# Technical Report (W14) (1).pdf

*by* Zhi Yi Ooi

**Submission date:** 22-Jan-2025 04:34PM (UTC+0800)
**Submission ID:** 2568934632
**File name:** Technical_Report_W14_1_.pdf (6.83M)
**Word count:** 9091
**Character count:** 50379

# Defects Prevention & Monitoring System for Assembly Line

[1]Chong Zhi Sheng, [2]Muhammad Amirul Naqil Bin Mohd Nohan, [3]Ooi Zhi Yi, [4]Ahmed Abdulgader Salim Assagaf, [5]Dr. Nurulaqilla Binti Khamis, [6]Flextronics

Universiti Teknologi Malaysia, Malaysia

Flextronics Shah Alam Sdn. Bhd.

[1]chongsheng@graduate.utm.my; [2]muhammadamirulnaqil@graduate.utm.my; [3]ooiyi@graduate.utm.my; [4]Salim@graduate.utm.my; [5]nurulaqilla@utm.my; [6]kangshen.guy@flex.com

*Abstract*— ***The Defects Prevention & Monitoring System for Assembly Line is an innovative solution designed to enhance the accuracy and efficiency of assembly processes in smart manufacturing. Using AI-based computer vision techniques like YOLO model for object detection and user-friendly hardware such as Raspberry Pi with the integrated cameras allows the system to identify assembly step errors, monitor operator productivity, and reduce cycle time inconsistencies. With customizable assembly steps and a scalable design suited to different stations, the system aims to improve the assembly steps accuracy, provide real-time alerts for immediate corrective actions, and show the step's correctness on an interface that allows a more sustainable and cost-effective manufacturing environment.***

***Keywords— monitoring, camera, assembly step, interface, configure, detection***

## 1. Conceive

### *1.1 Background Introduction*

Assembly plays a crucial role in the manufacturing production line. It involves the process of how the operators selecting, positioning or fitting a component into the product. However, evaluating operators' efficiency in terms of accuracy and speed in assembly tasks can be challenging. Therefore, the Defects Prevention and Monitoring System is created to be implemented at assembly stations, ensuring verification of step completion and monitoring the operator's performance.

### *1.2 End-user Input Data*

In order to understand the problem better, a group discussion has been conducted to identify what is the key focus on the project. Questions such as the existing solution for the problem, the expected solution and the technologies planned to be implemented. An industrial visit has been conducted to the factory to have a better understanding of the problem statement. For this project case study, the end user input data was obtained from the interview session with the engineer of Flextronic Sdn Bhd.

### *1.3 User Needs Analysis*

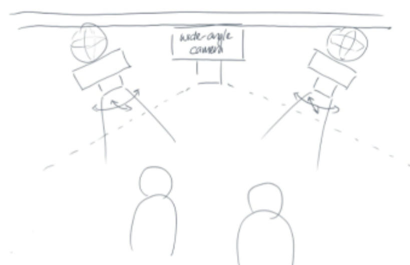
From the industrial visit and interview, it was concluded that the assembly station is experiencing inconsistent cycle times and frequent assembly errors by operators, leading to inefficiency. Therefore, a system capable of verifying assembly steps and monitoring operators is necessary. Additionally, the solution must be adaptable to various stations with different products.

### *1.4 Personification and Design Statement*

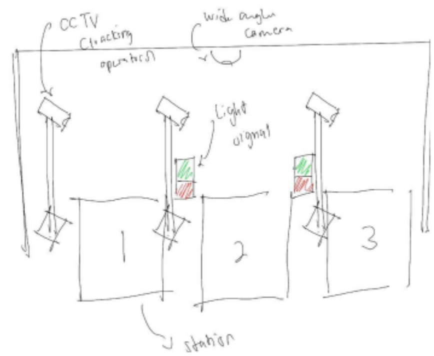
In proceeding with the product and solution design, a persona named Mr. John is introduced. At 40 years old, Mr. John is a supervisor on a manufacturing line encountering difficulties in assessing operator performance and ensuring assembly accuracy. Each station handles different products, making it challenging to trace assembly errors effectively. Therefore, he requires a system to assist in monitoring operators and validating the assembly steps to enhance overall efficiency and accuracy.

### *1.5 Idea Generation*

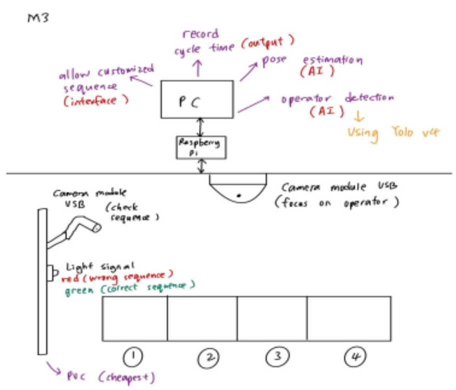
Our team begins by systematically identifying potential solutions tailored to effectively meet user needs. We engage in detailed discussions to thoroughly assess the pros and cons of each option, enabling us to determine the most optimal and efficient approach. We have opted for the implementation of an advanced AI model capable of object and hand pose detection, utilizing predetermined timing parameters to meticulously validate each step in the assembly process. Additionally, the integration of people detection capabilities will enable continuous monitoring of the assembly environment, safeguarding against any factors that might impede operator efficiency. This comprehensive approach ensures a precise and streamlined assembly process that remains adaptable to diverse operational conditions and challenges, ultimately enhancing productivity and accuracy.



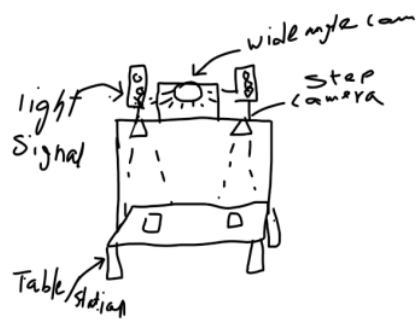

Conceptual Sketch by M1




Conceptual Sketch by M2




Conceptual Sketch by M3




Conceptual Sketch by M4

The final idea is developed by integrating the conceptual sketches from each of the members. The final idea does not include all features proposed by the members, but it is developed after discussion and agreed by the majority of the members.

*1.6 Conceptual Prototype Development*

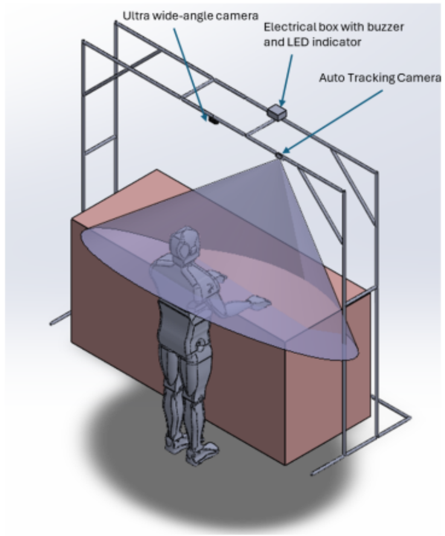



Figure 1.6: The design of prototype in SOLIDWORKS.

The prototype development involves constructing a structural frame for mounting cameras. This frame is made from PVC pipes with a diameter of ½ inch, measuring 3 ft x 6 ft x 7 ft. A wide-angle webcam serves as the primary camera, tasked with detecting the operator and products, and calculating the cycle time. Additionally, a rotatable camera with two degrees of freedom is utilized to monitor the operator's hands within preset regions. The duration for which hands are detected is compared against the required preset time to estimate step completion. Designed for versatile mounting, the rotatable camera can be adjusted to accommodate different assembly stations, where varying regions of interest may necessitate alternative viewing angles other than from the top.

*1.7 STEEP Analysis and SDG Requirements*

The implementation of this system significantly boosts productivity by allowing for detailed tracking of performance and assessment of workloads. This monitoring helps identify areas needing improvement and optimizes task assignments to ensure balance and efficiency across operations. Moreover, real-time validation enhances workplace safety by verifying that assembly steps are executed correctly, minimizing risks linked to the improper handling of equipment or materials. These advancements cultivate a more productive and safer work setting, promoting overall employee well-being.

This initiative provides substantial economic advantages by refining the assembly process, which results in increased productivity and enhanced operational efficiency. Reducing assembly mistakes cuts down expenses tied to rework, flawed products, and resource wastage. Additionally, the heightened precision and effectiveness elevate product quality, which in turn raises customer satisfaction and potential income. Businesses can also realize lasting cost savings by pinpointing inefficiencies and addressing errors early in the production cycle.

By minimizing the frequency of assembly errors, the system plays a crucial role in conserving materials, significantly cutting down on waste. This strategy not only alleviates environmental stress caused by discarded materials but also boosts energy efficiency by shortening production cycles and minimizing energy-heavy rework processes. Sustainable manufacturing techniques, improved through real-time validation and error reduction, align with corporate environmental stewardship objectives and support wider sustainability efforts.

TABLE I
STEEP analysis of the developed prototype

| Aspect | Contribution |
|---|---|
| Sociological | • Improves the ability to monitor productivity more effectively, thus assists in identifying areas for improvement.<br>• Workspace Safety and Efficiency. |
| Economic | • Increases productivity gains by optimizing the assembly process and reducing inefficiencies.<br>• Helps reduce costs by minimizing assembly errors and material waste. |
| Environmental | • Decreases material waste by reducing assembly errors.<br>• Energy Efficiency. |

### *1.8 Innovative Features*

The uniqueness of the proposed system is that it has multiple cameras to capture the movement of operators and their assembly steps. Multiple cameras can be used to monitor the stations with multiple operators, as well as to have multiple perspectives to reduce the blind spot of the camera view from top. The proposed system has cameras with free-mount features which make the system highly flexible and enables the user to mount the camera to different positions to fulfill the needs of different stations.

## 2. Design

### *2.1 Flowchart for Coding*

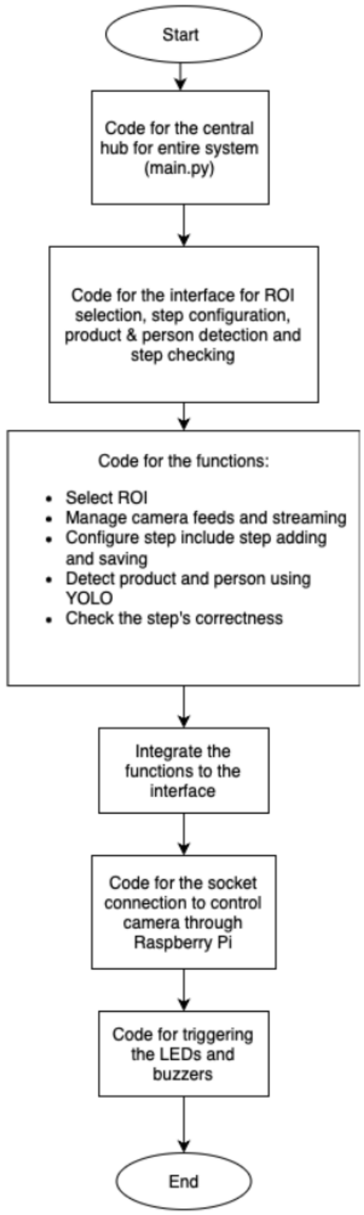



Figure 2.1: Flowchart for coding.

The code for all the interfaces and functions are completed using Python as the programming language and VS Code as the programming environment. Firstly, the central hub for the entire system which is main.py is coded where it consists of several buttons to go to several camera interfaces. It includes the interface for ROI selection (Wide-Angle Camera), step configuration (Rotatable Camera 1 & 2), product and person detection and step checking. Next, several functions are coded to be integrated to the interface created. Firstly, the first function is selecting ROI which can be used to define specific areas in the work station through video feed where the monitoring is required. The second function is managing camera feeds and streaming to ensure smooth video capture and display using OpenCV for both wide-angle camera and rotatable camera. The next function is step configuration including step adding and saving which allows the supervisor to define the steps required for operators to do their tasks. Moreover, the function for the detection of product and person using YOLOv8 is completed to allow the product and person during the assembly process to be detected and shown. The last function is the checking of step correctness including the validation of the steps performed using hand detection via MediaPipe.

After completing all the coding of the functions. the functions are integrated into their respective interfaces that we have created before. This is to ensure seamless interaction and monitoring. After checking and confirming all the functions work after integration to interface, the code for the socket connection to control camera through Raspberry Pi is completed as this connection is used to ensure the control over camera movements and settings to enable a flexible monitoring setup. Lastly, the final coding process involved triggering external hardware such as LEDs and buzzers to provide immediate alerts during the monitoring process. The LEDs are used to indicate correct or incorrect completion of assembly steps.

The tool used for the interface is Tkinter which is Python's standard library for creating graphical user interfaces as it is simple and easy to use for desktop applications. Additionally, OpenCV library is used to capture video streams from the camera, display the real-time video feeds in the interface and process video frames for object detection and tracking. Next, Ultralytics library is used to implement the YOLO to detect and track products and persons. Moreover, MediaPipe which is a framework for building perception pipelines is used to detect hand movements in video feed for step correctness checking process. Next, Pandas which is a data analysis and manipulation library is used to store the process with configured steps, selected ROI coordinates through the writing and reading of CVS files. Besides, a Python imaging library for image processing which is PIllow is used to convert the video frames into a format suitable for display in the Tkinter interface. Lastly, the library Socket is used to enable the communication between the program with Raspberry Pi for controlling the cameras.

### *2.2 Block Diagram / System Architecture*

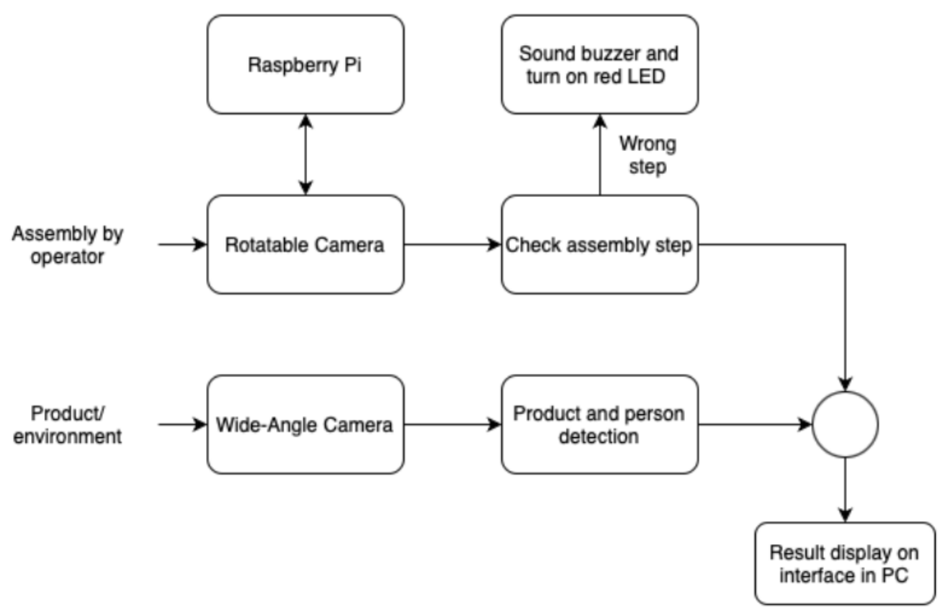



Figure 2.2: System architecture.

According to the block diagram above, two cameras play an important role in our overall system. The first camera is the rotatable camera which is for the monitoring and checking the assembly process by the operator. For the process of checking step using rotatable camera, Raspberry Pi is used for controlling the angle of the rotatable camera based on the input from the computer where it acts as a path to send the angle from computer to rotatable camera. Additionally, for the step configuration by the supervisor, Raspberry Pi also used to send the angle from camera to computer to record the angles for the steps configured to be used to compare with the steps completed by the operator to check the step correctness. Moreover, during the checking of the assembly step, the buzzer and LEDs have been used in which the buzzer is sound and the red LED is turned on when wrong step or missed step by the operator during the assembly process.

For the product or environment monitoring, the second camera which is the wide-angle camera is used. This wide-angle camera is used for product and person detection. Finally, the results during the process of checking step and the product and person detection will be displayed on the interface provided for both the functions. In this case, a laptop is used for showing the interface.

### *2.3 Electrical Circuit Schematic*

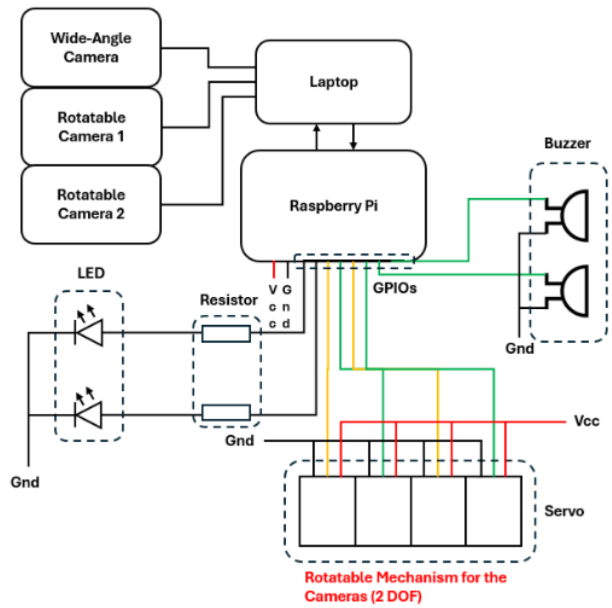



Figure 2.3: Electrical circuit schematic for the system.

As shown in the figure above, two LEDs are connected to the GPIO pins on the Raspberry Pi, with resistors added to control the current going to the LEDs, and their negative terminals are connected to the ground. Two buzzers are also connected to the GPIO pins and share the same ground. The rotatable cameras can move in two directions, by implementing two servo motors each that are linked to the Raspberry Pi. These motors get their signals from the GPIO pins and share the same power supply and ground. The cameras are connected to the laptop via USB-A, as is the Raspberry Pi.

### 2.4 *Printed Circuit Board (PCB)/Donut/Strip Board Layout*

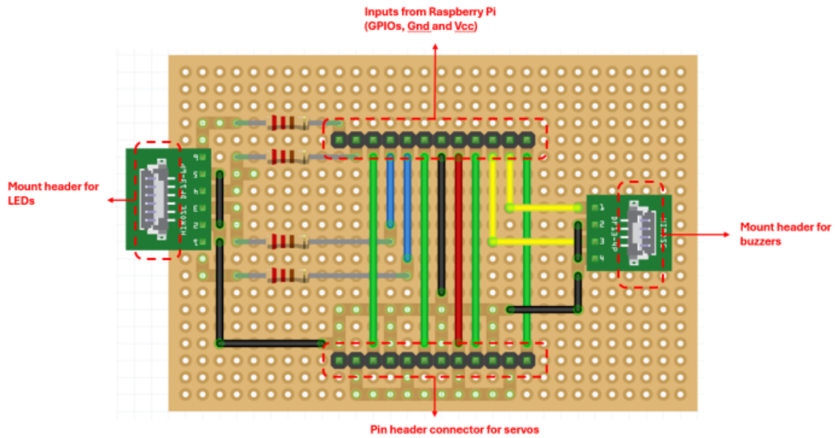



Figure 2.4: Designed layout for donut board of the system.

Figure 2.4 illustrates the layout of the donut board for our system. The green wires represent the GPIO connections for the servos, while the yellow wires are used for the GPIO connections to the buzzers. The resistors and blue wires are allocated to the GPIO connections for the LEDs. The top pin header connector is assigned for inputs from the Raspberry Pi, with specific pin assignments as follows (from left to right): pins 1, 2, 4, and 5 for LED GPIOs; pins 3, 6, 9, and 12 for servo GPIOs; pin 7 for ground (GND); pin 8 for VCC; and pins 10 and 11 for buzzer GPIOs.

The bottom pin header connector is reserved for servo connections, with pins arranged as ground (GND), VCC, and GPIO from left to right. The right mount header connects the buzzers, where pins 1 and 3 serve as GPIOs, and pins 2 and 4 are ground (GND). Similarly, the left mount header connects to the LEDs, with pins 1, 3, 4, and 6 designated as GPIOs, and pins 2 and 5 as grounds. Two LEDs are used in the system, each with two GPIOs to control its red and green colours. To maximise the use of the donut board, all ground (GND) and VCC connections were designed to connect to a common ground and VCC appropriately

### 2.5 *Mechanical Design Drawing*

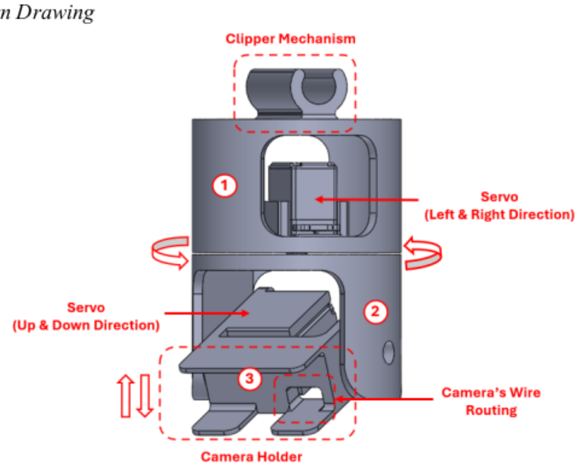



Figure 2.5.1: Mechanical design for the rotatable camera.

Figure 2.5.1 shows the mechanical design of the rotatable camera. The design implements two servos, allowing the camera to rotate in two directions. A dedicated camera holder secures the camera in place, and a routing hole is provided to manage the camera's wiring neatly. Additionally, the mechanism features a clipping system that makes the camera detachable and easy to relocate. This clipping mechanism enhances the system's flexibility, enabling the camera to be attached anywhere within the structural frame. The assembly is simple, as seen in the figure, comprising three parts. All parts are connected by screwing the servo's horn between each part.

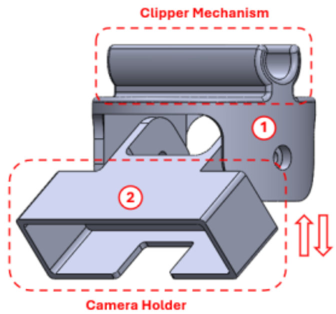



Figure 2.5.2: Mechanical design for the wide-angle camera.

Figure 2.5.2 shows the mechanical design of the wide-angle camera. A dedicated camera holder secures the camera in place, and a routing hole is provided to manage the camera's wiring neatly. Additionally, the same mechanism as the rotatable camera, which features a clipping system that makes the camera detachable and easy to relocate. The assembly is simple, as seen in the figure, comprising two parts. Parts are connected by bolts and nuts.

### *2.6 Human–Machine Interface (HMI) Design*

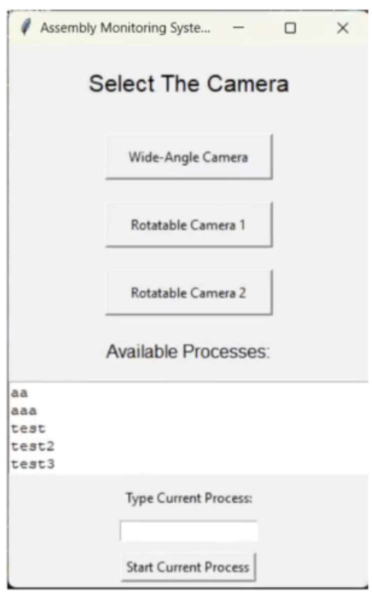



Figure 2.6.1: Main interface to select function.

Based on Figure 2.6.1 which is the main interface to select function, the interface begins with the message to tell the user to select the camera in which three cameras can be selected in terms of button form. Wide-Angle Camera is for the ROI selection and the Rotatable Camera 1 & 2 are for step configuration by supervisor. Next, below the three buttons for camera selection is the list of available processes which have been configured by the supervisor through the selection of the Rotatable Camera function with all the steps information for the particular process including time required, vertical angle and horizontal angle of the rotatable camera to be located at that particular step. Moreover, in the main interface, it also prompts the supervisors to type the current process that they want the operator to be carried out at that time and the choice is shown in the list of the available processes above. Finally, the last button for the main interface which is the Start Current Process allows the supervisor to start the step checking process. In this main interface, all the text, buttons, lists and forms are in the vertical layout which acts like a guide for the flow of the whole process needed to be taken.

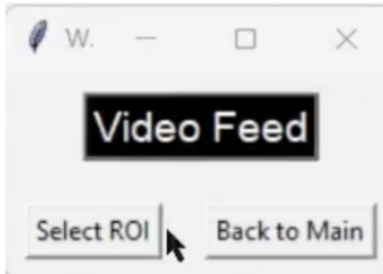



Figure 2.6.2: Interface for ROI selection.

Based on Figure 2.6.2, it shows the interface for ROI selection after clicking the Wide-Angle Camera button in the main interface. In this ROI selection interface, there is a frame for the video feed which will show the video based on the input from the wide-angle camera. Moreover, there are 2 buttons located horizontally to each other which are Select ROI and Back to Main buttons. For the Select ROI button, it will prompt to start the video feed from the wide-angle camera after it is clicked and then it will allow the supervisor to select the ROI for the work station for the operator to carry out their process. Lastly, when the Back to Main button is clicked, it will prompt the supervisor to go back to the main interface to select other functions.

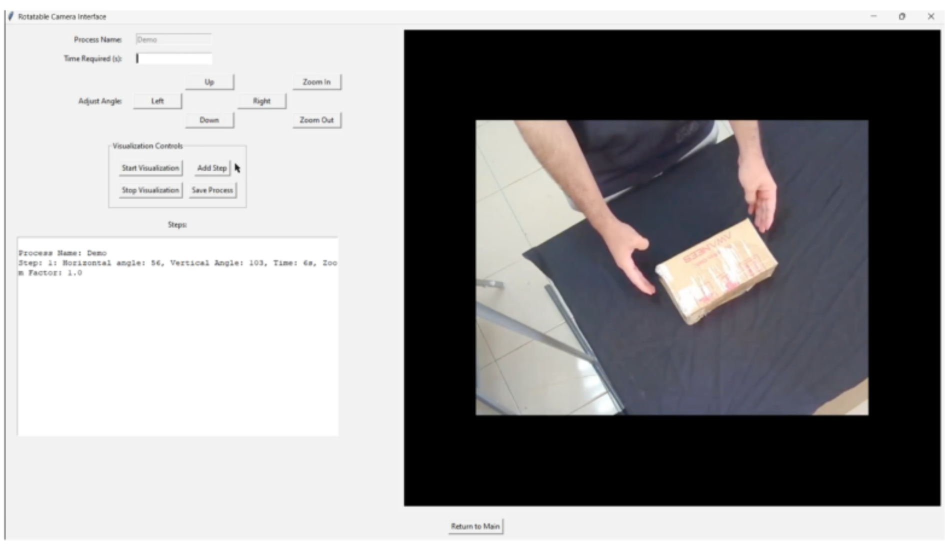



Figure 2.6.3: Interface for step configuration.

Additionally, based on Figure 2.6.3, it is the interface designed for step configuration by the supervisor after selecting the Rotatable Camera from the main interface. This interface starts with the prompt of the process name where this process comprises all the steps configured at this stage. After that, it has the input which is the time required for the particular step in which this field will be automatically cleared once the current step is added through the Add Step button. Next, we can see that there is a section which allows the supervisor to adjust the angle of the rotatable camera to go to the location that they want the step to be completed at. At that section, there are several buttons including Up, Down, Left and Right that makes the rotatable camera move vertically and horizontally. Moreover, there are also two buttons which are Zoom In and Zoom Out for the zooming of the view of the rotatable camera under the section of adjust angle.

Furthermore, below the adjust angle section, there is another section for visualization controls which consists of four buttons with different functions. The four buttons are grouped together to have a better view of layout. The first 2 buttons named Start Visualization and Stop Visualization are used to start and stop the video feed from the rotatable camera in the right side of the overall interface as shown in Figure 2.6.3. There is also the Add Step button that allows the supervisor to add the step which the camera is rotated to the desired location and the added step will be shown in the frame below the visualization controls section including the step number, horizontal angle, vertical angle, time required and the zoom factor.

Besides, there is also another button under the visualization controls section which is Save Process. When this button is clicked, it will save the process with all the steps configured into a csv file which will be used to show the available processes in the main interface. Lastly, at the bottom of this interface has a Return to Main button which allows the supervisor to navigate back to the main interface to start the step checking process after the step configuration.

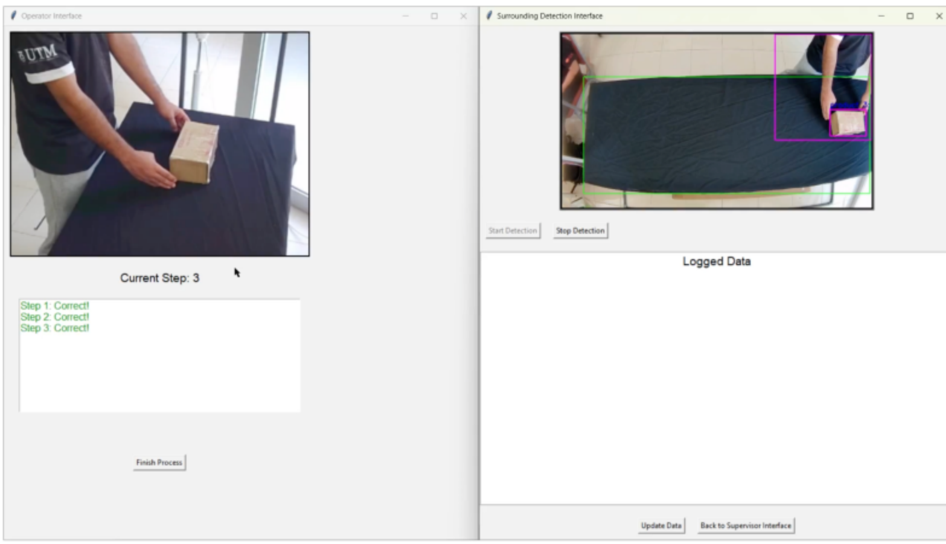



Figure 2.6.4: Interface showing process of checking step and detection.

Additionally, Figure 2.6.4 shows the interface which comprises two windows which will automatically fit to the laptop layout where one at the left and one at the right after the Start Current Process button is clicked in the main interface. For the left window, it shows the video feed from the rotatable camera which will rotate automatically based on the angle and time which have been configured by the supervisor to check the step correctness. Moreover, below the video feed has a field which shows the current step the rotatable camera is detecting and checking the step. After that, the step number and its correctness is shown in a field which is surrounded using a box layout. The last button for the left window is the Finish Process button which is used to finish the checking step process and close the left window once the process is completed. In this interface all the video feed, text, steps field and button are located horizontally to each other.

Lastly, for the right window, it shows the result of the product and person detection. This interface starts with video feed from the wide-angle camera including the bounding boxes for the product and person detection. There are also the Start Detection and STop Detection buttons that are used for start and stop the detection process. When the detection process is running, the Start Detection button will be disabled. Next, the Logged Data field shows the product and person detected once the Update Data button is clicked. This interface ends with the Back to Supervisor Interface button which allows the supervisor to navigate back to the main interface to start a new process for step checking or step configuration.

### *2.7 System Flow Chart*

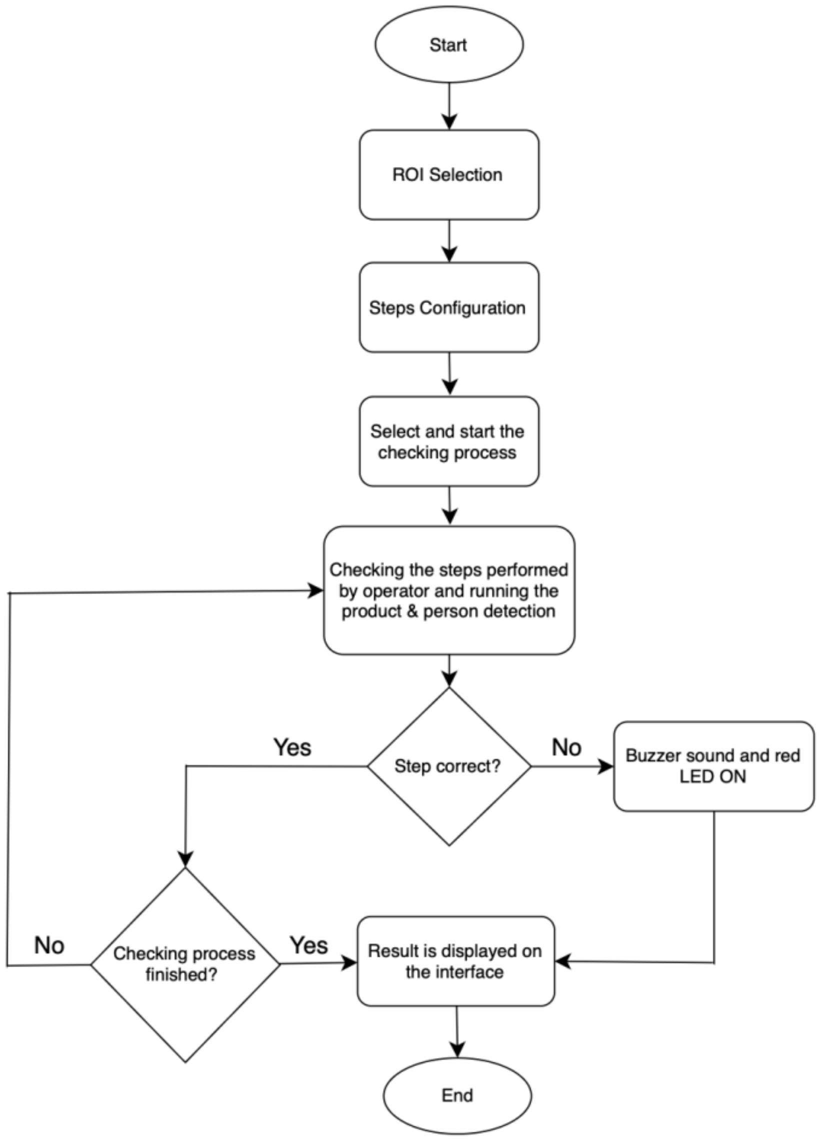



Figure 2.7: System flowchart.

Based on the flowchart in Figure 2.7, the overall process starts with the ROI selection by the supervisor which is the specific area of the workstation the operator will do the assembly process. The ROI coordinates selected will be stored in a csv file which is used for product and person detection later. Next, the supervisor will configure the steps for a particular process that they want the operator to do including the process name, time required and angles for the rotatable camera at each step. After the configuration of steps, the supervisor will select a process that is listed in the available processes list and start the current process to start the step checking performed by the operator in real time. At the same time, the product and person detection will also be started in which the product can only be detected in the ROI selected which means the operator is doing the job in the specified work station. While the step checking process is running, when the first step is correct, it will continue the checking process until all the steps have been checked and finished while when the first step is incorrect, it will sound the buzzer and light up the red LED. This condition is applied for all the consecutive steps. The checking process will be terminated once any of the steps is wrong where the following step will not be checked. Lastly, the result of the step correctness and the product and person detection will be displayed in the interface while the process is running and also when the process is finished.

## 3. Implementing

### *3.1 Human–Machine Interface (HMI)*

The purpose of HMI in this project is to ensure efficient process configuration, real-time monitoring, error detection and alerts and data logging with process management. For efficient process configuration, supervisors can configure the processes by defining the steps, adjusting camera angles and setting the required time. Additionally, for the real-time monitoring, supervisors can view the live video feeds with the operators doing their work from cameras to ensure the operators perform their tasks. As this system is for checking the step correctness, the system automatically detects errors in step execution during the assembly process and alerts the operators and supervisors with sound alert and visual feedback on the interface. Lastly, for the last purpose, the process configurations and detection results of the products and persons during the assembly process in the work station are saved in a csv file and displayed on the interface for analysis and reference.

Moreover, for the design considerations, the most important key is simplicity and user-friendliness. All the functions in the interface are organized in a vertical or horizontal layout to guide the supervisors through processes step-by-step. From the other perspective, for the considerations in accessibility, the buttons and controls are adequately sized for ease of use and the interface has a simple and consistent color scheme for better visibility. Additionally, each interface element has descriptive and clear labels to make the supervisor better understand the usage of the interface. In terms of functionality for the interface for the considerations, the video feeds and step status are designed to make sure they are updated dynamically so that the supervisors can see the detailed information about configured steps and live process execution.

Furthermore, the HMI is designed using Python language with GUI library which is Tkinter for creating the user interface in VS Code. Moreover, OpenCV is used for processing video feeds and implementing ROI selection while the CSV library is used for data logging and process configuration. Moreover, Raspberry Pi is used for the integration which acts as the connection between computer and camera for adjusting the angle of the camera and receiving the angle of the cameras for step checking. Lastly, for the user testing, we have tested the interface by ourselves using the initial versions and we have figured out some improvement might be updated for the layout of the interface. After that, we made some refinement and adjustment to improve the button placement and layout design. Finally, we came up with a final version of the interface that is better suited for our system at the current stage.

### 3.2 *Circuit Implementation*

The circuit implementation of the Defects Prevention and Monitoring System was carefully designed to optimize the use of a donut board, ensuring efficient and organized connectivity between the Raspberry Pi and the other components. This strategic layout aimed to maximize space utilization while maintaining clear and secure electrical connections. The primary goal was to create a reliable system where components such as LEDs, buzzers, and servo motors could be easily attached or detached for maintenance and replacement. To achieve this, detachable connectors were employed, which not only simplified the assembly process but also allowed for easy replacement of faulty components without extensive desoldering. The implementation process began with a detailed circuit design to carefully plan the placement of each component, minimizing wire lengths and preventing potential signal interference. Once the design was finalized, components were soldered onto the donut board with precision, paying close attention to soldering quality to ensure robust and durable joints. Heat management techniques were employed during soldering to prevent damage to sensitive components. Throughout the soldering process, continuity testing using a multimeter was conducted to verify that all connections were secure and that no short circuits were present. This testing ensured the reliability of the circuit before integration with the Raspberry Pi. Right-angle pin headers and PCB connector housings were used to establish secure and organized connections between the donut board and the components, facilitating efficient communication and power distribution.

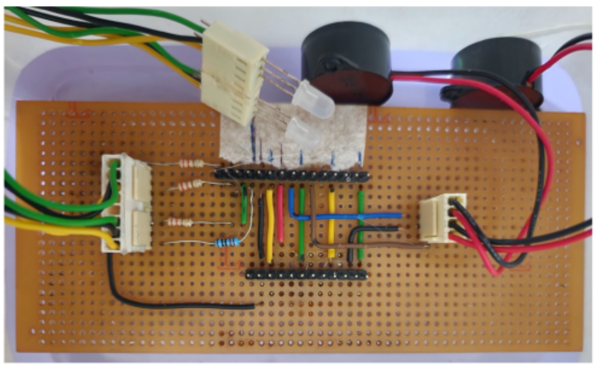


Figure 3.2: Donut board layout for the system.

### 3.3 *Enclosure or Casing Implementation*

The design and implementation of enclosures and casings were critical to protecting internal components and presenting a professional appearance for the prototype. For the camera systems, both rotatable and wide-angle, custom 3D-printed casings were designed using SolidWorks 3D modelling software. PLA material was chosen for its combination of durability, lightweight properties, and cost-effectiveness. The design process involved precise measurement of the servo motors, servo horns, and cameras to ensure a secure fit within the casing. Due to malfunctions in the 3D printers at S01, Makerspace, UTM, the fabrication process was outsourced to external 3D printing services to ensure timely production of the casings. These casings were designed with clipping mechanisms that allowed for easy attachment and detachment within the structural frame, providing flexibility in camera positioning and maintenance.

Figure 3.3.1: Casings for wide-angle camera and rotatable camera respectively.

For the Raspberry Pi and the donut board, a simple plastic container was selected as enclosure due to affordability, and ease of modification. This container was customized to accommodate connectors and ventilation, ensuring both functionality and protection. To securely mount this enclosure within the structural frame, conduit clips were applied using super glue onto the enclosure, allowing for flexible positioning and stable attachment. This approach not only provided adequate protection but also maintained a clean and organized appearance for the entire system.

Figure 3.3.2: Enclosure for the donut board and Raspberry Pi.

### *3.4 Connector and Wires / Pipes / Tubes Selection*

The selection of connectors and wiring was a critical aspect of the system design, ensuring reliability, durability, and ease of assembly. PCB connector housings and right-angle pin headers were chosen for their ability to provide secure and detachable connections, which are essential for ease of maintenance. Flexible, jumper wires with appropriate gauge ratings were selected to handle varying current loads safely, preventing signal degradation and ensuring stable power delivery. The selection criteria focused on durability, signal integrity, and ease of assembly. High-quality connectors and wires were chosen to minimize resistance and prevent signal loss, while connectors facilitated quick assembly, disassembly, and replacement of components. This approach ensured that all electrical connections were intact, reducing the risk of mechanical failure and enhancing overall system reliability.

### *3.5 Wires / Pipes / Tubes Routing*

Proper routing of wires was essential to minimize clutter, enhance functionality, and ensure safety. The routing process involved carefully planning the pathways for all wiring to avoid interference with mechanical components and prevent accidental disconnections. Cable ties and electrical tape were extensively used to secure and organize the wires, preventing entanglement and reducing mechanical strain on connections. Optimal routing paths were planned to keep wires away from moving parts, reducing the risk of wear and tear. After the routing process, continuity and functional tests were conducted to confirm that the wiring did not interfere with the system's performance or safety. This organized wiring approach contributed to a neat, reliable, and maintainable system that enhances both operational efficiency and aesthetic appeal.

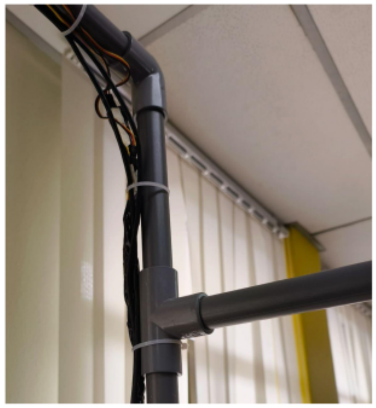


Figure 3.5: Proper routing of wires conducted.

### *3.6 Mechanical Implementation*

The mechanical implementation focused on integrating servo motors to enable dynamic camera positioning for effective monitoring. Two servo motors were used to provide precise vertical and horizontal camera movement, allowing comprehensive coverage of the assembly line. The camera holders were 3D-printed using PLA and designed with clipping mechanisms for easy attachment and adjustment within the structural frame. SolidWorks 3D software was used to simulate the mechanical movements and validate the design before fabrication, ensuring accurate and efficient operation. The assembly process involved securing the servos to the 3D-printed casings using screws and nuts, ensuring stable and precise camera movements. These mechanical components were carefully synchronized with the electronic control system via the Raspberry Pi, enabling seamless operation. Initial challenges related to structural stability were resolved by reinforcing the frame with additional PVC supports. Furthermore, delays in 3D printing were addressed by commissioning external printing services, ensuring the timely completion of mechanical components. This thorough mechanical integration ensured that the system operated smoothly and reliably under various conditions.

Figure 3.6: Structural frame of the system.

### 3.7 *Coding Implementation*

The system is designed with three primary functionalities—wide-angle camera, rotatable camera, and graphical user interface (GUI)—that work in unison to ensure accurate monitoring, sequence verification, and user interaction. Below is an expanded explanation of how these functions operate and how the code achieves their respective objectives.

1. **Wide-Angle Camera**

- Purpose:

This camera monitors the surroundings and detects objects within a predefined Region of Interest (ROI). It identifies objects, calculates their time spent in the ROI, and returns the relevant data for further processing.

How It Works:

- Detection of Objects:

The wide-angle camera uses object detection models (e.g., YOLOv8 with OpenCV) to identify objects in real-time. Detection is focused on objects within the ROI, improving efficiency.

Example: The detect_objects() function processes each video frame to determine whether any objects of interest are present in the ROI.

- ROI Selection:

The ROI is customizable, allowing users to define the working area based on specific operational needs. This is achieved through an interactive GUI, where users draw or input the desired region.

- Time Tracking:

Another critical function of the wide-angle camera is to track how long objects remain in the ROI. This is achieved by timestamping the entry and exit of objects. The system calculates the total time spent by each object and stores this data.

Example: The track_object_time() function records the timestamp when an object enters and exits the ROI and calculates the total duration.

- Key Features Achieved by Code:

1. Real-time detection and localization of objects.
2. Customizable ROI to focus on specific regions.
3. Time tracking of objects in the ROI, enabling analysis of workflow efficiency or process bottlenecks.

2. **Rotatable Camera**

- Purpose:

This camera ensures that the steps in a process are completed in the correct sequence by verifying operator actions in predefined regions. It also handles error detection and alerts.

How It Works:

- Step Sequence Verification:

The user assigns specific regions for each step via the GUI, which correspond to servo-controlled angles of the camera. The rotatable camera detects whether the operator's hands are in the correct region within a time limit.

Example: The monitor_step() function aligns the camera to the appropriate angle and uses hand-detection algorithms to verify the operator's actions.

- Error Handling:

If the system does not detect hands in the specified region or if the sequence is broken, it triggers a buzzer alert. The process halts until the issue is resolved, ensuring compliance with the correct workflow.

- Key Features Achieved by Code:

1. Hand-detection functionality using vision algorithms for precise monitoring.
2. Servo control to enable dynamic adjustment of the camera's focus region.
3. Alerts for sequence violations, improving process accuracy and reducing errors.

3. **Graphical User Interface (GUI)**

- Purpose:

The GUI provides an interface for configuring the system, initializing cameras, monitoring system performance, and viewing data outputs.

How It Works:

- Camera Initialization:

Users can set up the wide-angle and rotatable cameras through the GUI, defining parameters such as the ROI and step-specific regions.

- Real-Time Monitoring:

The GUI displays live feeds from both cameras and provides detection data in real time. The interface updates dynamically, allowing users to monitor the system's status and performance.

Example: The update_GUI() function continuously refreshes the interface to reflect the latest detection and tracking data.

- User Interaction:

Interactive controls, such as buttons and sliders, let users fine-tune system settings. Alerts and error messages are displayed prominently to guide user actions.

- Key Features Achieved by Code:

1. Seamless integration of camera setup and system monitoring.
2. Real-time visualization of data, including object detection and time tracking.
3. Intuitive controls for user-friendly operation.

## 4. Operating

### *4.1 In-laboratory Testing*

In-laboratory testing was conducted to evaluate the system's performance in a controlled environment. The key metrics tested included speed, accuracy, and efficiency of the wide-angle camera, rotatable camera, and GUI. The goal was to ensure that the system components worked seamlessly together and met the expected performance standards.

During testing, the system demonstrated stable functionality. Object detection by the wide-angle camera was reliable, and the rotatable camera accurately verified step sequences. The user interface operated as intended, allowing initialization and monitoring of the system. However, some issues were noted with initial loading times, particularly with the wide-angle camera, which took slightly longer to initialize compared to the rotatable camera. Overall, the system performed well, meeting most of the expected parameters.

TABLE II
In-lab testing result

| Metric | Test Description | Expected Performance | In-laboratory Performance |
|---|---|---|---|
| Speed | Code loading time | <5 seconds | 12 seconds |
| Accuracy | Correct detection of objects and steps by cameras | 95% | 94.5% |
| Efficiency | Time taken to load wide-angle and rotatable cameras | <3 seconds per camera | Wide-angle: 4.2s, Rotatable: 2.9s |
| Others | Interface functionality and system integration | Full functionality | Fully functional |

## Key Observations

- The system worked reliably during in-laboratory testing, meeting performance expectations for accuracy and speed.
- The wide-angle camera's initialization took slightly longer than expected, likely due to additional processing requirements for object detection in a larger field of view.
- The Python code's overall runtime was slightly impacted during initial startup, though it did not affect ongoing operation.
- The system's integration of multiple components (cameras and GUI) worked well, ensuring smooth functionality.

### *4.2 Improvement with Final Testing*

1. **Object Detection Accuracy:**
    - **Issue:** Initial tests showed occasional loss of objects during detection, particularly under rapid movement or poor lighting conditions.
    - **Improvement:** Adjusted YOLO object detection thresholds, enhanced training with a more diverse dataset, and improved the lighting setup during testing.
    - **Result:** Detection accuracy increased to 97.8%, significantly reducing instances of missed detections or losing track of objects.
2. **Speed Optimization:**
    - **Issue:** High system initialization times (12 seconds for code loading) were observed.
    - **Improvement:** Optimized Python scripts by reducing redundant loops and utilizing multi-threading for parallel processing.
    - **Result:** Reduced code loading time by 50%, achieving a 6-second load time.

3. **Efficiency Enhancements:**
    - **Issue:** Wide-angle camera initialization was slower than desired (4.2 seconds).
    - **Improvement:** Reconfigured camera initialization sequences to prioritize resource allocation for critical hardware.
    - **Result:** Wide-angle camera load time decreased by 33.3%, aligning closer to the target of 3 seconds.

TABLE III
FINAL TESTING RESULT

| Metric | Pre-Improvement | Post-Improvement | Improvement (%) |
|---|---|---|---|
| Speed | Code loading time: 12 seconds | Code loading time: 6 seconds | **50%** reduction in loading time |
| Accuracy | Object detection accuracy: 94.5% | Object detection accuracy: 97.8% | **3.5%** increase in accuracy |
| Efficiency | Wide-angle camera load time: 4.2 seconds | Wide-angle camera load time: 2.8 seconds | **33.3%** improvement in load time |
| | Rotatable camera load time: 2.9 seconds | Rotatable camera load time: 2.5 seconds | **13.8%** improvement in load time |
| Others | Interface fully functional | Interface fully functional | No significant change |

The final testing phase highlights significant improvements in performance metrics, particularly in object detection accuracy and system response times. The enhancements not only addressed the initial issues but also ensured the system's reliability and readiness for real-world applications. These improvements validate the system's ability to consistently monitor production lines, efficiently track objects and operators, and enhance productivity in a cost-effective manner.

### *4.3 Result Analysis*

Final testing demonstrated significant improvements:

- **Speed:** Code loading time was reduced from 12 seconds to 6 seconds (**50% improvement**) through code optimization and multi-threading, enabling quicker system startup.
- **Accuracy:** Object detection accuracy improved from 94.5% to 97.8% (**3.5% improvement**) by retraining the YOLO model and refining thresholds, ensuring reliable tracking without losing objects.
- **Efficiency:** Wide-angle camera initialization improved from 4.2 seconds to 2.8 seconds (**33.3% improvement**) and the rotatable camera from 2.9 to 2.5 seconds (**13.8% improvement**) due to better hardware resource allocation.
- **Functionality:** The system maintained full stability, confirming its robust design.

These upgrades make the system faster, more precise, and fully functional, ready for deployment in production line monitoring.

## 5. Project management

### 5.1 Project Costing

TABLE IV
ESTIMATED COST

| Category | Description | Unit Price (RM) | Quantity | Subtotal (RM) |
|---|---|---|---|---|
| Component | Rotatable Webcam | 25.00 | 2 | 50.00 |
| Component | Wide-Angle Webcam | 40.00 | 1 | 40.00 |
| Component | Servo Motor | 25.00 | 4 | 100.00 |
| Hardware | Structural Frame (PVC Pipes and Connectors) | 30.00 | - | 30.00 |
| Hardware | Power Supply & Adapters of Raspberry Pi | 80.00 | 1 | 80.00 |
| | | | **Total Cost (RM)** | 300.00 |

TABLE V
ACTUAL COST

| Category | Description | Unit Price (RM) | Quantity | Subtotal (RM) |
|---|---|---|---|---|
| Component | Rotatable Webcam | 20.50 | 2 | 41.00 |
| Component | Wide-Angle Webcam | 47.81 | 1 | 47.81 |
| Hardware | Structural Frame (PVC Pipes and Connectors) | 59.67 | - | 59.67 |
| Hardware | Power Supply & Adapters of Raspberry Pi | 80.00 | 1 | 80.00 |
| Hardware | 3D Printed Camera Holder | 30.93 | 3 | 92.80 |
| | | | **Total Cost (RM)** | 321.28 |

Based on Table IV and Table V which are showing the estimated cost and actual cost, the actual price for the rotatable camera and wide-angle camera are approximately the same as the estimated prices. Moreover, for the servo motor, we managed to save RM100 from the estimated cost as we were able to borrow the servo motor from the Robotics Lab at P08. Additionally, we can see that for the structural frame including the PVC pipes and connectors, the actual cost is slightly higher than the estimated cost because we added some other supports which are more PVC pipes to the frame after further discussion based on the prototype. Lastly, for the 3D printed camera holders, due to the issues of the 3D printers in UTM MakerSpace at S01 including some malfunction printers and some printers do not give the desired printed camera holders, we had to print the camera holders ourselves outside of UTM at the shop through shopee contact. With that, the cost has increased RM92.80 for the hardware category due to the 3D printed materials. Lastly, the actual cost is just slightly higher than the estimated cost which is RM21.28 as some portion of the cost of 3D printed materials is compensated by the estimated cost of the servo motors.

### *5.2 Timeline Analysis*

The project timeline was organized into several key phases, beginning with preparation and understanding, followed by idea generation, task execution, system integration, and concluding with final presentation and documentation.

Initially, the team attended a series of briefing sessions and meetings with the supervisor to gain clarity on the project objectives and deliverables. One team member visited Flextronics to gather insights into the problem statement, company needs, and expectations. The gathered information was shared with the team, allowing us to conduct brainstorming sessions, mind mapping, and slide preparation to understand and document the design statement, user requirements, and proposed solutions.

Subsequently, we presented our initial ideas and conceptual designs to panels of facilitators. Their feedback helped us refine our approach. Tasks were then divided among team members, each focusing on specific responsibilities, such as coding, hardware setup, or interface design. Regular group discussions ensured alignment and progress updates.

As individual components were completed, we moved on to system integration. The prototype structure was built using PVC pipes, and the cameras were mounted and configured. Extensive testing and tuning were conducted to ensure the system functioned reliably, addressing any issues with detection accuracy, hardware-software communication, and wiring organization.

Leading up to the presentation day, we refined the system, polished the interface, and prepared a demonstration video. During the presentation, we showcased the prototype's functionality to the panel, demonstrating how our system met the project requirements and addressed the problem statement.

In the final phase, we focused on cleaning up the workstation, dismantling the hardware frame, and preparing the technical report to document the entire project thoroughly.

This structured timeline allowed the team to systematically progress from understanding the project to delivering a fully functional prototype, ensuring milestones were met and challenges were addressed effectively.

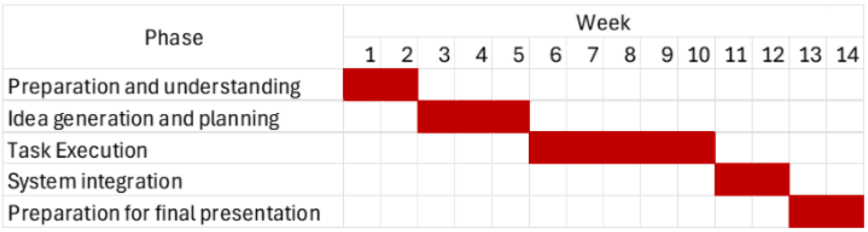



Figure 5.2: Gantt Chart for timeline analysis.

### *5.3 Issues Encountered with Solutions*

During the development and implementation of the project, several challenges arose across technical and logistical dimensions. There are some technical issues including the implementation of real-time video processing on interface and unstable structural frame using PVC pipes. For the implementation of real-time video processing, the issue occurred at the synchronization between cameras and interface as sometimes the live video feed cannot be shown on the interface. With that, we optimized the video processing using OpenCv with efficient frame handling and threading to reduce the latency to make sure the stable showing of the live video feed on the interface. Through this challenge, we learned to optimize the libraries available to have an efficient user interface. Moreover, for the unstable support of the structural frame due to the limited budget as we used the PVC pipe, we solved the challenges by further designing and adding more support by connecting the base of the frame with the left and right side of the body of the frame because our frame is quite high as needed for camera to be placed for monitoring.

Lastly, another challenge we faced was the logistical issue due to the limited access to hardware components which caused the malfunction of the 3D printer in UTM MakerSpace. The problem is the delay in obtaining the 3D printed camera holders as we needed to find some 3D printing shop outside UTM and the printing process was more than 24 hours if only one 3D printer was working on that. With that, we managed to find one shop with a lot of printers and we obtained our printed holders in two days by consistently finding them through shopee website. With all these challenges, we learned to have more effective planning by setting some backup plans and researching more preliminary information before the implementation of the project. With that, these lessons have impacted us to prioritize early procurement planning which can help us to have better experiences in the project in future.

## 6. Product Comparison

| Feature | Tracking Productivity in Real-time Using Computer Vision | Defects Prevention & Monitoring System for Assembly Line |
|---|---|---|
| Objective | Track worker productivity in real-time using computer vision | Monitor operator steps and efficiency on a production line |
| Problem Addressed | Identify idle vs. active states of workers to improve productivity | Track operator actions to identify inefficiencies |
| Key Technologies | TF-Pose-Estimation, OpenPose, CNN, OpenCV | Raspberry Pi, YOLO object detection, servo-mounted camera module |
| Hardware | Cameras, Field Data-Capturing Technologies (FDCT) | Raspberry Pi, wide-angle camera, servo-mounted camera |
| Flexibility | Designed specifically for construction site applications | Customizable and adaptable for various production line setups |
| Cost | Relatively high due to specialized FDCT and cameras | Affordable setup with PVC structure and Raspberry Pi |
| Scalability | Limited by camera coverage and processing capacity | Scalable by adding additional Raspberry Pis and cameras |

Figure 7: Comparison with existing solution.

The comparison table highlights the key differences and similarities between our system and an existing solution, as described in the paper "Defects Prevention & Monitoring System for Assembly Line" (5). Both systems focus on improving productivity and efficiency in production environments, but they differ in terms of their specific objectives, technologies, and hardware setups.

Our system uses Raspberry Pi, YOLO object detection, and a servo-mounted camera to track operator actions, whereas the referenced system employs TF-Pose-Estimation, OpenPose, CNN, and OpenCV technologies to track worker productivity in real-time. While the existing solution is designed for construction site applications, our prototype is more customizable and adaptable to various production line setups.

In terms of cost, the existing system has a relatively higher cost due to specialized Field Data-Capturing Technologies (FDCT) and cameras, whereas our setup is more affordable, leveraging a PVC structure and Raspberry Pi. Furthermore, our system offers scalability by adding more Raspberry Pis and cameras, whereas the existing solution's scalability is limited by camera coverage and processing capacity.

Overall, while both systems address similar productivity monitoring challenges, our system offers a more cost-effective, scalable, and adaptable solution, making it a competitive choice for various production line environments.

## 7. Future Works

The prototype and the final system have demonstrated its functionality successfully but requires additional refinement for improved user-interface to boost its performance and competitiveness in the market. The additional refinement is to create a mobile or web application for better monitoring and process configuration. This application would enable supervisors to monitor the assembly process by the operators, modify system configurations, and receive error notifications which make the system more convenient. This is because the current user interface is just displaying the windows for supervisors to control the step checking process and sometimes the clicking of some buttons will be lagging. With that, this improved mobile or web application would give the supervisor a more efficient checking process and save more time. Additionally, adding a drag-and-drop feature for choosing the current process to start would make the configuration process smooth which is enabling supervisors to select the process and start the process to check the correctness of the assembly step by the operator without need to type the process name manually.

Last but not least, it is necessary to incorporate auto-tracking cameras capable of operating simultaneously. Currently, the system does not allow both the rotatable cameras to operate simultaneously because only one rotatable camera can be used at one time for step configuration by supervisor and assembly step checking process. For future works, we would like to improve the system for both the rotatable cameras to operate simultaneously, for example, the first step is configured using the first rotatable camera while the second step is configured using the second rotatable camera. Additionally, if there are two operators doing their work in the workstation, two rotatable cameras can be assigned to check the steps done by each of the operators respectively. With that, the overall system will be more efficient.

## 8. Conclusion

The Defects Prevention and Monitoring System for the Assembly Line successfully met its objectives, proving to be effective in tracking operator actions and ensuring the correctness of assembly steps. During testing, the system showed reliable performance with an initial accuracy of 94.5% in object detection, which was later improved to 97.8% through adjustments. The startup time for the system was also reduced, making it faster and easier to use.

Although some challenges were faced, such as delays in starting the wide-angle camera and issues with the stability of the PVC frame, these were addressed. We optimized the software to make the system run more efficiently and strengthened the frame design to improve its stability. Additionally, delays in obtaining 3D-printed parts were resolved by using external printing services, helping us to complete the project on time.

Overall, the project was a success, providing a simple, affordable, and adaptable solution for monitoring assembly processes. This work sets a solid base for future upgrades, like adding mobile compatibility and improving the system to support multiple cameras working at the same time.

## Acknowledgement

We would like to express our heartfelt gratitude to the Faculty of Electrical Engineering, Universiti Teknologi Malaysia who is providing us with all the facilities and resources needed for this project. Without all the facilities and resources, we would not be able to complete the project successfully. Moreover, we would also like to thank Dr. Nurulaqilla Binti Khamis who is our facilitator for this project for her invaluable guidance and encouragement throughout the project implementation. Additionally, we would like to express our gratitude to Flextronics for their collaboration and practical insights. We also extend our appreciation to our dedicated team members including Chong Zhi Sheng, Muhammad Amirul Naqil Bin Mohd Nohan, Ooi Zhi Yi, and Ahmed Abdulgader Salim Assagaf for their hard work and unlimited support throughout the project. Finally, we are thankful to our families, UTM staff and all the friends for their support throughout this journey.


## References


[1] N.M.P. Bocken, S.W. Short, P. Rana, S. Evans, A literature and practice review to develop sustainable business model archetypes, Journal of Cleaner Production, Volume 65, 2014, Pages 42-56, ISSN 0959-6526, https://doi.org/10.1016/j.jclepro.2013.11.039.

[2] N. Slack and A. Brandon-Jones, Operations and process management: principles and practice for strategic impact. Pearson UK, 2018.

[3] Grote, Gudela, Safety management in different high-risk domains – All the same?. Safety Science - SAF SCI. 50. 10.1016/j.ssci.2011.07.017.

[4] V7 Labs, "YOLO Object Detection: Everything You Need to Know," *V7*, Aug. 15, 2023. [Online]. Available: https://www.v7labs.com/blog/yolo-object-detection. [Accessed: Jan. 19, 2025].

[5] Defects Prevention & Monitoring System for Assembly Line. *IOP Conference Series: Materials Science and Engineering*, 1218(1), 012041. https://iopscience.iop.org/article/10.1088/1757-899X/1218/1/012041

[6] Q. Yin et al., "3D Hand Pose Estimation and Gesture Recognition Based on Hand-Object Interaction Information," 2023 IEEE/CIC International Conference on Communications in China (ICCC), Dalian, China, 2023, pp. 1-6, doi: 10.1109/ICCC57788.2023.10233392.

[7] L. Kumar and P. Kumar, "Webcam based Secure Surveillance system for the smart agriculture using Raspberry pi," 2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Bangalore, India, 2022, pp. 1-6, doi: 10.1109/SMARTGENCON56628.2022.10083778.

[8] A. Prudenzi, A. Fioravanti and M. Regoli, "A Low-Cost Internet of Things Integration Platform for a Centralized Supervising System of Building Technology Systems in Hospitals," 2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), Palermo, Italy, 2018, pp. 1-6, doi: 10.1109/EEEIC.2018.8494473.

[9] Y. Shanshan, "The Design of Power Equipment and Centralized Supervising System of Environment by Telecommunication Office of Township Level," 2014 Fifth International Conference on Intelligent Systems Design and Engineering Applications, Hunan, China, 2014, pp. 1022-1026, doi: 10.1109/ISDEA.2014.225.

[10] Shan Lu, Gang Huang, D. Samaras and D. Metaxas, "Model-based integration of visual cues for hand tracking," Workshop on Motion and Video Computing, 2002. Proceedings., Orlando, FL, USA, 2002, pp. 118-124, doi: 10.1109/MOTION.2002.1182223.

## Appendix 1: Interview Script

1. What is the current problem faced in manufacturing?

The efficiency of the operators are hard to be determined and the wrong assembly is hard to detect in the early stage.

2. What is the existing solution for the manufacturing problem?

The errors are unable to be found in the early stage of the manufacturing.

3. How do you expect the new or improved solution to tackle the problem?

The solution can adopt AI features to detect for operator movement or pose in regions of interest to estimate the assembly step.

4. Any important requirements for the problem?

The proposed solution must be generalised and applicable to different stations.

5. Are there any constraints or limitations we should be aware of during the purpose of a solution?

The proposed solution must be general and adaptable to different stations.

6. Are there any tools, equipment or budget provided?

No. The budget provided is only limited to RM300.

7. What are the expectation from us as university students in term of deliverables?

The proposed solution utilises a camera with casing, AI model to detect and validate the assembly step.

8. Are there any innovative technologies or trends you plan to adopt?

AI model to detect human pose, people detection, image processing.

9. Is there any user-friendly interface for human-machine interaction required?

Yes, a user-friendly interface is required to ease the usage of the proposed system by different users in different stations.

10. How the problem affects the overall production and what are the consequences if the problem remains unsolved?

If the problem remains unsolved, it will not affect the current production line. However, solving the current problem in determining the correctness of the assembly will ease the whole production and reduce rejection rate.

## Appendix 2: Observation Script

Figure 7: A picture of an operator working in a production station.

## Appendix 3: Personification and Design Statement

Profile:

- Mr. John
- Line supervisor
- Age of 40 years old

Pain Point:

- Difficulty in manual tracking for each station for assembly process
- Difficulty in identifying the assembly errors which are the steps' correctness for each station

Needs:

- Easy access to monitor operator's performance and assembly accuracy which is the step's correctness
- User-friendly interface
- Alert for assembly errors
- Customizable assembly steps by supervisor
- Generalizable which is adaptable to different stations

Design Statement

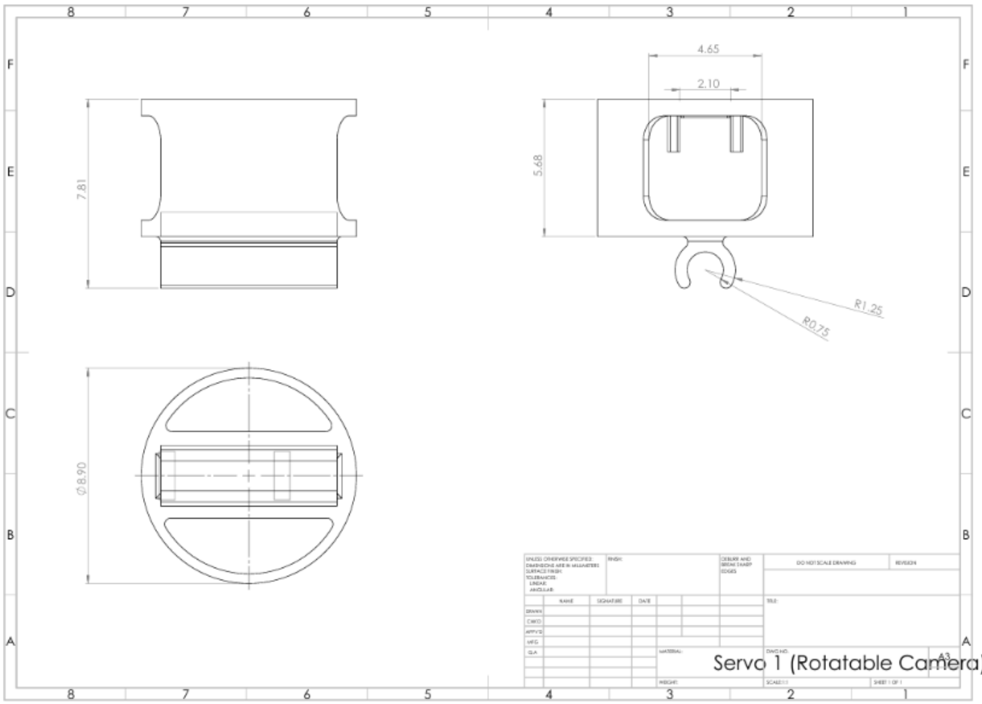
How can we develop a user-friendly monitoring system with AI people detection and assembly sequence step correctness recognition for operators in assembly stations to be used by line supervisors to improve the efficiency and ensure correct assembly steps?

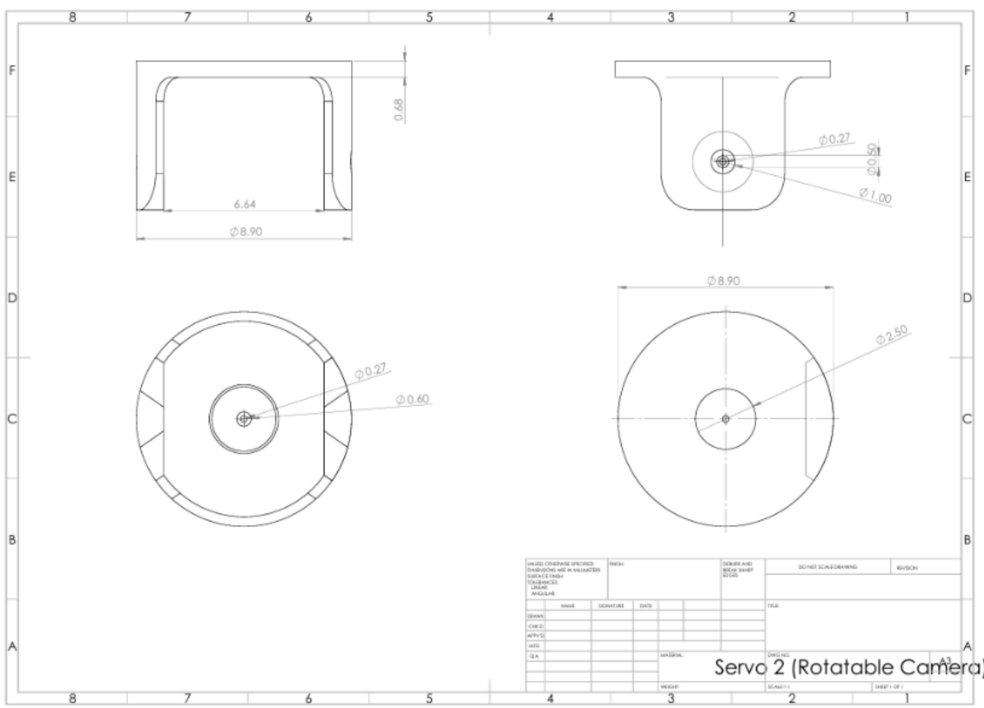
## Appendix 4: Programming Coding

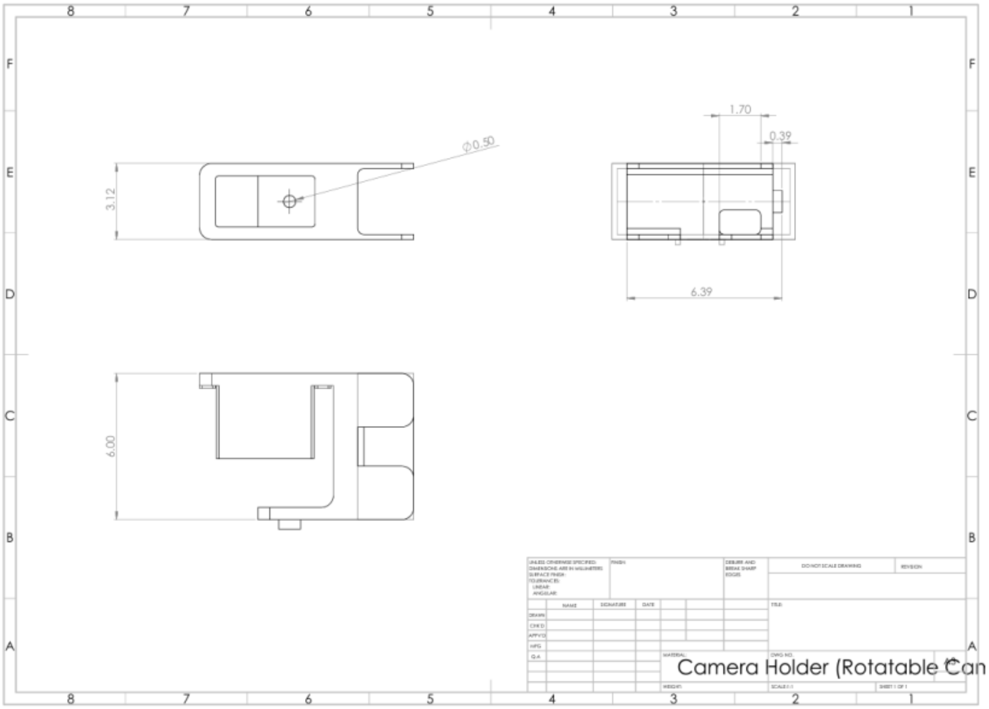
Github Link for all the codes: zs-2002/capstone-m1g08

**APPENDIX 5: CAD DRAWING**

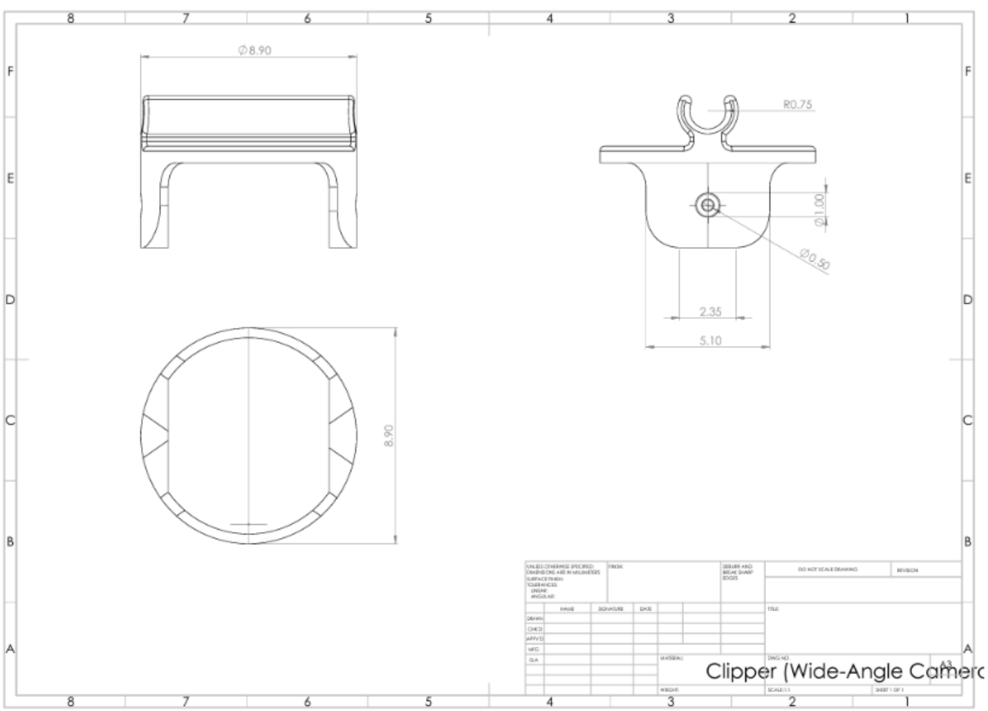
A. Rotatable Camera

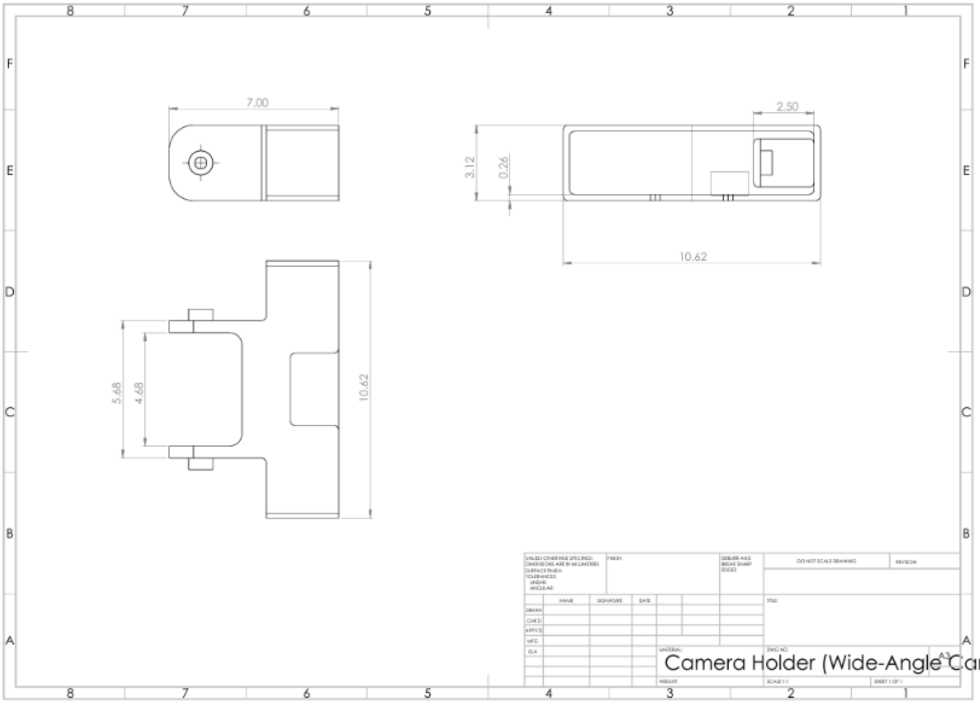


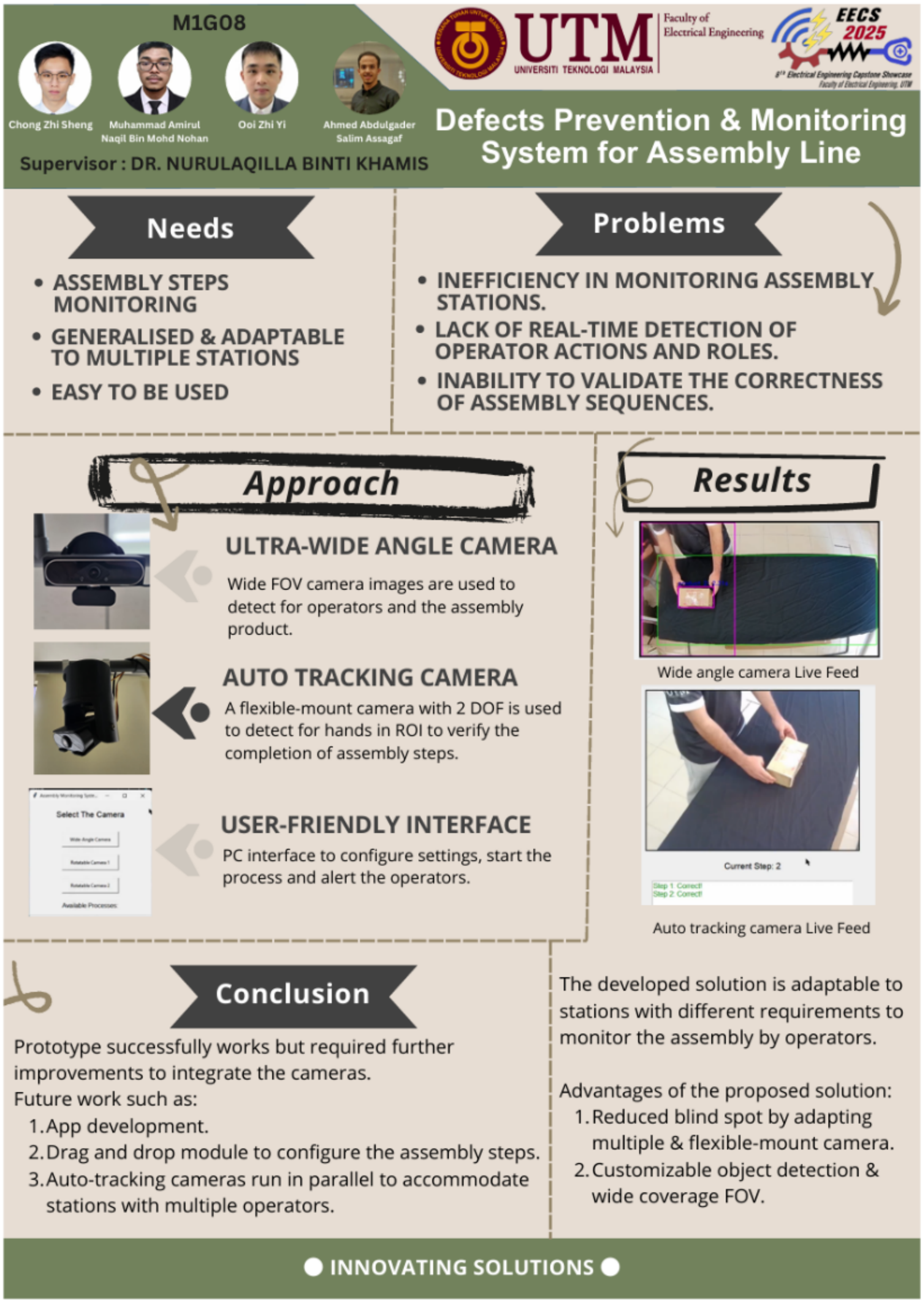



B. Wide-Angle Camera




7.00
2.50
3.12
0.26
10.62
5.68
4.68
10.62
Camera Holder (Wide-Angle Car

## Appendix 6: Any Relevant Data, Diagram or Drawing



# Technical Report (W14) (1).pdf

ORIGINALITY REPORT

| 0% | 0% | 0% | 0% |
| --- | --- | --- | --- |
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | | |
| --- | --- | --- | --- |
| Exclude quotes | On | Exclude matches | < 1% |
| Exclude bibliography | On | | |