

*Real-time Vision-based 6-DoF Pose Detector for Robotic  
Assembly Systems*  
**Test Plan**

*Group 45 – Team Four Explore*

*List of your Names:*

<b>Name</b>	<b>Position</b>	<b>Email</b>	<b>Phone</b>
<b>Daniel Leong</b>	Team Leader	<a href="mailto:102589132@student.swin.edu.au">102589132@student.swin.edu.au</a>	0431691326
<b>Parth Patel</b>	Coordinator	<a href="mailto:101604584@student.swin.edu.au">101604584@student.swin.edu.au</a>	
<b>Preenit Kshirsagar</b>	Support Manager	<a href="mailto:102486657@student.swin.edu.au">102486657@student.swin.edu.au</a>	0404655250
<b>Dylan Were</b>	Monitor	<a href="mailto:102102160@student.swin.edu.au">102102160@student.swin.edu.au</a>	

*EAT40003, FINAL YEAR RESEARCH & DEVELOPMENT PROJECT 1, SEMESTER 1 2022*

## Table of Contents

1	Resources Required .....	4
2	Features to be Tested .....	4
3	Functional Test Cases .....	4
4	Non-Functional Test Cases (Taken from SRS) .....	5
4.1	Functional Suitability: .....	5
4.2	Performance Efficiency: .....	5
4.3	Compatibility: .....	5
4.4	Usability: .....	6
4.5	Reliability: .....	6
4.6	Maintainability: .....	6
5	Strategy .....	7
5.1.1	Unit Testing: .....	7
5.1.2	Integration Testing: .....	7
5.1.3	System Testing: .....	7
5.1.4	Acceptance Testing: .....	7
5.2	Roles and Responsibilities .....	8
5.2.1	Unit testing: .....	8
5.2.2	Integration testing: .....	8
5.2.3	System testing: .....	8
5.2.4	Acceptance testing: .....	8
5.3	Test Deliverables .....	8
5.4	Schedule .....	8
5.5	Risk and Contingency with Testing .....	8
5.6	Testing Tasks .....	9
6	Pass/Fail Criteria .....	9
6.1	Product Level .....	9
6.1.1	Initial Stage of Product Development .....	9
6.1.2	Middle Stage of Product Development .....	9
6.1.3	Final Stage of Product Development .....	10
6.2	Suspension Criteria and Resumption requirements .....	10
6.2.1	Suspension Criteria: .....	10
6.2.2	Resumption Requirements .....	10
7	Test Results .....	11
7.1	Hardware Used .....	11

7.2	Functional Test Cases.....	11
7.2.1	Camera Selection .....	11
7.2.2	Calibration.....	12
7.2.3	Generate Tags .....	13
7.2.4	Detect and Follow Pose.....	14
7.2.5	Execution with Simulation .....	15
7.3	Non-Functional Test Cases.....	16
7.3.1	Functional Suitability:.....	16
7.3.2	Performance Efficiency: .....	19
7.3.3	Compatibility:.....	26
7.3.4	Usability: .....	26
7.3.5	Reliability:.....	26
7.3.6	Maintainability:.....	27

## 1 Resources Required

- PC/Laptop
- Built-in/External Camera
- IDE to initially run code
- Users to perform actions
- Measuring device for pose
- Deployment Device (Microcontroller)

## 2 Features to be Tested

- Detection of Marker/s
- Precision of Marker/s
- Efficiency of software

## 3 Functional Test Cases

Taken from Software Requirements Specifications.

Test Case/s:	Steps	Expected Results
<b>Detect objects using camera</b>		
<b>1) User display marker/s to camera connected to device.</b>	<ol style="list-style-type: none"> <li>1. User turns on device</li> <li>2. System automatically tries to establish a connection with the connected device</li> <li>3. User will display marker/s to the camera's field of view.</li> </ol>	Device will display current pose information to the users via a user interface.
<b>Capture numerical data from object and Perform algorithms</b>		
<b>1) User wants to know the current pose of the displayed marker/s</b>	<ol style="list-style-type: none"> <li>1. User will show marker/s to the camera.</li> <li>2. User will move marker/s around.</li> <li>3. System will display pose data.</li> </ol>	Device will display current pose information to the users via a user interface.
<b>Execute on microcontroller based on internal calculations</b>		
<b>1) User operates on device that is deployed on microcontroller</b>	<ol style="list-style-type: none"> <li>1. User will turn on the microcontroller.</li> <li>2. User will display marker/s to the camera's field of view.</li> <li>3. Microcontroller will respond in a programmed manner to the pose of the markers.</li> </ol>	Software will operate the same regardless of device used. Ideally, the mechanism should mimic the movement of the maker.
<b>View and edit live functional parameters of system</b>		
<b>1) User modifies the parameters of the system</b>	<ol style="list-style-type: none"> <li>1. User modifies parameters using the user interface. These</li> </ol>	Software will adapt to modified parameters and operate according to the new

	<p>are camera selection and calibration parameters, selection of marker dictionary and specification of marker/checkerboard dimensions.</p> <p>2. User will display marker/s to the camera's field of view.</p>	<p>parameters for greater pose estimation precision. Improving overall performance and flexibility.</p>
--	---	---

## 4 Non-Functional Test Cases

Taken from Software Requirements Specifications.

### 4.1 Functional Suitability:

Requirements	Test Case/s	Result
<b>The accuracy of the 6-DoF data should be within the allowed error range of <math>\pm 1</math> millimeter for all x, y, z coordinates and the pitch, yaw, and roll.</b>	User places a marker at different fixed positions and orientations.	Placed markers will produce near identical results when placed in the same location multiple times.
<b>System should provide 100% capacity to obtain and display the relevant 6-DoF data and send parameters to the system for both eye-in-hand and eye-to-hand variations.</b>	User places a marker at all different positions and orientations possible.	As Requirement states, data should be available for all times in any orientation.

### 4.2 Performance Efficiency:

Requirements	Test Case/s	Result
<b>Detection and response are almost instantaneous to the human eye, at computational level processing delay should be less than 800 milliseconds.</b>	User moves a marker systematically within the camera field of view.	Pose value updates dynamically with little to no delay as observed by the human eye.
<b>Under normal operation, no more than 50% of a system's full memory should be utilized.</b>	User moves a marker systematically within the camera field of view.	Observed memory should be less than 50% of capacity.

### 4.3 Compatibility:

Requirements	Test Case/s	Result
<b>System must be able to successfully transfer required information to relevant software and hardware components working in the</b>	<p>User swaps the camera to a different one.</p> <p>Application is transferred to different systems with</p>	System will successfully operate the exact same regardless of the varied hardware components.

<b>same environment with 100% compatibility and efficiency to avoid data loss.</b>	required runtime environment installed  Various microcontrollers can be controlled by minimal adjustment needed.	
<b>More than 50% of the system's software/hardware components should be able to function independently.</b>	Running the system without the integration of a microcontroller.	As Requirement states.

#### 4.4 Usability:

Requirements	Test Case/s	Result
<b>Use of universal codes, symbols, characters and adhere to general language/cognitive standards so that users can recognize them in less than 5 seconds.</b>	N/A	As Requirement states.
<b>Users should be able to learn to operate the system in less than 1 day.</b>	Users unfamiliar to the software, operate the software.	Users are able to briefly describe the operations of the system.
<b>Allow users to make less than one error per 10 operations</b>	Users operates with the system.	As Requirement states.
<b>User interface is practically laid out and easy to navigate.</b>	Users should be able to navigate the system easily.	As Requirement states.
<b>Can be operated by colour-blind people as well as deaf people.</b>	User wearing noise-cancelling headphones, operates the system	Users can operate the system proficiently.

#### 4.5 Reliability:

Requirements	Test Case/s	Result
<b>System must be available 100% or minimum 99% of the time when in operation.</b>	System is left on for a period, the user will sparingly display a marker.	As Requirement states.
<b>Error rate &lt; 1%.</b>	N/A	As Requirement states.
<b>Expected time between failures (downtime) is limited to 5-7 minutes.</b>	Leave the system on for a long period of time.	As Requirement states.

#### 4.6 Maintainability:

Requirements	Test Case/s	Result
<b>Testing satisfies &gt;90% of the test criteria.</b>	N/A	As Requirement states.
<b>No more than 18 days needed to implement, test, and deploy the modification.</b>	Transfer to another system and attach to a different microcontroller.	As Requirement states.

<b>More than 50% of the system's software/hardware components should be able to function independently.</b>	N/A	As Requirement states.
---	-----	------------------------

## 5 Strategy

The Project Plan briefly iterated over the critical success factors which put an indirect focus on the testing strategy that will be taken forward. To ensure the success of most software systems, multi-billion dollar companies have adopted the four-level testing strategy, thus, we have decided too, as follows:

The hierarchy is default in the order which testing should be carried out:

### 5.1.1 Unit Testing:

Due to the vast and multi-disciplined nature of the system, it is crucial to test incremental development of particular units of functionality independently to allow detection of errors/bugs in the vulnerable initial stages of the development process. To implement this practice, unit tests will be designed to test small fragments of code/functionality.

### 5.1.2 Integration Testing:

The logical next step following Unit testing, is integration testing. This is a point in the testing process where after ensuring individual components function as intended, the software components should be collaborative, thus should be tested as a cluster. This will be accomplished by testing various components of the systems by designing effective test modules.

### 5.1.3 System Testing:

As the name suggests, system testing is the testing of the nearly complete integrated product, in other words, to ensure that the product satisfies majority/all of the requirements. To reiterate, requirements consist of functional and non-functional requirements, thus, while the functional requirements might have been met in previous stages, it is essential to ensure test cases are designed to test non-functional test cases.

### 5.1.4 Acceptance Testing:

This testing stage is mainly to ensure that the business requirements have been catered for by the software product. This will involve client collaboration with the finalized product to evaluate client/stakeholder satisfaction.

## 5.2 Roles and Responsibilities

Below are the stated roles and responsibilities in-line with the testing strategies stated above:

### 5.2.1 Unit testing:

As unit testing deals with small fragments of the components/codes, it is a good practice that the team members/developers working on the fragments actively test their creation as an ongoing process by designing relevant test cases deemed suitable by themselves throughout the process.

### 5.2.2 Integration testing:

As integration testing is concerned with the unifying functionality of all software modules, this step can be executed with the input and presence of more than one developer and/or by the managing body and/or testing team (if applicable).

### 5.2.3 System testing:

It is preferred that this type of testing be carried with personnel specially appointed for testing (users) in the presence of the team leader/manager. The suitability of the testing group will depend on their amount of experience with software systems.

### 5.2.4 Acceptance testing:

At last, this type of testing should be conducted with the presence of the leader/manager and the testing personnel mentioned in the previous step. However, this time, the whole process will be under the eyes of the client which is necessary for them to evaluate the system with them possibly having no experience in developing software systems.

## 5.3 Test Deliverables

- Test Plan
- Test Cases
- Test Reports
- Usability Test Guide

## 5.4 Schedule

System Features	Estimated Testing Period
<b>Object detection</b>	Semester 2 (early)
<b>Pose detection</b>	Semester 2 (early)
<b>3D location with rotation output</b>	Semester 2 (mid)
<b>Operation in mobile setting (eye-in-hand, eye-to-hand)</b>	Semester 2 (mid-end)
<b>User Interface</b>	Semester 2 (end)

## 5.5 Risk and Contingency with Testing

Risk	Contingency
<b>Team members leaving/unable to make substantial contributions</b>	Alteration of scope if required/distribution of tasks to other available team members
<b>Scope variations</b>	Updating and recording changes from client. Harnessing agile development methodologies.
<b>Inadequate risk management</b>	Perform a risk analysis and keep an updated backlog.



<b>Incompatible devices (camera, microcontroller)</b>	Research various devices with compatible devices. Find a universal device that can handle different connections
<b>Client/stakeholder engagement</b>	Frequent user acceptance testing along with other tests, such as unit testing
<b>Time management with development and testing</b>	Run Tests alongside software development to reduce excess time.

## 5.6 Testing Tasks

- Effective test cases will be derived from real life testing scenarios in the intended operational environment of the product.
- According to testing scenarios, unique test cases will be designed to test particular functions/features of the system.
- Resource analysis for testing will be conducted.
- Test results along with errors/bugs observers will be properly documented along with changes made to fix them.

## 6 Pass/Fail Criteria

### 6.1 Product Level

#### 6.1.1 Initial Stage of Product Development

Testing Criteria	Pass Criteria	Fail Criteria
<b>Object Detection:</b> The system should be able to detect a moving object in its' field of vision	90% Objects Detected with minimal interference	Unable to Detect simplest of shapes or moving tags
<b>Pose Detection:</b> The system should be able to detect the pose of the tags and illustrate axis upon it	95% Tags should have an axis on them in field of camera and detect any changes in pose of the tags	Unable to determine the changing pose of the tags
<b>3D location with rotation output:</b> The system is able to provide an output of the pose of the tag in (x, y, z, pitch, yaw, roll)	System is able to provide the requested output in relative quick cycles (10 - 100 ms)	It can provide some output but not all. It is providing all the outputs but there is lot of latency making the system inefficient

#### 6.1.2 Middle Stage of Product Development

Testing Criteria	Pass Criteria	Fail Criteria
<b>Operation in Mobile Setting:</b> The system is deployed on a microcontroller and functions normally	System is functional within the microcontroller.	System does not work within the microcontroller.
<b>Eye-in-hand:</b> The system should be able to be mounted on a robot and work properly	This system needs to be able to calibrated on the fly to be able to match the movement skill.	This system is unable to be used for mounted operation

<b>Eye-to-hand:</b> <b>The system should be able to use normally at a fixed position</b>		
---	--	--

### 6.1.3 Final Stage of Product Development

Testing Criteria	Pass Criteria	Fail Criteria
<b>Speed of the system:</b> <b>The system is able to run fast and efficiently</b>	The system is able to run a relative cycle at 10 to 40 ms	The system is unable to run at the required speed
<b>Accuracy of the system:</b> <b>The system is able to be accurate</b>	The system is able to be accurate to 1 to 4 mm	The system is not accurate to the standards
<b>Interface for operations:</b> <b>The interface is easy to use and integrate with other systems</b>	The interface is user-friendly and is able to integrate with client specified robotic arm	The system does not have good interface

## 6.2 Suspension Criteria and Resumption requirements

### 6.2.1 Suspension Criteria:

- The system is found to have a critical operational error.
- The client requirement has changed for the proposed module
- The testing has found irregular issues in operational and results.
- The system is not suitable for the final product

### 6.2.2 Resumption Requirements

- All critical issues have been resolved.
- The system matches client's vision for the system
- The proposed system is suitable for the final product.
- The client has verified the use of the selected design and system.

## 7 Test Results

### 7.1 Hardware Used

	<b>Member Conducting</b>	<b>Hardware Used</b>
1	Daniel Leong	Raspberry Pi 4 Model B, 8GB Raspberry Pi Camera Module 2, Logitech StreamCam
2	Daniel Leong	Lenovo ThinkPad 13 Type 20GK Logitech StreamCam
3	Preenit Kshirsagar	Raspberry Pi 4 Model B, 8GB Raspberry Pi Camera Module 2
4	Preenit Kshirsagar	Dell Inspiron 15 5000

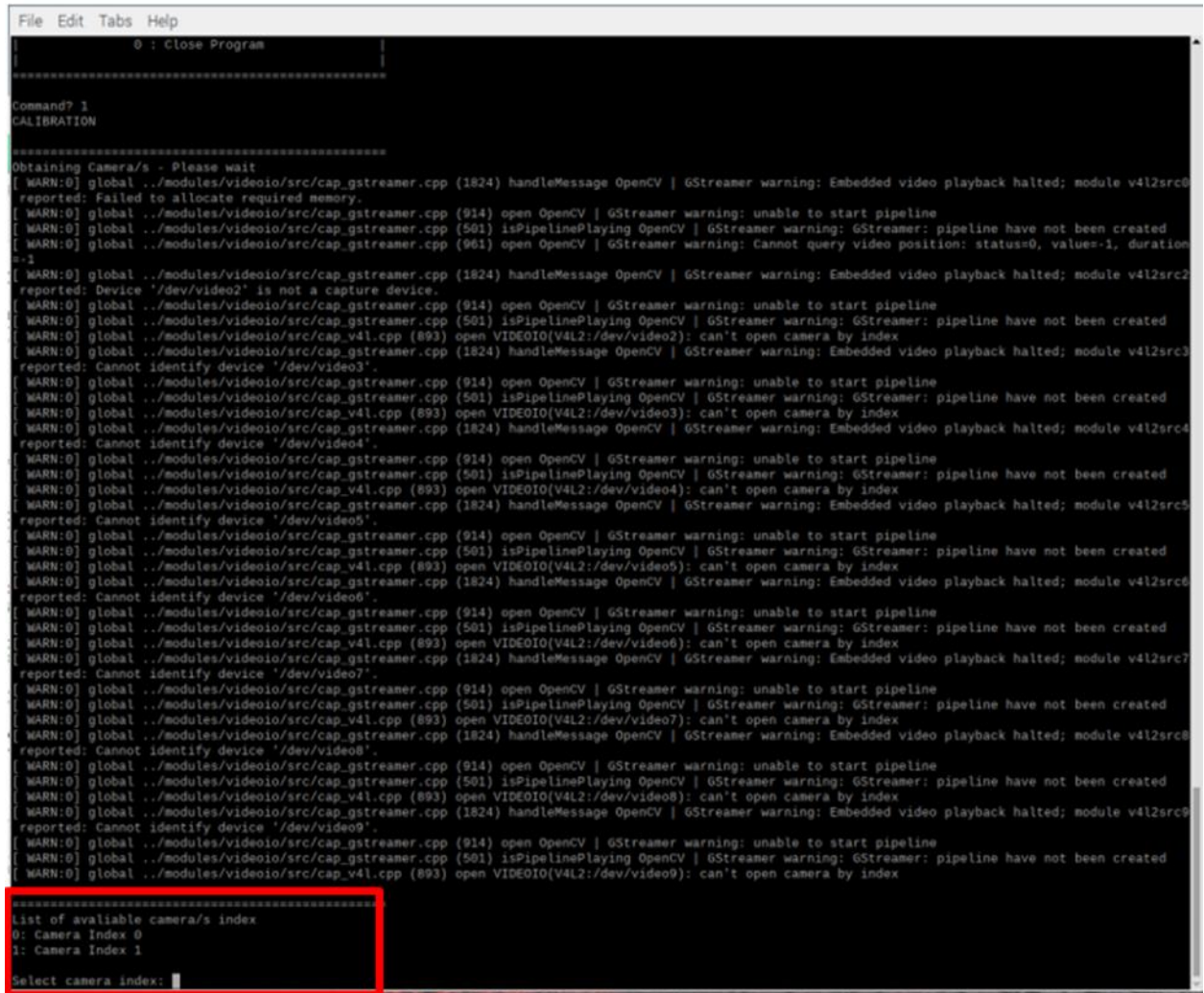
### 7.2 Functional Test Cases

	<b>Detect Objects using Camera</b>	<b>Capture data from object and Perform algorithms</b>	<b>Execute on microcontroller based on internal calculations</b>	<b>View and edit live functional parameters of system</b>
1	True	True	True	True (not live)
2	True	True	True	True (not live)
3	True	True	True	True (not live)
4	True	True	True	True (not live)

#### 7.2.1 Camera Selection

Video Reference – [https://youtu.be/8f4Gk\\_oZHdY](https://youtu.be/8f4Gk_oZHdY), <https://youtu.be/1HWdiYyCuaU>

Following from the core selling properties of the application, it was designed to accommodate various hardware components, the most important being cameras. Some users of the application may need to attach cameras with custom specifications and/or may have multiple cameras attached and want to choose a specific source camera. This provides successful evidence that the application is able to detect all cameras attached to it and provide users with an option which one to choose. Once the camera is selected, its specifications will be used for further functionality (frame width, frame height, fps etc). Below figure shows evidence of the application running the camera select feature:



```
File Edit Tabs Help
0 : Close Program

Command? 1
CALIBRATION

Obtaining Camera/s - Please wait
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (1824) handleMessage OpenCV | GStreamer warning: Embedded video playback halted; module v4l2src0
reported: Failed to allocate required memory.
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (914) open OpenCV | GStreamer warning: unable to start pipeline
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (501) isPipelinePlaying OpenCV | GStreamer warning: GStreamer: pipeline have not been created
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (893) open VIDEOIO(V4L2:/dev/video2): can't open camera by index
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (1824) handleMessage OpenCV | GStreamer warning: Embedded video playback halted; module v4l2src0
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (1824) handleMessage OpenCV | GStreamer warning: Embedded video playback halted; module v4l2src2
reported: Device "/dev/video2" is not a capture device.
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (914) open OpenCV | GStreamer warning: unable to start pipeline
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (501) isPipelinePlaying OpenCV | GStreamer warning: GStreamer: pipeline have not been created
[ WARN:0] global .../modules/videoio/src/cap_v4l.cpp (893) open VIDEOIO(V4L2:/dev/video2): can't open camera by index
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (1824) handleMessage OpenCV | GStreamer warning: Embedded video playback halted; module v4l2src3
reported: Cannot identify device "/dev/video3".
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (914) open OpenCV | GStreamer warning: unable to start pipeline
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (501) isPipelinePlaying OpenCV | GStreamer warning: GStreamer: pipeline have not been created
[ WARN:0] global .../modules/videoio/src/cap_v4l.cpp (893) open VIDEOIO(V4L2:/dev/video3): can't open camera by index
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (1824) handleMessage OpenCV | GStreamer warning: Embedded video playback halted; module v4l2src4
reported: Cannot identify device "/dev/video4".
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (914) open OpenCV | GStreamer warning: unable to start pipeline
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (501) isPipelinePlaying OpenCV | GStreamer warning: GStreamer: pipeline have not been created
[ WARN:0] global .../modules/videoio/src/cap_v4l.cpp (893) open VIDEOIO(V4L2:/dev/video4): can't open camera by index
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (1824) handleMessage OpenCV | GStreamer warning: Embedded video playback halted; module v4l2src5
reported: Cannot identify device "/dev/video5".
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (914) open OpenCV | GStreamer warning: unable to start pipeline
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (501) isPipelinePlaying OpenCV | GStreamer warning: GStreamer: pipeline have not been created
[ WARN:0] global .../modules/videoio/src/cap_v4l.cpp (893) open VIDEOIO(V4L2:/dev/video5): can't open camera by index
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (1824) handleMessage OpenCV | GStreamer warning: Embedded video playback halted; module v4l2src6
reported: Cannot identify device "/dev/video6".
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (914) open OpenCV | GStreamer warning: unable to start pipeline
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (501) isPipelinePlaying OpenCV | GStreamer warning: GStreamer: pipeline have not been created
[ WARN:0] global .../modules/videoio/src/cap_v4l.cpp (893) open VIDEOIO(V4L2:/dev/video6): can't open camera by index
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (1824) handleMessage OpenCV | GStreamer warning: Embedded video playback halted; module v4l2src7
reported: Cannot identify device "/dev/video7".
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (914) open OpenCV | GStreamer warning: unable to start pipeline
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (501) isPipelinePlaying OpenCV | GStreamer warning: GStreamer: pipeline have not been created
[ WARN:0] global .../modules/videoio/src/cap_v4l.cpp (893) open VIDEOIO(V4L2:/dev/video7): can't open camera by index
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (1824) handleMessage OpenCV | GStreamer warning: Embedded video playback halted; module v4l2src8
reported: Cannot identify device "/dev/video8".
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (914) open OpenCV | GStreamer warning: unable to start pipeline
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (501) isPipelinePlaying OpenCV | GStreamer warning: GStreamer: pipeline have not been created
[ WARN:0] global .../modules/videoio/src/cap_v4l.cpp (893) open VIDEOIO(V4L2:/dev/video8): can't open camera by index
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (1824) handleMessage OpenCV | GStreamer warning: Embedded video playback halted; module v4l2src9
reported: Cannot identify device "/dev/video9".
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (914) open OpenCV | GStreamer warning: unable to start pipeline
[ WARN:0] global .../modules/videoio/src/cap_gstreamer.cpp (501) isPipelinePlaying OpenCV | GStreamer warning: GStreamer: pipeline have not been created
[ WARN:0] global .../modules/videoio/src/cap_v4l.cpp (893) open VIDEOIO(V4L2:/dev/video9): can't open camera by index

List of available camera/s index
0: Camera Index 0
1: Camera Index 1

Select camera index: 
```

Figure 1 - Checking Available Cameras (2 connected in this case)

Referring to figure 14, the test was conducted (on raspberry pi computer) by connecting two different cameras to the system:

- 1) Raspberry Pi Camera Module 2
- 2) Logitech USB camera

When the camera selection function is executed, the program scans for all the available cameras and correctly lists the number of cameras available for selection (see red box).

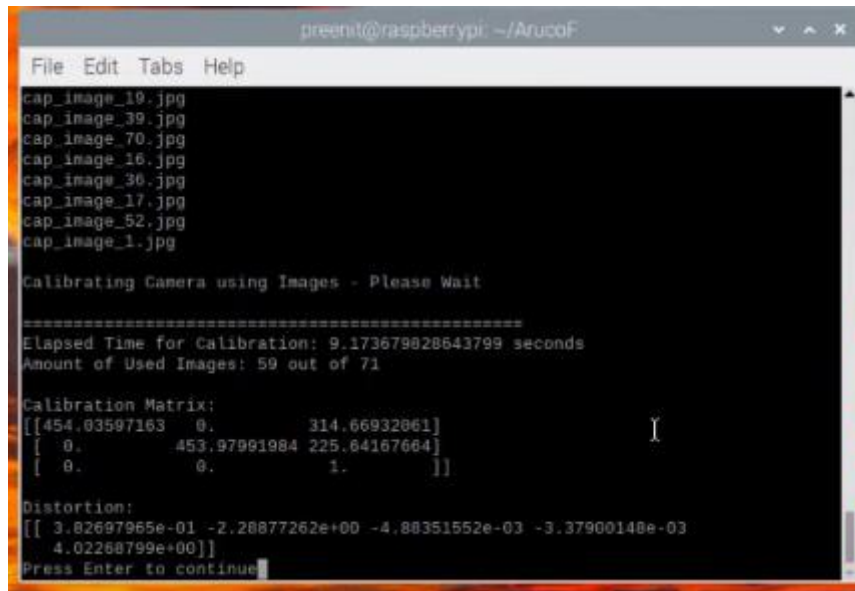
The outcome of the test was successful. Refer to the video for full test coverage.

## 7.2.2 Calibration

Video Reference – <https://youtu.be/OTkNknL9HQ8>, <https://youtu.be/Arq2QwTZxfw>

To be able to detect the pose of an object through a camera, it is essential to consider the issues with using hardware, in particular the camera lens, this has been discussed in the section architecture design. The purpose of the functional test is to verify that we are able to calibrate different cameras based on selection however many times we like repeatedly. As discussed, the camera has intrinsic and extrinsic properties which affect the image in terms of stretching, radial/tangential distortion etc which incorrectly modifies the image in the camera and thus we receive a poor image which negatively impacts the pose detection algorithm. The purpose of

calibration is to capture these properties of the camera (camera matrix and distortion matrix) and use them to reverse the effects of the camera on the image, so we get a perfect image. The following figure 15 shows evidence of producing the required matrices after doing the calibration using a raspberry pi camera module 2, note that different cameras will have different matrices:



```
preenit@raspberrypi: ~/ArucoF
File Edit Tabs Help
cap_image_19.jpg
cap_image_39.jpg
cap_image_70.jpg
cap_image_16.jpg
cap_image_36.jpg
cap_image_17.jpg
cap_image_52.jpg
cap_image_1.jpg

Calibrating Camera using Images - Please Wait

=====
Elapsed Time for Calibration: 9.173679028543799 seconds
Amount of Used Images: 59 out of 71

Calibration Matrix:
[[[454.03597163  0.          314.66932061]
 [ 0.          453.97901984 225.64167604]
 [ 0.           0.           1.          ]]]

Distortion:
[[[ 3.82697965e-01 -2.28877262e+00 -4.88351552e-03 -3.37900148e-03
  4.02268799e+00]]]
Press Enter to continue
```

Figure 2 - Creating of Relevant Matrices from Calibration

As can be seen, the test was successful. Refer to the video for full test coverage.

### 7.2.3 Generate Tags

Video Reference – <https://youtu.be/H31N1WGUrUA>, [https://youtu.be/kk40EY\\_mUFQ](https://youtu.be/kk40EY_mUFQ)

To provide a modular design for the application as well as a lower dependency of external applications required, the system has been equipped with its own feature to generate the markers that can be used for detection. This will allow the user to select markers to create from a range ArUco dictionaries and save them in a file directory, these markers can be then used to detect the pose of an object on which these markers are attached to. The following figure 3 shows the creation of markers from different dictionaries:

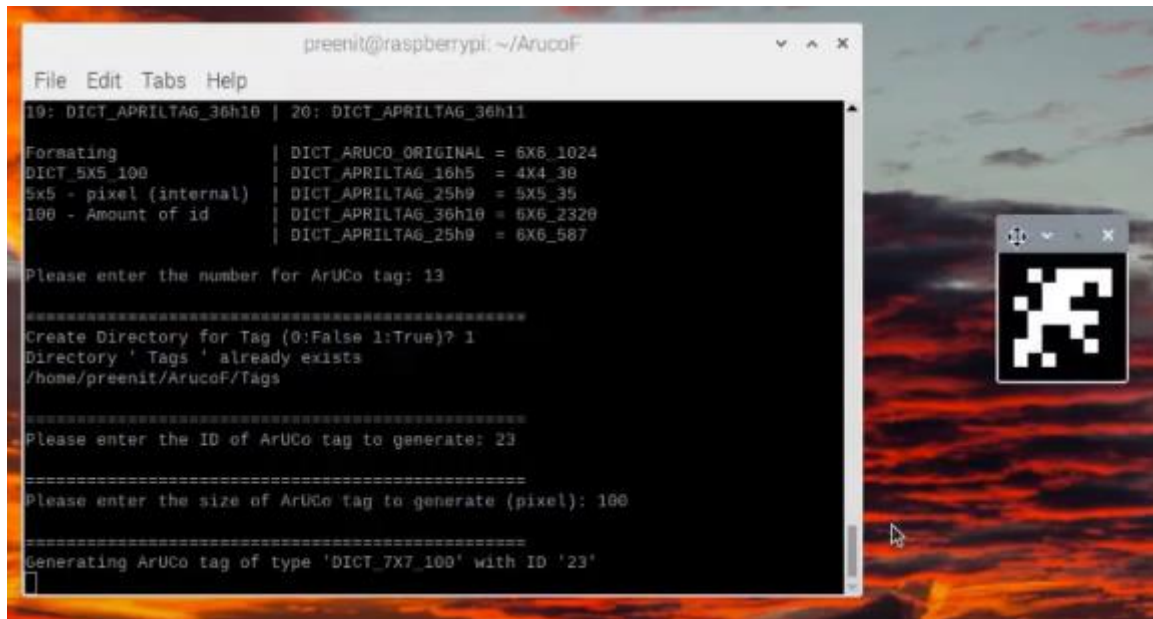


Figure 3 - Generating Tag

As can be seen from above figure 3, a marker from dictionary 13 (DICT\_7x7\_100) is generated by specifying the size to allow for further flexibility as well as greater accuracy when detecting the pose due to the fact that the user will know the exact size of the marker which they can provide to the Pose Detector module increasing the accuracy of the estimation.

The outcome of this test was successful. Refer to the video for the test.

#### 7.2.4 Detect and Follow Pose

Video References – [https://youtu.be/uSWewK\\_Ha7s](https://youtu.be/uSWewK_Ha7s), <https://youtu.be/sgTzLg3-OME>, <https://youtu.be/4sl-ZR5JbAE>

With the use of the camera, information collected during calibration and a tag generated to detect, we now test the actual algorithm for pose detection. This test aims to verify if the application can detect the marker generated using the previous modules and verifies the detection algorithm works as we are able to get accurate results in terms of x, y, z coordinates and pitch, roll and yaw Euler angles. The figure 4 below shows the evidence that we have selected a camera to detect a pose from, the calibration exists (both tests completed previously):



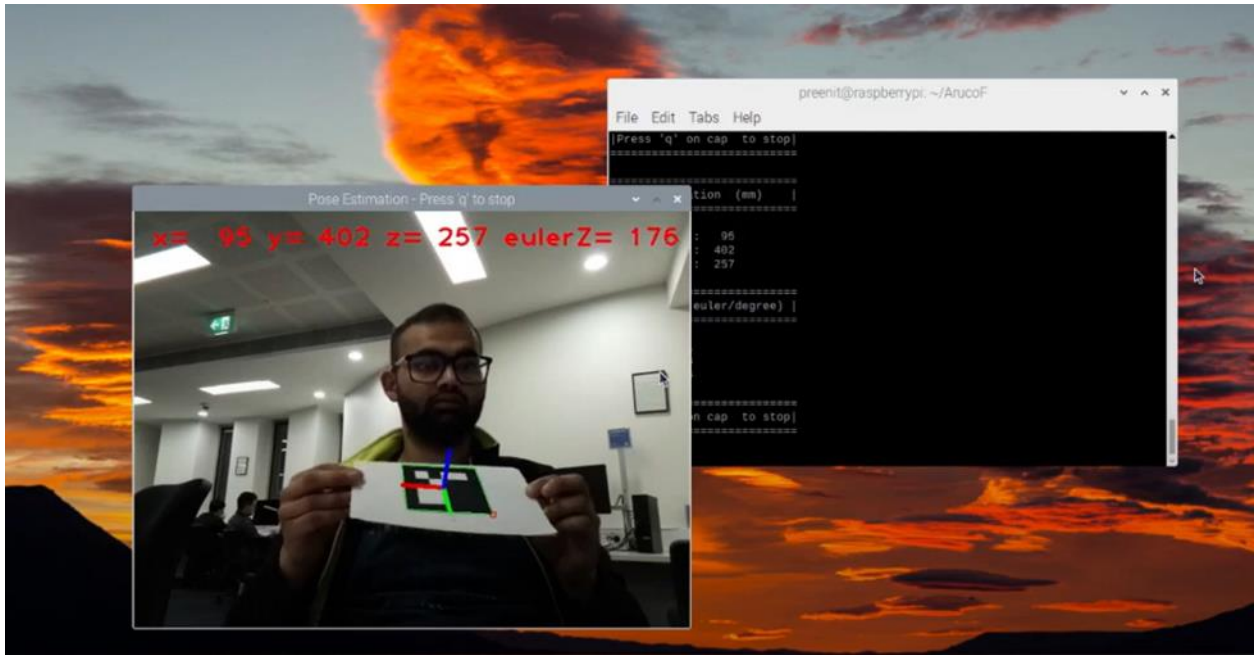


Figure 4 - Detection of Pose

Referring to Figure 17, it can be verified that we are able to detect the pose of the marker accurately, this verifies that we have access to our selected camera, the camera and distortion matrix are calculated as we do not have a distorted image (image is clear). The algorithm discussed in the architecture design is implemented correctly as can be seen from the pose detection values (in red) at the top of the window. Refer to the video for the full test.

#### 7.2.5 Execution with Simulation

Video Reference – [https://drive.google.com/file/d/1WeoGCFUby\\_Qc9EIIt4B97O0Z-4lvVXkj/view?usp=sharing](https://drive.google.com/file/d/1WeoGCFUby_Qc9EIIt4B97O0Z-4lvVXkj/view?usp=sharing)

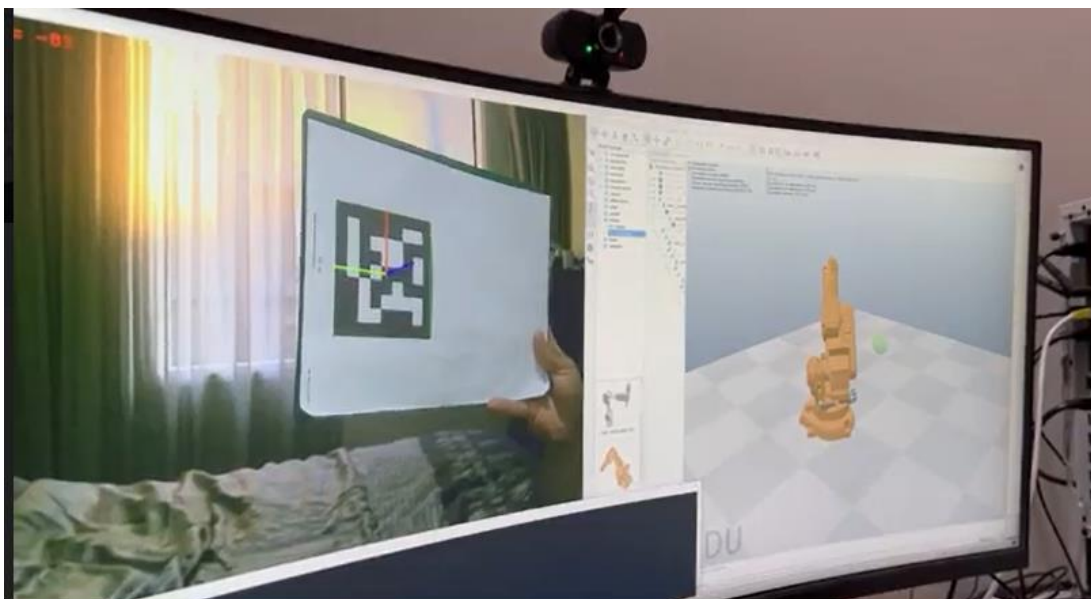


Figure 5 - Pose Detection with Simulation

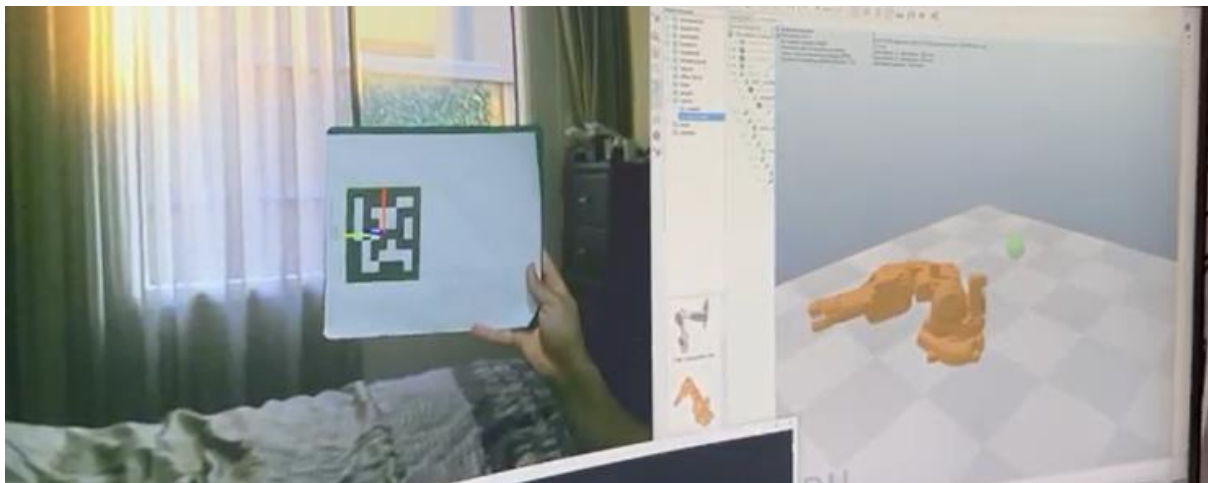


Figure 6 - Another Pose Detection with Simulation

To demonstrate the pose data that has been calculated can be put to use, we have developed a robotic arm simulation which mimics a robotic arm in the real world, the ‘head’ of the arm moves to align itself with the center of the marker by using the output pose values (x, y, z, pitch, yaw and roll) to reference its position in 3D space.

7.3 Non-Functional Test Cases

7.3.1 Functional Suitability:

	<b>The accuracy of the 6-DoF data should be within the allowed error range of +- 1 millimeter for all x, y, z coordinates and the pitch, yaw, and roll.</b>	<b>System should provide 100% capacity to obtain and display the relevant 6-DoF data and send parameters to the system for both eye-in-hand and eye-to-hand variations.</b>
1	True	True
2	True	True
3	True	True
4	True	True

Since the system is used to detect pose estimation values, there is no real way to test the accuracy of the system without having another system to compare the values with. Therefore, we are limited here in terms of the hardware available to test the system.

There exists however a mathematical formula which, provided with x, y coordinates gives us the ‘ideal’ z coordinate in mathematical terms without actually reading the z coordinate from the lens. This methodology is not 100% reliable due to the fact that we are still depending on reading the x and y coordinates from the camera to calculate z. However, the deviation from both the scenarios will tell us whether there is some discrepancy in the readings and thus this will prove the error rate.

The mathematical formula below has been embedded into the PoseDetector module to calculator z based on x and y:

Distance(z) = sqrt(x^2 + y^2 + z^2)

Where x, y and z are the coordinates obtained from the translation vectors.



Below is the code fragment that was added in python to get distance using numpy's square root (sqrt) function.

```
#Translation (mm)
x = realworld_tvec[0]
y = realworld_tvec[1]
z = realworld_tvec[2]

#greater accuracy distance calc formula
distance = np.sqrt(x ** 2 + y ** 2 + z ** 2);
```

Figure 7 - Getting Mathematical Distance

During the test, a ruler was used (millimeter accuracy) to place the marker at specific length from the camera of the system. The marker was placed 27 centimeters (270 millimeters) from the camera and the display observed for the z distance.

Video Reference – <https://youtu.be/tuX77j06DSg>

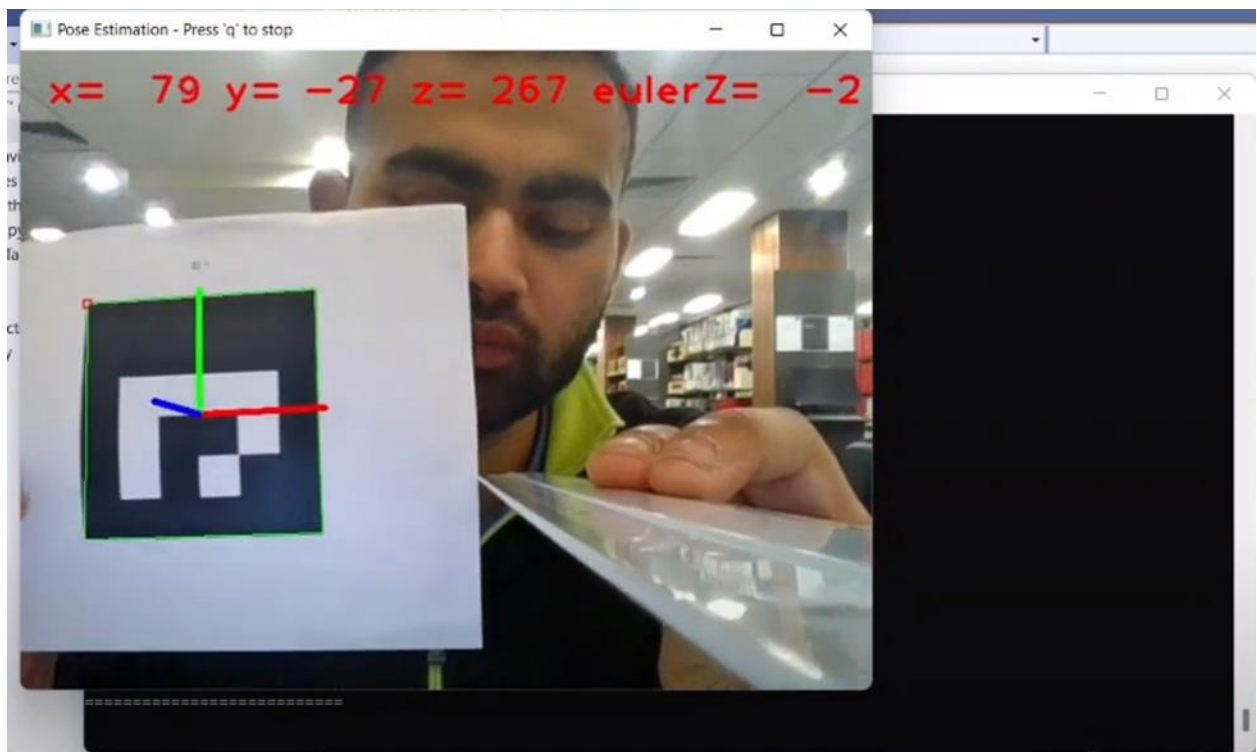


Figure 8 - Distance at 270 mm without using Formula

Video Reference – <https://youtu.be/0yxqPmT83dc>

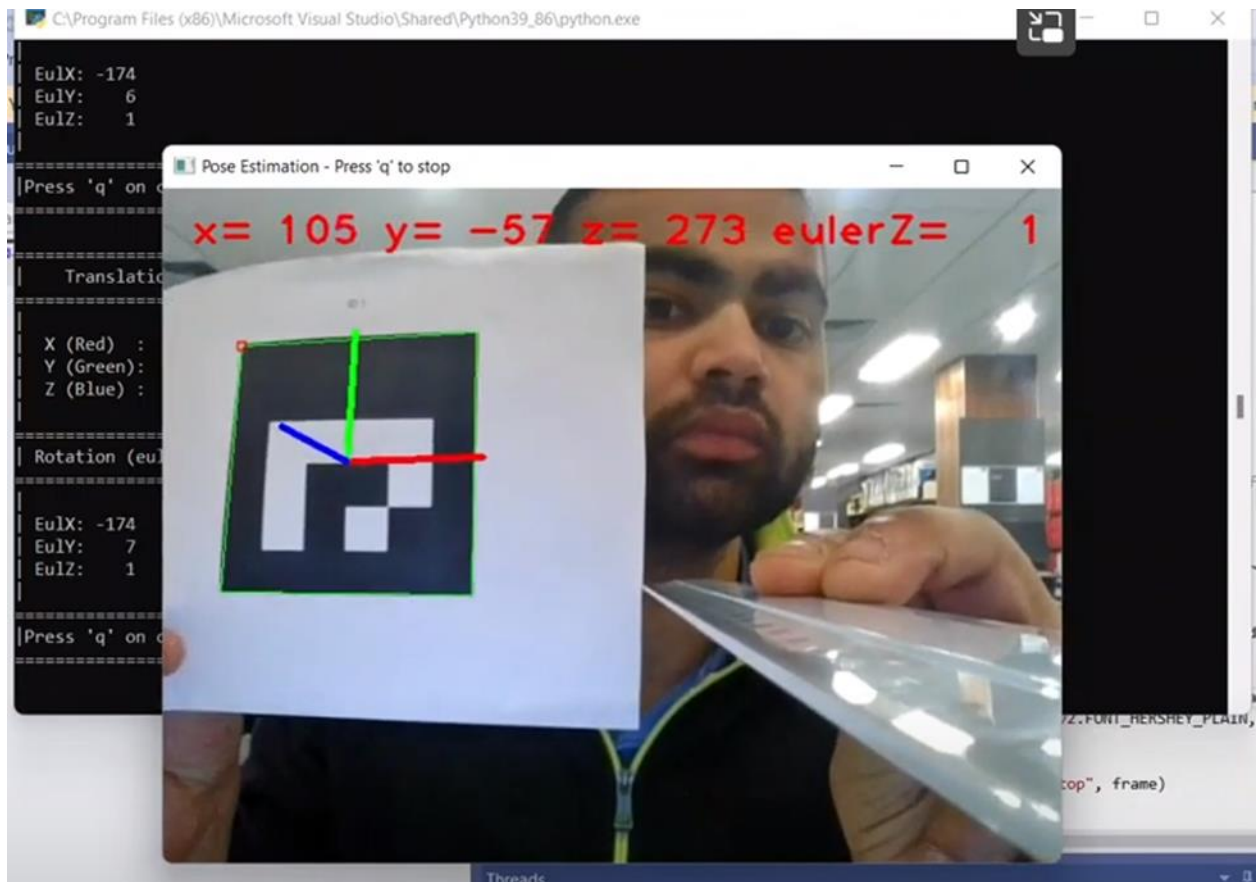


Figure 9 - Distance at 270 mm with using Formula

Therefore, in conclusion as can be seen from the measurements, the distance without using the algorithm was short by 3 mm and the distance obtained by using the algorithm overshoot by 3 mm.

Therefore, a rule can be formed where the program takes the average of the two numbers and then outputs that value as the z coordinate.

This ensures that the error remains within the acceptable range of  $\pm 1$  mm.

Refer to the videos for full test coverage.

### 7.3.2 Performance Efficiency:

**Detection and response are almost instantaneous to the human eye, at computational level processing delay should be less than 800 milliseconds.**

**Under normal operation, no more than 50% of a system's full memory should be utilized.**

1	True	True
2	True	True (max ~40%)
3	True	True
4	True	True

#### 7.3.2.1 Process Delay

Video Reference – <https://youtu.be/n6LNN677em4>

For ideal real-time pose detection, it is important that the processing delay of the application from detecting the object, calculating its pose in 3D and outputting the result must be minimal as any significant delays defies the purpose of the application for obvious reasons. The processing speed depends majorly on the hardware that the application is run on and minorly on the software design. The software design has been designed with modularisation techniques (discussed in design and architecture) to allow for efficient code and optimal code refactoring. Therefore, the requirement was reasonable to say that the processing delay of each frame (mostly hardware dependent) should be no more than 800 milliseconds at which point it can be detected by the naked eye. This test was conducted on a raspberry pi 4B with 8 Gb of RAM. The following figures 5 and 6 show the code embedded into the application source code for testing the time delay between capturing a frame and outputting pose values:

```

PoseDetector.py *% PanTilt.py % Calibrate.py %
121 cap.set(cv2.CAP_PROP_FRAME_HEIGHT, cameraSetting["height"])
122 cap.set(cv2.CAP_PROP_FPS, cameraSetting["fps"])
123
124 while True:
125     #start time stamp
126     timeStart = time.time()
127     #
128     ret, frame = cap.read()
129
130     #Flip Vertically and Horizontally
131     if flip: frame = cv2.flip(frame,-1)
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
```

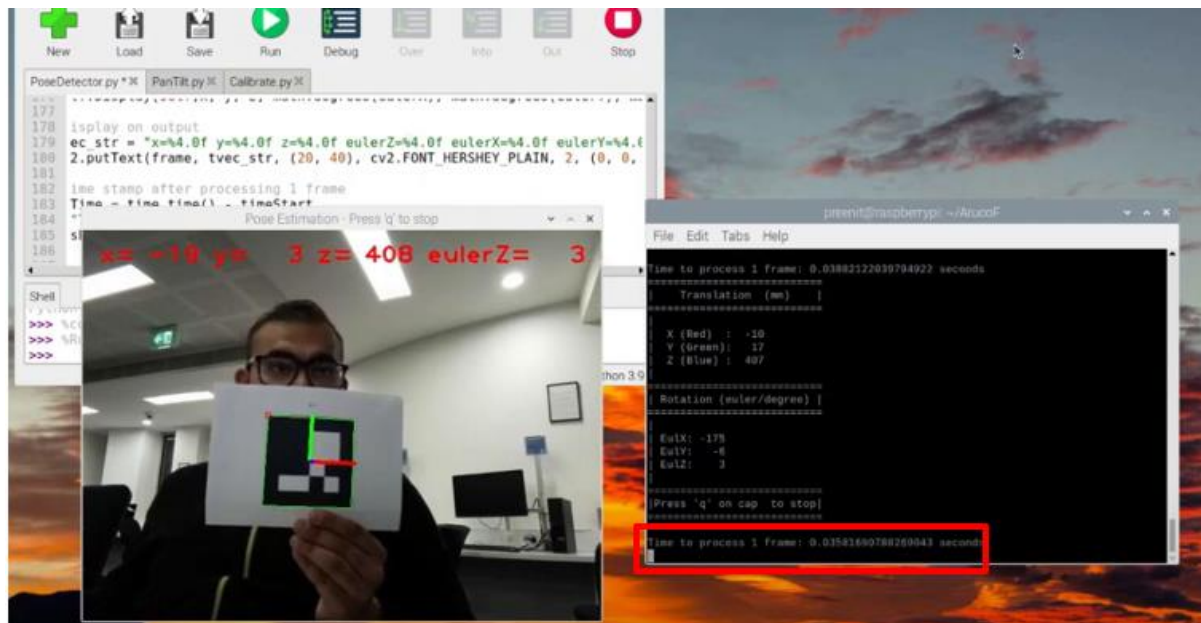


Figure 11 - Processing Delay Output

As can be seen in figure 11 above, the time taken to process one frame is around  $\sim 0.036$  seconds, roughly equal to 36 milliseconds.

Therefore, we can confirm that this is less than 800 milliseconds and hence the test was successful. Refer to the video for full test coverage.

### 7.3.2.2 Memory Usage

The following installation package was needed to be installed which allowed us to view hardware utilization using htop:

```
sudo apt-get install htop
```

As briefly mentioned in the previous test that the processing time was a critical aspect to efficient pose estimation and hence all factors that affect the speed of execution whether it be software or hardware need to be thoroughly tested to ensure that the memory being occupied by the any functionality in the program does not exceed 50%, this is based on the trend observed by team members of FourExplore based on their experience with computer systems. Generally, the performance of a computer is seen to decline as more and more memory is used. CPU utilization is also another metric that is considered in this test to verify that only the program that is executing is utilizing the majority of the CPU and not any other program that does not need to be run. The final metric that will be monitored is RES, this tells us basically how much of the programs memory is kept in the RAM which is arguably a better measure for memory utilization as modern computer systems make use of concepts such as spatial locality to store most of the program memory closer to the CPU (cache, RAM).

The following will test each functionality and attempt to capture the related metrics:

## Usage during Camera Select

Video Reference – <https://youtu.be/RG6srBUkX-g>

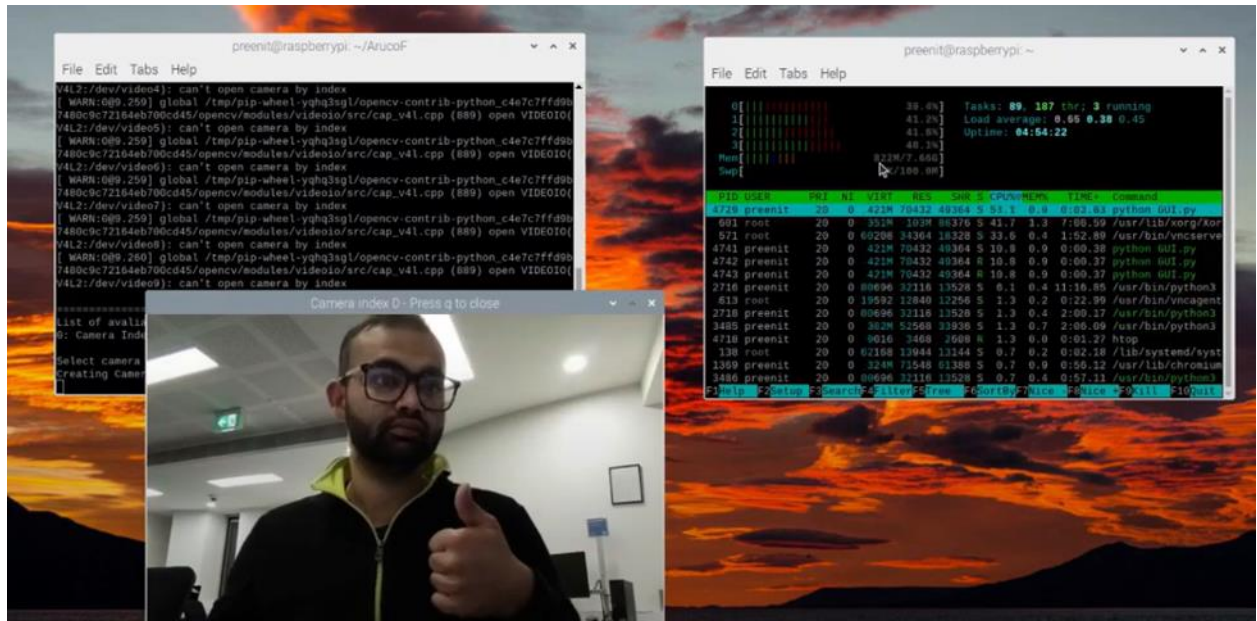


Figure 12 - RES, CPU and MEM statistics

It should be noted that the RAM size for the raspberry pi 4B under test is 8 gb.

As can be observed from running the camera preview, the RES is roughly 70432 which means that the task that is executing (camera select) requires around 70 Megabytes of space in the memory, this is significantly less than 50% of the total memory size hence we have passed the test in this case/

The %CPU represents the percentage of basically how much CPU time the program that is executing takes. Here is a simple diagram to visualize this parameter:

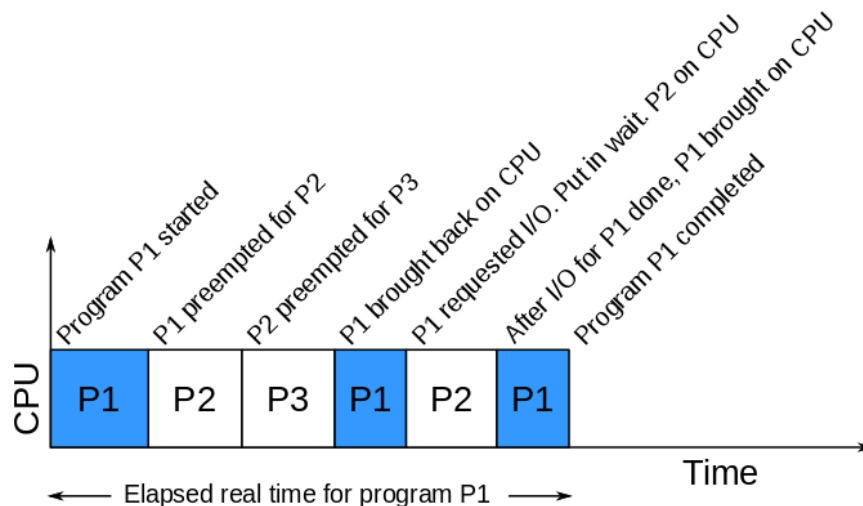


Figure 13 - Representation of CPU time (Wikipedia Contributors 2021)

As can be seen, the more dramatic the changes in between executions, the harder the and longer the CPU is working to accommodate for the task.



For the camera select scenario, the jump between root and preenit is around 10, thus conveying that the process is not putting load on the CPU which is ideal. Hence this test for this metric has also passed.

The %MEM is simply the percentage of memory that has been occupied by the program (out of 8 gb). As seen in the figure 7, only 0.9% of memory is being occupied by the current program hence this test also passes for this metric.

Therefore, all our tests are successful for the camera select feature. Refer to the video for real-time full test coverage.

## Usage during Calibrate

Video Reference – <https://youtu.be/BMLuGATUxpl>

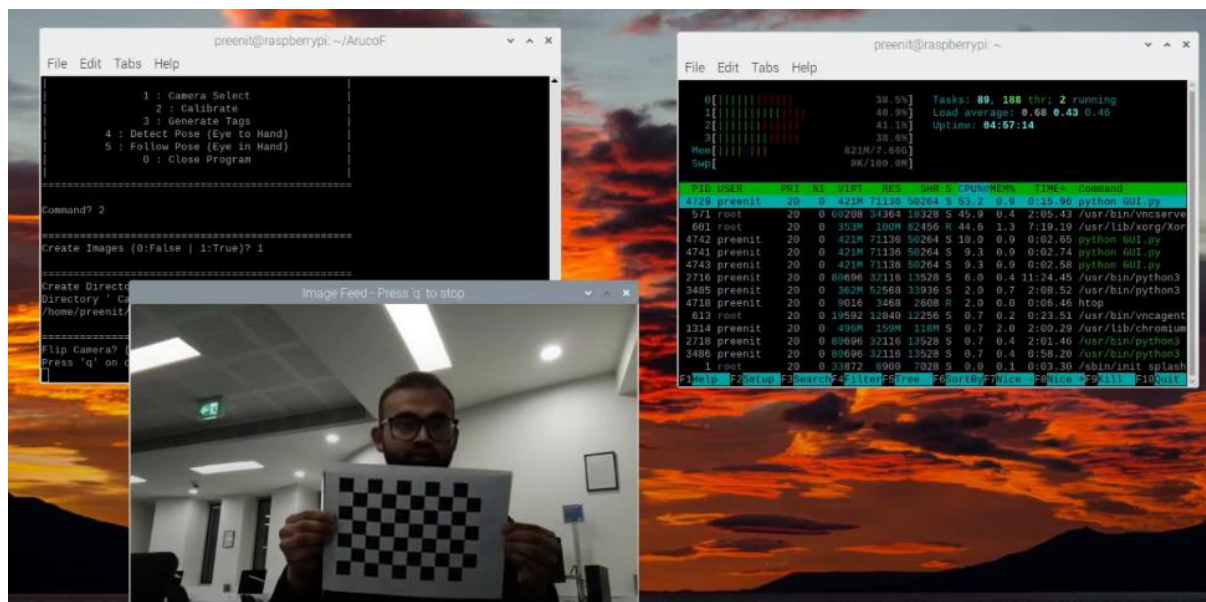


Figure 14 - Metrics when Capturing Images of Chess Board

As seen in Figure 14, the RES value is roughly 71 mb, meaning that the program running in the screenshot above is roughly occupying 71 mb of the memory. Therefore, this is less than 50% of total available memory, thus we have passed this test for this functionality.

There are no surprisingly large jumps in CPU% hence conveying that the CPU is working efficiently without much overhead. This test also passes for this instance.

The MEM% is roughly 0.9% meaning that during this action, only 0.9% of the memory is being utilized. This is significantly less than 50% of total memory (8 gb). Hence this test has also passed for this instance.

Refer to the video for real-time full test coverage.

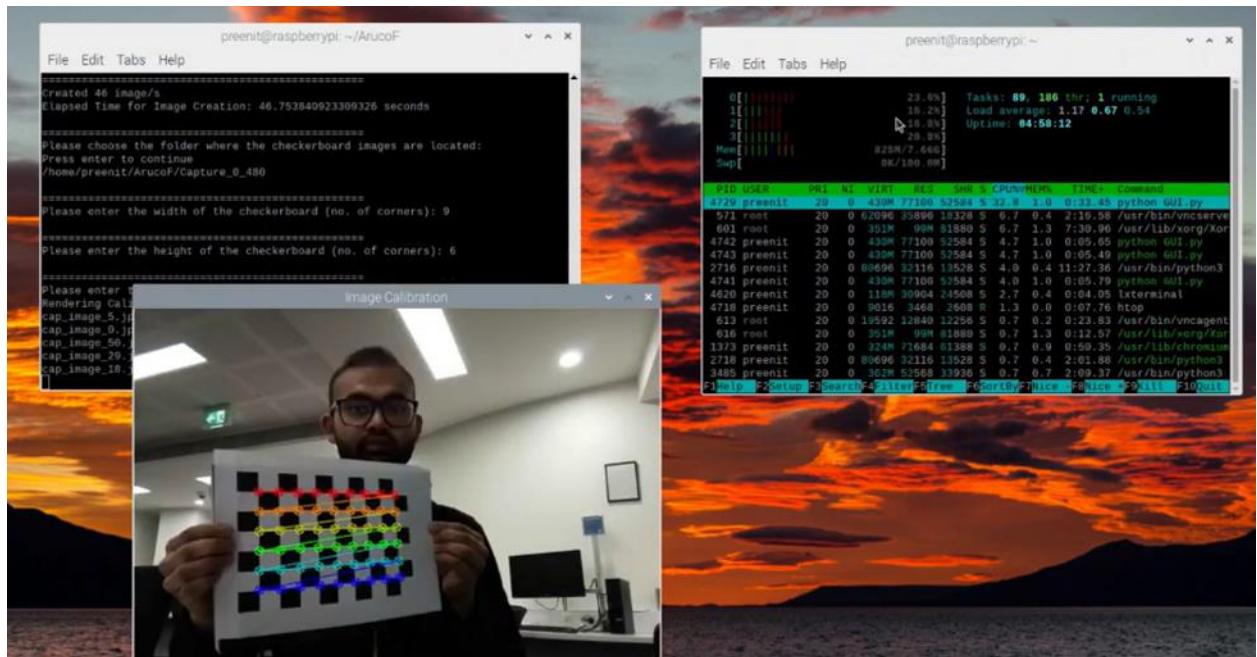


Figure 15 - Metrics when Performing Calibration on Images

Referring to figure 15 above:

- RES: slight hike to 78 mb of memory being occupied by the current program. This is still less than 50% of total memory and thus the test has passed.
- CPU%: here we are also able to see that the current running process is occupying more of the CPU time than for previous processes. However, the program is still operating efficiently. This test has also passed.
- MEM%: The memory being used here is roughly 1%, which is significantly less than 50% of the total available memory (8 gb). Therefore, this test has also passed.

Refer to the video for real-time full test coverage.

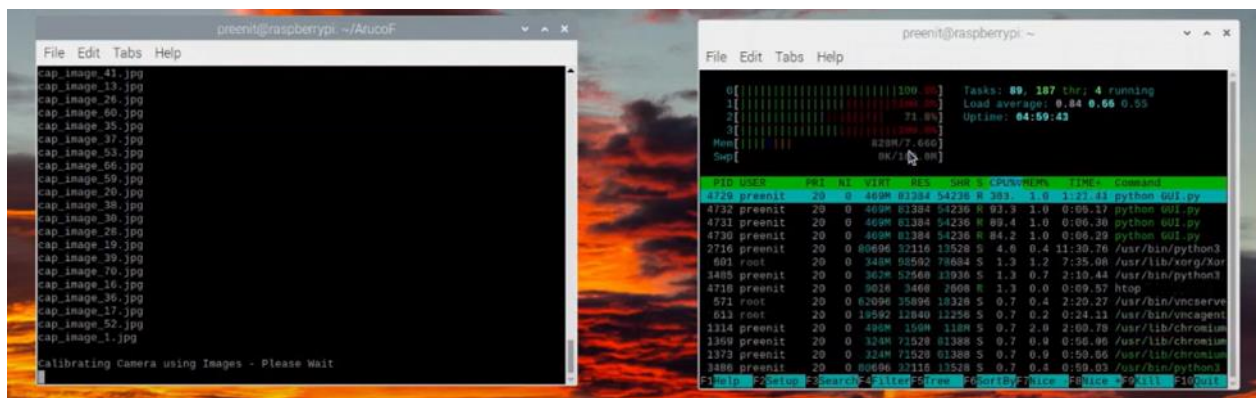


Figure 16 - Metrics when Producing Useful Results from Calibration

Referring to figure 16 above:

- RES: slight hike to 85 mb of memory being occupied by the current program. This is still less than 50% of total memory and thus the test has passed.
- CPU%: here we are also able to see that the current running process is occupying more of the CPU time (greater than double) than for previous processes. However, the program is still operating efficiently. This test has also passed.

- MEM%: The memory being used here is roughly 1.1%, which is still significantly less than 50% of the total available memory (8 gb). Therefore, this test has also passed.

Refer to the video for real-time full test coverage.

## Usage during Generate Markers

Video Reference - <https://youtu.be/VSQzocP82l8>

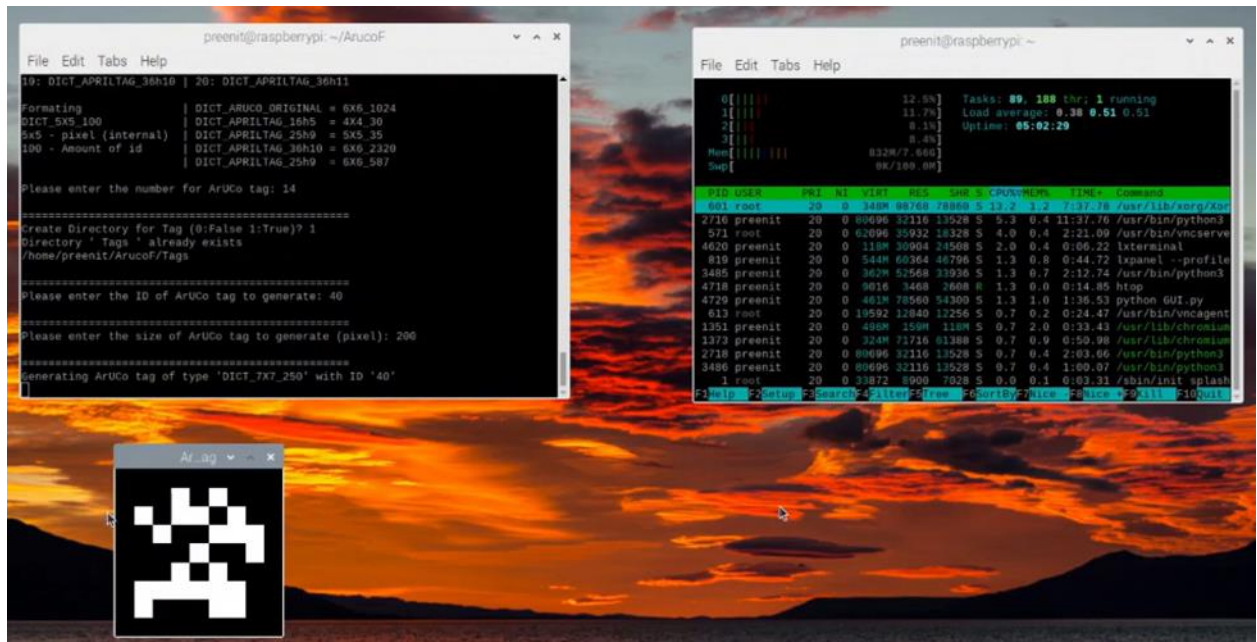


Figure 17 - Metrics Marker Creation

Referring to figure 17 above:

- RES: slight hike to 95 mb of memory being occupied by the current program. This is due to the fact that we are saving the created marker to the file system (memory) This is still less than 50% of total memory and thus the test has passed.
- CPU%: here we are also able to see that the current running process is occupying significantly less of the CPU time as there is not much processing that needs to be done. The program is still operating efficiently. This test has also passed.
- MEM%: The memory being used here is roughly 1.2%, which is still significantly less than 50% of the total available memory (8 gb). Therefore, this test has also passed.

Refer to the video for real-time full test coverage.



## Usage during Pose Detection

Video Reference – [https://youtu.be/cjKDq\\_VEvKo](https://youtu.be/cjKDq_VEvKo)

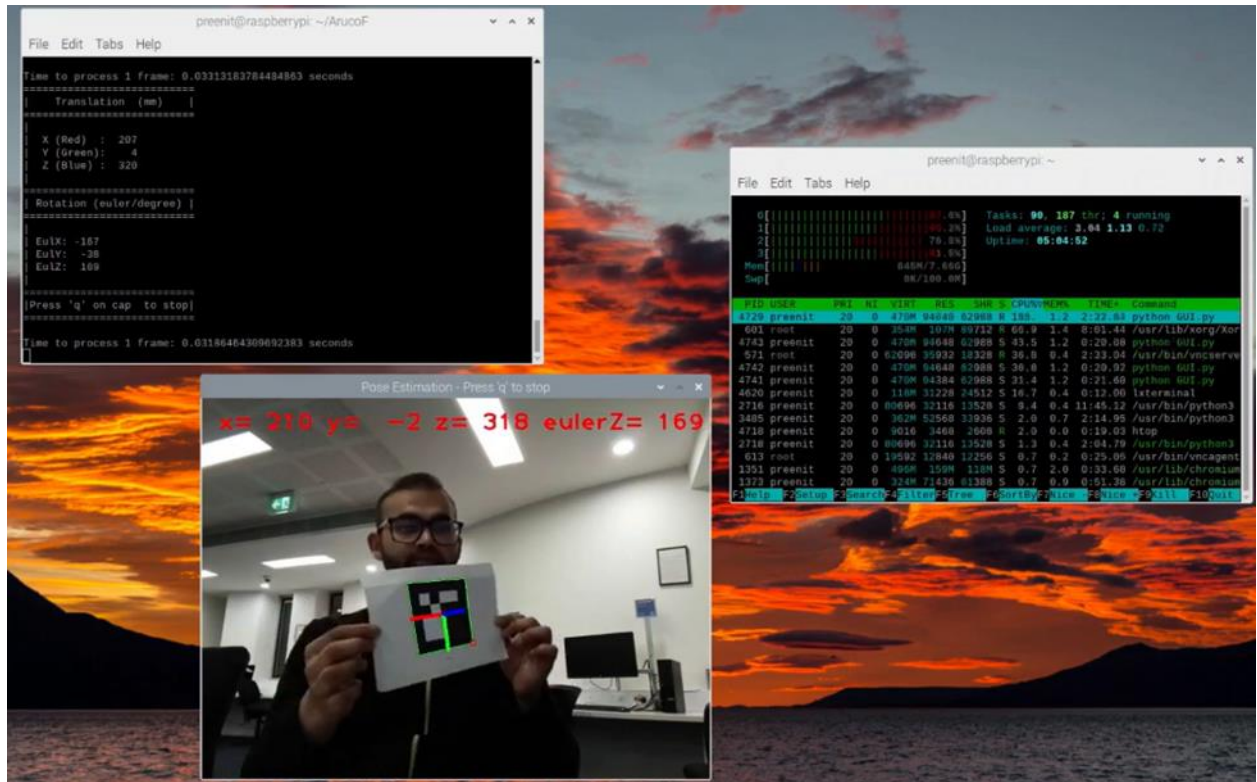


Figure 18 - Metrics Marker Creation

RES: slight hike to 94 mb of memory being occupied by the current program. This is due to the fact that this module makes access to memory for the calibration data and to store the values it has received to perform mathematical calculations on. This is still less than 50% of total memory and thus the test has passed.

CPU%: here we are also able to see that the current running process is occupying significantly greater of the CPU time as there is much processing that needs to be done (calculations using algorithms). The program is still operating efficiently. This test has also passed.

MEM%: The memory being used here is roughly 1.4%, which is still significantly less than 50% of the total available memory (8 gb). Therefore, this test has also passed.

Refer to the video for real-time full test coverage.

### 7.3.3 Compatibility:

**System must be able to successfully transfer required information to relevant software and hardware components working in the same environment with 100% compatibility and efficiency to avoid data loss.**

**More than 50% of the system's software/hardware components should be able to function independently.**

1	True	True
2	True	True
3	True	True
4	True	True

### 7.3.4 Usability:

**Use of universal codes, symbols, characters and adhere to general language/cognitive standards so that users can recognize them in less than 5 seconds.**

**Users should be able to learn to operate the system in less than 1 day.**

**Allow users to make less than one error per 10 operations**

**User interface is practically laid out and easy to navigate.**

**Can be operated by colourblind people as well as deaf people.**

1	True	True	True	True	N/A
2	True	True	True	True	N/A
3	True	True	True	True	N/A
4	True	True	True	True	N/A

### 7.3.5 Reliability:

**System must be available 100% or minimum 99% of the time when in operation.**

**Error rate < 1%.**

**Expected time between failures (downtime) is limited to 5-7 minutes.**

1	True	True	True (Calibration)
2	True	True	True (Calibration +Camera)
3	True	True	True
4	True	True	True

Refer to the video links from above testing which show that the system was available almost 100% of the time when in operation (there is no evidence of unavailability through the extensive testing).

The downtime when waiting for the system to do some processes for which the speed of the system does not matter still has a downtime no more than 5-7 minutes, examples are waiting for the camera view to open, calculating camera and calibration matrices etc.

7.3.6 Maintainability:

**Testing satisfies >90% of the test criteria.**

**No more than 18 days needed to implement, test, and deploy the modification.**

**More than 50% of the system's software/hardware components should be able to function independently.**

1	True	True	True
2	True	True	True
3	True	True	True
4	True	True	True

Through extensive testing, we can conclude that >90% of the tests were successful and the changes that were implemented as a result of the testing (e.g. the user interface needs to be more user friendly) took no more than 18 days to implement, test and deploy.