G52GRP Final Group Report

Semi-automated Sorting System for Germinated Oil Palm Seeds

28th April 2022
Group ID: 7b

Name	Student ID	Username
Alden Sia Zheng Heng	20196637	hfyas6
Yoong Jia Yin	20218749	hfyjy5
Justin Leuk Khen Wah	20216554	hfyjl13
Ku Moeun	20181569	saymk7
Yan Chee Chung	20203792	hfycy3
Gerryn Goh Jo-Vee	20129220	hfygg2
Hoh Waii Kong	20113452	hcywh2

Supervisor: Dr Iman Yi Liao

Abstract
The increasing demand for palm oil worldwide over the past 50 years has caused the rapid rise of palm oil manufacturing [1]. Therefore, the quality testing of palm oil seeds is essential to produce high-quality palm oil trees with high yields. Traditionally, palm oil seeds are quality checked and sorted manually by human workers. This leads to inevitable human error which will cost the industry more money and time. In this project, we will be focusing on producing a system that is able to classify the quality of palm oil seeds as good or bad using artificial intelligence (AI). As palm oil seeds are delicate and fragile in nature, they need to be handled with utmost care especially during sorting. The system we have designed will be semi-automatic, whereby human workers are required to handle some parts of the operation such while the rest will be automated by machines. Unlike traditional and conventional quality checking and sorting methods, our system aims to replace humans in the process of seed classification with a trained AI model. As humans lose productivity over long working hours, the speed and accuracy of the seed quality checking will be affected. To counteract this, our system will only require humans to operate the arrangement of seeds on trays, the loading of trays onto the conveyor belt, the sorting of seeds using AI prediction results and the unloading of the trays. By implementing hardware and software components into our system, we were able to build a system to prove the concept of a semi-automated sorting system that can replace the traditional fully manual sorting system.

Table of Contents

1. Introduction

1.1 Purpose

The objective of this project is to create a prototype sorting system that can take an image of a tray of seeds on a conveyor belt, predict the condition of the seeds and display the results of the prediction. This prototype is aimed to prove the system's concept of design and to be expanded by our client. The purpose of the system is so that the workers are able to easily replace the bad seed with a good seed based on the result. The sorting system is meant for quality control right before the seeds are sent out to the customers.

1.2 Client: Advanced Agriecological Research Sdn. Bhd. (AAR)

Advanced Agriecological Research Sdn. Bhd. is a wholly-owned subsidiary of Applied Agricultural Resources Sdn. Bhd. AAR has emerged as a leading tropical plantation crop research centre since its establishment in July 1986. ARR has more than 65 research executives and 650 dedicated staff members and workers currently serving more than 400,000 hectares of oil palm estates belonging to their Principals and commercial clients in Malaysia, Indonesia, Liberia, and Colombia. [2]

1.3 Project Scope and Deliverables

Our project involves the development of semi-automated germinated oil palm seeds. It is a quality control system which involves Image Processing, Machine Learning, Human-Computer Interaction, Edge devices, hardware and electronic components. The project is focusing on constructing a small-scale prototype that can be implemented for seeds classification based on its quality and workers can sort the seeds manually based on the seed prediction with a laser illumination system.

1.4 Background Research

1.4.1 Existing Solutions

1.4.1.1 Manual seed sorting system

The AAR company since its establishment has been using the traditional method of sorting palm oil seeds using human workers as part of their quality control process. The trained workers will be sorting and identifying these seeds based on their training and experience in sighting the physical condition of the seeds. There should be sufficient human workers available to increase the production of sorted seeds, and these human workers will have to be trained to ensure the quality of these seeds is monitored and maintained. The disadvantages of such a manual process would be affected by human behaviours and social interactions both at home and during work. Therefore the welfare and health of the human workers are equally important for producing a good yield of quality seeds.

1.4.1.2 Egg sorting system using Computer Vision

According to 8 students from the Department of Energy Saving and Automation, Kazakh National Agrarian University and the Department of Automatics and Mechatronics, University of Ruse [3], human workers sorting eggs manually may cause human errors or inaccuracy while detecting whether the eggs have any defects or identify the weight of eggs, and working manually may cause the reduction of productivity to less than 1000 eggs per hour. Therefore, the only solution for most manufacturers is to use machines and different techniques to sort eggs based on different parameters. Artificial Vision, which is part of the Computer Vision, consists of an Image Processing Algorithm that can analyse and calculate the shape of the eggs in order to be passed in as a geometric parameter to classify the defective eggs from the non-defective ones. This technique not only improves the production rate but also provides high work efficiency and high accuracy in identifying the quality of eggs. Whereas another technique called the Regression Analysis (not part of Computer Vision) can also take in the geometric parameters (the shape of eggs) calculated by the Image Processing Algorithm to sort the eggs based on their weight.

1.4.1.3 Agricultural products sorting systems using Image Processing Algorithm

The quality of agricultural products are graded and classified using the Image Processing Algorithm, based on their different inner (sizes, colours, shapes, etc.) and outer parameters (sweetness, acidity, freshness, etc.). According to Ambrose et al. (2016), the Image Processing Algorithm consists of a technique called the Partial Least Squares - Discriminant Analysis (PLS-DA) that is applied for the classification of treated and untreated corn seeds.[4] Another implementation of the Image Processing Algorithm includes the use of the Image Segmentation technique and Otsu's Method [5] in the sorting and grading of apples in a multi-coloured channel space, where Otsu's Method is used for performing image thresholding automatically from the pixel values distributed [6]. Other than that, there is a system that is developed for sorting and grading long-shaped fruit, such as the cucumber. [7] This kind of system utilises not only the Image Processing Algorithm but also the Neural Network technology, where after being trained with different sample fruit patterns or shape objects, other kinds of fruits could also be classified no matter what shape the fruit is. In a ginseng sorting system, the shape and weight of ginseng are parameters taken in by a Weight-Estimating Algorithm, and a Shape Analysis Algorithm using an Image Processing Algorithm. The ginsengs are divided into Grade 1, Grade 2, and Grade 3 after being analysed [8]. Through implementing the Image Processing Algorithm, the quality of products is improved and more time is saved, where the speed of the sorting system increases and the image analysis operates efficiently.

1.4.2 Literature Review

In the production line, sorting is one of the important tasks as it will affect the homogeneity of products [9]. Jahanbakhshi et al. reported that the shape of carrot non-homogeneity remains in the market for a long time leading to an increase in material loss. Even the undesirable shapes of carrots do not have any problems with their nutritional properties and taste, but customers commonly do not reach out to them in the market. Therefore, an appropriate method must be used for sorting and packaging carrots to increase their marketability and reduce wastage [10]. In the past, qualitative evaluation of agricultural products was done by the eyes and hands of human inspectors. However, the performance is slow and the cost keeps increasing with the increase in consumers' demands. Hence, the need to have higher quality products and faster sorting procedures has appeared.

1.4.2.1 Sorting aspect

Many scholars have been interested in the study related to the sorting system of agricultural products using image processing to increase productivity. With the development and application of image analysis and computer machine vision, the sorting method of agricultural products and efficiency are getting improved. There are many aspects to grading the quality of a product through outer and inner parameters. Most of the sorting aspect is through outer parameters such as size, colour intensity, shape, stem identification, surface texture and mass [7]. For example, in a study by Khojastehnazhand et al., an image processing technique is used to estimate citrus fruits' physical attributes such as diameters, volume, mass and surface area [11].

The study of M G NORSAZWAN, et. al. evaluates different oil palm seed colours (black, semi-white and white) during germination and nursery evaluation and reported seed colour does not implicate abnormality. To date, some practices remove the semi-white and white seeds, but no scientific studies conducted to determine the performance of white seeds as compared with the black coloured seeds in terms of their germination, as well as seedling growth and development. Overall, the colour of oil palm seed is not related to germination capacity and seedling performance in field planting. The quality of white seeds in terms of germination and seedling characteristics are equal to that of black seeds. Therefore, colour is not an important aspect for us to determine the quality of oil palm seed. [12] The study by Pornsuriya et al., 2013 reported oil palm are faced with several diseases including seed rot and brown germ during the seed stage. Brown germ is a disease with symptoms that appear as the seed germinates. Normally, light brown spots develop on the elongating radical as it emerges through the micropyle and these lesions merge to produce a solid discolouration of its tip. Seeds affected by the brown germ typically have a stubby radical with brown lesions mostly midway on the plumule and radical [13]. Hence, the outer parameters are the sorting aspect of our system through taking pictures of the germinated seeds to observe the seeds' radicals.

Unlike oil palm seed, the colour and size of lemon are the features for sorting fruit. The study inKhojastehnazh et al. developed a grading of the lemon system and implemented it in a visual basic environment which captured the surface of lemons. To provide uniform lighting, well illuminated and free from shadows images of fruit, four fluorescent tubes were placed and adjusted above the conveyor. White cardboard was used as a background to simplify the segmentation part later for removing the background in the later image processing session. An image of the background is captured to remove the background from fruit images. The R, G and B values from all pixels of background fruit images are calculated and stored in the database. Once the image of the fruit is captured, The RGB values in the fruit image will be compared with all the pixel values of the background image available in the database. All these RGB values and estimated volumes of lemon are extracted and stored in a database. The final grade of the passing fruit is by comparing the information during the sorting phase with the available information inside the database. [11] Seed sorting systems do not sort depending on seed colour. Therefore, no background picture has been taken to compare with the image of seeds. However, the background used for the sorting system must be the same as the background used to train the sorting system program.

1.4.2.2 Infrared (IR) Sensor for seed detection

Infrared (IR) sensors are used in the system to detect the presence of a seed object by sensing a kind of thermal radiation (which is invisible to our eyes) emitted from the seed. IR sensors work by sending out light rays to the target seed object, which then reflects the light rays to the sensor. If the reflected light rays are detected, it indicates that the seed object is present. For example, a strawberry sorting system used a photoelectric sensor, which uses Infrared to perform object detection on the strawberries. The strawberries are placed randomly on the conveyor belt [7], therefore the position is difficult and uncertain for the computer to capture an image. However, with the help of photoelectric sensors, the position of strawberries could be determined by finding out the calyxes on the stems. Another automated sorting system that operates based on the colour and thickness of waste management [14] used ultrasonic sensors for object detection instead of IR (or photoelectric) sensors. Instead of sending out light rays, ultrasonic sensors [15] send out sound waves to detect the presence of an object. Both sensors also have the functionality for measuring distances between the sensor and the object, but ultrasonic sensors are most commonly used as they can transmit very high-frequency ultrasonic waves, which is something IR sensors could not do. However ultrasonic sensors are not as sensitive as IR sensors when it comes to object detection and defining the edges of an area. Ultrasonic sensors also do not perform as well when detecting soft objects. But since the distance between sensor and object is not required information in the seed detection part of the project, an IR sensor is good enough as not only it is the most popular sensor [16] compared to other sensors (sonar sensor, laser sensor, etc.), it is also more reliable for a cost which allows the saving of budget.

1.4.2.3 Computer Vision and Image Processing Algorithm

Computer Vision is used for seed measurement in this project. The input image of seeds is taken in by the system, where the seed segment detection is carried out and the seeds on the image are cropped with the setting up of bounding boxes. With the bounding boxes, the width and height of the seed object could be easily determined for measuring each seed. The cropped image of the seeds is then analysed for the prediction of their quality using a trained model. The implementations of the Computer Vision and Image Processing Algorithm are much more complicated in the egg sorting system compared to the seed sorting system, as the shape of the egg requires information to analyse if an egg is defective or not.

1.4.2.4 Lighting system

Candling is another process for sorting out the defective eggs by applying light to an egg [17]. As the light makes the eggshell transparent, primarily it is used for checking the development of embryos, but it is also able to examine internal components (air cell, yolk, blood spots etc.) which allows the use of this process for quality checks of the eggs. Candling is done in a darkened room with light that penetrates the eggs on the conveyor belt. As the eggs are rolled while passing through the conveyor belt, it allows them to examine all sides of the eggs, increasing the accuracy of sorting. [18] As this process is only able to be used when the testing objects can be transparent when light is applied, it is not available for the seeds that are impenetrable to examine inner components.

1.4.2.5 Operations and speed of the sorting system

From the study of Ahmad et al., the green coffee bean sorting system have a discharging section, the program signals each LED to identify the green coffee bean corresponding to each quality. It is similar to our system. We use a laser diode to project the prediction results of each seed onto the seed tray. [19]

2. Project

2.1 Project Description

The proposed project is aimed to develop a proof-of-concept semi-automated sorting system. This system detects individual germinated oil palm seed(s) from a tray in a digital image captured in a controlled environment. The system will then classify the individual seeds detected in the image into good or bad seeds using AI. The main scope of the project is to utilise hardware and software components to capture images of the germinated oil palm seed(s), instantaneously process the images, accurately classify the seeds detected, and display the results of the classification to be manually sorted. The arrangement of the seeds in the tray will be done manually before being placed onto the conveyor belt and sorted at the end of the conveyor belt by manual labour. An edge device will be used for processing and connecting hardware to software. A GUI will be used as an interface for the user to control the system. The system is able to demonstrate the processing and classification of multiple seeds on a tray by using a tray with 2 by 2 slots for a total of 4 seeds. A laser illumination system will be used to project the results of the classification onto the tray of seeds by illuminating the positions of good seeds on the tray.

2.2 Client Requirements

The client's main requirement is to design a semi-automated sorting system prototype for germinated oil palm seeds using computer vision and prove its effectiveness. This system should reduce the human error caused by sorting the seeds manually by looking at the seeds with human eyes and increase the work efficiency. Some light projection system is requested to overlay the prediction results onto the tray of seeds to ease the workers when sorting instead of having to look up at a monitor screen. The system must be expandable to larger scales for industrial use.

2.3 Design Implementation Constraints

The design of the project will be implemented in several different stages as this is a large project that incorporates different technologies. The client did not give any suggestions about the design of the User Interface but clarified that the user interface should be interactive, clear, and easy to operate. The client pointed out that they would like for some light illumination system to display the results of the seed classification directly onto the tray of seeds to ease the workers when sorting. The client prefers the system to be independent of internet connectivity due to the poor internet access in the palm oil estates. Therefore, an edge device is preferred for seed processing and prediction instead of cloud computing.

2.4 Target Audience

Currently, the system is developed to only be used by our client, Advanced Agriecological Research Sdn. Bhd (AAR) based on the specification provided by them. Our client, ARR, will be using this system as a form of quality control to ensure the quality of the oil palm seeds before sending them out to their customers. This system of using image processing for quality control may be used or further developed for other companies or industries that require a similar sorting system.

3. Requirement specification

3.1 Functional requirement

1. I/O Device	● The system should have the necessary I/O ports for the components used. ● The system should accept input operations with a keyboard, mouse, camera and IR sensor as input devices. ● The system should display output operations with a display device and laser diodes as the output devices. ● The camera should capture an image whenever the IR sensor detects an object.
2. Image Capturing	● The camera should capture an image of the seeds in high resolution, well-lit and not blurry.
3. Local File Storage	● The system should store the data of seed trays that have been processed into a .csv data file. ● The data file should be accessible and viewable through the GUI or local files. ● The resulting images should be stored in an output folder for future reference
4. Image Processing	● The system should be able to perform the detection and segmentation of seeds on the tray.
5. Seed Classification	● The system should be able to perform the prediction of seeds detected based on their condition: good, bad or invalid almost instantaneously. ● The results from the seed classification should be recorded for future reference.

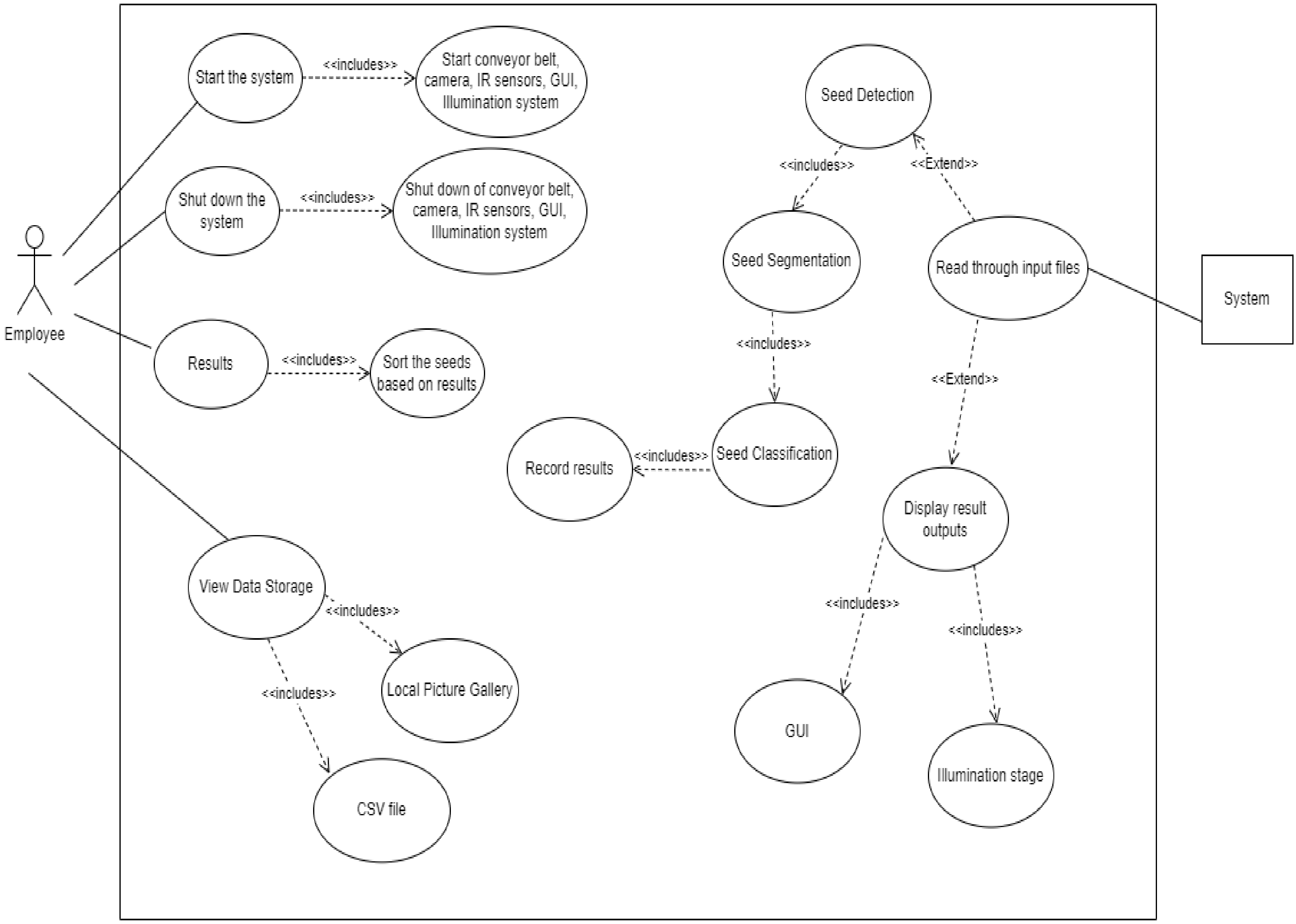
6. Human-Computer Interaction	• Users should be able to interact with the system through a dedicated GUI by using a keyboard and mouse. • The GUI should be intuitive and interactive to use.

3.2 Non-functional requirement

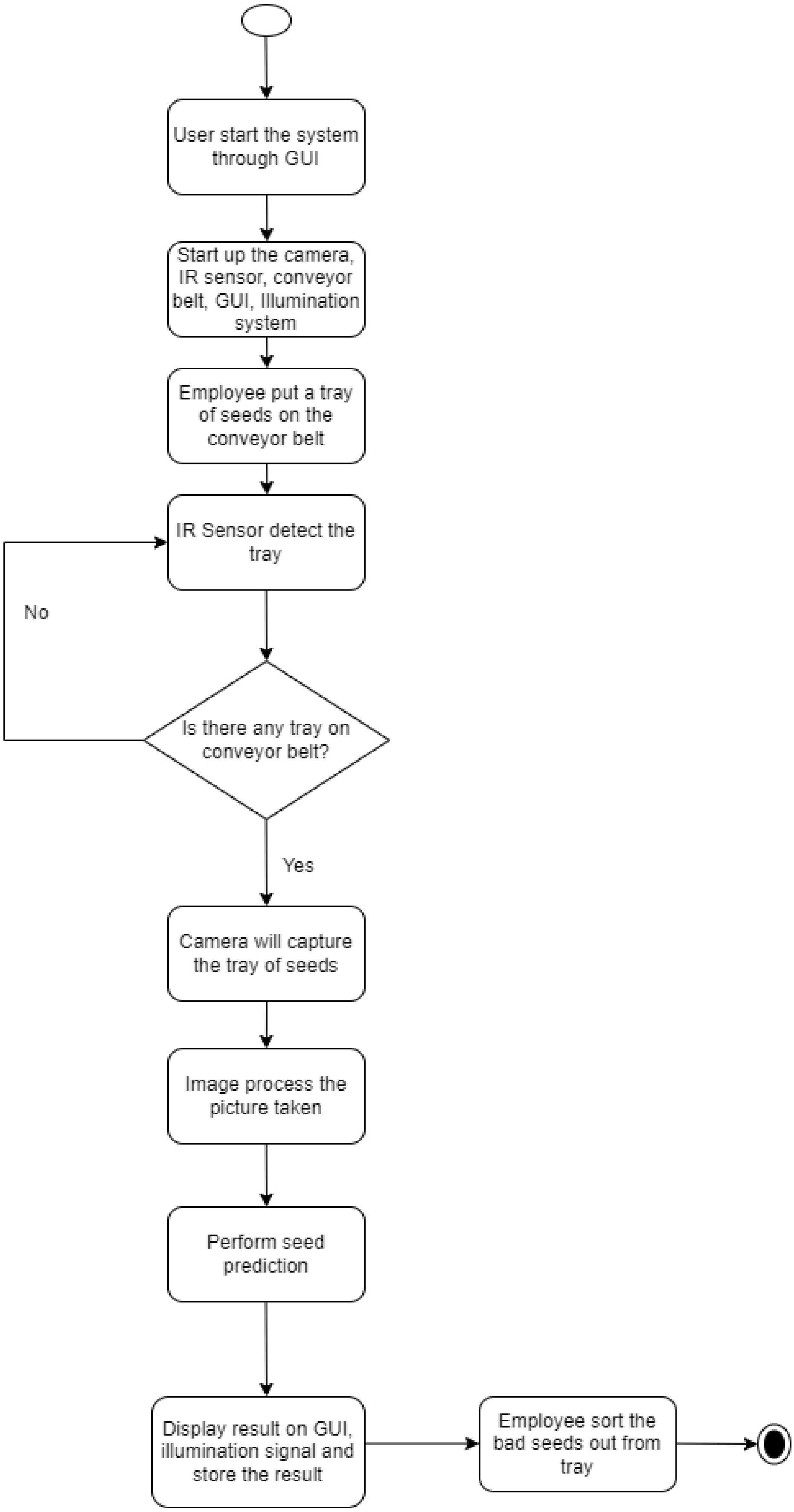
1. Performance requirement	• The software system should be able to run efficiently on an edge device and produce results instantaneously.
2. Programming language requirement	• Python programming should be used for the GUI design implementation, seed detection and classification algorithms.
3. Edge device requirement	• The system should be able to run on a Raspberry Pi device and Raspberry Pi OS.
4. Error handling	• The system should handle errors without crashing • The system should provide error messages to the user for troubleshooting
5. Input data type requirement	• Image input must be in .jpg file format

3.3 System Design

3.3.1 Use Case Diagram



3.3.2 Activity Diagram



4. System Design and User Interface

4.1 Hardware Design Overview

This system will be run on the Raspberry Pi. The Raspberry Pi will handle all the input and output from or to the hardware. Figure 4.1 provides an overview of the hardware components of the system.

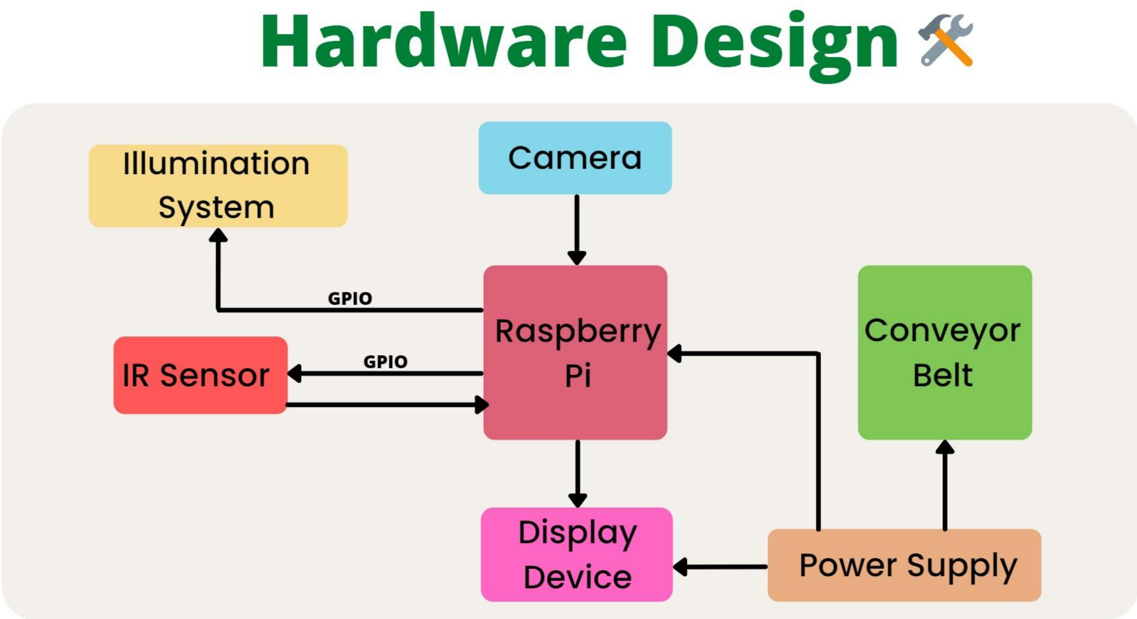


Figure 4.1: Hardware Design Diagram

4.1.1 Raspberry Pi

The Raspberry Pi is used to run the software program and connect the hardware components to the software. The camera, IR sensor and illumination system will be connected to it during implementation. This includes the main functionality of the system to perform seed detection, segmentation, classification and prediction.

4.1.2 Infrared(IR) Sensor

The Infrared sensor is used to detect the seed tray as an input when it passes by. It will return a signal when an object is detected and send it back to the raspberry pi.

4.1.3 Illumination system

The illumination system is used to illuminate the position of good seeds on the tray processed.

4.1.4 Camera

The camera is used to capture the image of the tray when detected by the IR Sensor.

4.1.5 Display Device

The display device is an output device for Raspberry Pi that displays information in pictorial or text form.

4.1.6 Conveyor Belt

The conveyor belt is used to transport the tray of seeds from one end to another end.

4.2 Software Design Overview

Figure 4.2 provides an overview of the graphical user interface (GUI) and the backend implementation.

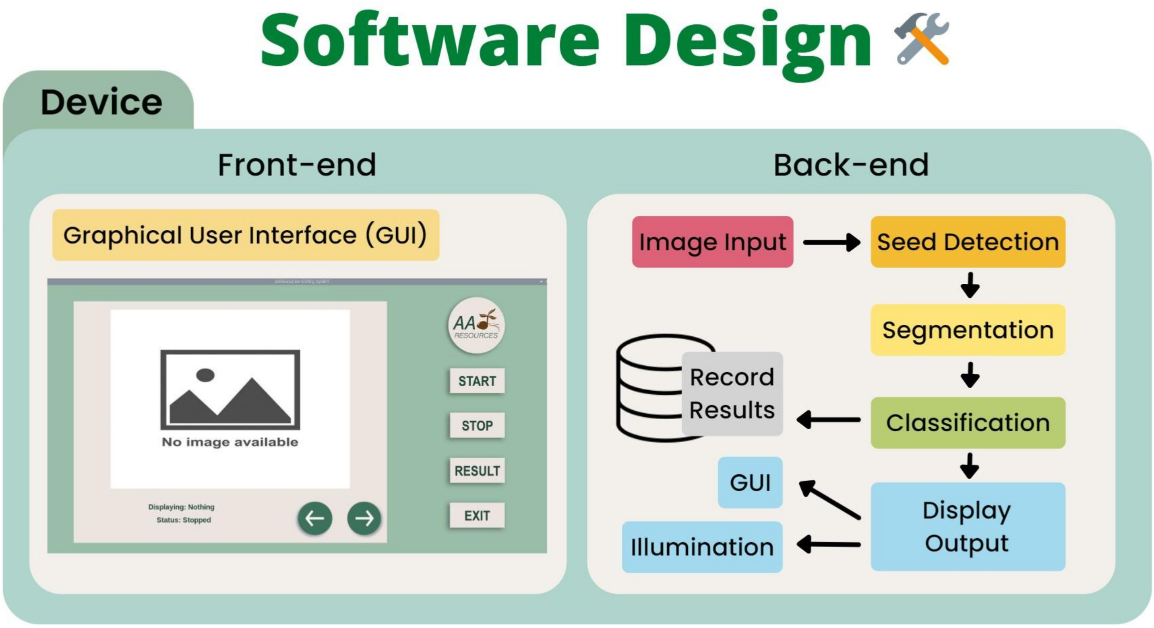


Figure 4.2: Software Design Diagram

Front-end software

4.2.1 GUI

The GUI will be the main interface for the user to interact with the system. This window consists of a dynamic output image placeholder, 4 main buttons: Start, Stop, Result, Exit and 2 image navigation buttons: Left Arrow, Right Arrow. This GUI allows the user to start, pause, terminate the system and view the results during their session.

Back-end software

4.2.2 Seed Prediction

The seed prediction handles the processing of input images captured by the camera. It consists of the seed detection and seed classification algorithms that are used to perform segmentation and prediction of the seeds in the input images.

4.2.3 Output Display

The output display handles the display of results obtained from the seed classification. Output images of seed trays overlaid with coloured bounding boxes are stored in the output folder. The output images of the current session can be viewed through the GUI. The user is also able to see the results directly on the tray using the laser illumination system.

4.2.4 Data Record

The data of the results from the seed classification are recorded in a CSV file format. This includes information such as the processed trays with their tray id, number of good/bad seeds, and date/time. A table with this information can be viewed through the GUI.

4.3 Hardware Components

4.3.1 Conveyor belt

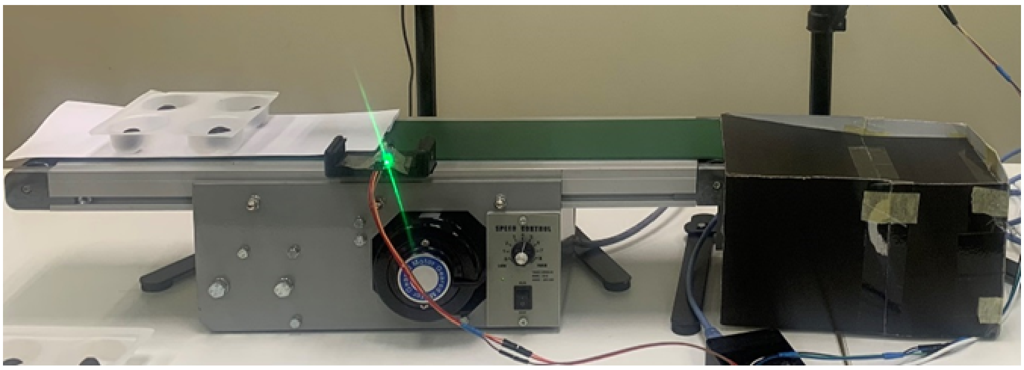
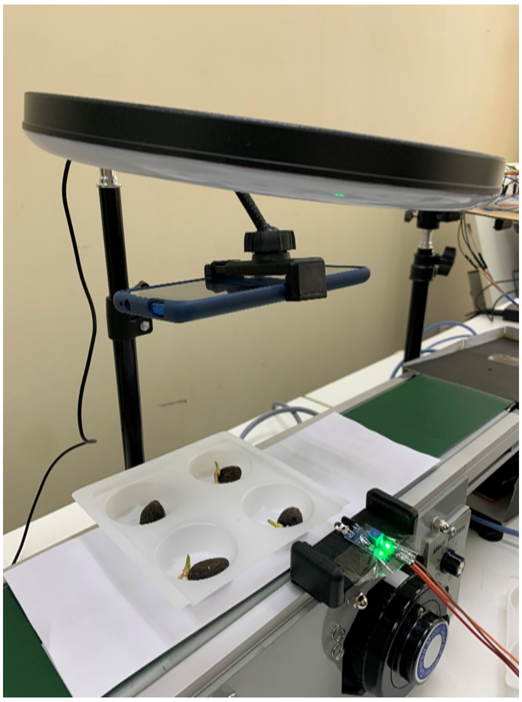


Figure 4.3.1: Conveyor belt with a tray of seeds

The conveyor belt is used to transport the tray of seeds from one end to the other end of the sorting system. The detection and image capture of the seed tray is done at the middle of the belt. The sorting of the seeds with the use of the illumination system takes place at the end of the belt.

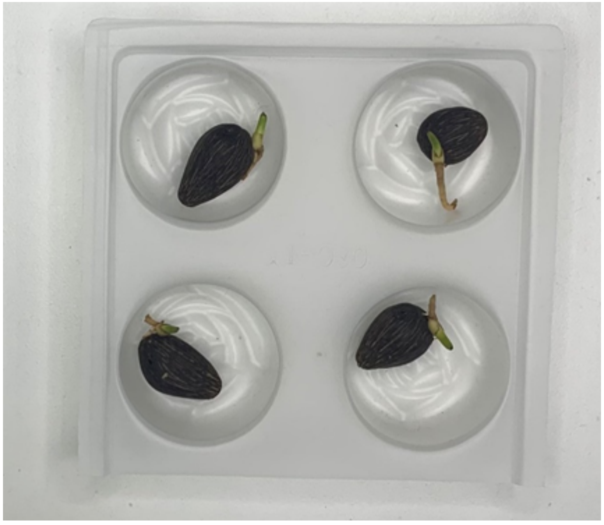
The conveyor belt has a speed control unit to adjust the speed of the conveyor belt. The speed of unit 4 is used for our system as the speed of the belt affects the timing of the image capture. There is also the main power switch to turn the conveyor belt on or off.

4.3.2 IR Sensor & Camera



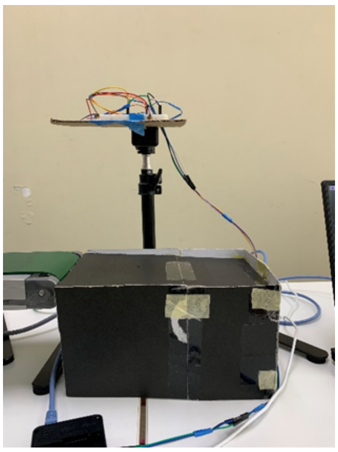
The infrared sensor is positioned towards the edge of the camera frame. This is to ensure that the image captured will have the entire tray within the camera frame when it passes by. When a tray is detected in front of the IR sensor, a signal will be sent to the Raspberry Pi and the camera will capture an image of the current frame. The camera is held using a phone holder attached to a light stand to improve the brightness of the images taken.

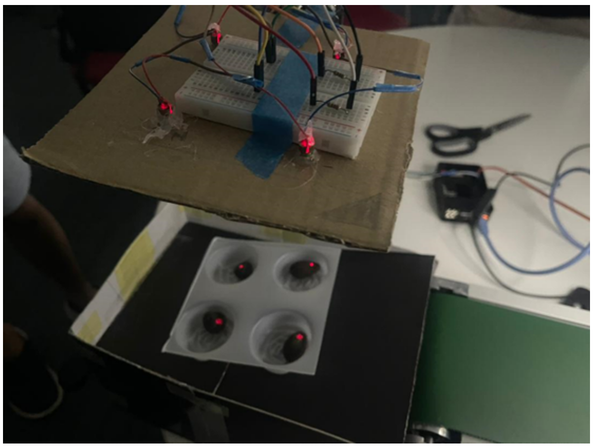
4.3.3 Tray



The trays used in our system have 2 rows and 2 columns to demonstrate that our system works with multiple seeds. The trays are white in contrast to the dark seeds to ease the segmentation of seeds in the trays.

4.3.4 Laser Illumination System & Sorting Station





The illumination system is located at the end of the conveyor belt alongside the sorting station. The laser illumination system is held using a stand and is parallel to the sorting station. This is to ensure that each laser diode is aligned with the corresponding positions of seeds in the tray right below it. The size of the laser diode array is the same as the number of slots in the tray.

4.4 User Interface

4.4.1 Main Menu

T

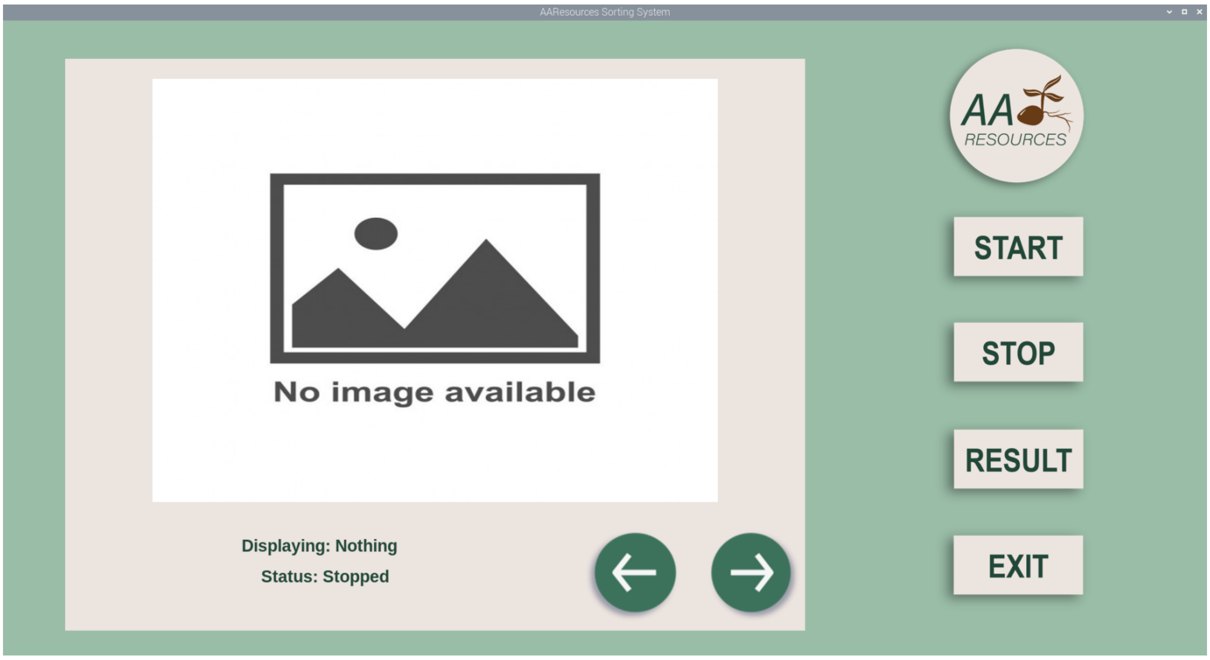


Figure 4.4.1.1: GUI window of Main Menu

The main menu contains the four main control buttons, two navigation buttons, an image placeholder and two status textboxes. The ‘Start’ button starts the system program. The ‘Stop’ button pauses the system program. The ‘Result’ button opens up the result window. The ‘Exit’ button terminates the program. The ‘Left Arrow’ and ‘Right Arrow’ buttons navigate through the previous and next output images from the session.

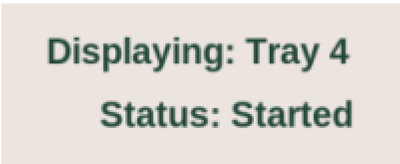


Figure 4.4.1.2: Updated status textboxes when program is running

The ‘Displaying:’ textbox shows the tray_id of the current image displayed. The ‘Status:’ textbox shows the running status of the program.

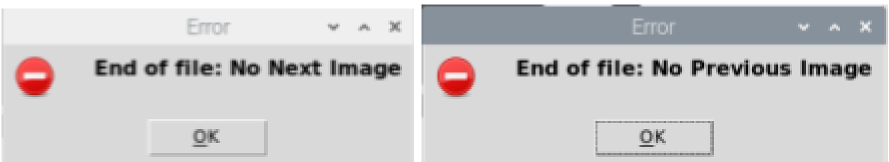


Figure 4.4.1.3: Error window when no more next or previous images

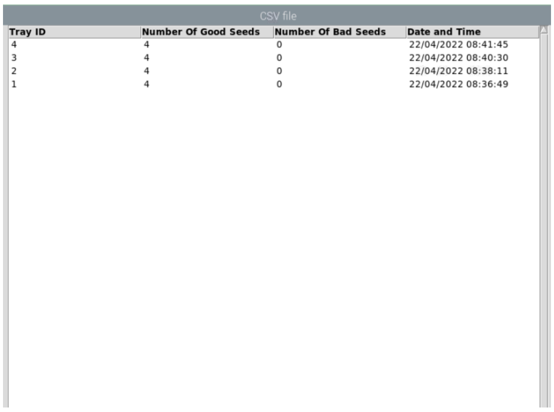
The previous and next image buttons: ‘Left Arrow’ and ‘Right Arrow’ will prompt the user with an error message when there are no more next/previous output images to display in the image placeholder.

Figure 4.4.1.4: Error window when image taken is invalid

While the system program is running, if an invalid object or tray is detected by the camera, an error message will prompt the user to reinsert a new valid tray.

4.4.2 Result Window

When the ‘Result’ Button is pressed, the result window will be displayed. The resulting window displays the information of processed seed trays with the fields ‘Tray ID’, ‘Number of Good Seeds’, ‘Number of Bad Seeds’, ‘Date and Time’.


CSV file

Tray ID	Number Of Good Seeds	Number Of Bad Seeds	Date and Time
4	4	0	22/04/2022 08:41:45
3	4	0	22/04/2022 08:40:30
2	4	0	22/04/2022 08:38:11
1	4	0	22/04/2022 08:36:49

Figure 4.4.2: Result window showing information of processed seed trays

5. Implementation

5.1 Hardware Implementation

5.1.1 Conveyor Belt

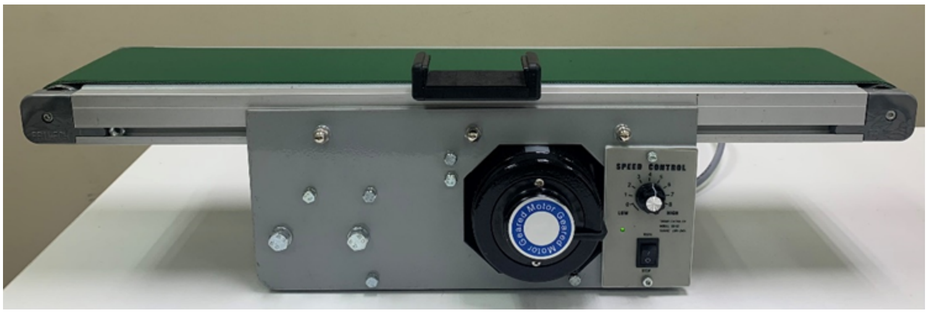


Figure 5.5.1: Conveyor belt

The conveyor belt is purchased from Primech Industry Supplies (M) Sdn Bhd. It is custom-built to our specifications based on our prototype size and budget. The size of the conveyor belt is 14cm x 60cm, with the belt being 10cm in width.

5.1.2 Infrared Sensor

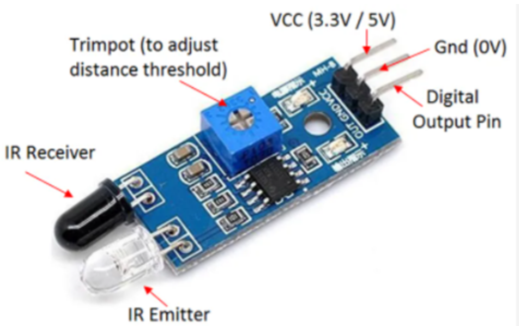


Figure 5.1.2.1: Infrared Sensor

The infrared (IR) sensor module has only three pins which are the voltage common collector (VCC), ground (GND) and digital output pin. The VCC, GND and output pins of the IR sensor are connected to pin numbers 4, 3 and 6 on the Raspberry Pi respectively using Jumper Wires. Once the pins are correctly connected to the Raspberry Pi, the IR sensor is ready to detect objects and send its output signal to the Raspberry Pi.

5.1.3 Laser Illumination System

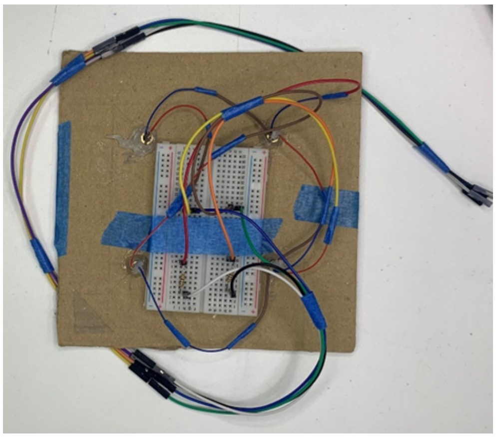


Figure 5.1.3.1: 2 by 2 Laser Illumination System

The laser illumination system is made using four 3V laser diodes, an electronic breadboard, jumper wires and two 75 ohm resistors. The laser diodes are laid out 6.5cm apart from one another on cardboard to be aligned to the spacing between the seeds on the tray.

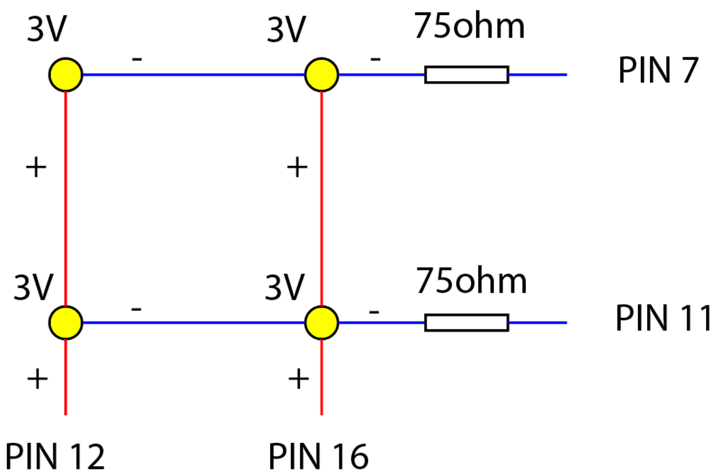


Figure 5.1.3.2: Laser Illumination system circuit diagram

The laser illumination system is built based on the schematics above. The laser diodes wires are soldered together to form a parallel circuit based matrix. 3V laser diodes are used as the maximum voltage supplied by the Raspberry Pi pins is 3.3V. 75-ohm resistors are used to limit the current flow and prevent the laser diodes from fusing. The resistors are connected in series at the end of the negative wires (anodes). The anodes will be the rows of the matrix while the cathodes will be the columns of the matrix. The anodes are connected to PIN 7 and PIN 11 while the cathodes are connected to PIN 12 and PIN 16 on the Raspberry PI respectively. This design of utilising rows and columns is called multiplexing, which switches between the output of every row at very small time intervals. The laser illumination system is easily expandable to larger sizes as multiplexing reduces the number of pins needed. For instance, a 10 x 10 laser diode matrix would require 100 pins by default, but with multiplexing that is reduced to only 20 (number of rows + columns).

5.1.4 Camera

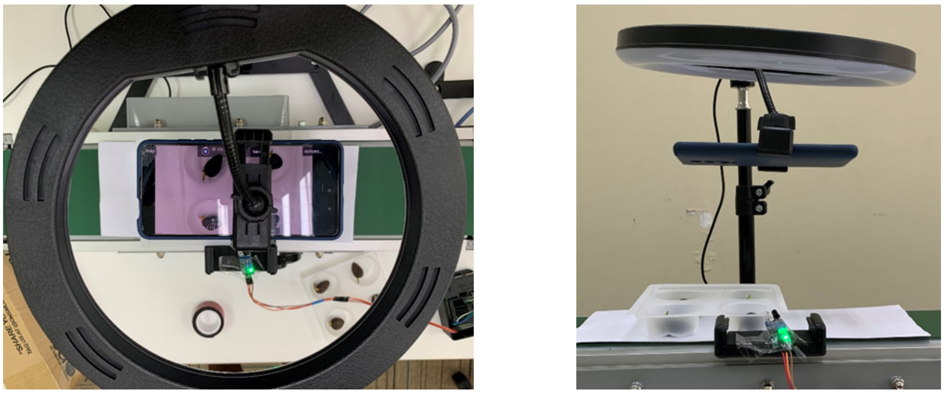


Figure 5.1.4.1: Top view and side view of phone camera mounted on camera stand

A phone camera is used instead of a Raspberry Pi Camera due to the limited 32-bit OS support for the camera. A ring light and a camera stand are used to ensure that the tray is evenly lit and a top-down view of the tray is captured. To connect the phone camera to the Raspberry Pi device, an app on the phone called “IP Webcam” is to be installed. Both devices must be connected to the same network and accessed using the server IP address provided by the app.

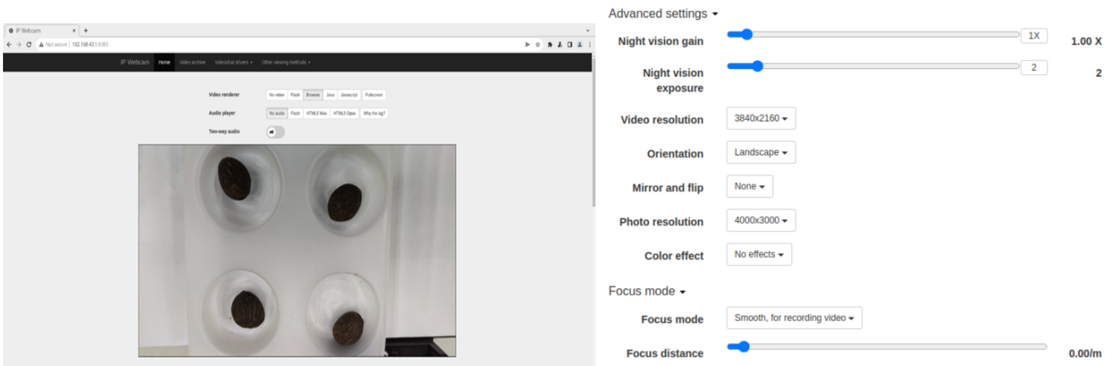
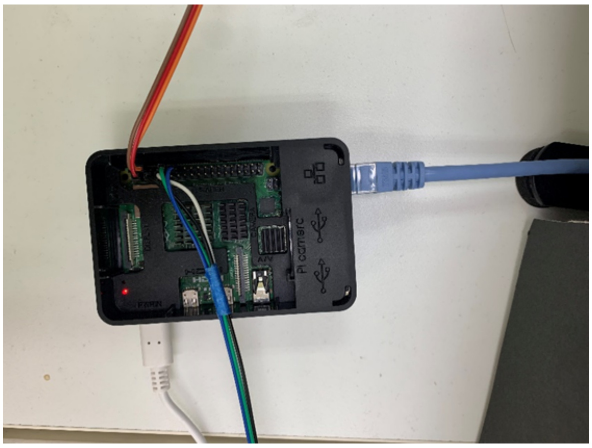


Figure 5.1.4.2: Viewing IP Camera website with camera configuration interface

The IP camera frame and configuration controls can be accessed by entering the IP address provided into the browser. The same IP address is also used in the system’s software to directly access the camera.

5.1.5 Raspberry Pi



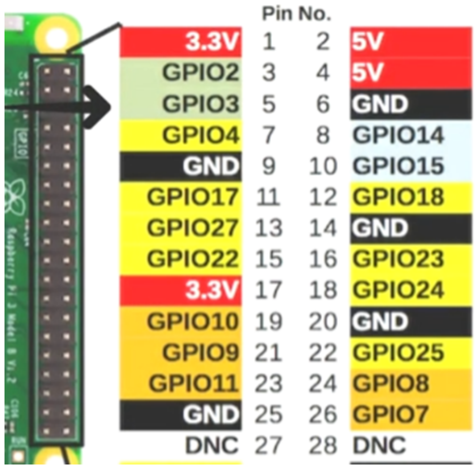
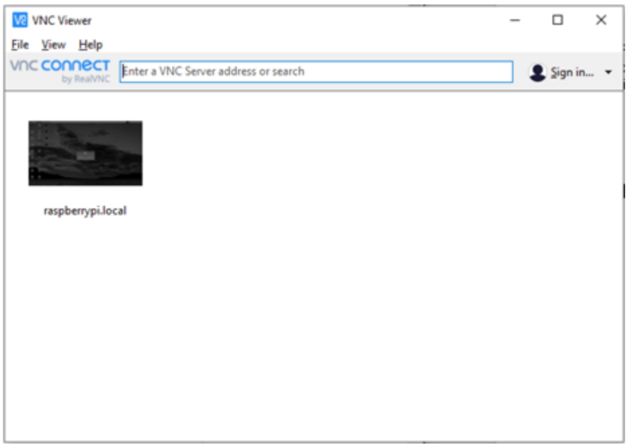


Figure 5.1.5.1: Raspberry Pi Figure 5.1.5.2: Raspberry Pi GPIO Pin

The Raspberry Pi 4 Model B with 4GB RAM is chosen as the edge device due to its processing power and it is the newest model. The device is convenient to work with as it has a lot of connection ports, removable/expandable storage and GPIO pins for hardware connectivity. The 64-bit version of Raspberry Pi OS is installed onto the 64GB microSD card inserted into the device using software called “Raspberry Pi Imager”. The 64-bit OS version is installed instead of 32-bit due to the applications of our system software such as the PyTorch and Torchvision modules requiring 64-bit compatibility.

5.1.6 Display Device



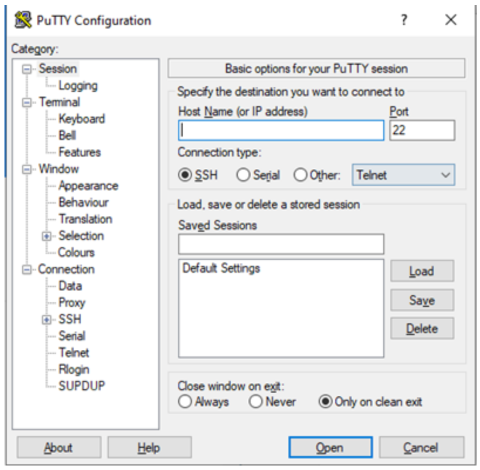


Figure 5.1.6.1: VNC Viewer Interface Figure 5.1.6.2: Putty Interface

The display of the Raspberry Pi can be viewed using a display device such as an external monitor. We chose our laptop as the display device instead of purchasing a new monitor to save cost. After connecting the devices with an ethernet cable, Putty is used to run SSH and establish a connection between the Raspberry Pi to the laptop. VNC viewer is used to view the display of the Raspberry Pi and control it remotely.

5.2 Software Implementation

5.2.1 main.py

The main.py file initialises and updates the GUI using Tkinter. It handles the behaviour of every button and updates the dynamic information in the GUI.
When the start button is pressed, it starts and runs the mainFunction() thread which endlessly loops until the program pauses or terminates.

```python
def startProgram():
    global starting
    global running
    global sleeping
    global stop_threads
    global paused
    paused = False
    running = True
    sys_status(True)

    #Start mainFunction thread only once when starting
    if starting:
        print("starting")
        start_Operation = threading.Thread(target=mainFunction)
        start_Operation.start()
        starting = False
```

Figure 5.2.1.1: startProgram() function when ‘Start’ button is pressed

The mainFunction() handles the coordination of the hardware processes: camera.py, ir_sensor.py and illumination.py with seed_detection.py. While running, it awaits the input from the Sensor() and calls the takeImage() function to take an image when the ir_sensor detects an object.

```python
def mainFunction():
    global running
    # if (image_taken = true), get new image file path
    # input will always be an image
    if running:
        #Read the current image input based on a queue system. Each new image input are named after the current tracking number.
        dirname = os.path.join(os.path.dirname(__file__), "resources/temp/current_input")
        f = open('resources/temp/tracking_number.txt', 'r')

        #Read the image file in the folder
        filepath = os.path.join(dirname, f.readline() + '.JPG')

        path = pathlib.PurePath(filepath)
        filename = path.name

        #If operation is not paused/stopped, check if the IR sensor detects an object.
        if paused == False:
            #If IR sensor detects an object, call takeImage function and pass the filename parameter to name the image taken
            if Sensor() == 1:
                takeImage(filename)

        if not os.path.exists(filepath):
            root.after(100, mainFunction)
```

Figure 5.2.1.2: mainFunction() when thread starts

Once the image captured is located in the current_input folder, it will then be passed to the run_seed_detection() function for processing. The returned matrix of the processed image is then used to update the laser() function through the laser_index.txt file. When the image is successfully processed, the tracking number will be incremented by 1 and the output image is displayed on the GUI using display_img().

```
elif os.path.isfile(filepath):
    output_matrix = run_seed_detection(filename)

    output_list.append("out/results/" + filename)
    output_image_file = filename
    fname = output_image_file

    lst.insert(tk.END, fname)
    lst.bind("<<ListboxSelect>>", display_img)

    os.remove(filepath)
    fileint = int(filename.split('.')[0])
    fileint += 1
    file = str(fileint)
    filepath = os.path.join(dirname, file + ".jpg")
    with open('resources/temp/tracking_number.txt', 'w') as f:
        f.write(str(fileint))

    with open('resources/temp/laser_index.txt', 'w') as f:
        f.write(str(output_matrix))

    #Update output image display on GUI
    display_img()

    #Update current output image index on GUI
    img_status()
    root.after(1000, mainFunction)
```

Figure 5.2.1.3: Continuation of mainFunction() code

When the stop button is pressed, the paused flag is set to True which prevents any new image input.

```
def stopProgram():
    global paused
    paused = True
    sys_status(False)
```

Figure 5.2.1.4: stopProgram() function when 'Stop' button is pressed

When the result button is pressed, the data in the results.csv file will be read and displayed in a table.

```
with open('out/results.csv') as f:
    reader = csv.DictReader(f, delimiter=',')
    for row in reader:
        tray_id = row['tray_id']
        number_of_good_seeds = row['number_of_good_seeds']
        number_of_bad_seeds = row['number_of_bad_seeds']
        date_time = row['date_time']
        tree.insert("", 0, values=(tray_id, number_of_good_seeds, number_of_bad_seeds, date_time))
```

Figure 5.2.1.5: openCSV() function when 'Result' button is pressed

When the exit button is pressed, the GUI is destroyed using root.destroy() and the system is terminated safely using sys.exit().

```python
def exitProgram():
    global release
    release = 1
    root.destroy()
    cv2.destroyAllWindows()
    with open('resources/temp/running_status.txt', 'w') as f:
        f.write("False")
    sys.exit()
```

Figure 5.2.1.6: exitProgram() function when 'Exit' button is pressed

5.2.2 seed_detection.py

The seed_detection.py file reads the image in the current_input folder and passes it to segment(). The segment() function crops the seeds in the image based on the seed edges and returns the seed_bound of the seeds. The bounds of the segmented seeds are passed into predict_seed() imported from seed_classification.py. The predicted seed results returned by the predict_seed() function are used to update the laser_matrix.txt file and results.csv file. The seed labels are used to generate an output image with an overlay of coloured bounding boxes: green for seeds labelled as good and red for bad.

```python
for i in range(len(seed_bound)):
    # cropped it out
    seed_segmented = image[seed_bound[i][1]:seed_bound[i][1] + seed_bound[i][3],
                     seed_bound[i][0]:seed_bound[i][0] + seed_bound[i][2]]
    cv2.imwrite(os.path.join(folder_name_cropped, imagename + '_' + str(i) + '.jpg'),
                cv2.cvtColor(seed_segmented, cv2.COLOR_RGB2BGR))
        if model:  # make prediction for each seed if a model is specified
            net, device, criterion = load_trained_model(model)
            label, score = predict_seed(seed_segmented, model=net, device=device, criterion=criterion)
        else:
            label = None
        rows.append([imgpath, seed_bound[i][0], seed_bound[i][1], seed_bound[i][0] + seed_bound[i][2],
                     seed_bound[i][1] + seed_bound[i][3], label])
```

Figure 5.2.2: Main loop in segment() function

5.2.3 seed_classification.py

The seed_classification.py file houses the CNN model classes and seed_predict() function. It utilises both torchvision.py and tensorboard.py packages to run the neural network. The seed_predict() function classifies the segmented seeds using the trained CNN model and returns the label of the seed (good or bad) based on its prediction confidence score.

```
def predict_seed(seed_img, model, device, criterion):
    if not torch.is_tensor(seed_img) and isinstance(seed_img, np.ndarray):
        seed_img = cv2.resize(seed_img, (256, 256))
        # resize the image here (256, 256)
        # convert seed_img to tensor
        seed_img = transforms.ToTensor()(seed_img).float()
        if len(seed_img.shape) == 2:
            # convert (H,W) to (B,C,H,W)
            seed_img = seed_img.unsqueeze(0).unsqueeze(0)
        elif len(seed_img.shape) == 3:
            # convert (C,H,W) to (B,C,H,W)
            seed_img = seed_img.unsqueeze(0)
    else:
        raise Exception('The input image for classification must be either a Tensor or an Numpy array!')
    seed_img = seed_img.to(device)
    model.eval()
    with torch.no_grad():
        output = model(seed_img)
        if isinstance(criterion, nn.BCELoss):
            predicted_label = (output.data > 0.5).float().view(-1)
            predicted_score = output.data.detach().view(-1)
        elif isinstance(criterion, nn.CrossEntropyLoss):
            _, predicted_label = torch.max(output.data.detach(), 1)
            predicted_score = F.softmax(output.data.detach(), dim=1)[:, 1]

    return predicted_label, predicted_score
```

Figure 5.2.3: predict_seed() function in seed_classification.py

5.2.4 camera.py

Loads the camera module using cv2.VideoCapture() and captures an image of the current camera frame using cap.read(). Writes image file into current_input folder using cv2.imwrite.

```
def takeImage(filename):
    #Start up camera module
    cap = cv2.VideoCapture('http://192.168.43.1:8080/video')
    ret, frame = cap.read()

    #Set folder path to save image captured in
    path = '/home/pi/Desktop/PI_SEGP/resources/temp/current_input'
    cv2.imwrite(os.path.join(path, filename),frame)

    #Turn off camera module
    cap.release()
```

Figure 5.2.4: takeImage() function in camera.py

5.2.5 ir_sensor.py

This script interacts with the input infrared sensor device. It will read the output of the sensor using GPIO.input() at the pin connected. Input from the GPIO pin returns 1 if nothing is detected, and 0 if an object is detected.

```
def Sensor():
    GPIO.setwarnings(False)
    GPIO.setmode(GPIO.BOARD)

    #Setup GPIO to read pin 3 (input) and 5 (output)
    GPIO.setup(3,GPIO.IN)
    GPIO.setup(5,GPIO.OUT)
    #Read output of IR sensor at PIN(3)
    val = GPIO.input(3)

    #If IR sensor returns 1, no object is detected
    if val == 1:
        GPIO.output(5, GPIO.LOW)
        sensor = 0

    #If IR sensor returns 0, an object is detected
    else:
        GPIO.output(5,GPIO.HIGH)
        sensor = 1

    return sensor
```

Figure 5.2.6: Sensor() function in ir_sensor.py

5.2.6 illumination.py

The illumination.py file Interacts with the output laser illumination device. It converts the matrix string stored in the laser_matrix.txt file into an integer matrix using the str2matrix() function.

```python
#Converts matrix string eg: "[1, 1]" to integer array [1,1]
def str2matrix(str_matrix):
    matrix = [[1,1],[1,1]]
    matrix[0][0] = int(str_matrix[2])
    matrix[0][1] = int(str_matrix[5])
    matrix[1][0] = int(str_matrix[10])
    matrix[1][1] = int(str_matrix[13])

    for i in range(2):
        for j in range(2):
            if matrix[i][j] == 1:
                matrix[i][j] = 0
            elif matrix[i][j] == 0:
                matrix[i][j] = 1


    return matrix
```

Figure 5.2.6.1: str2matrix() function in illumination.py

The converted matrix is then assigned to the GPIO output pins using GPIO.output(). Each element in the matrix corresponds to the position of the laser diodes in the illumination system. The laser diode turns on if its GPIO output value is 1, and turns off if its GPIO output value is 0. Due to multiplexing, the program loops between every row of laser diodes with a 0.001s delay between loops to create a seamless display of laser illumination.

```python
#Loop through the laser illumination while main system is running
while running:

    #While looping, read the latest laser_index.txt file which contains the latest results
    f = open('resources/temp/laser_index.txt', 'r')
    str_matrix = f.readline()

    #Convert matrix string to integer list (array)
    matrix = str2matrix(str_matrix)


    #Only 1 column can display at a time due to multiplexing
    #Cycle through columns with 0.001s delay to display seamless transition between columns
    for i in range(2):
        GPIO.output(cathodes[0], matrix[i][0])
        GPIO.output(cathodes[1], matrix[i][1])

        #Update current column
        GPIO.output(anodes[i], 1)

        #Set delay to 0.001s
        time.sleep(sleeptime)

        #Disconnect current column
        GPIO.output(anodes[i], 0)
```

Figure 5.2.6.2: Main loop in laser() function in illumination.py

6. Key Implementation Decisions

6.1 Programming Environment

The Python programming language is chosen for our system's implementation. Python is an object-oriented language and high-level programming language with dynamic semantics. It combines the dynamic typing and binding of high-level built-in data structures. It also contains a lot of other features and modules which are suitable for our system's application. The syntax of this programming language is intuitive and easy to pick up, reducing the time needed for learning. Furthermore, the AI algorithms provided are also coded in Python, making it the ideal choice.

6.2 Operating Device & System

Raspberry Pi 4 Model B is chosen as our operating device because of its processing power, compact form factor and extensive connectivity. It is convenient to work with and carry around when we work with our groupmates physically. Its extensive interface allows us direct connection to our devices and hardware components using the GPIO pins. The removable microSD storage enables us to opt for storage expansion and backups in the future.

The operating system used in our device is the Raspberry PI 64-bit OS. Through the use of Raspberry Pi OS, we are able to make use of the Python programming language and its modules that require 64-bit OS support. This operating system is efficient to run and having a similar design to Windows, it is easy to use.

6.3 Version Control System

We have decided to use GitHub as our version control tool to manage changes to source code over time. GitHub tools help us work faster and smarter. It offers free online private repositories storage and is also one of the popular version control systems used by others.

6.4 Integrated Development Environment (IDE)

For our Python Integrated Development Environment (IDE), we used Thonny. Thonny is an IDE that is widely used on Raspberry Pi devices for being lightweight and powerful for most usages.

6.5 Communication and Collaboration

As for our communication tools, we made use of Discord, WhatsApp chat and Microsoft Teams. These applications are free of charge and every of our group members are familiar with them.

For project collaboration and documentation, we made use of cloud services such as Google Drive and Google Docs. These services are free of charge and allow us to create, edit and share files instantaneously.

6.6 Data Storage Format (CSV)

We chose to record our system's results in a comma-separated values (CSV) file format. Each line in the CSV file corresponds to a row in a table, and the comma delimiters separate the columns in a table. CSV is a widely used storage file format in database and data storage systems as importing CSV files is fast and easy. By using CSV for our data recording, future database implementations and extensions can be done without having to modify the data structure. Due to the simplistic data structure of CSV files, it is easy for us to modify and read the files.

7. Problems Encountered

7.1 Incompatible camera

At the beginning of our implementation, we purchased a Raspberry Pi camera for image capturing. However during implementation, we found that the camera does not work with the 64-bit Raspberry Pi OS, which is still in beta test version. We are unable to downgrade the OS from 64-bit to 32-bit due to our Python modules requiring 64-bit compatibility.

As an alternative, we bought a webcam as a replacement to the Raspberry Pi camera. Implementing the webcam into our system's software is easy with the use of Python modules. However, we have noticed that the images taken by the webcam are blurry and have a low resolution of 640x480p. The poor image quality has led to the seed detection and classification algorithm to fail.

Due to the limited time and budget left, we settled on using our phone camera as a replacement to our previous cameras. We made use of a mobile app called "IP Webcam" that streams the phone's camera to a website hosted on a local network. By using our phone camera, we managed to get consistent and clear images taken without any additional cost.

7.2 Image Detection and Segmentation

The image segmentation and object detection algorithms used in our system are unable to accurately detect the seeds in the input image. There were bounding boxes at random locations in the image which indicates that the algorithms are not properly locating the seeds. This was later found to be the nature of the seed detection algorithm's trained model whereby the seeds must be placed in front of a white background. As the tray is placed on green conveyor belt, the non-white background is disrupting the algorithm from working as intended.

To counteract this problem, we placed a folded A4 paper under the tray of seeds to eliminate any background noises from being captured. As this is only a temporary solution, a permanent solution would be to replace the colour of the conveyor belt from green to white.

7.3 Poor Seed Prediction Results

Towards the end of our system implementation, we requested palm oil seeds from our client for testing purposes. From the testing with samples of good seeds and bad seeds given, we were unable to get accurate prediction results from our chosen seed prediction model. We tried multiple trained seed prediction models that were provided to us, however none of them performed as expected. Every seed detected was classified into being a good seed, albeit being tested with multiple trained prediction models.

Initially, we came up with the hypothesis that it could be that the algorithm failed due to having the incorrect versions of Python, PyTorch or TorchVision modules installed. Although we have tried downgrading and reinstalling the Python modules, we were unable to trace the cause of the problem. We also thought that the model may be underperforming because the Raspberry Pi does not have CUDA cores from a graphics card.

In the efforts of troubleshooting the model for performing below expectations, we tried testing the trained models using the unmodified seed detection and classification source code on our laptops with the provided dataset and the image samples of the seeds provided in trays. In this case, our laptops should have the CUDA cores to run the algorithms and an untouched seed detection program to test the input images, however the results were still the same.

At last, we contacted our supervisor who is the author of the algorithm and provided her with some seed sample images taken using our camera that are used in our system for testing. This was to identify if the problem was from our end or it could be that the model was not trained with enough dataset. The results of the sample images tested by our supervisor returned the same results as our system. She proposed that it could be the small seed sample size of our tray that may be affecting the results to skew towards poor results.

We have concluded that the subpar results of the prediction model may be due to the differences in many aspects of the images used in the training dataset and images captured using our system. Factors such as image resolution, brightness, sharpness and background noise may affect the prediction model from predicting accurately. The prediction model should work reliably if it is trained with datasets taken using our system's settings and environment.

7.4 System Robustness

During our system testing stage, we encountered a lot of errors with our system. The most common errors encountered were in the input images taken. At times, the camera may go out of focus resulting in a blurry image, or when the image taken does not have the entire tray within the frame. When an error occurs, the software program will crash as seeds cannot be detected for classification. This has been fixed by using try and except to allow the program to run and display an error message during exceptions without crashing. In this case, the invalid input images will not be processed and recorded into the results.

Apart from creating exceptions, we understand that our system has a flimsy build quality that can be further improved and calibrated. The problems faced such as blurry images can be solved by upgrading the camera to one that is able to capture images at a faster shutter speed without affecting the brightness. The system also needs to be calibrated upon setting up for the first time as the individual hardware components are placed without consideration to precision. A mere centimetre difference in placement may affect the image of the tray taken to be out of frame, or the tray colliding with other hardware components during operation. Like in an industrial setting, our system should work flawlessly provided that the placements of the components are calibrated and anchored.

8. Summary of Achievements

Our system is operable on an edge device, which means that our software has low operating requirements and is efficient even on a low-performance device. We have also successfully proven the concept of our proposed design and implementations as we have successfully developed a fully functional prototype while integrating the prediction model given and meeting the client's requirements. Another feat we have achieved is that the graphical user interface of our system is intuitive with a consistent design language without affecting the system's functionality and performance. Despite our limited knowledge in the fields of engineering, we managed the implementation of electrical components and designed our own laser illumination system. Aside from that, we were able to apply the AI techniques for seed detection and classification provided by our supervisor, albeit its complexity and ability to produce accurate results which are beyond our scope. Lastly, we managed to complete our project within the given time and budget despite the shortcomings and challenges faced.

9. Reflective Comments

Throughout the completion of our assigned project, our team has faced many challenges. One main issue was our deficiency in technical knowledge of hardware design and implementation. As Year 2 computer science students, our expertise only covers up to the software level of implementation. However, this does not excuse us from learning and overcoming our barriers in integrating hardware components.

At the beginning of our project, we were lost with the task given at hand and there was not much progress. During the first semester of our project, the inability to meet physically due to the pandemic has been detrimental to our progress. We were unable to design and test out physical prototypes for our system together. The lack of physical meetings due to distance between members has also hindered us from purchasing and starting the hardware implementation of our system. By the time we were clear on the direction of our project, we had other coursework and examinations to focus on. However, with guidance provided by our supervisor and proper time management, we were able to work things out during the first semester. We were given the opportunity to express our strengths.

On another note, we were also given the task to understand the seed prediction source code provided by our supervisor. As we were unfamiliar with Python and Neural Networks, we took a while to understand the source code to an extent where we can make use of it and integrate it into our main system. By consistently conducting meetings every week between members, we were able to approach problems together and make incremental progress over time.

Although there were conflicts between ideas and design choices among members at times, we all came to a common understanding and made decisions as a group. Every group member was assigned tasks based on their interest and was given the opportunity to express their strengths. Through teamwork and collaboration, we were able to cover each other's weaknesses and solve problems faced during the project.

10. References

[1] Ritchie, H., & Roser, M. (2021). Palm Oil. Our World in Data. https://ourworldindata.org/palm-oil#citation.

[2] AA RESOURCES official website. https://www.aarsb.com.my/

[3] Alikhanov, J., et al. (2019). 1319 Design and Performance of an Automatic Egg Sorting System Based on Computer Vision. TEM Journal, 8(4), 1319-1325. https://www.temjournal.com/content/84/TEMJournalNovember2019_1319_1325.pdf

[4] Ambrose, A., Kandpal, L. M., Kim, M. S., Lee, W. H., & Cho, B. K. (2016). High speed measurement of corn seed viability using hyperspectral imaging. *Infrared Physics & Technology*, *75*, 173-179. https://www.sciencedirect.com/science/article/pii/S1350449515300074

[5] Mizushima, A., & Lu, R. (2013). An image segmentation method for apple sorting and grading using a support vector machine and Otsu's method. *Computers and electronics in agriculture*, *94*, 29-37.

[6] Clinton, N. 2017. Otsu's Method for Image Segmentation. Google Earth and Earth Engine. https://medium.com/google-earth/otsus-method-for-image-segmentation-f5c48f405e

[7] Liming, X., & Yanchao, Z. (2010). Automated strawberry grading system based on image processing. *Computers and Electronics in Agriculture*, *71*, S32-S39.

[8] Jeong, S., Lee, Y. M., & Lee, S. (2017). Development of an automatic sorting system for fresh ginsengs by image processing techniques. *Human-centric Computing and Information Sciences*, *7*(1), 1-13. https://link.springer.com/article/10.1186/s13673-017-0122-5

[9] Tho, T. P., Thinh, N. T., & Bich, N. H. (2016, November). Design and development of the vision sorting system. In *2016 3rd International Conference on Green Technology and Sustainable Development (GTSD)* (pp. 217-223). IEEE. https://ieeexplore.ieee.org/abstract/document/7796652

[10] Jahanbakhshi, A., Momeny, M., Mahmoudi, M., & Radeva, P. (2021). Waste management using an automatic sorting system for carrot fruit based on image processing technique and improved deep neural networks. *Energy Reports*, *7*, 5248-5256. https://www.sciencedirect.com/science/article/pii/S2352484721006302

[11] Khojastehnazh, M., Omid, M., & Tabatabaeefar, A. (2010). Development of a lemon sorting system based on color and size. African Journal of Plant Science, 4(4), 122-127. https://academicjournals.org/journal/AJPS/article-full-text-pdf/CC3D97311502

[12] Norsazwan, M., G., Sinniah, U., R., Puteh, A., B., Namasivayamn, P., Appleton, D., R., Mohaimi, M., & Aminuddin, I., A. (2021, July). ASSOCIATION OF SEED COLOUR WITH GERMINATION, PHYSICAL AND PHYSIOLOGICAL GROWTH OF OIL PALM (Elaeis guineensis) SEEDLINGS. http://jopr.mpob.gov.my/wp-content/uploads/2021/07/DOI-Norsazwan-ASSOCIATION-OF-SEED-COLOUR-8-inpress.pdf

[13] Pornsuriya, Chaninun & Sunpapao, Anurag & Srihanant, Natthapach & Worapattam, Kanyapak & Kittimorakul, Jittra & Phithakkit, Supaporn & Petcharat, Vasun. (2013). A Survey of Diseases and Disorders in Oil Palms of Southern Thailand. Plant Pathology Journal. 12. 169-175. 10.3923/ppj.2013.169.175.

[14] Kute, U. V., et. al. 2021. Automatic Sorting System based on color and thickness of waste management. International Research Journal of Engineering and Technology (IRJET). https://www.irjet.net/archives/V8/i2/IRJET-V8I242.pdf

[15] Vachher, H. 2016. Right Sensors for Object Tracking @ IOT-part 2. HCL. https://www.hcltech.com/blogs/right-sensors-object-tracking-iot-part-2

[16] Burnett, R. 2017. Ultrasonic vs Infrared (IR) Sensors - Which is better? MaxBotix. https://www.maxbotix.com/articles/ultrasonic-or-infrared-sensors.htm

[17] Ramiro, V. C., Lourdes, R. S. C., & María, N. L. C. (2018). Egg Candling Analysis Equipment Design: A Safety Solution. Procedia-Social and Behavioural Sciences, 238, 501-506.

[18] The Poultry Site. (2006). Egg Grading. https://www.thepoultrysite.com/articles/egg-grading

[19] Ahmad, U., Seminar, K. B., Soedibyo, D. W., & Subrata, I. (2010). The development of automatic coffee sorting system based on image processing and artificial neural network. https://repository.ipb.ac.id/bitstream/handle/123456789/41672/The%20Development%20of%20Automatic%20Coffee%20Sorting%20System.pdf?sequence=1&isAllowed=y

11. Appendix

11.1 Test Case

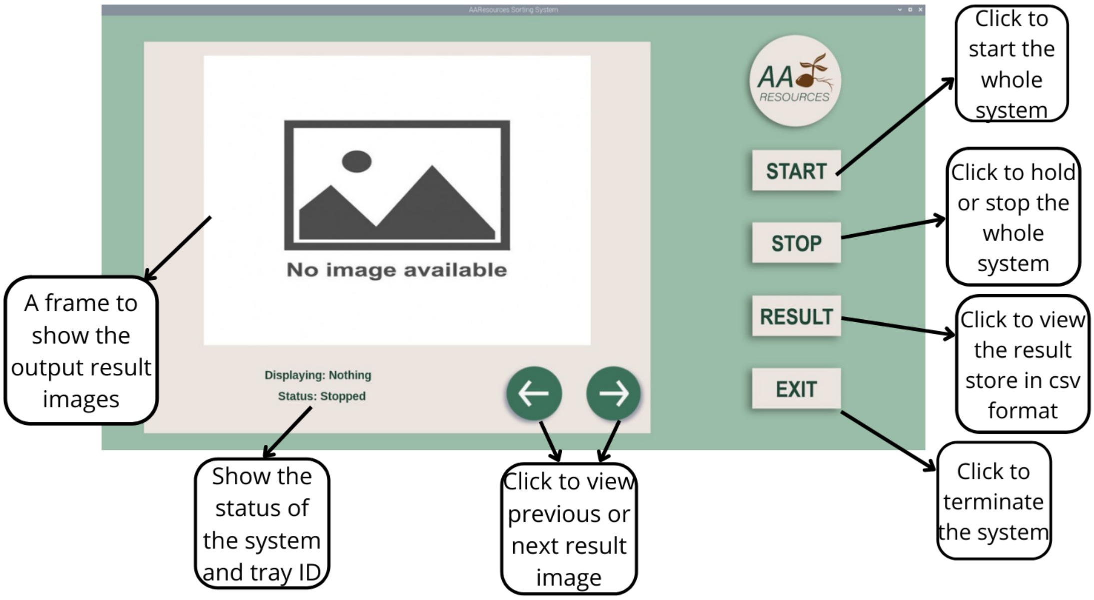
Test Case	Action	Expected Result	Pass/Fail
1	Start the program	Display interface of menu	Pass
		Display image placeholder	Pass
		2 text placeholder below image placeholder: - Index of image displaying - Status of software (Started/Stopped)	Pass
		2 buttons below image placeholder: - Left Arrow - Right Arrow	Pass
		4 buttons on the right of the menu: - Start - Stop - Data - Exit	Pass
2	Click on Start Button	IR sensor starts accepting input	Pass
		Update program status to Running	Pass
3	Click on Stop Button	IR sensor stops accepting input	Pass
		Update program status to Stopped	Pass
4	Click on Data Button	Display a table of results	Pass
		Program still runs in the background	Pass

5	Click on Exit Button	Terminate the program safely	Pass
6	Click on Next Button	Display next result image on the interface if any	Pass
		Update index of current result on the interface	Pass
7	Click on Next Button at the end of file	Display error box message that there is no next image to display	Pass
8	Click on Previous Button	Display previous result image on the interface if any	Pass
		Update index of current result on the interface	Pass
9	Click on Next Button at the end of file	Display error box message that there is no previous image to display	Pass
10	IR sensor detects a tray	Signal camera to capture an image of the tray	Pass
		Run segmentation algorithm	Pass
11	IR sensor detects an invalid tray or object	Signal camera to capture an image of the tray	Pass
		Prompt user with error message that the tray/object detected is invalid	Pass
		Delete the input and wait for next valid input	Pass

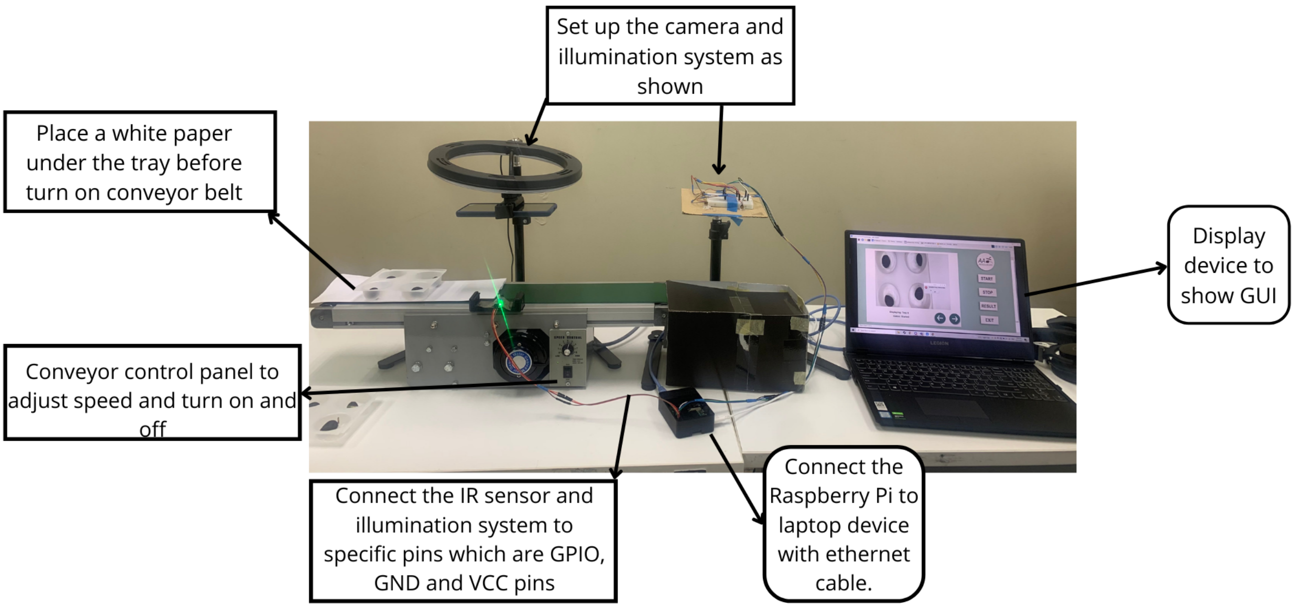
12	Segmentation algorithm	Crop the individual seeds in the image	Pass
		Run prediction algorithm using cropped seeds	Pass
13	Prediction algorithm	Accurately predict the condition of seeds	Fail
		Display image of prediction result on the interface and update index of current result	Pass
		Update Illumination System with result	Pass
14	Laser illumination system	Only illuminate positions of good seeds based on the results of prediction	Pass
		Turn off illumination when program exits	Pass

11.2 User Manual

11.2.1 GUI interface



11.2.2 Hardware Equipment



11.3 Meeting Minute

No.	**Meeting Minutes 01**
Date	30 Sept 2021
Prepared By	Ku MoEun
Attendees	All members and supervisor (Dr Iman)

Content:

- Objective
  - To meet members and supervisor
  - To know the objective and goals of the project
- Discussion
  - Self-introduction
  - Briefing about module and project title
  - Decide on project title to do

No.	**Meeting Minutes 02**
Date	6 Oct 2021
Prepared By	Ku MoEun
Attendees	All members, supervisor, and clients

Content:

- Objective
  - To know about client's backgrounds and requirements
- Discussion
  - Discuss the pros and cons of current oil palm seeds sorting system
  - Method to increase productivity
  - Discuss about system design
- To do
  - Further research on project title

No.	**Meeting Minutes 03**
Date	9 Oct 2021
Prepared By	Ku MoEun
Attendees	All members

Content:
- Objective
    - Discuss on client's requirement
    - Discuss the structure of the project description report
- Discussion
    - Discuss the outline and content of the project description report
    - Analyse the requirement from the client and design the structure of the system and model.
- To do
    - Further research on project title
    - Writing draft for project description report

No.	**Meeting Minutes 04**
Date	11 Oct 2021
Prepared By	Ku MoEun
Attendees	All members, supervisor

Content:
- Objective
    - Discuss on client's requirement
    - Discuss a rough idea on system design
- Discussion
    - Discuss and brainstorm about the research that we have done
    - Create a draft model of system design based on eggs sorting system.
- To do
    - Further research on project title
    - Working on the project description draft

No.	**Meeting Minutes 05**
Date	16 Oct 2021
Prepared By	Ku MoEun
Attendees	All members, supervisor, and client

Content:

- Objective
    - Further discuss about client's requirement
    - Getting information for writing project description
- Discussion
    - Present rough idea of system design and model
    - Clients provide feedback on the idea and model.
    - The supervisor gives a guide on the report writing
- To do
    - Research on the system to implement

No.	**Meeting Minutes 06**
Date	18 Oct 2021
Prepared By	Ku MoEun
Attendees	All members

Content:

- Objective
    - Discuss about project description report
- Discussion
    - Decided on the content of the project description report to write
- To do
    - Drafting project description

No.	**Meeting Minutes 07**
Date	20 Oct 2021
Prepared By	Ku MoEun
Attendees	All members, supervisor, and client

Content:

- Objective
  - Project Description Review
- Discussion
  - Discuss the content to include in the project description report
  - Discuss the progress in the project description report
- To do
  - Finalise and review of the project description report

No.	**Meeting Minutes 08**
Date	27 Oct 2021
Prepared By	Ku MoEun
Attendees	All members

Content:

- Objective
  - Discuss and finalize the version of system design
- Discussion
  - Decided about the tools and software for system design and implementation
- To do
  - Explore Python toolboxes
  - Explore the machine learning toolboxes

No.	**Meeting Minutes 09**
Date	28 Oct 2021
Prepared By	Ku MoEun
Attendees	All members, supervisor

Content:

- Objective
    - Discuss on sorting algorithm provided
- Discussion
    - Supervisor gives an explanation of the algorithm provided.
    - Give a demo run on the algorithm
    - Further explain on the project title
- To do
    - Run and modify the code to implement to the system designed

No.	**Meeting Minutes 10**
Date	3 Nov 2021
Prepared By	Ku MoEun
Attendees	All members and supervisor

Content:

- Objective
    - Further discuss on sorting algorithm provided
- Discussion
    - QnA session for clearing the doubt
    - Supervisor provides a details explanation of the algorithm
- To do
    - Learning the code and understand the algorithm provided

No.	**Meeting Minutes 11**
Date	5 Nov 2021
Prepared By	Ku MoEun
Attendees	All members, supervisor, and Dr Aazam

Content:

- Objective
    - Discuss about edge devices for the project
- Discussion
    - Dr Aazam provides the information about tools and methods to implement edge devices
    - Dr Aazam and the supervisor provide a guideline and explain about the details on the implementation stage
- To do
    - Start to implement the system

No.	**Meeting Minutes 12**
Date	15 Nov 2021
Prepared By	Ku MoEun
Attendees	All members and Mr Chin Jun Yuan

Content:

- Objective
    - Discuss about IDE to run the code
    - Discuss about the method to train model
    - Further discuss the sorting algorithm provided
- Discussion
    - Type of implementation
    - Mr Chin provided and explained about the Python tools such as Pytorch and TensorBoard for machine learning
    - Mr Chin provided a guide on the sorting algorithm provided
    - Clarification on seed sorting algorithm.
- To do
    - Modify the file path in the code to read the file and image processing the images read

No.	**Meeting Minutes 13**
Date	22 Nov 2021
Prepared By	Ku MoEun
Attendees	All members

Content:

- Objective
    - Discuss interim group report
    - Review on system implementation
- Discussion
    - Decided on the content in the interim group report
    - Separate the job for each of the members.
    - Progress update on the GUI design
- To do
    - Complete the part assigned for the interim group report

No.	**Meeting Minutes 14**
Date	4 Dec 2021
Prepared By	Ku MoEun
Attendees	All members and client

Content:

- Objective
    - Discuss and review the first prototype of the system design
- Discussion
    - Show and demonstrate to the client the first prototype
    - Ask for a suggestion for implementation
    - Client provides feedback on the first prototype
- To do
    - Planning for next stage of the implementation
    - Complete the interim group report

No.	**Meeting Minutes 15**
Date	6 Dec 2021
Prepared By	Ku MoEun
Attendees	All members

Content:

- Objective
    - Update and merge the interim group report.
- Discussion
    - Review on interim report draft.
    - Merging the group work.
- To do
    - Modify and add the content in the interim report.

No.	**Meeting Minutes 16**
Date	10 Dec 2021
Prepared By	Ku MoEun
Attendees	All members and supervisor

Content:

- Objective
    - Review and discuss the interim group report with the supervisor.
- Discussion
    - Supervisor provides feedback on the interim report, especially the literature review section.
    - Finalize and review the interim report with the group members.

No.	**Meeting Minutes 17**
Date	13 Dec 2021
Prepared By	Ku MoEun
Attendees	All members

Content:

- Objective
  - Review and finalise the interim report.
- Discussion
  - Review on interim report
- To do
  - Grammer checking

No.	**Meeting Minutes 18**
Date	24 Jan 2022
Prepared By	Ku MoEun
Attendees	All members

Content:

- Objective
  - Budget proposal
- Discussion
  - Check the price of the hardware parts
- To do
  - Write down the parts found with price

No.	**Meeting Minutes 19**
Date	10 Feb 2022
Prepared By	Ku MoEun
Attendees	All members

Content:
- Objective
    - Redesign hardware
- Discussion
    - Discuss what equipment needs to use for hardware
    - Redesign in software parts
    -
- To do
    - Contact with the head of Electrical engineering

No.	**Meeting Minutes 20**
Date	14 Feb 2022
Prepared By	Ku MoEun
Attendees	All members and supervisor

Content:
- Objective
    - Feedback on interim report
    - Discussion of projected design
- Discussion
    - Supervisor provides feedback on interim report submitted
    - Separate jobs for each member into coding part and report part
- To do
    - Correction on the interim report based on feedback given

No.	**Meeting Minutes 21**
Date	16 Feb 2022
Prepared By	Ku MoEun
Attendees	All members

Content:

- Objective
  - Discussion on interim report
  - Hardware equipment
- Discussion
  - Book appointment with the head of EE
  - Discuss on hardware parts
- To do
  - Research on literature background on sorting system
  - Correct grammatical error

No.	**Meeting Minutes 22**
Date	21 Feb 2022
Prepared By	Ku MoEun
Attendees	All members

Content:

- Objective
  - Discussion about purchasing hardware parts
- Discussion
  - Research on camera and Raspberry Pi
  - Volunteer on comparing camera
  - Separate jobs on purchasing materials
  - Decide who will go to see conveyor belt
- To do
  - Research on camera, literature background
  - Write budget proposal

    - Drive to shop to check the conveyor belt

No.	**Meeting Minutes 23**
Date	7 Mar 2022
Prepared By	Ku MoEun
Attendees	All members

Content:

- Objective
    - Choose a date to have client meeting
    - Implementation of hardware
- Discussion
    - Discuss the design of the hardware
    - Separate jobs on implementation of hardware
- To do
    - Modify literature review in final report

No.	**Meeting Minutes 24**
Date	12 Mar 2022
Prepared By	Ku MoEun
Attendees	All members

Content:

- Objective
    - Discussion about final report
- Discussion
    - Separate parts on final report
    - Create discord channel
- To do
    - Choose a part write in final report

No.	**Meeting Minutes 25**
Date	14 Mar 2022
Prepared By	Ku MoEun
Attendees	All members with supervisor and clients

Content:

- Objective
    - Update the progress to clients on software and hardware
    - Update current progress to client
- Discussion
    - Explain the progress and problems that we are facing now
    - Show demo video to the client
- To do
    - Get seed samples from the client

No.	**Meeting Minutes 26**
Date	21 Mar 2022
Prepared By	Ku MoEun
Attendees	All members

Content:

- Objective
    - Feedback on final report
- Discussion
    - Speak out what each member did past week
- To do
    - Finalise meeting minutes

No.	**Meeting Minutes 27**
Date	24 Mar 2022
Prepared By	Ku MoEun
Attendees	All members

Content:

- Objective
    - Discuss final report
- Discussion
    - Feedback on overall report
- To do
    - Make corrections

No.	**Meeting Minutes 28**
Date	31 Mar 2022
Prepared By	Ku MoEun
Attendees	All members

Content:

- Objective
    - Demonstration of hardware
- Discussion
    - Video recording
- To do
    - Finalise final report

No.	**Meeting Minutes 29**
Date	8 April 2022
Prepared By	Ku MoEun
Attendees	All members with supervisor

Content:

- Objective
    - Demonstrate the hardware to the supervisor
- Discussion
    - Get feedback from supervisor
    - Try some other suggestion
    - Try to fix errors caused during demonstration
    - Recorded demo video
- To do
    - Write missing points into final report

No.	**Meeting Minutes 30**
Date	9 April 2022
Prepared By	Ku MoEun
Attendees	All members with clients

Content:

- Objective
    - Feedback from clients after showing the demo video
    - Short meeting with members about final report
- Discussion
    - Overall feedback of the hardware
    - Points that we need to include in the final report

No.	**Meeting Minutes 31**
Date	11 April 2022
Prepared By	Ku MoEun
Attendees	All members

Content:
- Objective
  - Slies preparation
- Discussion
  - Divide parts for presentation slides
  - Divide parts for additional point in final report
  - Check on final report
- To do
  - Summarise final report
  - Create presentation slides

No.	**Meeting Minutes 32**
Date	20 April 2022
Prepared By	Ku MoEun
Attendees	All members

Content:
- Objective
  - Video recording and finalise slide presentation
- Discussion
  - Discuss about slide presentation
  - Video recording about the demonstration of software and hardware system
- To do
  - Finalise slide presentation

No.	**Meeting Minutes 33**
Date	22 April 2022 (Afternoon)
Prepared By	Ku MoEun
Attendees	All members with examiners

Content:
- Objective
    - Presentation day
- Discussion
    - Live demonstration
    - Slide presentation

No.	**Meeting Minutes 34**
Date	22 April 2022 (Night)
Prepared By	Ku MoEun
Attendees	All members

Content:
- Objective
    - Finalise final report
- Discussion
    - Divide parts for missing parts in final report
    - Check grammatical errors
- To do
    - Write out missing points in the report
    - Tide up the report

No.	**Meeting Minutes 35**
Date	25 April 2022
Prepared By	Ku MoEun
Attendees	All members

Content:

- Objective
    - Finalise final report
- Discussion
    - Check information
    - Check grammatical errors
- To do
    - Write out missing parts in the report
    - Tide up the report