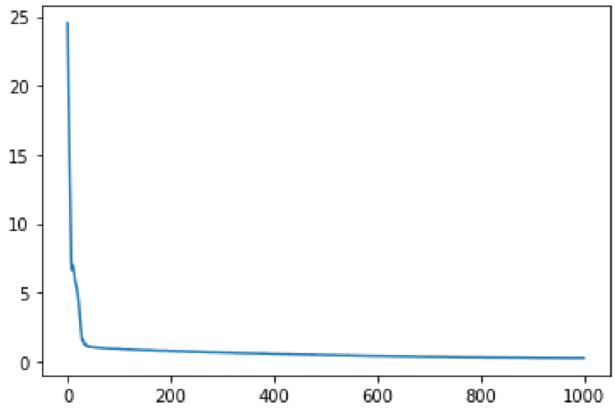


Figure 66: Loss function variation

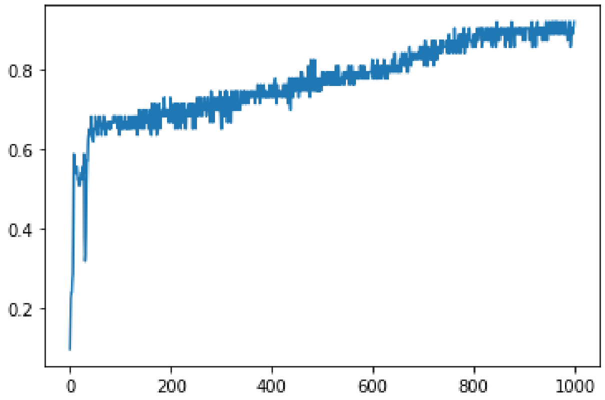


Figure 67: Trained accuracy variation

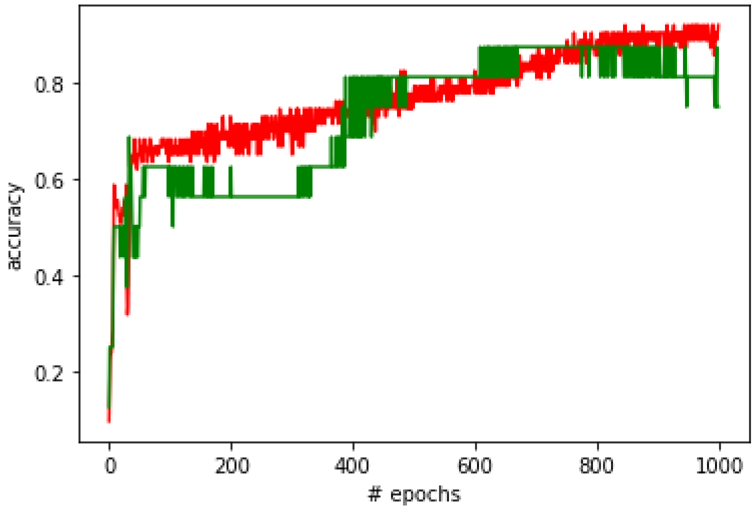


Figure 68: Validation vs accuracy

Testing was done with 10 mangoes with the **90%** accuracy and confusion matrix has been mentioned below [Tab. 15].

	True data				
Predicted data		**G1**	**G2**	**G3**	**G4**
	G1	4	0	0	0
	G2	0	1	0	0
	G3	0	0	2	1
	G4	0	0	0	2

Table 15: Confusion matrix for FFNN(for bottom images)

- **Support vector machine-based implementation**

The eight statistical features which were extracted in the feature extraction part, were used as the input feature vector [Tab. 5] to the SVM and labels were the G1,G2,G3 & G4. Testing was done with 9 mangoes with the **55.55%** accuracy and confusion matrix has been mentioned below [Tab. 16].

	True data				
Predicted data		**G1**	**G2**	**G3**	**G4**
	G1	5	0	0	0
	G2	1	0	0	0
	G3	2	0	0	0
	G4	1	0	0	0

Table 16: Confusion matrix for SVM(for bottom images)

- **K-Nearest neighbor-based implementation**

The eight statistical features which were extracted in the feature extraction part, were used as the input feature vector [Tab. 5] to the KNN and labels were the G1,G2,G3 & G4. Testing was done with 9 mangoes with the **66.67%** accuracy and confusion matrix has been mentioned below [Tab. 17].

		True data			
Predicted data		**G1**	**G2**	**G3**	**G4**
	G1	4	0	2	0
	G2	0	0	1	0
	G3	0	0	1	0
	G4	0	0	0	1

Table 17: Confusion matrix for KNN (for bottom images)

4.5.3.6 Mongo Size Analyzer

4.5.3.6.1 Feature Extraction and Selection

The relevant preprocessing parts which were used to detect the bounding rectangle, have been explained in detail under the above preprocessing explanation. According to the problem definition, the mangoes which were classified as "grade one" or "grade two" mango by spot analyzer, are further categorized by considering their size into the three group such as "grade one", "grade two small" and "grade two large". Therefore, height and width parameters were extracted from detected bounding rectangles as the features. We implemented the size with different deep learning and machine learning algorithms such as FFNN, SVM and KNN.

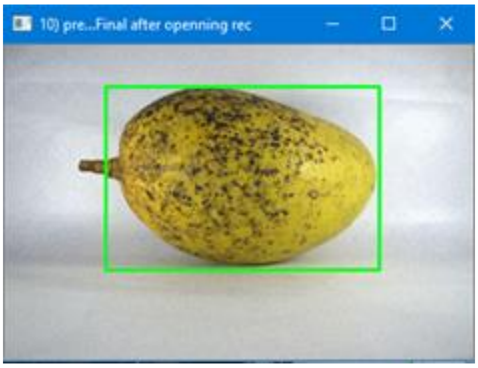


Figure 69: Detected bounding rectangle

4.5.3.6.2 Mongo Size Analyzer Implementation

- **Feed Forward Neural Network Based Implementation**

The feature vector consisted of two values(height & width) and final output is binary output(small or large). Relu activation function and softmax activation function were used inside the implemented deep learning architecture as the activation function for hidden and final layers. “Categorical cross entropy” function was used as the loss function and “adam” optimizer used as the optimization function. No dropout layers were used and all were dense layers.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_4 (Dense)              (None, 8)                 24
_________________________________________________________________
dense_5 (Dense)              (None, 4)                 36
_________________________________________________________________
dense_6 (Dense)              (None, 2)                 10
=================================================================
Total params: 70
Trainable params: 70
Non-trainable params: 0
_________________________________________________________________
```

Figure 70: FFNN architecture for size analyzing

Classifier training part was done with 2000 epochs. Accuracy variation charts are mentioned below.

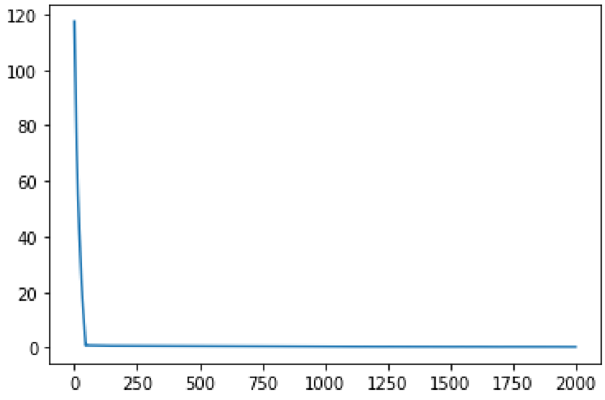


Figure 72: Loss function variation

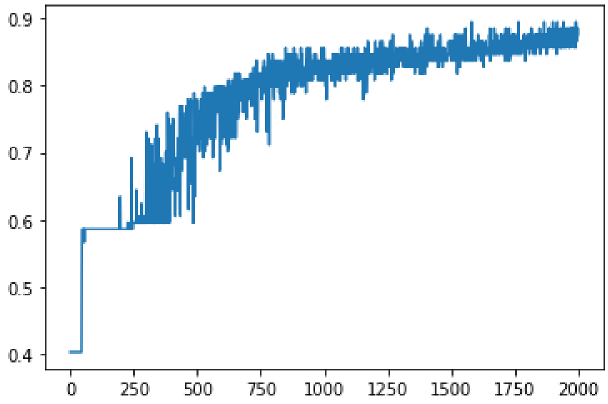


Figure 71: Trained accuracy variation

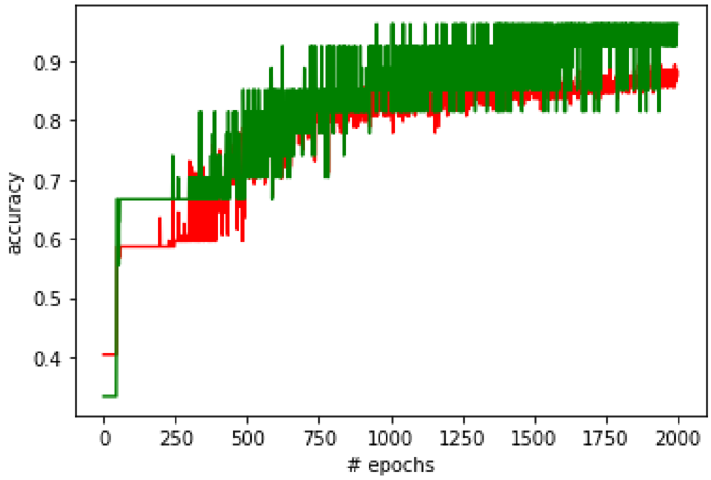


Figure 73: Validation vs accuracy

Testing was done with 15 mangoes with the **100%** accuracy and confusion matrix has been mentioned below [Tab. 18].

		True data	
Predicted data		**Small**	**Large**
	Small	11	0
	large	0	4

Table 18: Confusion matrix for FFNN(Spots analyzer)

- **Support Vector Machine Based Implementation**

The two features(height & width) which were extracted in the feature extraction part, were used as the input feature vector to the SVM and labels were the small and large. Testing was done with 15 mangoes with the **86%** accuracy and confusion matrix has been mentioned below [Tab. 19].

		True data	
Predicted data		**Small**	**Large**
	Small	12	0
	large	2	1

Table 19: Confusion matrix for SVM(Spots analyzer)

- **K-Nearest Neighbor Based Implementation**

The two features(height & width) which were extracted in the feature extraction part, were used as the input feature vector to the KNN and labels were the small and large. Testing was done with 15 mangoes with the **100%** accuracy and confusion matrix has been mentioned below [Tab. 20].

		True data	
Predicted data		**Small**	**Large**
	Small	4	0
	large	0	11

Table 20: Confusion matrix for KNN(Spots analyzer)

4.5.3.7 Integration of Classifiers for Final Grade

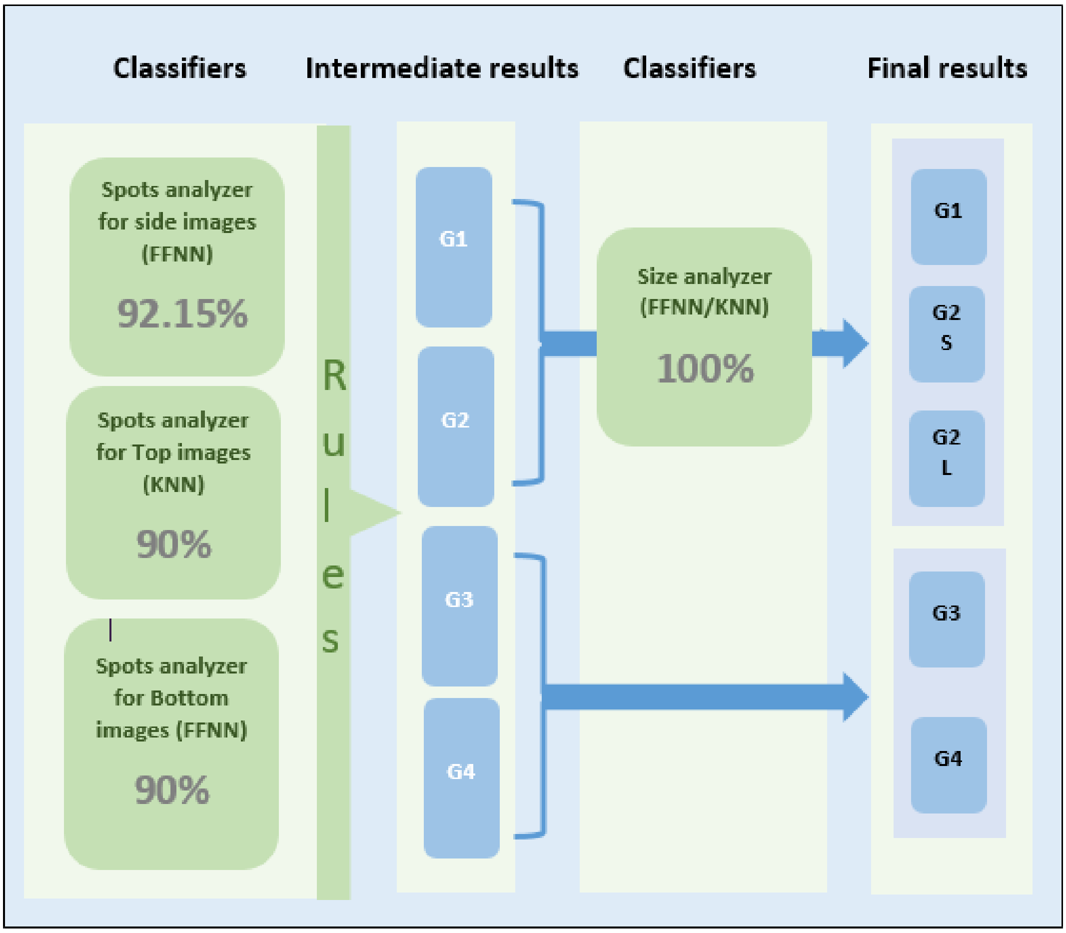


Figure 74: Final classifiers

By considering the classification accuracy of our implemented machine learning and deep learning classification algorithms, four algorithms were selected for each classification sub component. The final system consists FFNN based classifier with 92.15% accuracy for spots analyzing the side images, KNN based classifier with 90% accuracy for spots analyzing the top side images, FFNN based classifier with 90% accuracy for spots analyzing the bottom side images and FFNN/KNN based classifier with 100% accuracy for the sized analyzing classifier. Finally, to combine all of the classification algorithms, a rule-based classifier combination algorithm was implemented with the help of domain expertise.

Chapter 5 - Results and Analysis

5.1 Accuracy evaluation of classification approaches

This research was done by considering four classification algorithms FFNN, SVM, KNN, and CNN to classify six sides of each mango. The classifier of top-side images was implemented with an accuracy of 77.77%, 44.44%, 90%, and 40% for the above algorithms respectively. The bottom side images classifier was implemented for the above algorithms with an accuracy of 90%, 55.55%, 66.66%, 45% respectively. For the other four sides 93%, 64%, 91% and 72% accuracies were given by FFNN, SVM, KNN, and CNN algorithms respectively. The mango size analyzing classifier was implemented with three classification algorithms FFNN, SVM, KNN with the accuracy of 100%, 86%, 100% respectively. In addition to the main classifiers, a rule-based classifier combination algorithm was implemented to determine the final result.

Classification Algorithm	FFNN	CNN	KNN	SVM
Spots analyzer - side images	**93%**	72%	**91%**	64%
Spots analyzer - top images	77.77%	40%	**90%**	44.44%
Spots analyzer - bottom images	**90%**	45%	66.66%	55.55%
Size analyzer	**100%**	-	**100%**	86%

Table 21: Accuracy table for classifiers

Then the algorithms which gave higher accuracy were selected. In [Fig :64] has been mentioned those details. Overall accuracy of the entire grading process is **93.25%**

5.2 Execution Time Evaluation (Serial vs Concurrent execution)

The efficiency depends mainly on the prediction model. Prediction is the main bottleneck in the system, and multithreading was implemented to improve efficiency and expected to execute the spots analyzing models parallelly with six side images. Flowing results for dual core system. For getting real performance improvement of the classification model should be deployed in the machine which has more core.

```
127.0.0.1 - - [19/Mar/2021 07:10:29] "←[37mGET /resultserial HTTP/1.1←[0m" 200 -
calling spots_Analyzer => image name =>
['1.jpg']
S1 > Grade 1
calling spots_Analyzer => image name =>
['2.jpg']
S2 > Grade 1
calling spots_Analyzer => image name =>
['3.jpg']
S3 > Grade 1
calling spots_Analyzer => image name =>
['4.jpg']
S4 > Grade 1
Apply Size Analyzer
Final Grade => Grade 1
127.0.0.1 - - [19/Mar/2021 07:10:42] "←[37mGET /resultserial HTTP/1.1←[0m" 200 -
```

Figure 75: System Log for Serial execution

```
127.0.0.1 - - [19/Mar/2021 07:08:41] "←[37mGET /result HTTP/1.1←[0m" 200 -
calling spots_Analyzer => image name =>
calling spots_Analyzer => image name =>
['1.jpg']
['2.jpg']
calling spots_Analyzer => image name =>
['3.jpg']
calling spots_Analyzer => image name =>
['4.jpg']
S2 > Grade 1
S1 > Grade 1
S3 > Grade 1
S4 > Grade 1
Apply Size Analyzer
Final Grade => Grade 1
127.0.0.1 - - [19/Mar/2021 07:09:01] "←[37mGET /result HTTP/1.1←[0m" 200 -
```

Figure 76: System Log for concurrent execution

Multithreaded architecture [Fig:67] and execution times [Tab: 21] have been mentioned below. The six threads were implemented for spots analyzers of each side separately and size analyzer has to wait until spots analyzing thread are completed. By considering component wise, the Spots analyzer, size analyzer and final grader evaluator are executing serially.

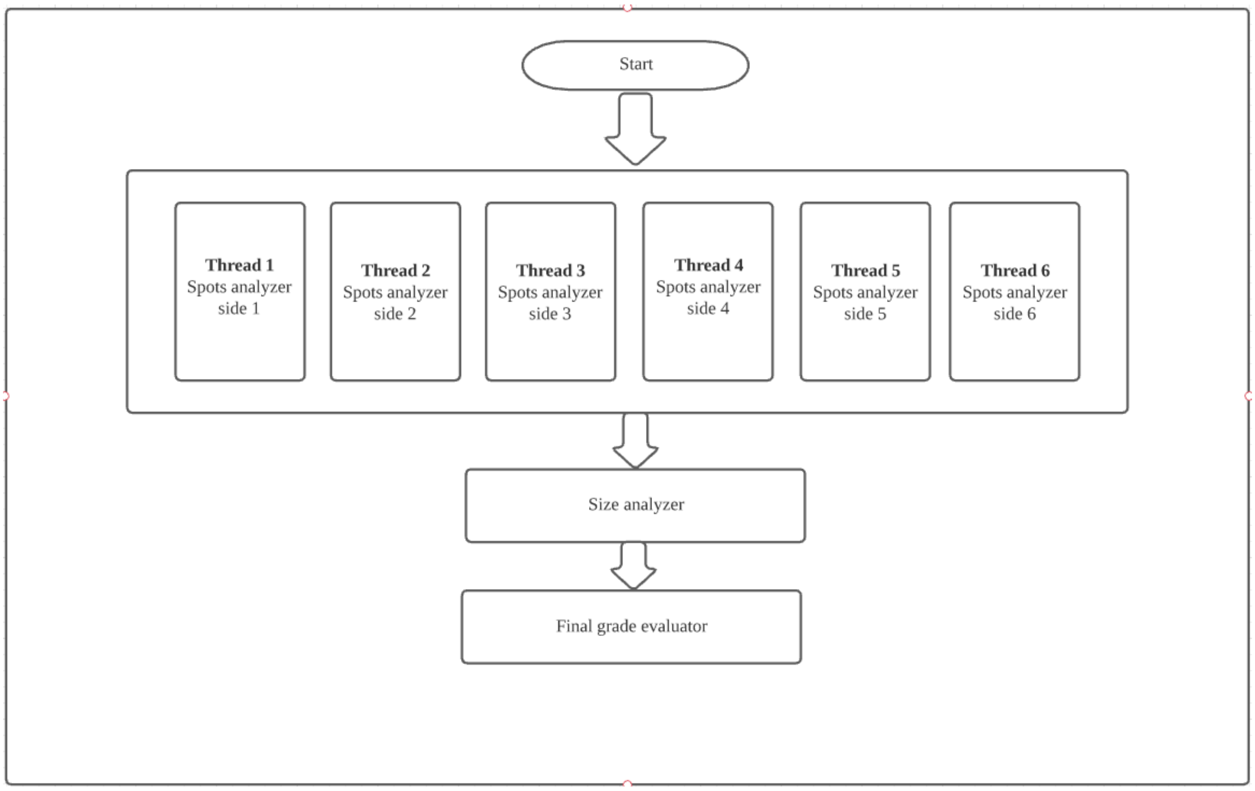


Table 22: Multithreaded implementation

Execution method	Execution time
Serial execution	13 seconds
Concurrent execution using threads	09 seconds

Table 23: Execution time

5.3 CPU Utilization Evaluation (Serial vs Concurrent execution using threads)

Diagrams which relate to CPU utilization when the system is executing serially and parallel, have been included below [Fig: 77][Fig: 79].

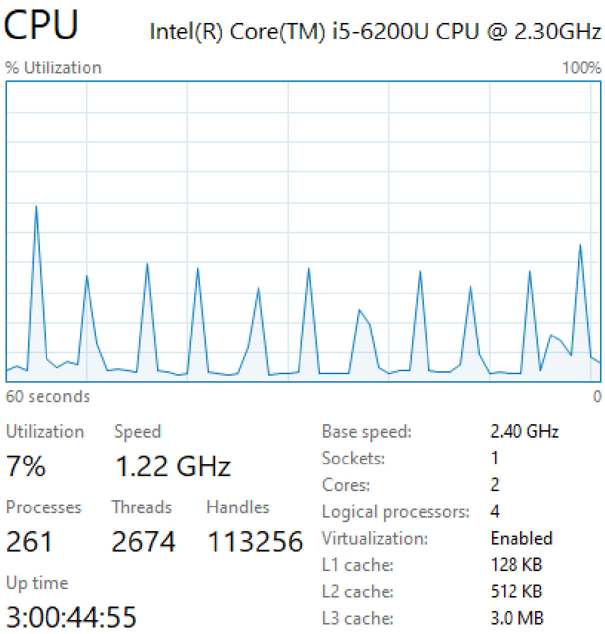


Figure 78: CPU utilization in idle state

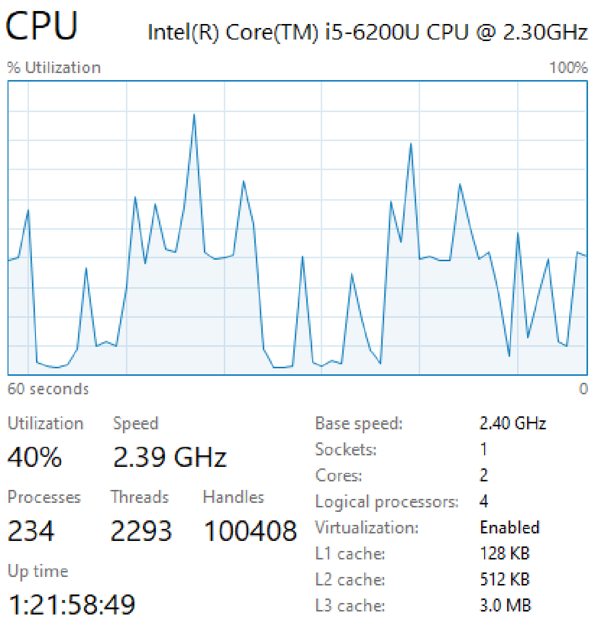


Figure 77: CPU utilization for Serial execution

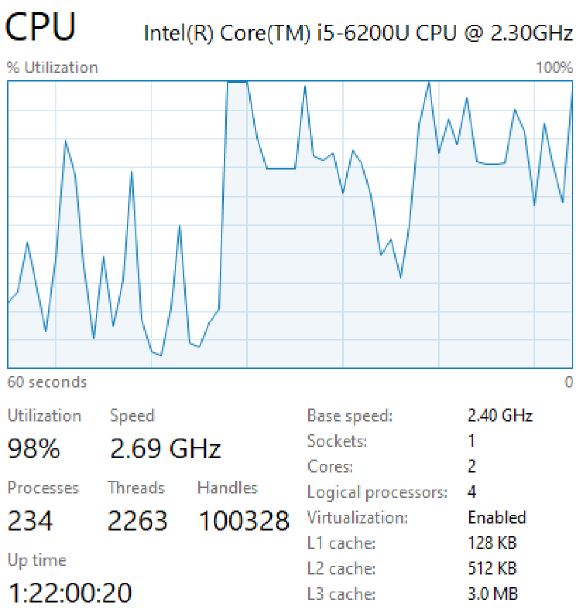


Figure 79: CPU utilization for concurrent execution

Chapter 6 - Testing and Evaluation

6.1. Test Approach

6.1.2 Unit testing

Individual items were tested separately and identified issues among them. Necessary changes were made before doing the integration.

Individual test items

1. Test the correct process ID is shown in the user interface.
2. Test the created process ID is successfully written in the database.
3. Test the shown grade is correct with respect to the loaded images of a mango.
4. Test the bar chart gives the correct mango count of the current process.
5. Test the pie chart gives the correct mango count of all processes in each grade.
6. Test the accordion’s data matches with the bar chart and pie chart.

6.1.3 Integration testing

Each tested component which had been refined in the unit tests were integrated one by one and tested the dependencies between each component so that integrated components will function properly. After training the model with a considerable accuracy, the final grade classifier component was integrated and tested to verify that the system still classifies the image with the same accuracy and the generated grade is correct. Integration of user interface with the classification model was tested to determine responsiveness of user interfaces and the correctness of the output.

6.1.4 System testing

After integrating all the system components and user interfaces, system testing was performed and checked whether the system fulfills the functional requirements intended by the requirements.

6.1.5 Regression testing

Performed regression testing to tests the issues in the system that will cause due to bug fixes and changes such as user interface changes, and new trained data in order to ensure that the system will not be prone to bugs due to those changes.

6.1.6 Beta testing

Acceptance testing was performed with our client by giving him a full demonstration of the approach the project took, and his feedback was taken into account to improve the accuracy of TJC mango grading.

6.1.7 Performance testing

Performance needs not to be tested thoroughly as this project is carried out as a proof of concept and the final output will be a prototype with fewer machinery resources and power. But in order to prove that this technique can be used in this scenario with a good performance level a performance test is needed to perform.

Preprocessed images are tested because incorrectly preprocessed images may lead to training the classification model wrongly. Therefore, the implementation under preprocessing and the preprocessed images used to train the model were tested separately. Tests in the training-model can be examined by using the trained model for prediction. The first phase of the model testing was carried out for the FFNN and KNN. The operations and sending images from cameras to the software system need to be tested in a specific manner after developing the hardware prototype.

Final grades of the mangoes which were input to the system need to be visualized in the dashboard with relevant details with high responsiveness and it should be tested with the aspects of UI.

6.2. Test Cases

Test Case ID	01
Test title	Verify functionality of creating a new process
Test steps	1. Navigate into the process creation page 2. Click on the ‘Create new process’ button 3. A pop will be displayed, click on the ‘ok’ button
Expected result	Once the user clicks on the ‘create new process’ button, a pop up should be displayed with the process id and the time stamp. Once the ‘ok’ button is clicked, a toast message should be displayed as ‘Success! Process created successfully’
Actual result	System successfully creates a new process with the process id and time stamp

Test Case ID	02
Test title	Verify functionality of not creating a new process

Test steps	1. Navigate into the process creation page 2. Click on the ‘Create new process’ button 3. A pop will be displayed, click on the ‘no’ button
Expected result	Once the user clicks on the ‘create new process’ button, a pop up should be displayed with the process id and the time stamp. Once the ‘no’ button is clicked, a toast message should be displayed as ‘Failed! Process is not created’
Actual result	System doesn’t create a new process. The process id and the time stamp will be the most recently successfully created process.

Test Case ID	03
Test title	Verify start functionality
Test steps	1. Click on the ‘start’ button

Expected result	Once the user clicks on the ‘start’ button, 6 images of a mango should be displayed. A toast message should show the relevant grade of the mango And also, the bar chart as well as pie chart should be displayed
Actual result	System successfully shows 6 images of a mango and its correct grade as well as will display the bar chart and the pie chart.

Test Case ID	04
Test title	Verify repetition of the start functionality
Test steps	1. Click on the ‘start’ button repetitively
Expected result	Once the user clicks on the ‘start’ button repetitively, the bar chart and the pie chart should be updated.
Actual result	System successfully updates the bar chart and the pie chart when clicking on the start button repetitively.

6.3. Testing Tools

6.3.1 Backend API testing - Postman

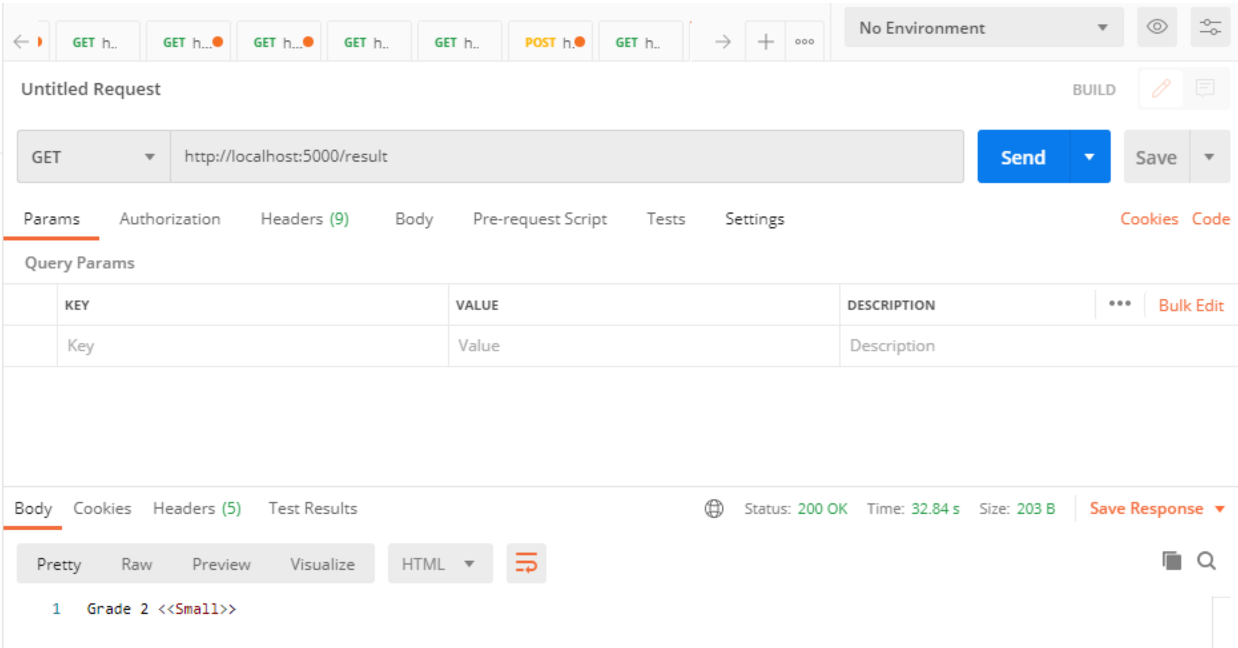


Figure 80: Back-end Api testing 1

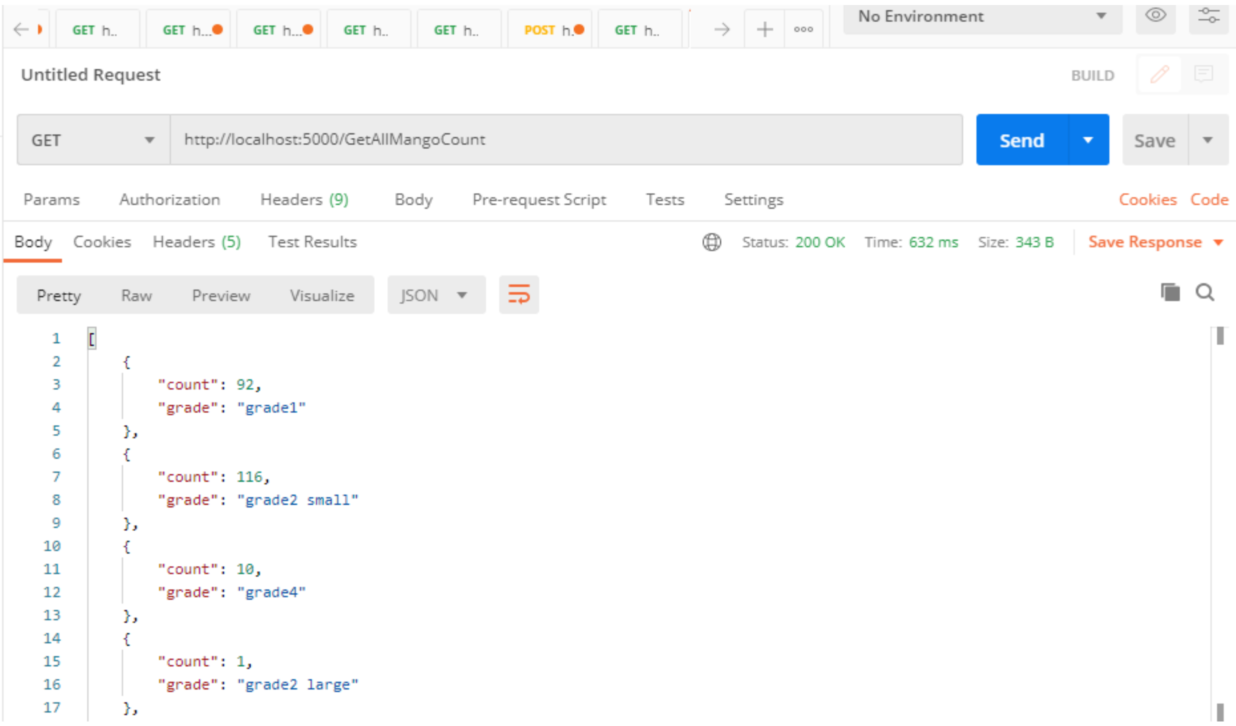


Figure 81: Back-end Api testing 2

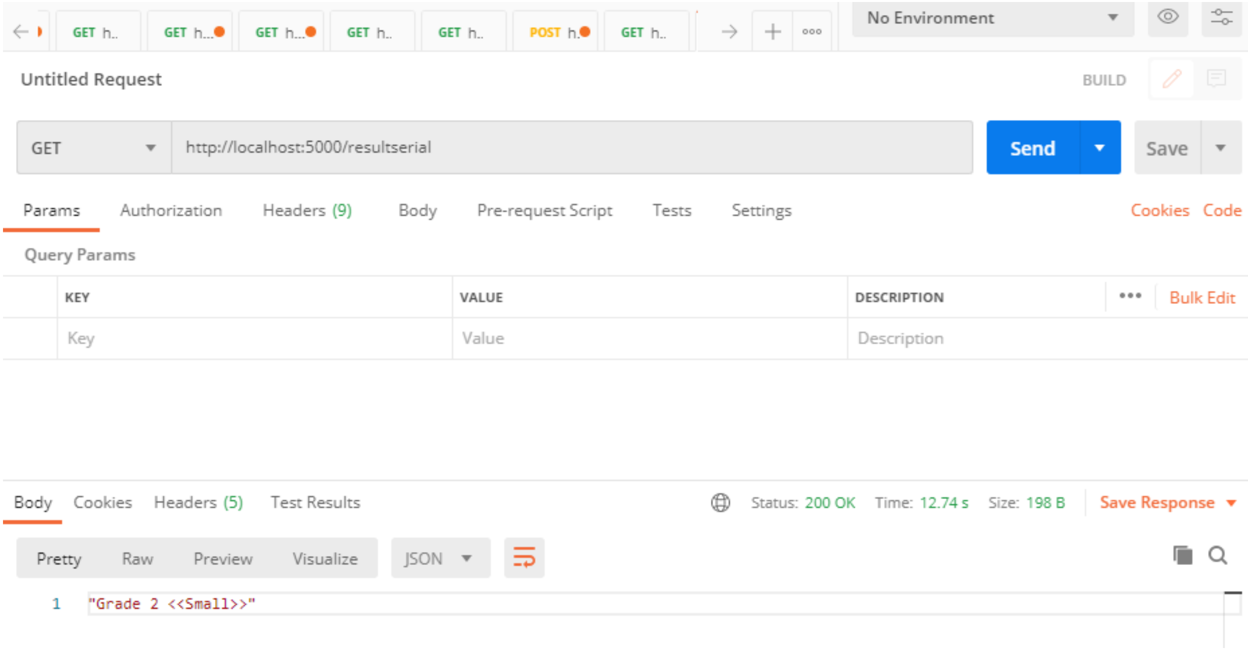


Figure 82: Back-end Api testing 3

6.3.2 Front-end automated testing - Karma

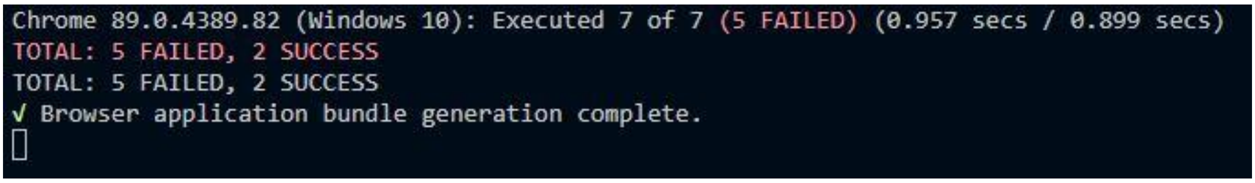


Figure 83: Front-end automated testing(phase 1)

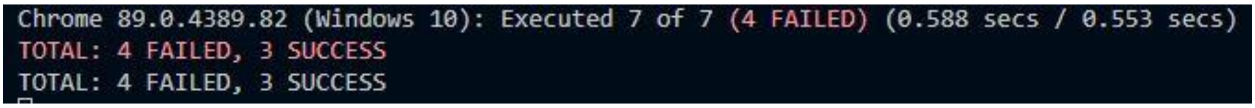


Figure 84: Front-end automated testing(phase 2)

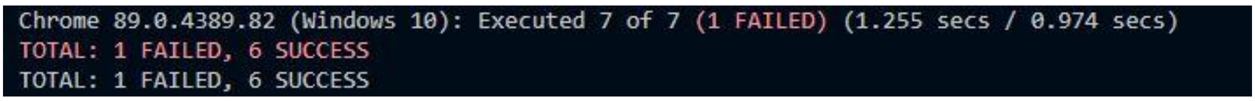


Figure 85: Front-end automated testing(phase 3)

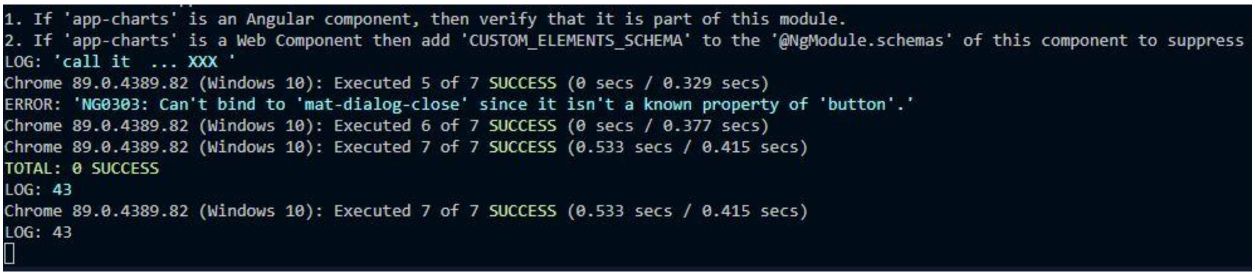


Figure 86: Front-end automated testing(phase 4)

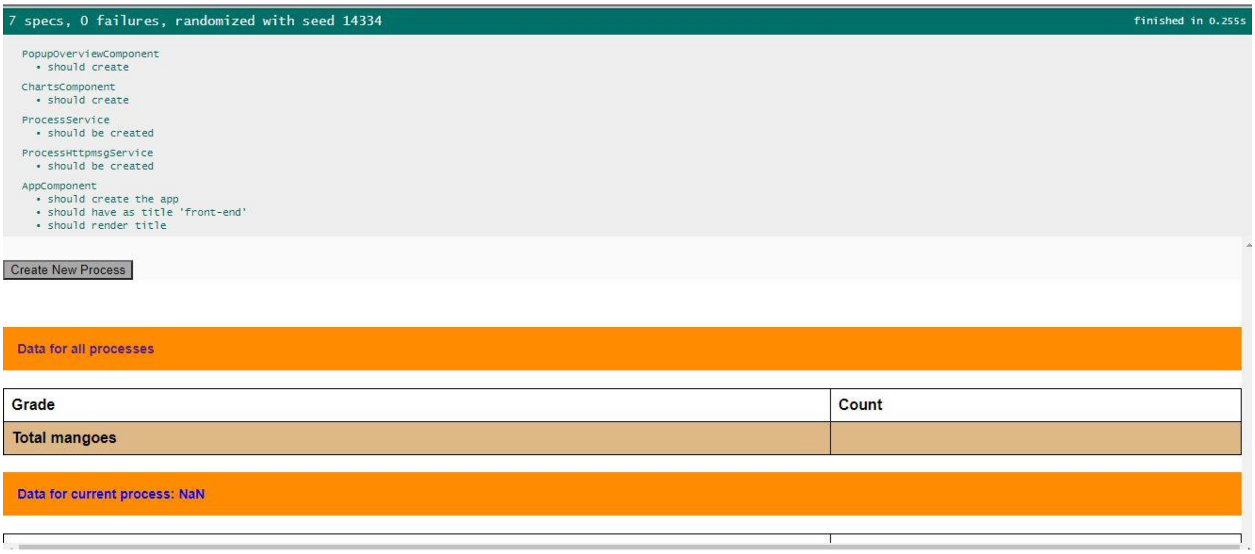


Figure 87: Angular Karma testing output 1

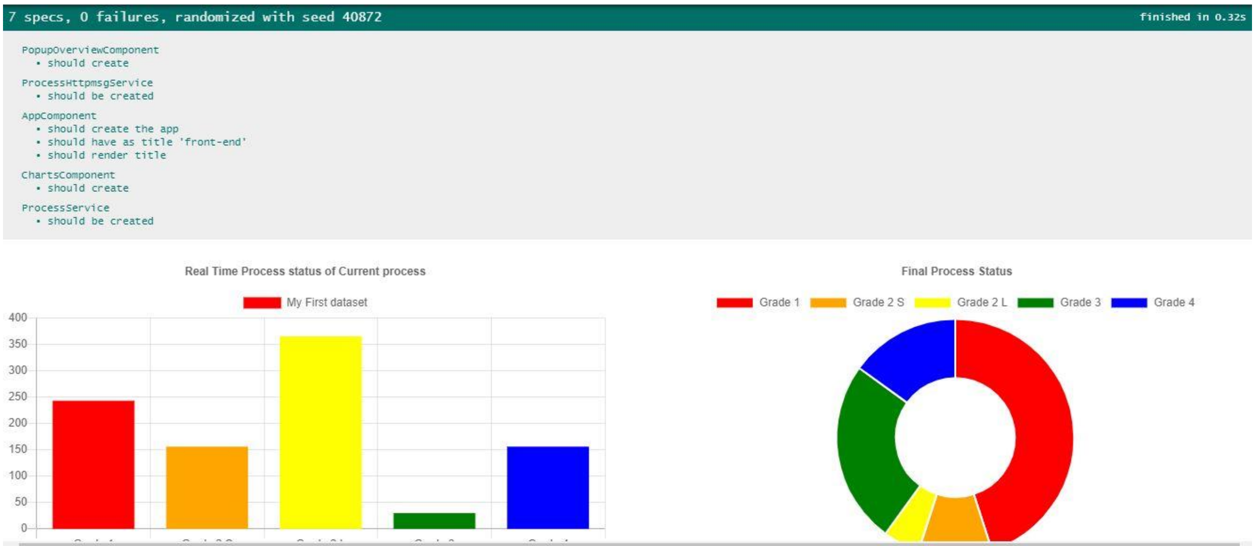


Figure 88: Angular Karma testing output 2

Chapter 7 - Conclusion

This project was undertaken to generate the final grade of a TJC mango by considering surface spots, size and weight to overcome the inconsistency and accuracy issues in current manual and machine-based approaches. The model consists of 4 sub models separately to classify mangoes based on spots in top-side, bottom-side, other 4 sides and based on size. According to the accuracies given by SVM, FFNN, KNN, CNN which were tried out in the research, FFNN was taken as the best classifier to analyze spots in side images, bottom images and to analyze size of mangoes as it gives the best accuracy and performance. KNN was taken as the best classifier to classify mangoes based on spots in top side images of mangoes. Hence FNN and KNN classifiers were selected as the best classifiers over SVM, CNN in this machine learning based model which gives 93% average accuracy.

7.1 Limitations

In this study, it doesn't take into consideration the internal damages that exist in mangoes when classifying. In a warehouse where different mango varieties are being graded, this system can’t be used. Another limitation in our research is, when classifying mangoes, the classification models don’t consider the surface color of a mango. Therefore, it’s not capable of determining the maturity of mangoes.

7.2 Future Works

7.2.1 Suggested system design

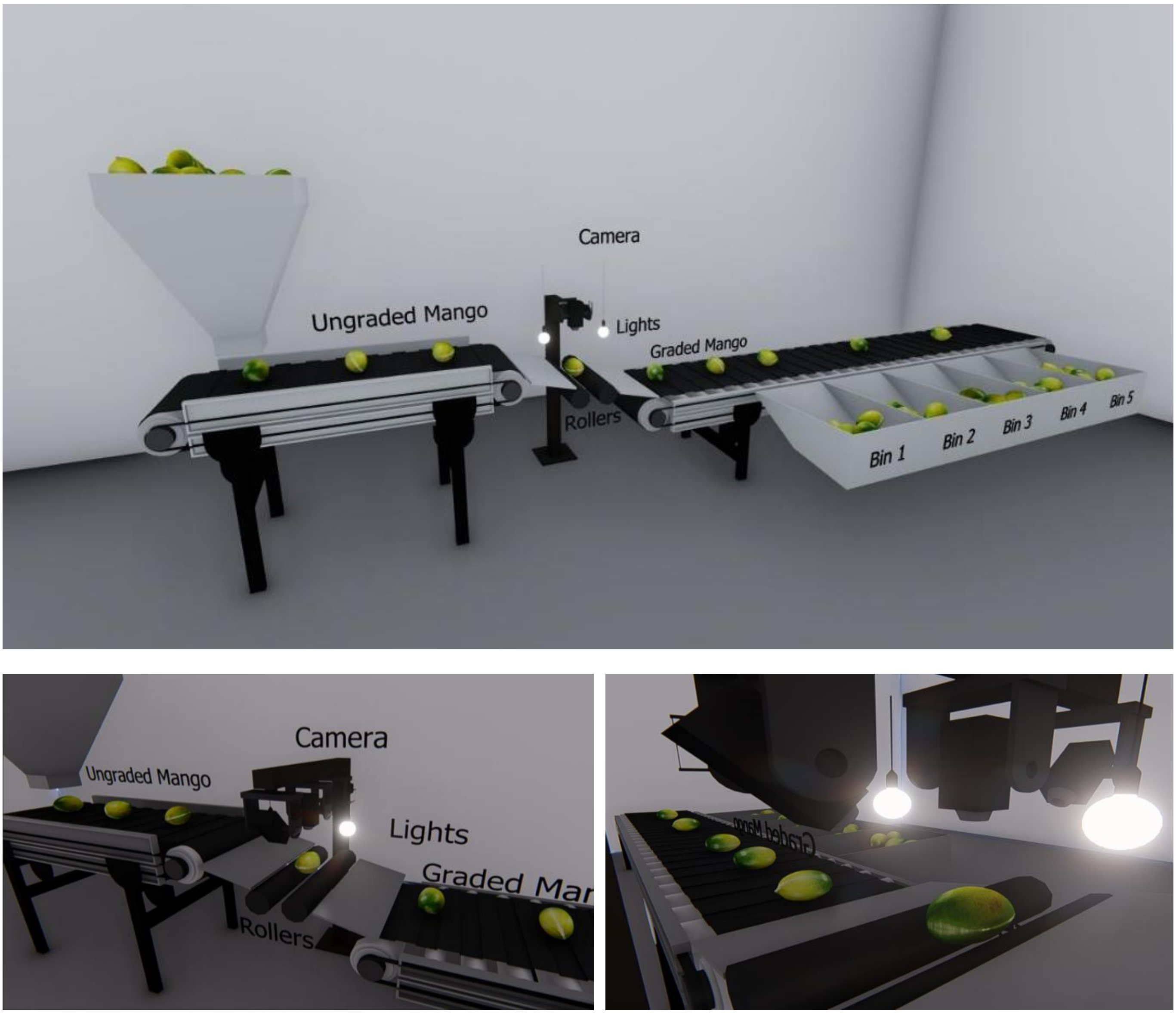


Figure 89: Suggested system design

The above image [Fig. 89] depicts the design of the suggested system. As a future work, a machine designed as in the image [Fig. 89] can be integrated to the classifier which will be introduced at the end of this project.

In this process ungraded mangoes are moved one by one through a low-speed conveyor belt and rolled on to two rollers. These two rollers are rotating in 90^0 at a time as well as they are made with a high coefficient of friction . There are three cameras fixed from three sides to capture images. One camera will take images from four sides at 90^0. Other two cameras take images from the top and bottom sides of the mango. Using these images, the classifier will classify each mango into the correct grade. Then the mango will pass through the next conveyor belt. It will have a sensor to put mango into the correct bin.

7.2.1.1 Suggested system design assumptions

- In the mango grading process, mangoes rotate uniformly 360^0 along with the roller without tilting and slipping.
- Captured 4 sides were taken by rotating exactly 90^0 . Therefore, each mango covered 360^0 view.
- Captured 6 sides for every mango in the dataset are captured in the same angle.
- Mango doesn’t rotate on the z-axis and y-axis when capturing images. Therefore, the size of the bounded contour will not be affected.
- The distance between the camera and the object and the lighting conditions we considered when capturing images, is the same as in the real system as well.

Chapter 8 - References

1. *Central Bank of Sri Lanka* 2019." Annual Report 2017" [Online]. Available: https://www.cbsl.gov.lk/en/publications/economic-and-financial-reports/annual-reports/annual-report-2017.[Accessed: 08 May-2020].
2. *PriceWaterhouseCoopers* – Sri Lanka. "Technical Assistance Consultant's Report Sri Lanka: Small and Medium-Sized Enterprises Line of Credit Project" [Online]. Available: https://www.adb.org/sites/default/files/project-documents/49273/49273001-tacr-en_5.pdf.[Accessed: 08-May-2020].
3. "Cbi market survey". The EU market for mango (2009) [Online]. Available: http://www.siicex.gob.pe/siicex/resources/estudio/887993922rad726FD.pdf [Accessed: 08-May-2020].
4. *Department of Agriculture*. "Mango cultivation" [Online]. Available: https://doa.gov.lk/ICC/images/publication/Books/Books/Mango%20BOOK.pdf [Accessed: 08-May-2020]
5. *K.H.S.Peiris*. "The Mango in the Democratic Socialist Republic of Sri Lanka" [Online]. Available: https://www.researchgate.net/publication/316881889_The_Mango_in_the_Democratic_Socialist_Republic_of_Sri_Lanka[Accessed: 09-May-2020]
6. "A Fruitful Pursuit" [Online]. Available: https://echelon.lk/a-fruitful-pursuit/[Accessed: 09-May -2020]
7. *M.A. Momin, M.T. Rahman, M.SSultana,C. Igathinathane, A.T.MZiauddin,T.E.Grift. "*Geometry based mass grading of mango fruits using image processing" [Online]. Available: http://dx.doi.org/10.1016/j.inpa.2017.03.003[Accessed: 09-May-2020]
8. "Mango weight grading machine, mango weight sorting machine" [Online]. Available: https://www.fruitpittingmachine.com/mango-weight-grading-machine.html[Accessed: 09-May-2020]
9. *Anuja Bhargava, Atul Bansal*. "Fruits and vegetables quality evaluation using computer vision" [Online]. Available:https://doi.org/10.1016/j.jksuci.2018.06.002 [Accessed: 09-May-2020]
10. *The world banks. Sri Lanka* "GDP share of agriculture - data, chart" [Online]. Available: https://www.theglobaleconomy.com/SriLanka/share_of_agriculture [Accessed: 10-May-2020]
11. *H.M.S. Heenkenda*. "Present status of fruit industry and issues on organic fruit production in Sri Lanka" [Online]. Available: http://itfnet.org/Download/tfnetsymposium2015/11-SriLanka.pdf [Accessed: 11-May-2020]
12. *Department of Census and Statistics (*2013). "Performance report of the department of census and statistics for the year 2016" [Online]. Available: https://www.parliament.lk/uploads/documents/paperspresented/performance-report-department-of-census-and-statistics-2016.pdf [Accessed: 11-May-2020]
13. *W.A.G. Sisira Kumara*. "Sri Lankan Agriculture" [Online]. Available: http://www.fao.org/fileadmin/templates/rap/files/uploads/ESF_Presentations/Sri_lanka_e_solutions_Sisira_Kumara.pdf [Accessed: 11-May-2020]
14. *Ministry of Agriculture, Rural Economic Affairs, Irrigation, and Fisheries,And Aquatic Resources Development.* "Sri lanka Overarching Agricultural Policy" [Online]. Available:

http://www.agrimin.gov.lk/web/images/Information_Act/Development/2019_08_19_Draft_OAP.pdf[Accessed: 11-May-2020]

15. *Kondo, Naoshi,* 2003. “Fruit grading robot. In: Proceedings of the 2003 IEEE/ASME, International Conference on Advanced Intelligent Mechatronics” (AIM 2003), pp. 1366–1371.

16. *Gay, Paolo & Remigio, Berruto*. (2004). “Innovative techniques for fruit color grading”. [Online]Available: https://www.researchgate.net/publication/228761300_Innovative_techniques_for_fruit_color_grading

17. *Laykin, S., Alchanatis, V., Fallik, E., Edany*, 2002. Image processing algorithms for tomatoes classification. Transactions of the American Society of Agricultural Engineers (ASAE) 45 (3), 851– 858.

18. *Polder, G., van der Heijden, G.W.A.M., Young,* I.T., 2000. Hyperspectral image analysis for measuring ripeness of tomatoes. ASAE International Meeting, ASAE Paper No. 003089.

19. *Unay, Dervim,* 2006. Multispectral Image Processing and Pattern Recognition Techniques for Quality Inspection of Apple Fruit. Ph.D. Thesis, Facult´e Polytechnique de Mons, Belgium.

20. *Aguilera, Cristhian, Ramos, Mario, Roa, Gabriel,* 2006. An automatic grading system for panels surfaces using artificial vision. International Journal of Computers, Communications and Control I (2), 15–22.

21. *Abdesselam, Abdelhamid, Abdullah, Rahmat Choo*, 2000. Pepper berries grading using artificial neural networks. In: TENCON 2000. Proceedings vol. 2, pp. 153–159.

22. *Saito, Yasuo, Hatanakat, Toshiharu, Uosakit, Katsuji, Shigetot, Kazuhide*, 2003. Eggplant classification using artificial neural networks. Proceedings of the International Joint Conference on Neural Networks 2, 1013–1018.

23. *YousefAl Ohali,* Computer vision-based date fruit grading system: Design and implementation. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157810000054?via%3Dihub

24. *Olaniyi, Ebenezer & Oyedotun, Oyebade & Khashman, Adnan*. (2016). Intelligent Grading System for Banana Fruit Using Neural Network Arbitration. Journal of Food Process Engineering. [Online]Available:https://www.researchgate.net/publication/291553678

25. *Medjahed, Seyyid Ahmed.* (2015). A Comparative Study of Feature Extraction Methods in Images Classification. International Journal of Image, Graphics and Signal Processing vol. 7, pp. 16-23. 10.5815/ijigs.2015.03.03. http://www.reseracgate.net/publication/272367419

26. *Sri Lanka Rupavahini*. 2019. [video] Nugasewana: Nugasewana Ministry of Primary Industries & Social Empowerment. [Online]Available: https://www.youtube.com/watch?v=1Bs8O4xrEow

27. *Myas, Ankur & Talati, Bijal & Naik, Sapan*. (2014). Quality Inspection and Classification of Mangoes using Color and Size Features. International Journal of Computer Applications.vol 98, pp.1-5. [Online]Available: https://www.researchgate.net/publication/272863257.

28. *Payman Moallem, Alireza Serajoddin* (2016) *Hossein Pourghassem,* Computer vision-based apple grading for golden delicious apples based on surface features,Information Processing in Agriculture,vol. 4, pp. 33-40. [Online]Available: https://doi.org/10.1016/j.inpa.2016.10.003.

29. *Ebenezer Obaloluwa Olaniyi ,Oyebade Kayode Oyedotun ,Khashman Adnan,* Intelligent Grading System for Banana Fruit Using Neural Network Arbitration [Online]Available: https://onlinelibrary.wiley.com/doi/full/10.1111/jfpe.12335

30. *R. Pandey, N. Gamit and S. Naik, (2014),* "A novel non-destructive grading method for Mango (Mangifera Indica L.) using fuzzy expert system," 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), New Delhi, pp. 1087-1094 [Online]Available: https://ieeexplore.ieee.org/document/6968366

31. *C. S. Nandi, B. Tudu and C. Koley*, (Aug.15, 2016)"A Machine Vision Technique for Grading of Harvested Mangoes Based on Maturity and Quality," in IEEE Sensors Journal, vol. 16, no. 16, pp. 6387-6396. [Online]Available: https://ieeexplore.ieee.org/document/7490390

32. *V. Ashok and D. S. Vinod, (*2016)"A comparative study of feature extraction methods in defect classification of mangoes using neural network," 2016 Second International Conference on Cognitive Computing and Information Processing (CCIP), Mysore, pp. 1-6. [Online]Available: https://ieeexplore.ieee.org/abstract/document/7802873

33. *Mohammed A.H.Ali, KelvinWongThai* Automated fruit grading system[Online]Available: https://ieeexplore.ieee.org/abstract/document/8231734 [Accessed 11 May 2020]

34. Unece Standard(2017).Unece Standard Ffv-45 Mangoes [Online].Available: https://www.unece.org/fileadmin/DAM/trade/agr/standard/standard/fresh/FFV-Std/English/45_Mangoes.pdf [Accessed: 11-May-2020]

35. *Mizuho Nishio, Richard Kinh Gian Do, Kaori Togashi* Convolu*tional neural net*works: an overview and application in radiology [Online]. Available: https://www.researchgate.net/publication/325932686_Convolutional_neural_networks_an_overview_and_application_in_radiology [Accessed: 11-May-2020]

36. Mango Project Overview https://roomifoods.pk/index.php/mango/ [Accessed 11 May 2020]

37. *P. Nistrup, "*What is .npy files and why you should use them*", towardsdatascience.com, 2019. . .*https://towardsdatascience.com/what-is-npy-files-and-why-you-should-use-them-603373c78883 [Accesses: 25-March-2021]

38. "Design Pattern - Factory Pattern - Tutorialspoint", *Tutorialspoint.com*, 2021. [Online]. Available: https://www.tutorialspoint.com/design_pattern/factory_pattern.htm. [Accessed: 25-Mar-2021].

39. "Design Pattern - Singleton Pattern - Tutorialspoint", *Tutorialspoint.com*, 2021. [Online]. Available: https://www.tutorialspoint.com/design_pattern/singleton_pattern.htm. [Accessed: 25-Mar-2021].

40. "Design Patterns - MVC Pattern - Tutorialspoint", *Tutorialspoint.com*, 2021. [Online]. Available: https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm. [Accessed: 25-Mar-2021].

41. "Design Patterns - Observer Pattern - Tutorialspoint", *Tutorialspoint.com*, 2021. [Online]. Available: https://www.tutorialspoint.com/design_pattern/observer_pattern.htm. [Accessed: 25-Mar- 2021].

42. "What is web socket and how it is different from the HTTP? - GeeksforGeeks", *GeeksforGeeks*, 2021. [Online]. Available: https://www.geeksforgeeks.org/what-is-web-socket-and-how-it-is-different-from-the-http/. [Accessed: 25- Mar- 2021].

Appendix A: Diagrams

qanad@nelna.lk
to Punya, me, Upul

Dear Mr. Sasadara,

We grant the permission to collect the required data for your research in requested dates.

Best regards,

Nuwan Wijewardhana
Assistant Manager - Quality Assurance
(M) +94 76 9753166
(T) +94 47 5678245
(F) +94 47 5679600
(E) qanad@nelna.lk
www.nelnamango.com/ www.nelna.lk

Save a tree. **Don't print** this e-mail unless it's really necessary.

IMPORTANT: The contents of this email and any attachment are confidential They are intended for the named recipient(s) only. If you have received this mail by mistake, please notif

Nelna Agri Development (Pvt) Ltd
Hathduwa Estate, Ranwala, Meethrigala, Sri Lanka
Kadurugasara, Thunkama, Embilipitiya, Sri Lanka

Figure 90: Permission letter for data gathering

mango dataset information

	A	B	C
1	image_number	weight(g)	grade
2	424	595	1
3	430	695	1
4	436	606	1
5	442	755	1
6	448	679	1
7	454	722	1
8	460	608	1
9	466	705	1
10	472	790	1
11	478	703	1
12	484	534	1
13	490	515	1
14	496	850	1
15	502	502	1

Grade 1 | Grade 2L | Grade 2S | Grade 3 | Grade 4

Figure 91: Dataset of grade 1 mangoes

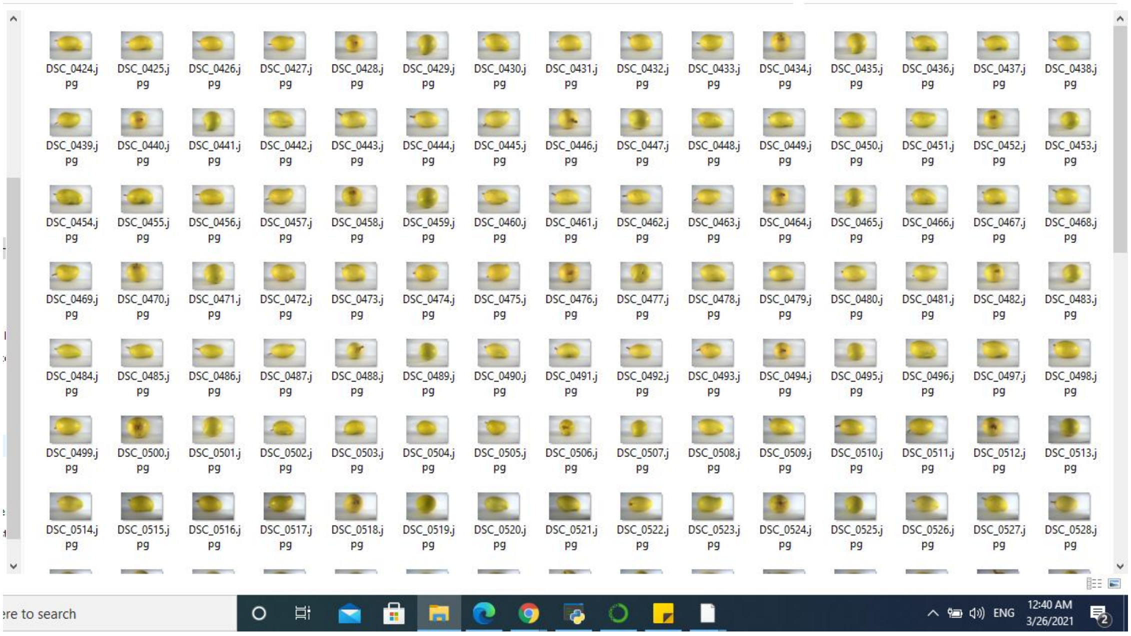


Figure 92: Grade 1 mango images

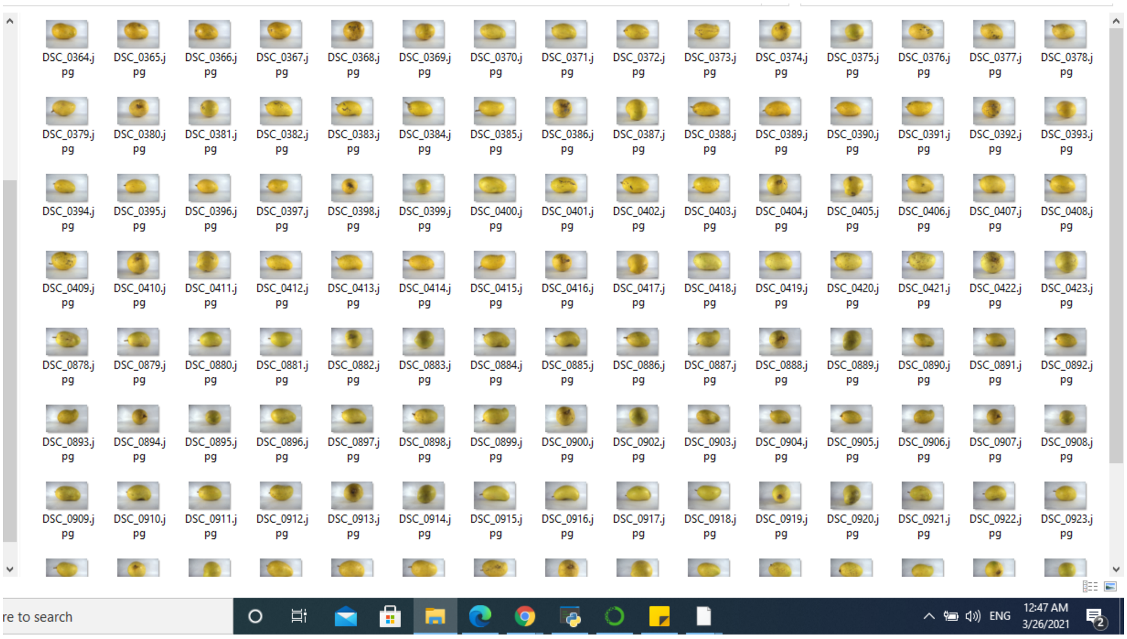


Figure 93: Grade 2 large mango images

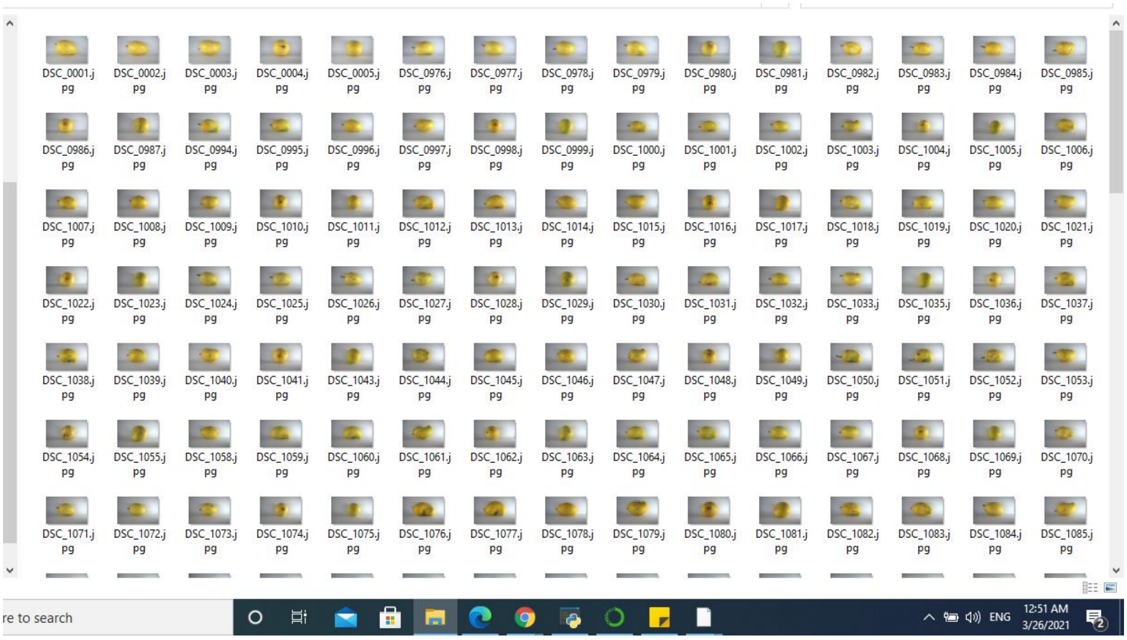


Figure 94: Grade 2 small mango images

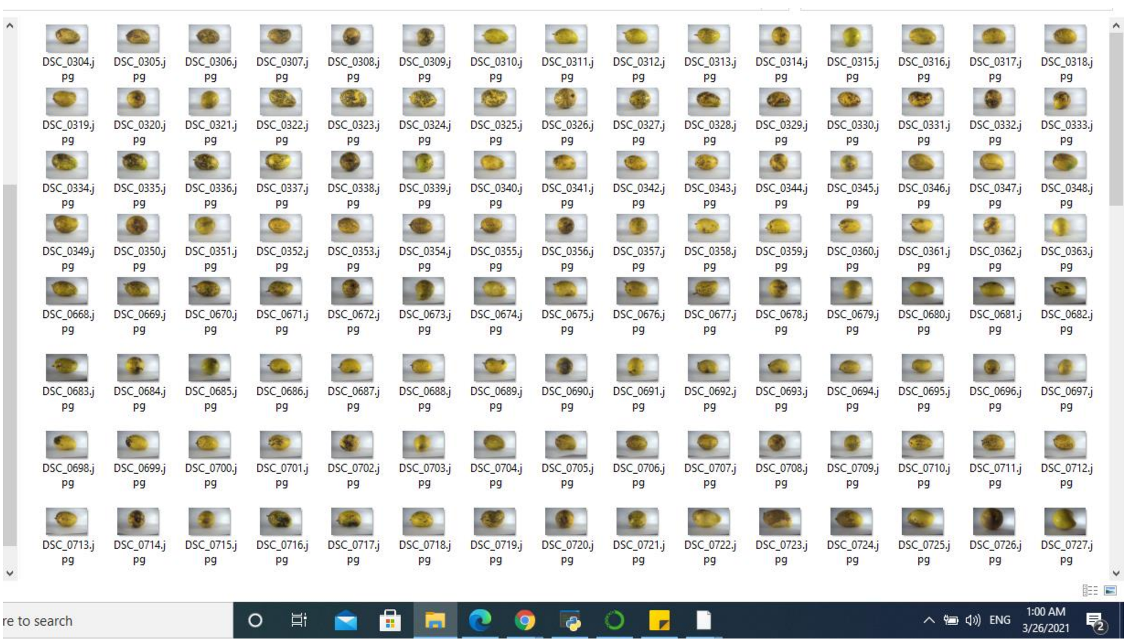


Figure 95: Grade 3 mango images

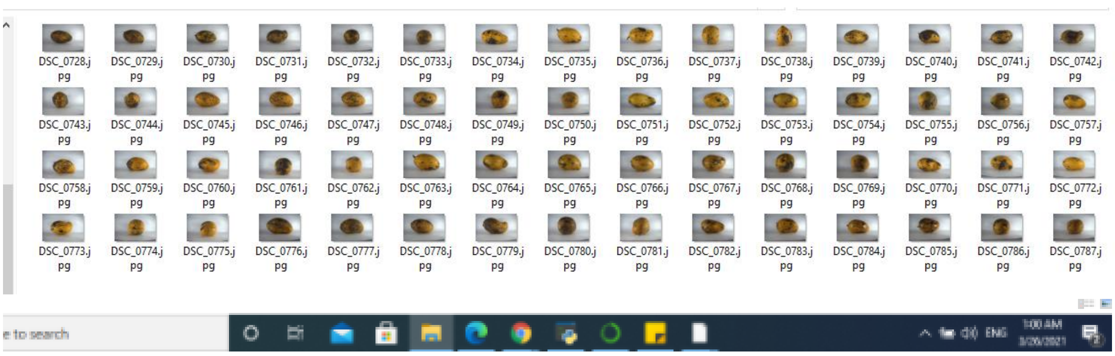


Figure 96: Grade 4 mango images

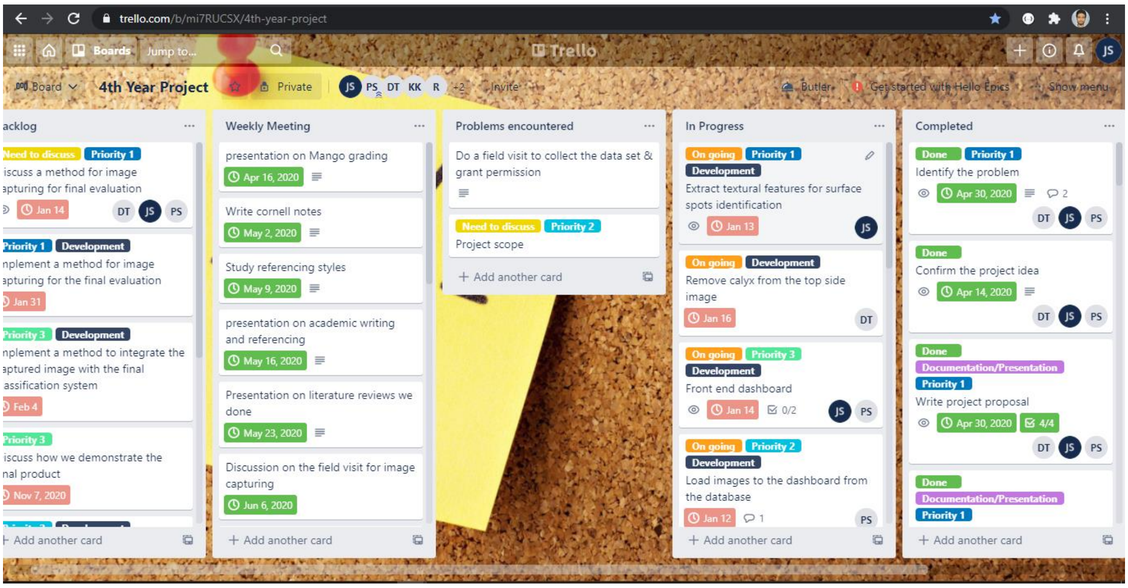


Figure 97: Grade 4 mango images

Appendix B: Contribution

In accordance to the problem definition in this dissertation, all the three team members were equally contributed in finding an approach to achieve the proposed goal. Different components in the project were implemented by different members and some components were implemented collectively by all the members.

Contributor 1: Patabendige S.S.J

- Literature review on mango grading
    - Based on Spot analyzing
    - Based on size
- Data gathering
    - Background study on TJC mango grading process
    - Data gathering and dataset creation
- System architecture designing and design the suggested system
- Image preprocessing
    - Bounding rectangle detection
    - Stem reduction
    - Detected the region around the stalk for top side images using K-Means and K-Medoids clustering algorithms.
- Feature extraction
    - Detect the HOG in the image and calculate the mean and standard deviation in HOG
    - Features were extracted relevant to size analyzing classification model
- Feature selection
    - Selected the optimal subset of features using SFS
- Classification model built based on surface spots is trained, tested and validated for side images, top images, bottom images, using FFNN
- Classification model built based on surface spots is trained, tested and validated for side images, top images, bottom images, using SVM
- Classification model built based on the size is trained, tested and validated, using SVM

- Rule based algorithm was optimized in a multi-threaded environment.
- Python Flask APIs and web socket implementation
- Implementing the software dashboard
    - pie chart/ Bar chart

Contributor 2: A.V.P.Sewwandi

- Literature review on mango grading
    - Based on Spot analyzing
    - Based on size
- Data gathering
    - Background study on TJC mango grading process
    - Data gathering and dataset creation
- System architecture designing and design the suggested system
- Image preprocessing
    - Image smoothing
    - Contour detection
- Feature extraction
    - Identify intensity distribution of YCBCR channels and calculate mean and standard deviation for each channel separately.
- Feature selection
    - Selected the optimal subset of features using SFS
- Classification model built based on surface spots is trained, tested and validated for side images, top images, bottom images, using CNN
- Classification model built based on the size is trained, tested and validated, using KNN
- Database designing
- python flask backend API Implementation
- Implementing the software dashboard
    - Show statistical data from the database
- Implement the process management software components
- Automation testing was carried out using KARMA to test the frontend.

Contributor 3: D.D.Tharaka

- Literature review on mango grading
    - Based on Spot analyzing
    - Based on size
- Data gathering
    - Background study on TJC mango grading process
    - Data gathering and dataset creation
- System architecture designing and design the suggested system
- Image preprocessing
    - Image enhancing
    - Image segmentation
    - Contour detection
- Feature extraction
    - Identify intensity distribution of RGB channels and calculate mean and standard deviation for each channel separately.
- Feature selection
    - Selected the optimal subset of features using SFS
- Classification model built based on surface spots is trained, tested and validated for side images, top images, bottom images, using KNN
- Classification model built based on the size is trained, tested and validated, using FFNN
- In order to determine the final grade, a rule-based algorithm built to combine results obtained from the classification models which are developed based on surface spots and size.
- Implementing the software dashboard
    - pie chart/ Bar chart
- Test case writing and manual testing was carried out for each individual test items