DEVELOP AN OIL PALM HARVESTING PLAN

R.M.S.S. Herath

179463G

Faculty of Information Technology

University of Moratuwa

July 2021

DEVELOP AN OIL PALM HARVESTING PLAN

R.M.S.S. Herath

179463G

Dissertation submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka for the fulfillment of the requirements of Degree of Master of Science in Information Technology.

July 2021

Declaration

We declare that this is our own work and has not been submitted to another degree or diploma at a university or other higher education institution. Information obtained from published or unpublished work by third parties is acknowledged in the text and a list of references is provided.

Name of Student	Signature of Student
R.M.S.S Herath	……………………..
	Date: ……………….

Supervised by

Name of Supervisor	Signature of Supervisor
Mr.Saminda Premarathna	……………………….
	Date: ………………..

Acknowledgements

First of all, I would like to thank Mr. Saminda Premarathna, a lecturer at the Faculty of Information Technology at Moratuwa University, for his valuable time in teaching, monitoring, consulting and research projects.

In addition, I would be glad to thank Dr. Mohamed Fildos for teaching me research methodology and literature review, as well as my thesis in which the subject was written on which this study was based.

Thanks also to all the colleagues in IT degree program and my office staffs who gave us valuable comments in order to improve the results of our research throughout my life, especially the support that I provided.

Contents

List for Figures

Page No

List for Tables

Page No

Abstract

Fruit harvesting techniques that are both safe and quick are becoming more popular in the agricultural sector. Traditional grading systems are still widely employed today, despite the advancement of technology. However, high prices and certain discrepancies force the post-harvesting sector to use automation in categorization activities due to the high expenses.

Recently, businesses have begun to turn to automation technologies in order to increase working capacity while simultaneously lowering operating expenses. When automated grading methods are utilized, the inconsistencies that are connected with hand grading are reduced. As a result, the mistake rate and expenses are reduced, while the speed is increased. As is well known, the size, shape, color, and tissue of the object are the fundamental factors in the categorization process. An investigation into automated color grading of oil palm utilizing digital cameras and computerized image processing methods was conducted in this research. The constructed system has accomplished the most fundamental duties, but it still needs additional development.

CHAPTER I

1. Introduction

1.1 Background

Image processing is a vital technology that is utilized in a variety of different sectors. Nowadays, due to the technological impact on farming, image processing is being used in a variety of sectors for the detection of insects, bacterial factors, weeds, crop quality and other variables. Image processing is also being used in the pharmaceutical industry. Digital image processing is the process of manipulating pictures with the use of digital technologies. The number of people who use it has grown dramatically in recent decades. Medicine, entertainment, geological processing, and remote sensing are just a few of the applications. Multimedia systems, which are the basis of the contemporary information society, rely significantly on digital image processing to function properly.

In the area of digital image processing, both digital signal processing methods and image-specific techniques are used to create digital images. A image is defined as a function f (x, y) of two variables that are constants, x and y. It must first be sampled and transformed into a number matrix before it can be digitally processed. Because a machine can only show numbers with a limited degree of precision, the numbers must be represented digitally. Image processing in digital form requires dealing with accuracy numbers that are limited in quantity. Digital image processing may be split into four categories: enhanced pictures, restored images, analyzed images, and compressed images. Improved images are the most common kind of image processing. When it comes to image enhancement, heuristic methods are used to alter pictures in such a way that human viewers can extract valuable information from them. Image restoration methods are intended to handle damaged pictures in which the degradation has been statistically or mathematically characterized in order to reverse the deterioration and restore the original image. Imaging methods allow for the processing of images in such a way that information may be retrieved from them without the need for a human to do so. Image segmentation includes the study of pictures, edge extraction, texture analysis, and motion analysis, to name a few examples. The vast quantity of information required to depict pictures is a crucial characteristic of the images themselves. The bit count for even a grayscale image with modest resolution, such as 512 to 512, is 512 to 512 to 8 or 2 106 bits, depending on the kind of picture. If one wants to save and transmit digital pictures in a realistic manner, one must employ some kind of picture compression, which takes use of the redundancy of images in order to minimize the number of bits needed to represent them.

Oil palm farming has been practiced in Sri Lanka for a long time. This plant was first grown in Nakiyadeniya, Sri Lanka, and introduced to the country in 1968. Construction on the Kalutara extension began in 1996, but was abandoned due to widespread public resistance and economic concerns in the surrounding area. In Kegalla district, however, the new administration has promised to sow 20,000 hectares of crops. There were widespread public protests in the area. Large planting firms have started to root rubber and grow palm oil, as well as to root other crops. The destruction of the forests inside the rubber plantations and river forest regions started in earnest. Wild animals that cause harm to the palm oil plantations have also been targeted and killed. Prior to this decision, there had been insufficient feasibility, social, and environmental evaluations conducted.

The Namunukula Plantations PLC is a wholly owned subsidiary of the Richard Peiris Group. A variety of crops are grown at this operation, including tea, rubber, oil palm, coconut, and cinnamon. For the last ten years, the company has been engaged in the agribusiness industry.

The oil palm (Elaeis Guineensis) is a commercial crop native to West Africa that is grown for its oil. It was first introduced in Sri Lanka in 1969, and is mostly found in the districts of Kalutara, Galle, and Rathnapura, among other places.

Sri Lanka produces crude oil for export after processing the mature fruit bunches via steaming and oil extraction operations. The estates provide the factory with a ready bunch of raw foam that has a reasonable profit margin. This is the most important element in the manufacturing process since the rate of oil extraction (OER) changes depending on the maturity of the fruit bunch throughout the harvesting phase. Harvesting is usually done manually using harvesting knives and fake poles, which is the most common technique. In theory, orange bunches having at least 10 loose fruit are regarded to be ripe bunches of fruit.

Growing a successful sparkling bunch is a time-consuming process that takes at least 36 months from inception to harvest. Harvesting a cluster of palm leaves after they have been treated at the palm's top may take four to six months.

Normally, 12-16 bunches are produced each year, with 3-5 bunches being grown in various stages of development in the crown at any one time.

The majority of the bunches are covered with thighs. Color and condition of the appropriate bunch in high palms are difficult to discern with the naked eye since they are ready to be sprayed in the crown at any moment when the bunches are mature and thorny in appearance. Precision in harvesting is a matter of skill.

1.2 Problem Statement

It is no easy task to grow oil palm, but with the proper attention and work, it can be done. Not only that, day by day cost of harvesting cost is going up to due to inappropriate harvesting plan .The harvesting cost included mainly crop loss and harvesting related other charges . Many are reported to be the result of negligence within the business while others appear to be linked to the facility itself. Many species of oil palm may be found in a color pattern of four hues (Yellow, Orange, Dark red, and Black). It is recommended that this variety be harvested only once it has achieved a particular hue. Sri Lanka does not have any labors that are skilled at harvesting all time. It is unknown if they have adequate understanding to grasp particular colors. As a result of those harvesting oil palms when they are mature instead of at the ideal period produces a certain proportion of wastage. There is yet another issue, and that is that harvesting palm oil does not happen every week. But in the case of many oil palms, rot occurs because it is not possible to harvest the palms that need to be harvested within the allotted time period. And because this harvesting procedure is done by people, a number of trees may be missed. Due to all of the concerns with palm oil mentions, it may be estimated that around 7-8% of oil palm residues might be squandered.

1.3 Research Questions

It would be interesting to see whether it is possible to determine the maturity of an oil palm bunch based on its color and age as a research topic.

- How does the color change in oil palm bunch varied according to its maturity?
- How to predict estimated harvesting date?
- How train an optimized model?

1.4 Aim and Objectives of the study

1.4.1 Aim

- To provide proper Harvesting plan based on analyzing the color of bunches.

1.4.2 Objectives

- To collect images of oil palm bunches in each age.
- To analyze the color of oil palm bunches.

- To predict the date of harvesting of each bunch.

1.5 Scope of the study

This research attempts to identify the well matured oil palm bunches by considering it color to harvest the crop at the very best time period.

- Consider the color changes in bunch in different maturity periods.
- Consider the relationship between color and maturity.

1.6 Proposed solution

Initially planning to take images of every formed bunch of every oil palm tree, by using Android mobile phone in field level .After that processing those images, a proper harvesting plan will be provided to management .

CHAPTER II

2. Literature Review

2.1 Oil palm harvesting in Sri Lanka

In the sorting and classification of kiwi fruit (Fu et al., 2016; Rashidi & Seyfi, 2008), strawberries , tomatoes (Clement et al., 2012), Oil palms Altisent, and Valero (2008) categorized olive fruits according to the presence or absence of external faults. After classifying the fruits based to their color and external flaw form, the experts split them into seven groups. Furfure developed a machine vision system for categorizing olive fruits based on their ripeness and the presence or absence of external defects. Liming and Yanchao (2010) developed a method for automated image-based strawberry sorting that is still in use today. The computer was capable to distinguish between the physical features of strawberries based on their image characteristics. They obtained an accuracy of 88.8% when sorting by color attributes and 90% when sorting by shape attributes. Mousavi (2012) used an image processing method created at the University of California to distinguish and categorize cherry cherries based on their size, ripeness, and defects. Elmasry et al. designed and tested a machine vision system that is quick and accurate method for differentiating irregular from normal form potatoes (2012). They discovered that when used to sort potatoes by size, maturity, and defect-based sorting, the system had a functional accuracy of 96.2 percent. Mohammadi et al. (2015) classified date plum fruit according to their maturation and color using an image processing technique. The researchers discovered that image analysis showed a substantial variation in the majority of image properties, including the R, G, and B channels, as well as the gray level, across different degrees of ripeness. Additionally, they demonstrated that quadratic discriminant analysis (QDA) was capable of accurately classifying fruits 90% of the time. Kheiralipour and Pormah (2017) used image processing techniques and machine learning networks to sort cucumber fruits with a 97.1 percent accuracy using the best sorting model developed by neural networks. According to a study of the available research on the topic, no research on carrot sorting based on shape has been published. Thus, the purpose of this study’s aim to identify the outer shape of carrots makes use of machine vision, which is beneficial for carrot sorting in order to expansion of product marketability while also decreasing waste.

Brazil's fruit and vegetable industry, as well as related sectors, are experiencing substantial losses due to a lack of suitable technologies. The quality of vegetables is often characterized in terms of its size, shape, mass, hardness, color, and bruising, which are all used to classify and grade vegetables. However, owing to the technological implementation in this field, most companies are unable to pay software, infrastructure, and operating costs. Because optical approaches, such as moiré, are quick and do not need physical contact with the specimens, they are especially helpful for geometric characterization and sorting fragile goods such as eggs, fruits, and other plant parts. This is because they are inexpensive and enable automation.

It was founded by Wayne Rasband of the National Institute of Mental Health's Research Services Branch in Bethesda, Maryland. ImageJ is a program that enables the viewing of pictures in a number of formats. The letter J stands for Java, a computer language. Version 0.50 was launched on September 23, 1997, while version 1.31 was released in February 2004. Apart from being freely available, the program runs on all major operating systems, has an intuitive user interface, and is capable of performing a broad range of picture changes, showing a large and competent user community. The ImageJ program supports a variety of photo acquisition devices, including scanners, different video sources, cameras (including TWAIN-compatible cameras), FireWire, Cooke frame grabbers, National Instruments, and PixelSmart.

Images may be altered in a number of ways, including reading and writing files, performing pixel operations on single pixels and picture areas, manipulating whole images, and manipulating volumes. Volumes organized as image sequences may be used in their entirety. Apart from that, it is capable of performing basic operations on image collections, including convolution (compression), edge detection (fourier transformation), histogram (histogramme), editing (color alteration), dilation, and mathematical operations (multiplication and/or division). The visualization techniques include color space conversions from RGB to Hue Saturation color space, 2D and 3D charting, surface and volume rendering, and a variety of other approaches and tools. ImageJ is compatible with a wide range of operating systems, including Microsoft, Macintosh, and Linux.

The program's high limit enables users to create Macros to automate often repeated tasks, yet doing so is easy - and does not need knowledge of Java. Plug-ins are extra programs that expand the capabilities of ImageJ. The majority of them are built in the Java programming language. The third method of increasing software's functionality is to add new features to it. software is via the API (application programmer's interface), or the user interface.

According to the literature, fruits may be classified by mass with a precision of one gram, diameter with a precision of one mm, color, bruising, shape, and density. Additionally, they may be categorized by size with an accuracy of one gram. Electrical sorters may be set with 16 distinct sizes, eight distinct colors, and four distinct criteria, totaling 512 distinct criteria. Typically, this kind of equipment includes a sensor, lens, light source, filter, and personal computer (VON BECKMANN and BULLEY, 1978; SARKAR and WOLFE, 1985; HAHN, 2002). KONDO and TING provide a simple data collecting setup that incorporates color, mass, and scale (1998). Despite the fact that the authors will consider the components of a simple design, they will rely on reliable software to distribute the product to the right networks. Modern technology has the ability to sort up to ten fruits each second according to color, size, shape, spots, bruises, stem presence, and other fruit features. According to JAHNS et al. (2001), the absolute blunder in mass-sorting tomatoes using image analysis was about 2.06%.

Color is associated with physiological maturity in a greater number of fruits and may be used as a sorting pattern in a greater number of fruits. According to ARIAS et al. (2000), the color of the surface of the tomato is a key aspect in determining the fruit's ripeness.. JAHNS et al. (2001) further assert that at the pixel level, color, spots, and bruises may be identified. In 2002, HAHN reported a success rate of 85 percent utilizing a multi-color scheme to identify tomatoes that were immature physiologically. POLDER et al. (2003) assert that they have established a substantial link among spectral photoes and tomato lycopen material, which is responsible for the fruit's red color and changes in intensity according on the degree of ripeness at which the fruit is picked.

The majority of sorting equipment has an algorithm that selects a certain color range, as is usual practice. According to KADER (2002), several photographs were needed to determine the diameter of the fruit, implying that video footage may be utilized to evaluate the fruit's appearance. According to VON BECKMANN and BULLEY (1978), by reducing the amount of fruit handled, sorting fruit by size and color would save time.

2.2 Image Processing

Image processing is a method of improving or extracting usable information from a photograph by executing specific operations on it. It is used to improve the image's quality or to retrieve valuable data from it. It is a kind of processing of signal in which an image is used as the input and the result may be either a picture or a collection of characteristics. Image processing is one of the most quickly evolving technical areas today. It is also critical as a primary topic of study in engineering and computer science.

In general, image processing consists of the following main steps:

- Picture input through image capture tools;

- Image analysis and modification; and
- Output, which may include a modified image or report based on image analysis.

When it comes to image processing, there are two methods to consider: analog pictures and digital images. Analog image processing, such as printing and sketching, may be utilized to create tangible copies of artwork. When image analysts use these visual methods, they employ a variety of distinct perception principles. Using digital image processing techniques, computers are able to alter digital pictures more effectively. All types of data are exposed to three main stages throughout the digital technology process: pre-processing, improvement, and presentation of information extraction, to name a few.

During this lecture, we will cover a variety of basic topics, including image, digital image, and digital image processing, among others. The presentation will cover a variety of digital picture sources, with examples for each source being provided. From image identification to computer vision, the topics covered in this lecture will cover the whole spectrum. Finally, we will discuss the acquisition of pictures as well as the many types of image sensors.

Image processing is a method of conducting specific operations on a picture in order to get a better image or to extract valuable information from it. It is used to enhance the image or to recover useful information from selected image. It is a kind of signal processing in which the input is an image and the output may be either a picture or a set of features. The processing of pictures is now one of the most rapidly developing technological fields. It also serves as a focal point for research in the areas of engineering and computer science, among other things.

Basically, image processing consists of three steps:

- Image import by image acquisition tools;
- Image modification and analysis;
- Output in which the image or report based on image processing may be changed.

When it comes to image processing, there are two methods to consider: analog pictures and digital images. Analog image processing, such as printing and sketching, may be utilized to create tangible copies of artwork. When image analysts use these visual methods, they employ a variety of distinct perception principles. Using digital image processing techniques, computers are able to alter digital pictures more effectively. All types of data are exposed to three main stages throughout the digital technology process: pre-processing, improvement, and presentation of information extraction, to name a few.

During this lecture, we will cover a variety of basic topics, including image, digital image, and digital image processing, among others. Various digital picture sources will be discussed in detail, with examples provided for each kind of source. From image identification to computer vision, the topics covered in this lecture will cover the whole spectrum. Finally, we will discuss the acquisition of pictures as well as the different types of image sensors available.

2.3 Image Processing in Harvesting Fruits and Vegetables

In the sorting and categorization of kiwi fruit , strawberries, tomatoes, and aple , and tomatoes , . Image processing has been widely used to determine the height, shape, and defects of fruits and vegetables categorized olive fruits based on the form of their external defects and found that this categorization was correct. Specialists first categorized the fruits into seven categories, following which they were further classed according to color and external defects. Carfagni (2010), for example, developed a visioning system for classifying olive fruits according to their degree of maturity and the presence or absence of external defects. Liming (2010) released a paper in 2010 describing an automated strawberry selection method based on image analysis. The device was able to detect the physical characteristics of strawberries based on the image's features. The accuracy of the evaluation was 88.8 percent for color characteristics and 90 percent for shape attributes. Mousavi Balestani (2012) used image analysis to discriminate and classify cherry fruits based on their height, maturity, and defects. Sorting was found to be accurate to within 96 percent, 92 percent, and 90 percent of the fruits' sizes, ripeness, and defects, respectively. Elmasry et al. (2012) developed a quick and reliable visual method for distinguishing irregular and normal form potatoes, determining that the system has a functional precision of 96.2 percent (the ability to distinguish between shapes accurately). Mohammadi et al. (2015) classified persimmon fruit according to their ripeness degree using an image processing technique. Their study discovered a substantial variation in the majority of image characteristics such as red, green, and blue, as well as the gray level, across different maturity levels. Additionally, the accuracy with which the QDA analysis classified the fruits was noted (90.24 percent). Pormah (2017) used image and artificial neural networks to classify cucumber fruits and claimed that their neural network generated the best sorting model with a 97.1 percent accuracy. According to the results of the literature review, there is no published research on carrot sorting based on form. To improve the product's marketability and control, the goal of this research was to determine the form of carrots using machine vision, which is beneficial for carrot sorting and therefore marketability.

Brazil's fruit and vegetable production, as well as related industries, are incurring substantial losses as a result of a lack of suitable technology. The quality of a vegetable is often described in terms of its size, shape, mass, hardness, color, and contouring, which may all be used to grade and categorize the vegetables. However, technological adoption in this industry is not feasible owing to a lack of suitable applications, infrastructure, and operating costs. Because optical technologies such as moiré methods enable rapid virtual interaction with specimens, low cost, and automation, they are particularly useful for geometric characterizations and the implementation of sorting techniques for delicate items such as eggs, vegetables, and other plant organs.

On the basis of these factors, the research project proposes to modify free software to enable the classification system to identify color, shape, volume, and potentially blemishes in a single look, resulting in the creation of reliable and low-cost methods for sorting of fruits.

2.4 Image Pre-Processing

Data cleaning, also known as data pre-processing, is a critical stage in the machine learning engineering process. For many machine learning engineers, this represents a significant amount of time and effort; they devote several hours on data cleaning and pre-processing. There are a variety of pre-processing methods that you may use to improve the quality of data, including outlier identification, management of missing values, and removing unnecessary or noisy information.

Another way of putting it is that image processing is the discipline of dealing with images at the most basic level of abstraction possible. Neither of these operations increases nor decreases the amount of information contained in the picture, depending on whether or not entropy is regarded as information. When pre-processing images, the aim is to enhance the image data in such a manner that undesirable distortions are reduced to a bare minimum. Alternatively, the goal is to maximize picture characteristics that will be useful in future processing and analysis activities.

Image Pre-Processing is a term that refers to a group of four different pre-processing methods that are discussed below.

1. Brightness transformations/corrections for pixels
2. Transformations of Geometry
3. Filtering and Segmentation of Images
4. Image restoration and Fourier transform

1. Pixel brightness transformations (PBT)

In-between brightness changes alter the brightness of individual pixels, and the properties of a pixel are taken into consideration throughout these transformations. In PBT, no other pixel value has any effect on the pixel value that is shown on the screen. The same can be said for the operators that enable users to make changes to things like brightness and contrast, in addition to color correction and transformations.

Contrast augmentation is one of the most important areas of image processing for both human and computer vision, and it is one of the most challenging. The technique is used in medical image processing, as well as in speech recognition, texture generation, and a variety of other image and video processing applications.

2. Geometric Transformations

This article begins with an overview of past methods before delving into the subjects of color and contrast. Despite the fact that the geometric transformation is used to change the placements of pixels in an image, the colors of the image are not altered in anyway.

A geometric transform is a technique that allows for the minimization of geometric distortion that occurs when images are captured. Geometric transformation methods, such as rotation, scaling, and distortion (or distortion!) of images, are referred to as "geometric transformation procedures" in the general public lexicon.

There are two different processes that must be followed when doing geometric transformations:

For starters, the spatial transformation of the actual rearranging of pixels in the image takes place during the rendering process.

Gray level interpolation, which assigns grey levels to the changed image, is a technique used in computer graphics.

Methods of intermodal interpolation include:

After the transformation operations were completed, the new coordinates for the new point, x' and y', were computed. So let's pretend for the sake of argument that these extra points do not in general match the discrete raster of the output picture. In order to do this, interpolation techniques may be used to obtain the values of each pixel in the resultant image raster.

It is customary in the case of the brightness interpolation issue to provide it in a two-part format. Y' is the brightness value of the pixel (x',y') in the output image, and x' and y' correspond to discrete raster points on the output picture..

Interpolation techniques are available in a wide range of configurations.

Most basic technique, closest neighbor interpolation sampled the pixel values from the input vector or matrix using an interpolation method such as nearest neighbor, is the most common.

Using linear interpolation, four points around a given point (x,y) are assessed, with the assumption that the brightness function is linear in this neighborhood.

Bicubic interpolation, which is used in the 3D model of the brightness function, improves the approximation by applying a bicubic polynomial surface on the surface in a localized manner. Interpolation is performed on the target using sixteen nearby points that are used for calibration.

3. Image Filtering and Segmentation

In order to alter and/or enhance visual characteristics, as well as to collect essential data from pictures such as edges, corners, and blobs, filters are used in conjunction with image editing software. A kernel is a tiny array that defines a filter that is applied to each pixel in an image as well as all of the pixels in its immediate vicinity.

The methods listed below serve as the building blocks for more sophisticated filters. They include:

is a low-pass filter that lowers noise and smooths out peaks to the greatest extent possible. Low-pass filters are often used as a starting point for smoothing methods since they are simple to implement. Smoothing an image is accomplished by averaging the values of adjacent pixels, which reduces the variation between the pixel values in the picture.

Detection of edges (sharpening): High-pass filters may be used to enhance the sharpness of a photograph. For example, these filters, which are the inverse of the low-pass filter, emphasize the fine details in the picture; they are also known as high-pass filters. High-pass filtering, like low-pass filtering, is accomplished via the use of a convolution filter. However, instead of using the kernel of a low-pass filter, it makes use of a different one.

It is possible to utilize the first-order directional filter to calculate the first derivatives of an image using a directional filter of first order. The initial derivatives (or slopes) of a function are very apparent when there is a significant change between successive pixel values, as seen in the example. In certain cases, sound guides may be designed to steer sound in any direction within a specific region of interest.

When it comes to identifying the second derivatives of an image, Laplacian filtering, also known as Laplacian filtering, is a widely used technique. It performs its function by determining the rate at which the first derivatives change. This is necessary for determining if there is a change in adjacent pixel values as a result of an edge or a continuous progression. When using the Laplacian filter, negative values are often seen in the kernels, which are centered inside the array and spread out in a cross pattern. It is possible to have a corner with a value of zero or a positive integer. It is possible for the center value to be either a negative or a positive integer.

Image Segmentation

Image segmentation is a technique that is often used in the field of digital image processing and analysis. There are many different ways to split a picture into different parts or areas, and the most common method is to divide it depending on the characteristics of the pixels in the image. One of the methods used in picture segmentation is the separation of foreground and background pixels, as well as the grouping of pixels based on their similarity in color or shape.

Detecting the faces of individuals is a common use for this program.

- Radiography and magnetic resonance imaging
- Robotic vision, humanoid vision
- Unmanned Vehicle Autonomy

In the field of image segmentation, there are two kinds of techniques: supervised and unsupervised.

Because there is no need for contextual data, this thresholding method is easy to apply to non-contextual segmentation problems. A single threshold is applied to produce an image that includes both greyscale and color components, which may be used to create a binary picture of binary areas by combining them together. In the binary map, the two regions have potentially discontinuous borders, In both circumstances, pixels with input data values less than a threshold are included in one, while pixels with input data values equal to or greater than the threshold are included in the other. The thresholding methods described below are specific types of thresholding.

- Thresholding with a simple method
- Two: Adaptive thresholding
- Color thresholding

The Contextual Segmentation method divides pixels into groups based on a cutoff that does not take into account where the pixels are located on the image plane. Additionally, utilizing context-specific segmentation to distinguish different things is more successful than using general segmentation since it takes into account the closeness of pixels that belong to a single individual item. The two most important methods for contextual segmentation are those that are concerned with identifying discontinuities in data and those that are concerned with discovering commonalities between signals. Discontinuity-based methods attempt to build full borders that encompass fairly homogeneous areas by assuming abrupt signal changes on either side of the boundary on either side of the boundary. With this technique, the emphasis is on the similarities between pixels, and the goal is to create uniform regions by connecting pixels that meet specified similarity criteria. One technique has two sides that are mirror images of the other. For example: Using a total border, the two areas will be divided evenly in half. They were consuming the kind of Contextual segmentation described below.

- Connection in the form of pixels
- Region-to-region resemblance
- Rapidly expanding region
- Splitting and merging segmentation

In image analysis and computer vision, texture segmentation is a method for determining the features of a surface, such as its texture, that is extensively utilized. It is possible to divide the method categories developed for texture issues into four distinct groups.

1. An strategy based on structure
2. The use of statistical methods
3. Model-based approach to problem solving
4. The use of filters as a starting point

4. Fourier transform

Fourier Transforms are often employed in image processing for many purposes, such as decomposing an image into its sine and cosine components. Images translated to the spatial domain are the spatial domain equivalents of their frequency or Fourier-based counterparts. Fourier domain by a particular frequency, which may be discovered in the spatial domain image, may be found on each point in the image.

Image analysis, image filtering, picture reconstruction, and picture compression are some of the tasks that the Fourier Transform is used to do in a variety of fields, such as computer vision.

The DFT (Discrete Fourier Transform) is a sampled Fourier Transform, and as a result, it does not include all of the frequencies that make up an image, but rather a collection of samples large enough to properly describe the spatial domain picture. The total number of pixels in a spatial domain picture, which is also the same as the total number of pixels in a Fourier domain picture, will be obtained after taking into consideration all of the frequencies.

CHAPTER III

3. Technology Adapted

3.1 Introduction

During Chapter 2, the authors reviewed the past results that were relevant to this study and identified the most significant ideas, concepts, and techniques that might be used in this research in order to further the investigation. This chapter will explicitly outline the technology that will be utilized to address this issue.

3.2 Python.NET

This is the language in which the program is written. Python.NET (python net) is a Python package that enables interaction between Python programmers and the Microsoft.NET Framework 4.0+ Common Language Runtime (CLR) on Windows, as well as the Mono runtime on Linux and Mac OS X. Python.NET is a Python module that enables interaction between Python programmers and the Microsoft.NET Framework 4.0+ Common Language Runtime (CLR). Python.NET, a sophisticated application scripting tool, enables developers of.NET applications to create dynamic online applications. This package enables you to script.NET applications or creates whole programs entirely in Python while utilizing. Any language that targets the CLR can be used to create NET services and components (including C#, VB.NET, F#, C++/CLI, and other similar languages).

Consider Python is not implemented as a first-class CLR language in this package., meaning it is not transformed to managed code (IL) (as other packages do).Rather than the other way around, the Python engine is combined with either the.NET or Mono runtimes in this instance. While utilizing CLR services, may continue to utilize existing Python code and C-API extensions, and Python code will continue to execute at native Python execution rates. Individuals looking for a pure managed-code implementation of the Python programming language should look at the Iron Python project, which is currently in active development and well worth their time.

Python.NET in its current form is compatible with and has been tested with Python versions 2.7-3.8 as well as 3.6-x. The Python.NET website maintains a list of the most recent releases and offers links to those versions..NET mailing list, views the list's online archives, and unsubscribe from the list. The easiest method to report problems and defects is to use the Python.NET issue tracker.

3.3 SQLite

SQLite is a C-language library that provides a tiny and fast SQL database engine that is also self-contained, dependable, and feature-rich. SQLite is often known as SQLite for short. SQLite is the most commonly used database engine on the planet, with over a billion users. In addition to being incorporated into all mobile phones and most PCs, the SQLite database engine is packed into an infinite number of additional applications that people use on a daily basis. SQLite is a database engine that was developed by the company SQLite. There is more information accessible..

As stated by the SQLite file format's developers, They intend to keep the format stable, cross-platform, and backwards compatible until the year 2050. In addition to being extensively used as a container for transporting rich information between systems [1, 2, 3, 4], SQLite database files are also frequently utilized as a long-term data preservation format [4, 5]. There are now more than one trillion (1e12) SQLite databases in use [5, with the number growing on a daily basis] [5.

The SQLite source code is accessible in the public domain and may be used for any purpose by anyone at any time for any purpose without restriction..

https://www.sqlite.org/index.html

3.4 Tensor flow

In order to train and build the model, this was done. In this article, we will discuss TensorFlow, a machine learning software library that can be downloaded for free and without restrictions. In spite of the fact that it may be used to a wide range of applications, it is mainly concerned with the training and inference of deep neural networks. Dataflow and differentiable programming are the foundations of TensorFlow, a symbolic math toolbox that is a valuable tool for data scientists..

3.5 Android

This section is utilized to execute the mobile application. Android is a Google-developed mobile operating system that is built on top of a modified version of the Linux kernel and other open-source applications. It is primarily designed for touchscreen mobile devices such as smartphones and tablets.

3.6 HTML, PHP

PHP is a web programming language that is especially well convenient for the creation of web applications. It is extensively used.

3.7 Google core lab

Web application development is made easier using PHP, a general-purpose programming language that is particularly well suited for the production of web-based applications. A great deal of people utilizes it.

3.8 Google Drive

It was created as a cloud-based file storage and synchronization service for Google's users by the search engine behemoth. It was launched on April 24, 2012, and allows users to store files in the cloud (on Google's servers), synchronize data across many devices, and share files. Users may access their files from any computer or mobile device. In addition to a web-based interface, Google Drive offers apps that can be used offline on PCs running Windows and MacOS, as well as smartphones and tablets running Android and iOS. A collaborative editing environment is provided by Google Docs, Google Sheets, and Google Slides, which are all, part of the Google Docs Editors office suite. This allows for the collaborative editing of documents, spreadsheets, presentations (including PowerPoint), drawings (including 3D CAD models), forms, and other types of documentation. You may create and edit documents that are saved in Google Drive by using the Google Docs suite of tools.

CHAPTER IV

4. My Approach

4.1 Introduction

This chapter introduced the technologies that will be used to solve the research topic and detailed how to build an oil palm harvesting strategy. Building the proper module for identifying the Maturity stage of the collected data. I present my approach by high highlighting hypothesis, input, output, process, users, and features of the load cell prediction solution.

TensorFlow is a Python library implemented by the Google Brain team that is available as an open-source project. TensorFlow is a library that brings together a variety of techniques and models, allowing the user to create deep neural networks for use in tasks such as image identification and classification, as well as natural language processing. TensorFlow is free software. TensorFlow is a sophisticated framework that works by constructing a series of processing nodes, each of which represents a mathematical operation, and the whole series of nodes is referred to as a "graph."

In terms of Kera's, it is a high-level API (application programming interface) that may be used to access the functionalities of TensorFlow's underlying architecture (as well as other ML libraries like Theano). Kera's was created with the concepts of user-friendliness and modularity at the forefront of its design. The Kera's framework, in practice, makes it as easy as possible to implement the numerous powerful but sometimes complicated features of TensorFlow, and it's already set to operate with Python without the need for any further changes or configuration.

4.2 Hypothesis

The main hypothesis is to build a proper harvesting plan for harvest the oil palm crop at correct time. Apart from creation of harvesting plan, Identify the relationship among the time and color of the oil palm bunch. In Addition to that benefit to manger can allocate people and transport facility for harvesting.

4.3 Data Modeling

Experimentation and failure are frequent occurrences on the road map to machine learning success. In order to achieve the greatest accuracy possible, engineers and scientists who are fresh to the concept will constantly alter and update their algorithms and models.

Throughout this process, there may be problems faced, especially in the administration of data and the selection of the most suitable model.

For machine learning models to be developed, it is essential to understand that real-world data is imperfect and that different types of data need different techniques and tools. It is also important to understand that compromises will always exist when choosing the optimal machine learning model.

Experimentation and failure are frequent occurrences on the road map to machine learning success. In order to achieve the greatest accuracy possible, engineers and scientists who are fresh to the concept will constantly alter and update their algorithms and models. Throughout this process, there may be problems faced, especially in the administration of data and the selection of the most suitable model.

For machine learning models to be developed, it is essential to understand that real-world data is imperfect and that different types of data need different techniques and tools. It is also important to understand that compromises will always exist when choosing the optimal machine learning model.

4.4 Input

This record collection begins with 400 pictures and spans the period 2020-11-01 to 2020-12-31. It is divided into three categories. Row data was captured in JPG format with the assistance of highly skilled harvesters and filing offices.

4.5 Attribute selection

The color variations in bunches of grapes are used to identify the four main phases of maturing grapes.

Figure 1: Maturity stages of oil palm bunch

4.6 Data Pre-Processing Data

The concept of digital image processing is applied to digital pictures using computer algorithms. This is an example of image processing, which is a subset of image recognition. Conventional image processing has the same set of benefits and drawbacks as analog image processing. In contrast, digital image processing, which is a component of digital signal processing, provides many unique advantages. A consequence of this advancement is that algorithms used to process the incoming data may include a far wider range of options. Our goal is to enhance the picture data (features) using techniques such as blurring or sharpening in order to help our AI-Computer Vision models.

Developers may use the Kera's preprocessing layers API to build an input processing pipeline, which is supported natively by the Kera's platform. To get an idea of where they might be used, we'll say that non-procedures Kera's could be used as a separate preprocessing code, incorporated into the models created by Kera, or exported as a component of the Saved Model. In addition, they may be used in the Kera's Saved Model, which is an additional choice.

Models that are truly end-to-end are further built and exported by developers. All models handle feature normalization and feature value indexing by themselves. If provide a raw picture or raw structured data, you will get back a raw photo or raw structured data.

4.7 Input through GUI

Images are captured using an Android mobile smartphone and graphical user interfaces (GUIs).

4.8 Output

There is just one output per picture, which is selected from one of four categories. This output is the most essential since it allows you to estimate when each bunch will be harvested. Both outputs are delivered via a web-based solution.

4.9 Process

At the beginning of the process, the system will make use of the preprocessed data, and the tool will produce the outputs that have been suggested. The results are presented to the user in the form of a stage, and the data prediction idea is based on the Sequential model implemented in TensorFlow. Following its deployment in a live environment, this tool will produce outputs that are based on real-time information.

4.10 Users

There are primarily two kinds of users in a company: those who are employees and those who are not. As a result of this, implementation is linked to the acquisition of internal management decisions. Internal users include members of the operation management team, members of the estate management team, accountants, and internal decision-makers, among others.

4.11 Features

Aside from the functional needs, there are certain non-functional criteria, the most important of which is that the created tool should be capable of generating a web-based detail page for the user. In addition, all of the other criteria are met.

- Low cost
- User independent
- Browser independent
- Expandable
- High performance

CHAPTER V

5. Data analysis and Design

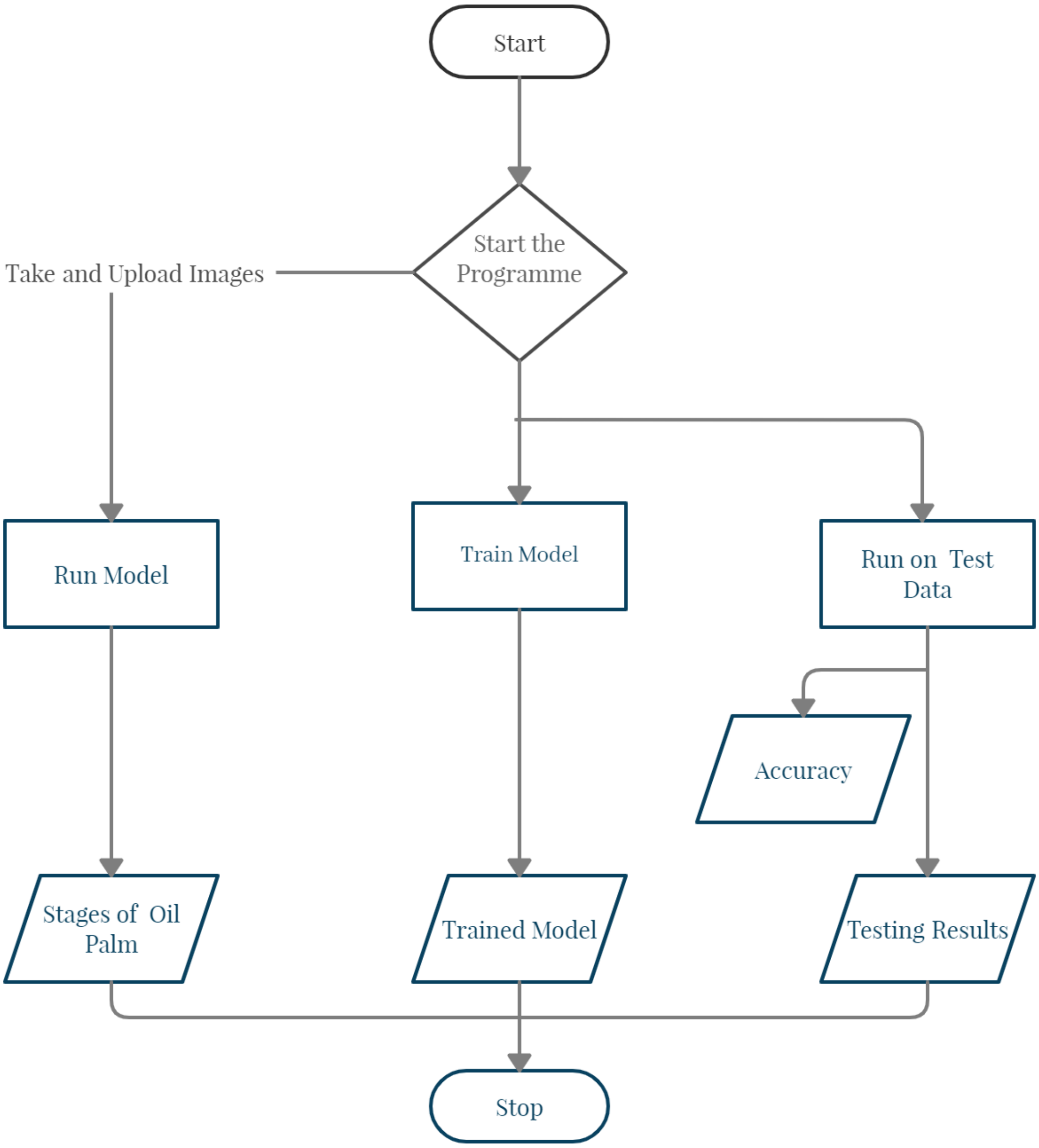

Figure 2 : Flow chart for interface design

5.1 Image classification

This section demonstrates how to categorize photographs of bunches. It uses a keras.Sequential model to generate an image classifier and preprocessing to load data into it. Image dataset from directory preprocessing function. Finding overfitting and implementing methods to minimize it, such as data augmentation and Dropout, are important tasks.

Fundamental machine learning process:

1. Create an input pipeline
2. Create the model
3. Train the model
4. Test the model
5. Improve the model and go through the process again.

Normally tensor flow contains following libraries.

```
import matplotlib.pyplot as plt
import numpy as np
import os
import PIL
import tensorflow as t
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.models
```

As per this study, Sequential models, for example, can be employed successfully in the case of a basic stack of layers with exactly one input tensor and one output tensor. Sequential models may be utilized successfully in the situation of a basic stack of layers where each layer has exactly one input tensor and one output tensor, for example. Because a Sequential model cannot accommodate large numbers of inputs or many outputs, it is not the ideal choice for many applications. Depending on the setup, any of layers may have a large number of inputs and/or a large number of outputs.

5.2.1 Creation of data set

The usage of a validation split while building our model is a recommended practice.

Suitable to use buffered prefetching to produce data from disk in the absence of causing the pipeline's input and output to become a bottleneck. When it comes to loading data, there are two essential techniques to consider.

After the pictures are loaded from disk during the first epoch, Dataset.cache() maintains them in memory until the next epoch. It supports prevent the dataset from becoming a bottleneck during the training of model. In the event that the selected dataset is too large to fit in memory, this technique can also be used to establish a performant on-disk cache.

Some parameters have been defined to use with loader:

- target_size=(300, 300),
- batch_size=16,

As per this study, Here can be mainly identified four categories, so 'catogorical ' class mode is used for dataset.

```
:
path = r'D:\Personal\Notebooks\Menaka\oil_1'
# All images will be rescaled by 1./255
train_datagen = ImageDataGenerator(rescale=1/255)

# Flow training images in batches of 128 using train_datagen generator
train_generator = train_datagen.flow_from_directory(
        path,  # This is the source directory for training images
        target_size=(300, 300),  # All images will be resized to 300x300
        batch_size=16,
        color_mode="rgb",
        # Since we use binary_crossentropy loss, we need binary labels
        class_mode='categorical')


path = r'D:\Personal\Notebooks\Menaka\oil_2'
# All images will be rescaled by 1./255
val_datagen = ImageDataGenerator(rescale=1/255)

# Flow training images in batches of 128 using train_datagen generator
val_generator = val_datagen.flow_from_directory(
        path,  # This is the source directory for training images
        target_size=(300, 300),  # All images will be resized to 300x300
        batch_size=16,
        color_mode="rgb",
        # Since we use binary_crossentropy loss, we need binary labels
        class_mode='categorical')

Found 320 images belonging to 4 classes.
Found 80 images belonging to 4 classes.
```

Figure 3: Data set and categorical classes

5.2.2 Dataset

The prefetch() function, which is used during training, overlaps data preparation and model execution.

5.2.3 Standardize the data

The values of the RGB channels are in the [0, 255] range. A neural network is not designed to operate in this manner, and strive to keep model's input values modest. By utilizing a Rescaling layer, you will be able to normalize values to be in the [0, 1] range.

In this project, I adhere to the procedures outlined below.

- from tensorflow.keras.optimizers import RMSprop
- from keras.preprocessing import image
- from tensorflow.keras.preprocessing.image import ImageDataGenerator

```
# All images will be rescaled by 1./255
train_datagen = ImageDataGenerator(rescale=1/255)
```

Figure 4: Standardizing the data set

5.2.4 Creation of the model

The model is made up of five convolution blocks, each having a max pool layer and ‘relu’ activation function is used to activate a completely connected layer with 512 units on top of it. The purpose of this project was to build an appropriate model for this study; hence it wasn't tweaked for high accuracy.

```
In [2]: model = tf.keras.models.Sequential([
            # Input shape is the desired size of the image 300x300 with 3 bytes color
            # This is the first convolution
            tf.keras.layers.Conv2D(16, (3,3), activation='relu', input_shape=(300, 300, 3)),
            tf.keras.layers.MaxPooling2D(2, 2),
            # The second convolution
            tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
            tf.keras.layers.MaxPooling2D(2,2),
            # The third convolution
            tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.MaxPooling2D(2,2),
            # The fourth convolution
            tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.MaxPooling2D(2,2),
            # The fifth convolution
            tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
            tf.keras.layers.MaxPooling2D(2,2),
            # Flatten the results to feed into a DNN
            tf.keras.layers.Flatten(),
            # 512 neuron hidden layer
            tf.keras.layers.Dense(512, activation='relu'),
            # Only 1 output neuron. It will contain a value from 0-1 where 0 for 1 class ('horses') and 1 for the other ('humans')
            tf.keras.layers.Dense(4, activation='softmax')
        ])
```

Figure 5: Sequential Model

Using the model's summary technique, then can visualize all of the network's layers as follows.

```
In [8]: # Getting the summery of the model (Model structure)
        model.summary()

Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_5 (Conv2D)            (None, 298, 298, 16)      448
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 149, 149, 16)      0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 147, 147, 32)      4640
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 73, 73, 32)        0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 71, 71, 64)        18496
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 35, 35, 64)        0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 33, 33, 64)        36928
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 16, 16, 64)        0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 14, 14, 64)        36928
_________________________________________________________________
max_pooling2d_9 (MaxPooling2 (None, 7, 7, 64)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 3136)              0
_________________________________________________________________
dense_2 (Dense)              (None, 512)               1606144
_________________________________________________________________
dense_3 (Dense)              (None, 4)                 2052
=================================================================
Total params: 1,705,636
Trainable params: 1,705,636
Non-trainable params: 0
_________________________________________________________________
```

Figure 6: Sequential Model summary

5.2.5 Compile the model

RMS prop algorithm has been utilized as an optimizer in the module compilation process in order to optimize the Neural network. In order to calculate the loss between the labels and predictions, the Categorical Cross entropy function is utilized. With the metrics parameter, you can view the training and validation accuracy for each training period for each training period.

```
In [9]: # Compiling the model please che
model.compile(loss='categorical_crossentropy',
              optimizer=RMSprop(lr=0.001),
              metrics=['accuracy'])
```

Figure 7: Compilation of the model

5.2.6 Training and visualize the data

Following the trains, the accuracy of the model has increased to 0.97.

Let's say that 80 percent of the pictures are used for training and 20 percent are used for validation.

```
In [50]:
epochs=15
verbose=1
steps_per_epoch=8
history = model.fit(
      train_generator,
      validation_data=val_generator,
      epochs=epochs)

Epoch 1/15
22/22 [==============================] - 213s 10s/step - loss: 1.3602 - accuracy: 0.3735 - val_loss: 1.0211 - val_acc
uracy: 0.5750
Epoch 2/15
22/22 [==============================] - 211s 10s/step - loss: 1.0003 - accuracy: 0.5618 - val_loss: 0.6987 - val_acc
uracy: 0.8250
Epoch 3/15
22/22 [==============================] - 205s 9s/step - loss: 0.6583 - accuracy: 0.7441 - val_loss: 1.0862 - val_accu
racy: 0.7250
Epoch 4/15
22/22 [==============================] - 204s 9s/step - loss: 0.5170 - accuracy: 0.8176 - val_loss: 0.4478 - val_accu
racy: 0.8625
Epoch 5/15
22/22 [==============================] - 205s 9s/step - loss: 0.4625 - accuracy: 0.8324 - val_loss: 0.3504 - val_accu
racy: 0.8875
Epoch 6/15
22/22 [==============================] - 200s 9s/step - loss: 0.4533 - accuracy: 0.8529 - val_loss: 0.4320 - val_accu
racy: 0.8500
Epoch 7/15
22/22 [==============================] - 200s 9s/step - loss: 0.3136 - accuracy: 0.8971 - val_loss: 0.8351 - val_accu
racy: 0.8625
Epoch 8/15
22/22 [==============================] - 203s 9s/step - loss: 0.4023 - accuracy: 0.8882 - val_loss: 0.7157 - val_accu
racy: 0.8500
Epoch 9/15
22/22 [==============================] - 218s 10s/step - loss: 0.2838 - accuracy: 0.8882 - val_loss: 0.5654 - val_acc
uracy: 0.8750
Epoch 10/15
22/22 [==============================] - 226s 10s/step - loss: 0.2136 - accuracy: 0.9324 - val_loss: 0.3991 - val_acc
uracy: 0.8750
Epoch 11/15
22/22 [==============================] - 202s 9s/step - loss: 0.2779 - accuracy: 0.8912 - val_loss: 0.5837 - val_accu
racy: 0.8750
Epoch 12/15
22/22 [==============================] - 201s 9s/step - loss: 0.1753 - accuracy: 0.9412 - val_loss: 0.6035 - val_accu
racy: 0.7625
Epoch 13/15
22/22 [==============================] - 198s 9s/step - loss: 0.1439 - accuracy: 0.9471 - val_loss: 0.4866 - val_accu
racy: 0.9000
Epoch 14/15
22/22 [==============================] - 201s 9s/step - loss: 0.3392 - accuracy: 0.8971 - val_loss: 0.8025 - val_accu
racy: 0.8500
Epoch 15/15
22/22 [==============================] - 196s 9s/step - loss: 0.1052 - accuracy: 0.9676 - val_loss: 0.4556 - val_accu
racy: 0.9000
```

Figure 8: Training the model

After the training the model, the model accuracy increased and loss of has been decreased, As follows.

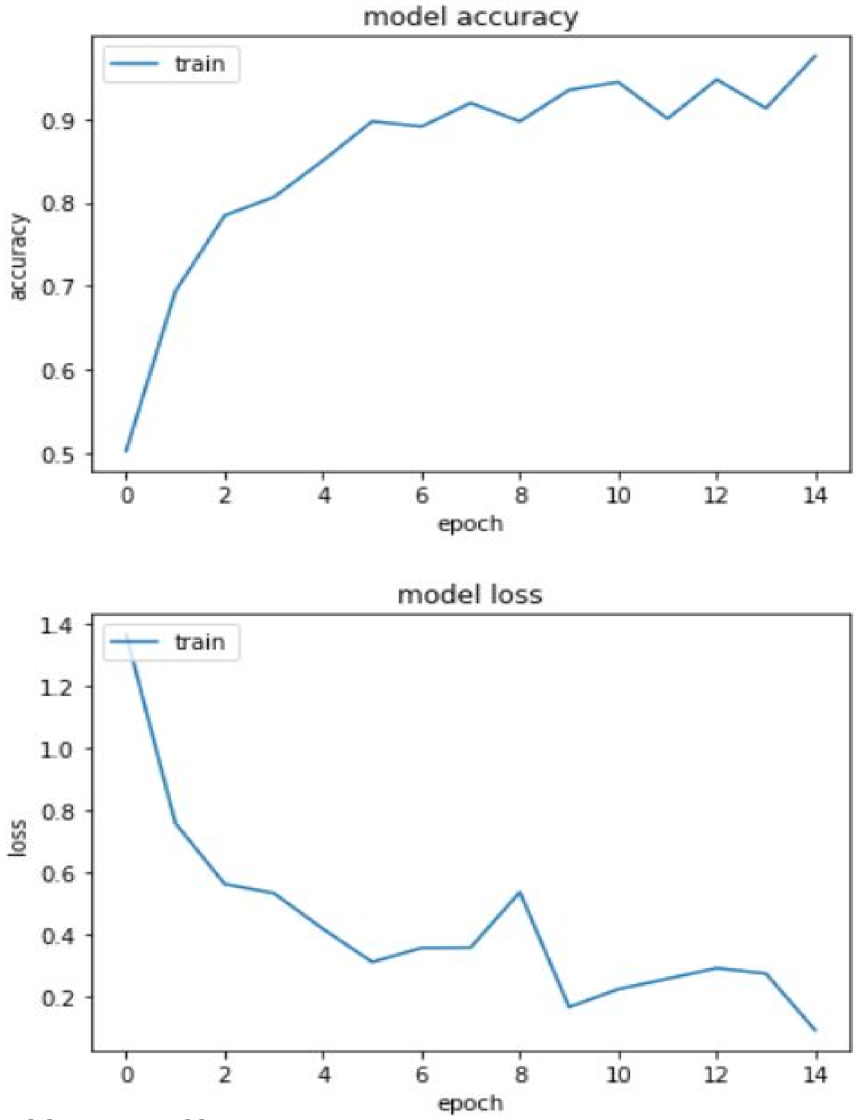

Figure 9: Model accuracy and loss

5.2.7 Evaluation the model

Using a validation split when developing our model is an excellent technique to follow. Following the training and validation of the model, the validation accuracy has improved to up to 90 percent, and the validation loss has been reduced to a minimum.

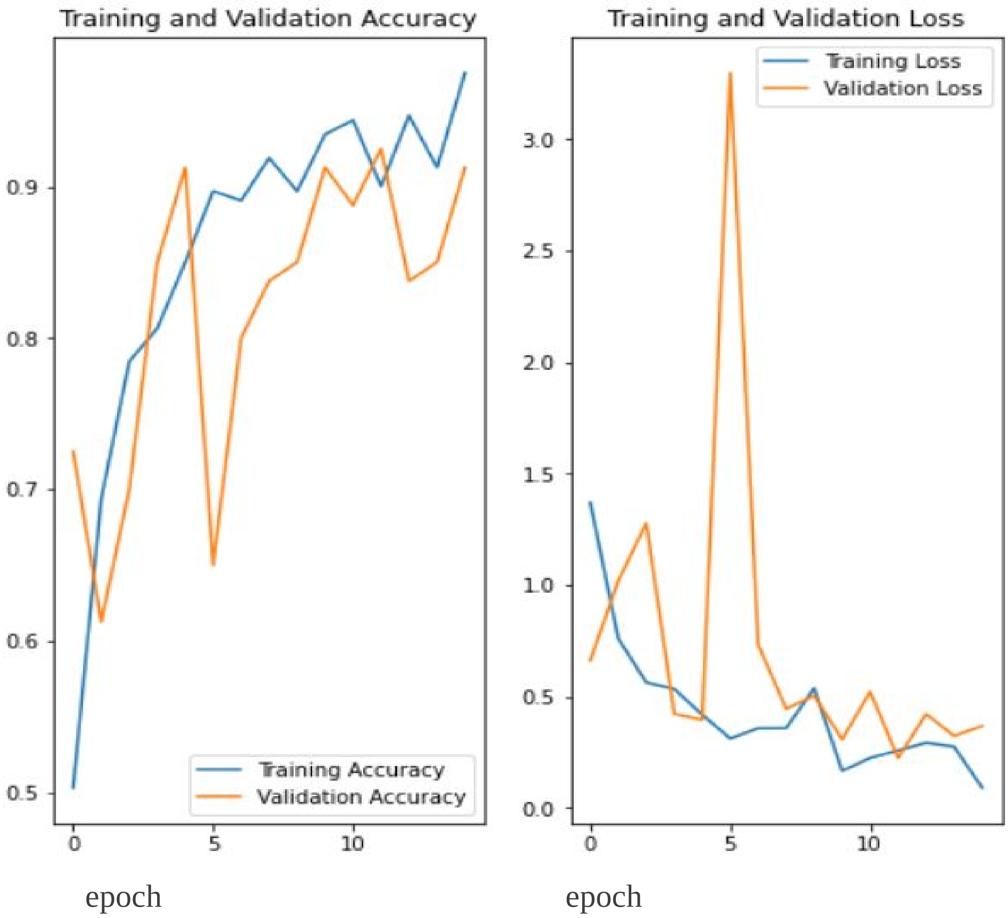

Figure 10: Model validation

Required to minimize the failures of model and need to increase the overall performance of model by identifying and correcting the issues. According to the training accuracy and validation accuracy, this model is suitable for predicting the maturity phases of the organism.

5.2.8 Data prediction

Finally, the model can be used for test to categorize images that were not part of the training or validation data sets.

```
# predicting images
path = '2.jpg'
img = image.load_img(path, target_size=(300, 300))
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)

images = np.vstack([x])
classes = model.predict(images, batch_size=10)
print('Palm tree is on type', np.argmax(classes, axis=1)[0], 'stage')

Palm tree is on type 2 stage
```

Figure 11: Prediction of images

Out put	Maturity Stage
O stage	Before 16 days
1 Stage	Before 12
3 stage	Before 8
4 stage	Before 4

Table: 01 Relationship among output and maturity stage

Following the results of the above table, the process may be divided into four phases, which are as follows: If the picture is at stage zero, it indicates that there are still 16 days left to harvest oil palms. If the recorded picture in stage one shows that you still have 12 days to harvest the specific tree, the system indicates that you have 12 days to harvest the tree. There are still 8 days left to harvest the picture if it is in the third stage. Final step is to inform that you are ready to harvest the oil palm fruit for that tree if the picture appears in stage four.

5.2.10 Integration with data base

After analyzing the fetched data, The trained model has been used save into a database file. For that purpose, following Tensoflow libraries have been used.

- import tensorflow as tf
- import pandas as pd
- import numpy as np

- from keras.preprocessing import image
- import sqlite3
- import os

The data base integration part is shows in Figure 12.

```
In [1]: import tensorflow as tf
        import pandas as pd
        import numpy as np
        from keras.preprocessing import image
        import sqlite3
        import os

In [2]: new_model = tf.keras.models.load_model('palmClassification.h5')

In [3]: conn = sqlite3.connect('palmTree.db')
        c = conn.cursor()

        c.execute(''' delete from Predictions  ''')

        #create table
        c.execute('''CREATE TABLE IF NOT EXISTS Predictions
                     (imageName text, predictedCatagory real)''')

Out[3]: <sqlite3.Cursor at 0x25d9b26b730>

In [4]: for img_path in os.listdir('TestImages'):
            path = f'TestImages\{img_path}'
            img = image.load_img(path, target_size=(300, 300))
            x = image.img_to_array(img)
            x = np.expand_dims(x, axis=0)

            images = np.vstack([x])
            classes = new_model.predict(images, batch_size=10)
            prediction = np.argmax(classes, axis=1)[0]
            print('Palm tree is on type', prediction, 'stage')

In [3]: conn = sqlite3.connect('palmTree.db')
        c = conn.cursor()

        c.execute(''' delete from Predictions  ''')

        #create table
        c.execute('''CREATE TABLE IF NOT EXISTS Predictions
                     (imageName text, predictedCatagory real)''')

Out[3]: <sqlite3.Cursor at 0x25d9b26b730>

In [4]: for img_path in os.listdir('TestImages'):
            path = f'TestImages\{img_path}'
            img = image.load_img(path, target_size=(300, 300))
            x = image.img_to_array(img)
            x = np.expand_dims(x, axis=0)

            images = np.vstack([x])
            classes = new_model.predict(images, batch_size=10)
            prediction = np.argmax(classes, axis=1)[0]
            print('Palm tree is on type', prediction, 'stage')

            # Inserting to the database
            c.execute("insert into Predictions (imageName, predictedCatagory) values (?, ?)",
                      (img_path, int(prediction)))
            conn.commit()

Palm tree is on type 1 stage
Palm tree is on type 1 stage
```

Figure 9: Data base integration

CHAPTER VI

6. Implementation

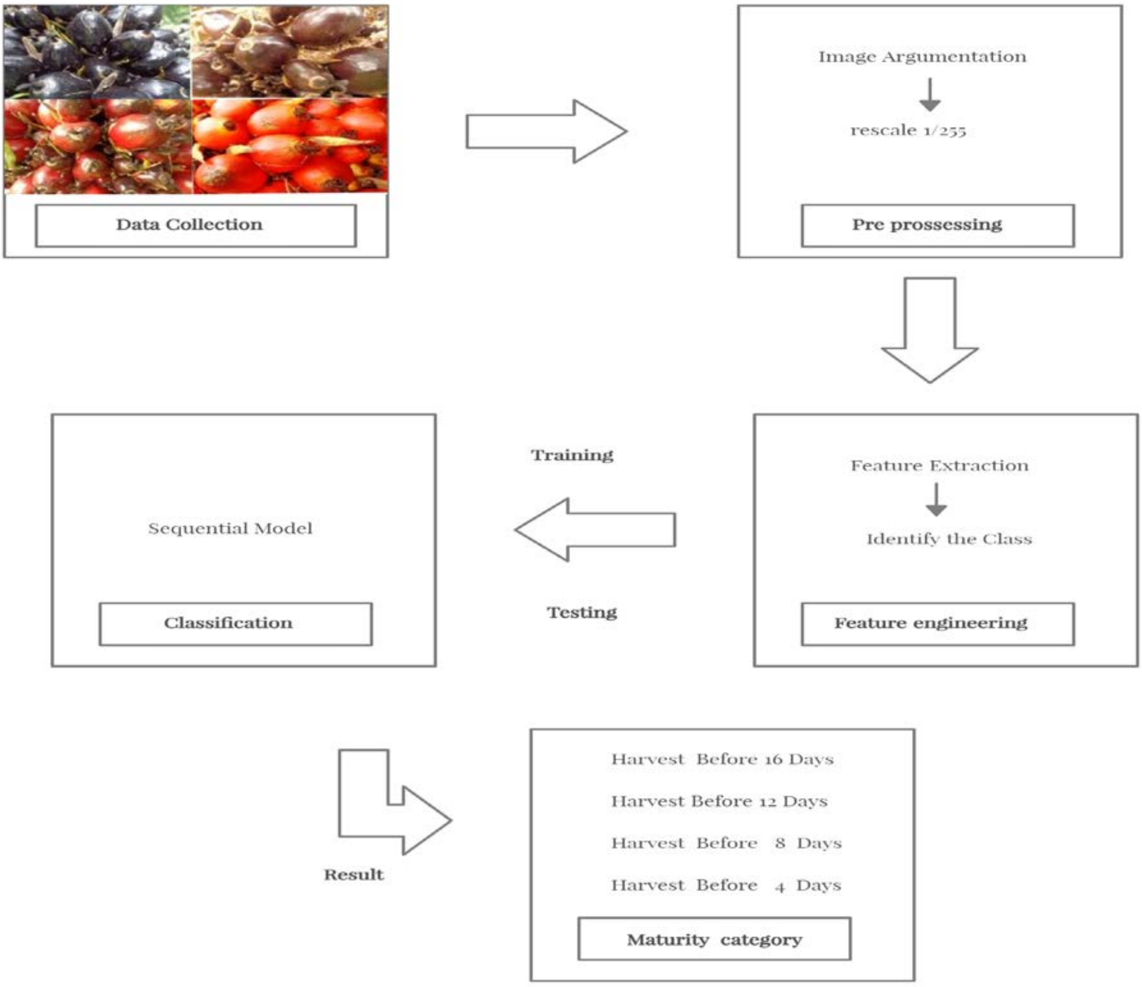

Figure 10: Implementation chart

6.1 Numbering the trees by Using QR Code

It is necessary to number each tree in order to make identification easier as the initial stage in the implementation process. In order to do this, I have created a QR code for each tree that contains the tree number, field number, and division number, as seen in the picture below. In the digital world, a QR code is a kind of barcode that can be read quickly by a digital device and that encodes information as an array of pixels in a square-shaped grid of pixels.

6.2 Taking images by using android application

The second stage is to gather all of the information we can on the oil palm trees in the field. I've created an Android application that will capture QR codes and gather data.. After scanning the QR code, the app will immediately launch the camera, allowing you to take pictures of the oil palm bunches as they are to be harvested. Then we'll be able to photograph each individual bunch of oil palm.

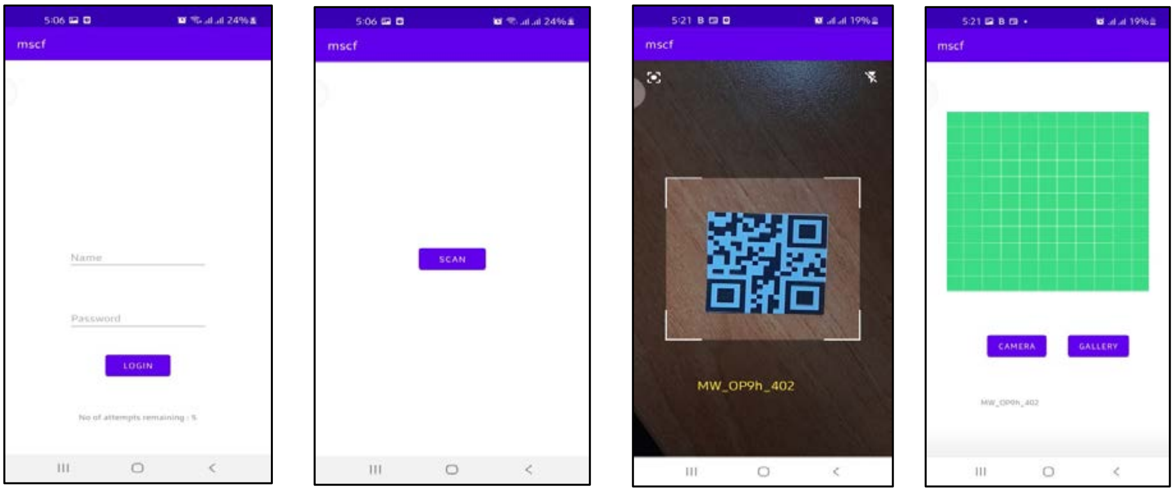

Figure 11: GUI's of image collection application

6.3 Sync images to web drive

The taken images can be synced with Google Drive. Google Drive is a cloud-based file storage and syncing service created by the search engine giant Google.

6.4 Processing images.

As fourth step after uploading all of the captured photos to Google Drive, you may start analyzing and standardizing them. We will process all of the pictures via Google Colab IDE and analyze them using the TensorFlow frame work and Python.

6.5 Sync produced data file with local PC

After Google Colab has finished analyzing all of the pictures, it will update the data base file that has been produced onto the local PC via Google Drive. A database is a collection of data that has been structured in such a way that it is easily accessible and managed, allowing for fast and simple data access. Database files are data files that are used to keep track of image details and maturity stages.

6.4 Displaying the harvesting plan

It will then use web technology to display the information from the database that has been created as a final step. In computer networking, web technology refers to the method through which computers interact with one another via the use of markup languages and multimedia applications. It provides us with a means of interacting with hosted information, such as webpages.

Logout

mm/dd/yyyy mm/dd/yyyy submit

Opened database successfully

Division	Field Number	Tree Number	Estimated Harvest Date
MW	0130c	003	2021-07-17
MW	0130c	012	2021-07-17
MW	0130c	050	2021-07-17
MW	0130c	061	2021-07-21
MW	0130c	075	2021-07-21
MW	0130c	085	2021-07-17
MW	0130c	110	2021-07-21
MW	0130c	110	2021-07-17
MW	0130c	118	2021-07-21
MW	0130c	120	2021-07-17
MW	0130c	120	2021-07-17
MW	0130c	131	2021-07-21
MW	0130c	140	2021-07-21
MW	0130c	162	2021-07-13
MW	0130c	170	2021-07-21
MW	0130c	177	2021-07-17
MW	0130c	178	2021-07-17
MW	0130c	182	2021-07-21
MW	0130c	192	2021-07-21
MW	0130c	195	2021-07-21
MW	0130c	195	2021-07-17
MW	0130c	195	2021-07-21
MW	0130c	200	2021-07-21
MW	0130c	205	2021-07-21

Figure 12: Dash board

CHAPTER VII

7. Conclusion

7.1 Limitations

Because I believe the maturity color and period differs from geographical area to area, I was subjected to certain limitations in my ability to do this task. And this must alter in response to changes in the climate, soil, NPK input and clone.

7.2 Further works

It is necessary to develop a Android application detect oil palm bunch only. Consequently, it is beneficial to the input validation section. It was also recommended to integrate the output findings with by evaluating the constraints stated above.

References

- Font, D.; Palleja, T.; Tresanchez, M.; Teixido, M.; Palacin, J. Preliminary study on color based nectarine variety classification. Proceedings of the 2012 IEEE International Instrumentation and Measurement Technology Conference (2012 IEEE I2MTC), 2187-2190, Graz, Austria, May 13-16, 2012. ISBN: 978-1-4577-1771-0.
- HAHN, F. Multi-spectral prediction of unripe tomatoes, **Biosystems Engineering**, Bedford, v.81, n.2, p.147-155, 2002.
- JAHNS, G.; NIELSEN, H. M.; WOLFGANG, P. Measuring image analysis attributes and modeling fuzzy consumer aspects for tomato quality grading. **Computers and Electronics in Agriculture**, Davis, v.31, p.17-29, 2001.
- KADER, A. A. Standardization and inspection of fresh fruits and vegetables. In: KADER, A. A. **Postharvest technology of horticultural crops**. 3. ed. Davis: University of California, 2002. p. 287-289.
- KONDO, N., TING, K. C. **Robotics in bioproduction systems**. ed. St. Joseph: ASAE, The Society for engineering in agricultural, food, and biological systems, 1998. 325p.
- PAULUS, I., SCHREVENS, E. Evaluating and modelling the size classifcation of apples. **Journal of Agricultural Engineering Research**, Bedford, v. 74, p. 411-419, 1999.
- PBMH - PROGRAMA BRASILEIRO PARA A MODERNIZAÇÃO DA HORTICULTURA. **Normas de classificação do limão (Lima ácida) Tahiti**. São Paulo: CEAGESP, 2000.
- PBMH - PROGRAMA BRASILEIRO PARA A MODERNIZAÇÃO DA HORTICULTURA. **Normas de classificação de Tomate. São Paulo**: CEAGESP, 2004.
- POLDER, G., VAN DER HEIJDEN, G. W. A. M.; YOUNG, I.T. Hyperspectral image analysis for measuring ripeness of tomatoes. In: ASAE International Meeting, 2000, Milwaukee. **Proceedings...** Milwaukee: ASAE, 2000, Paper Nº 003089.
- POLDER, G., VAN DER HEIJDEN, G. W. A. M.; YOUNG, I.T. Spectral image analysis for Measuring ripeness of tomatoes. **Transactions of the ASAE,** St. Joseph, v. 45, n. 4, p. 1155-1161, 2002.
- POLDER, G., VAN DER HEIJDEN, G. W. A. M.; YOUNG, I.T. Tomato sorting using independent component analysis on spectral images. **Real-Time Imaging**, London, v.9, p.253-259, 2003.
- SARKAR, N.; WOLFE, R. R. Computer vision based system for quality separator of

fresh-market tomatoes. **Transactions of the ASAE,** St. Joseph, v. 28, n. 5, p. 1714-1718, 1985

- TEOH, C. C., SYAIFUDIN, A. R. M. Use of image analysis for grading size of mango. **Acta horticulturae**, Leuven, v.710, p.485-490, 2006.
- VAN DER HEIJDEN, G. W. A. M., POLDER, G., GEVERS T. Comparison of multispectral images across the Internet. In: SPIE: Internet Imaging, 2000, Washington. **Proceedings...** Washington: SPIE, 2000. v.3964, p.196–206.
- VON BECMANN, J. W.; BULLEY, N. R. Electronic sizer and color grader for tomatoes. **Transactions of the ASAE**, St. Joseph, v.21, n.1, p.25-30, 1978.
- YIBIN, Y. New method to identify shape, size and surface defect of huanghua Pear with machine vision-Part–: Development of hardware and image preprocessing method. In: International Conference on Agricultural Engineering, 99, 1999, Beijing. **Proceedings...** Beijing: Natural Science Funds of China, 1999. p.79-86.
- Moreno, J.; Tresanchez, M.; Palleja, T.; Teixido, M.; Font, D.; Palacin, J. Desarrollo de un robot cartesiano de uso docente con 5 grados de libertad y una camara en la pinza. Annual seminary of Automatic, Industrial Electronic and Instrumentation (SAAEI 2013), Madrid, Spain, 10-12 July, 2