Pose Detector Guide

As of 11/10/2022

https://github.com/DSLeong/PoseDetectorPi

Introduction

This document contains instructions for the users of the Pose Detector. This document goes over:

- Installation Guide for both Visual Studio and Raspberry Pi
- Program Guide and Usage

Contents

Installation Guide

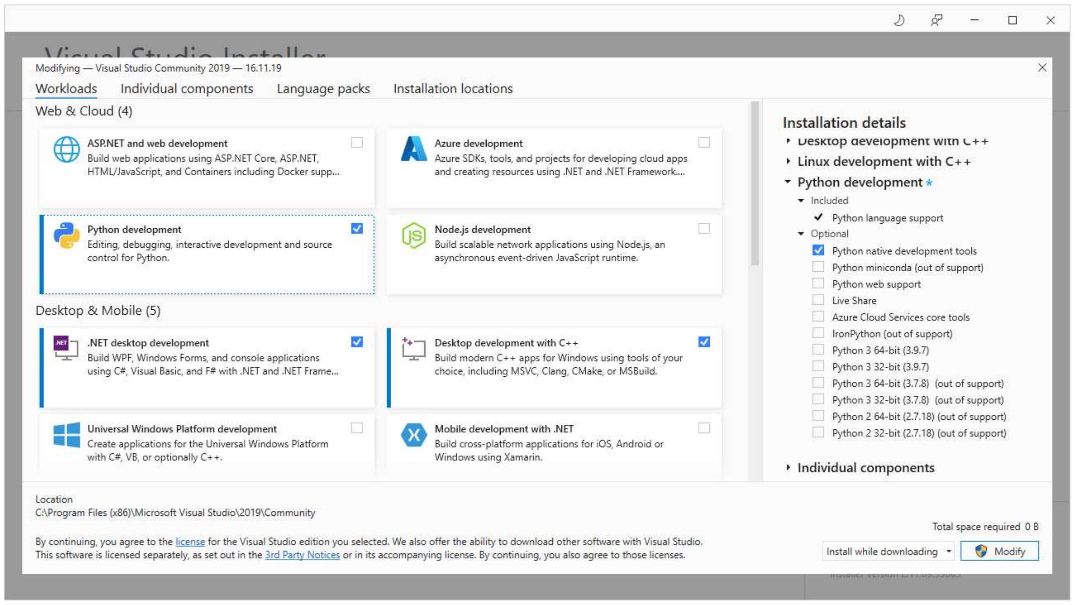
Visual Studio

Software/Material Needed

- Pose Detector Code (via https://github.com/DSLeong/PoseDetectorPi)
- Visual Studio

Installation Guide

1. Install Python development within Visual Studio Installer.



Modifying — Visual Studio Community 2019 — 16.11.19

Web & Cloud (4)

ASP.NET and web development
Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker supp...

Azure development
Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET and .NET Framework...

Python development [✓]
Editing, debugging, interactive development and source control for Python.

Node.js development
Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Desktop & Mobile (5)

.NET desktop development [✓]
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...

Desktop development with C++ [✓]
Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.

Universal Windows Platform development
Create applications for the Universal Windows Platform with C#, VB, or optionally C++.

Mobile development with .NET
Build cross-platform applications for iOS, Android or Windows using Xamarin.

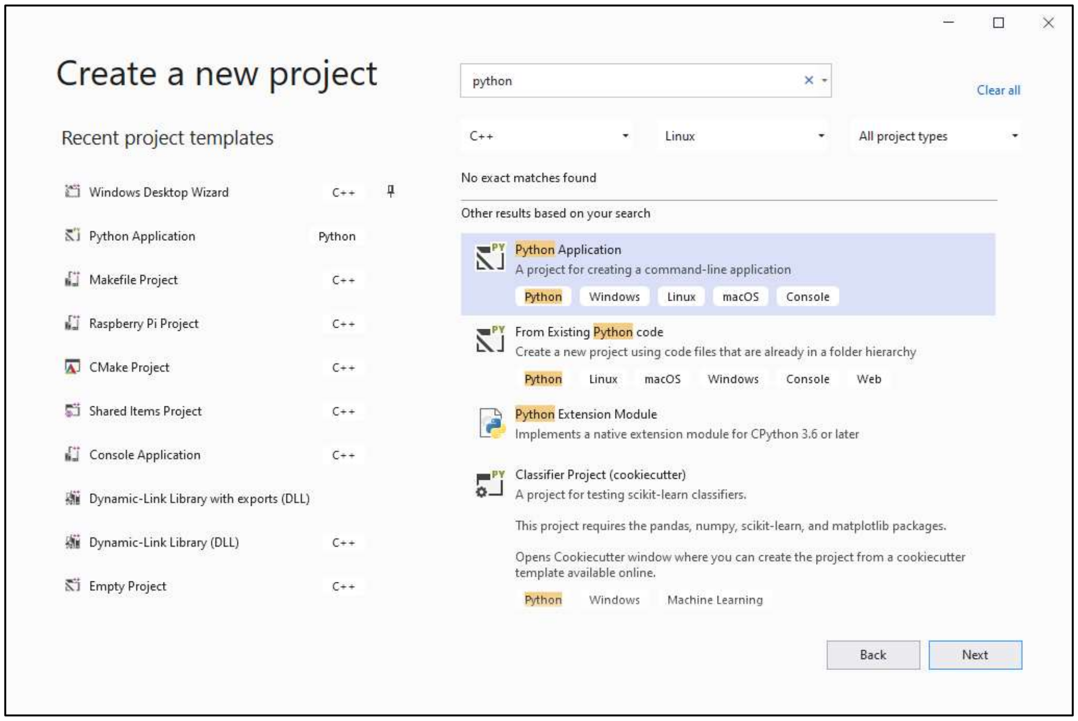
Installation details
- Desktop development with C++
- Linux development with C++
- Python development
  - Included
    - ✓ Python language support
  - Optional
    - [✓] Python native development tools
    - [ ] Python miniconda (out of support)
    - [ ] Python web support
    - [ ] Live Share
    - [ ] Azure Cloud Services core tools
    - [ ] IronPython (out of support)
    - [ ] Python 3 64-bit (3.9.7)
    - [ ] Python 3 32-bit (3.9.7)
    - [ ] Python 3 64-bit (3.7.8) (out of support)
    - [ ] Python 3 32-bit (3.7.8) (out of support)
    - [ ] Python 2 64-bit (2.7.18) (out of support)
    - [ ] Python 2 32-bit (2.7.18) (out of support)
- Individual components

Location
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 0 B

2. Create a new project with the Python Application template.



Recent project templates

Windows Desktop Wizard	C++
Python Application	Python
Makefile Project	C++
Raspberry Pi Project	C++
CMake Project	C++
Shared Items Project	C++
Console Application	C++
Dynamic-Link Library with exports (DLL)	
Dynamic-Link Library (DLL)	C++
Empty Project	C++

No exact matches found

Other results based on your search

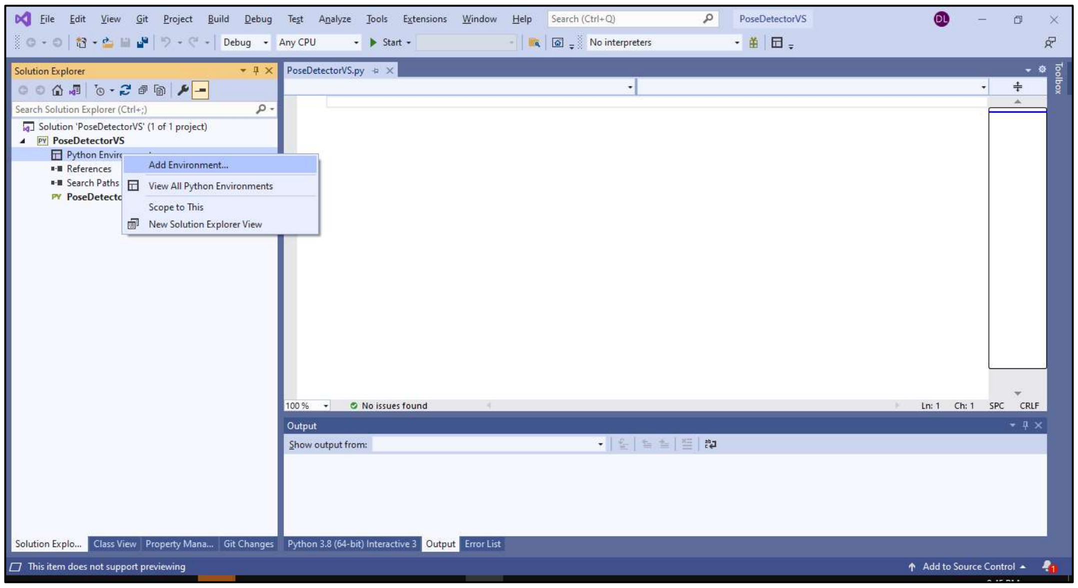
Python Application
A project for creating a command-line application
Python Windows Linux macOS Console

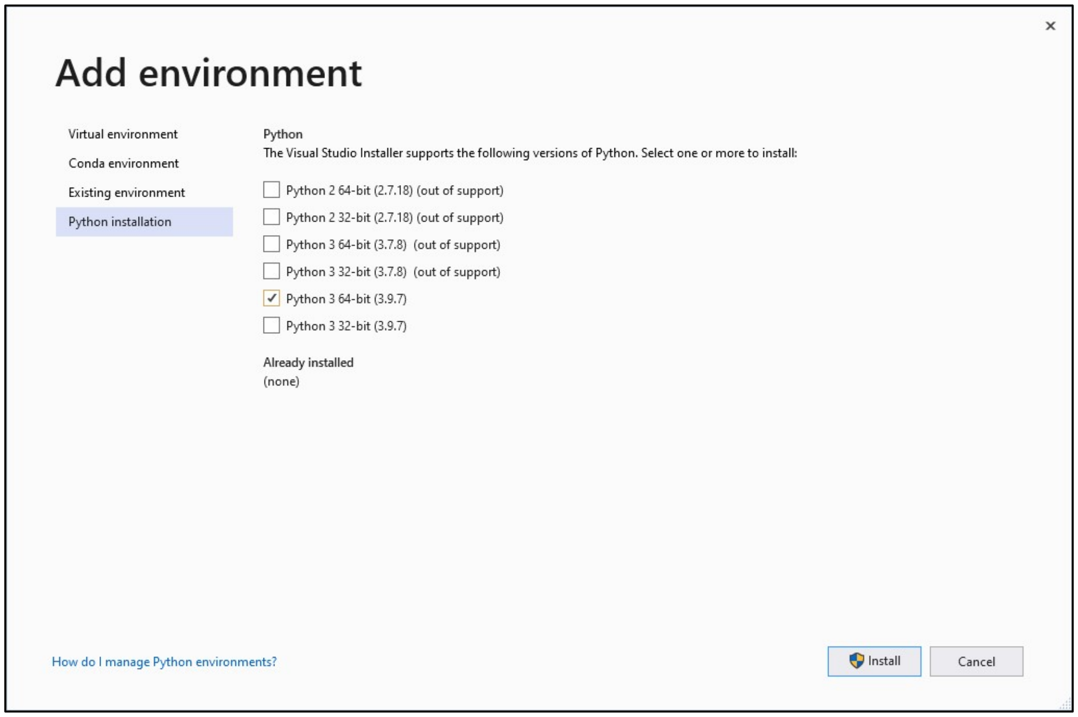
From Existing Python code
Create a new project using code files that are already in a folder hierarchy
Python Linux macOS Windows Console Web

Python Extension Module
Implements a native extension module for CPython 3.6 or later

Classifier Project (cookiecutter)
A project for testing scikit-learn classifiers.
This project requires the pandas, numpy, scikit-learn, and matplotlib packages.
Opens Cookiecutter window where you can create the project from a cookiecutter template available online.
Python Windows Machine Learning

3. Add New Python Environment (Python version > 3.0 and either 32-bit or 64-bit)




Add environment

Virtual environment
Conda environment
Existing environment
Python installation

Python
The Visual Studio Installer supports the following versions of Python. Select one or more to install:

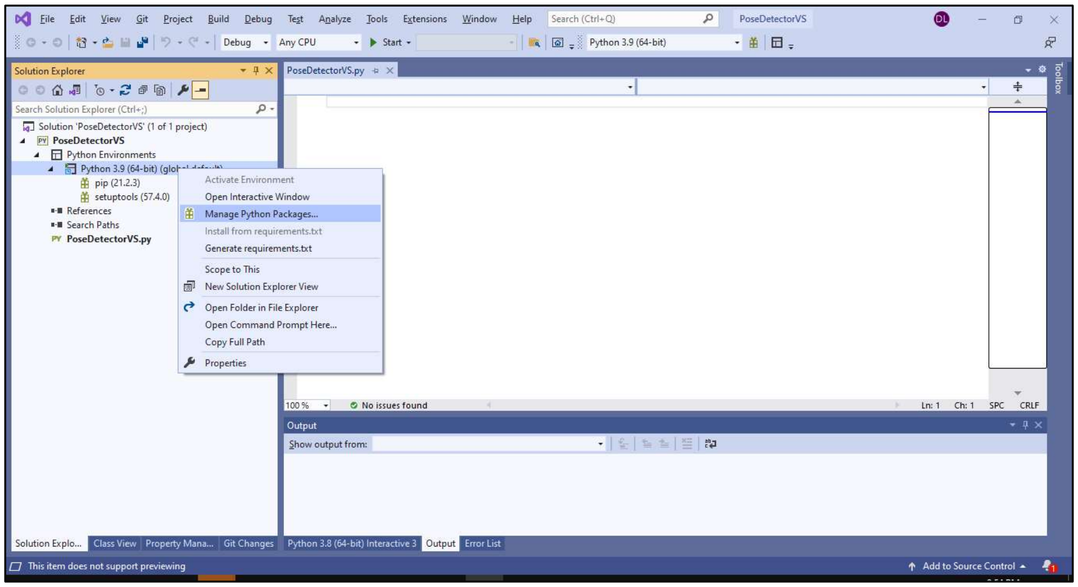
- [ ] Python 2 64-bit (2.7.18) (out of support)
- [ ] Python 2 32-bit (2.7.18) (out of support)
- [ ] Python 3 64-bit (3.7.8) (out of support)
- [ ] Python 3 32-bit (3.7.8) (out of support)
- [✓] Python 3 64-bit (3.9.7)
- [ ] Python 3 32-bit (3.9.7)

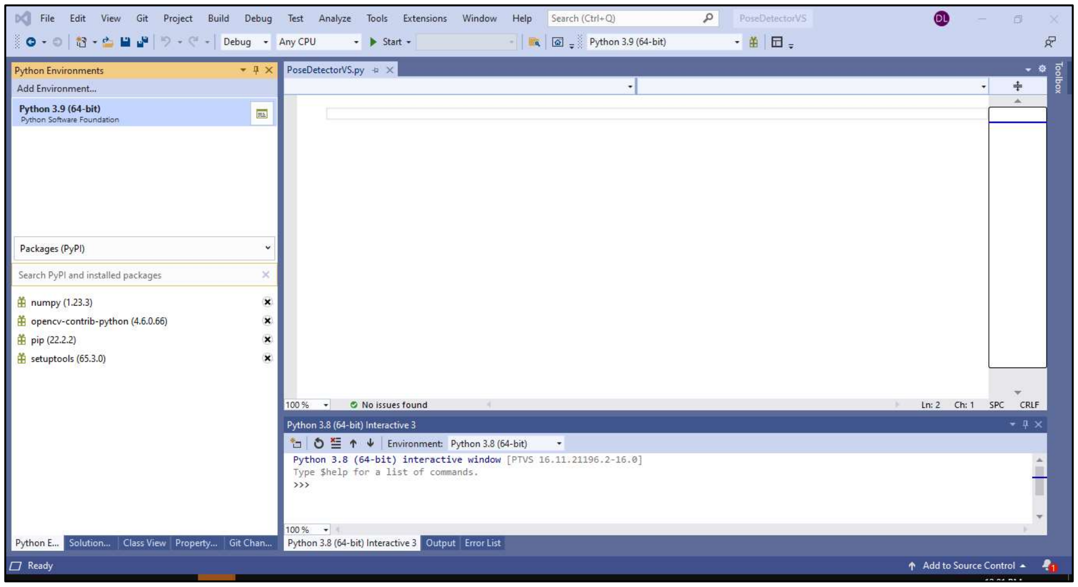
Already installed
(none)

How do I manage Python environments?

Install Cancel

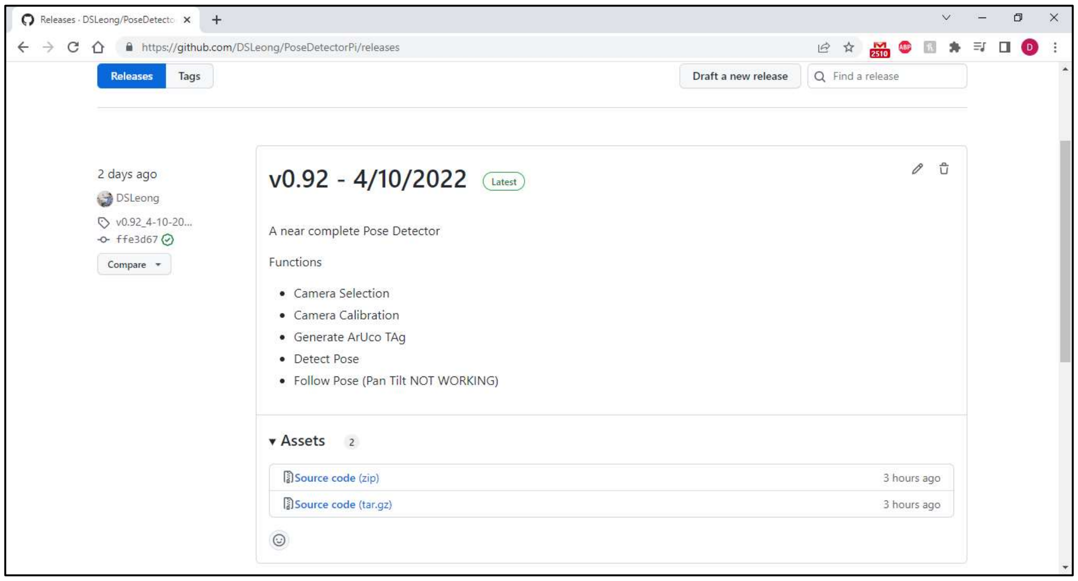
4. Install/Update Python Packages
    a. pip
    b. setuptools
    c. numpy
    d. opencv-contrib-python


Solution Explorer
Solution 'PoseDetectorVS' (1 of 1 project)
PoseDetectorVS
Python Environments
Python 3.9 (64-bit) (global default)
pip (21.2.3)
setuptools (57.4.0)
References
Search Paths
PoseDetectorVS.py
Activate Environment
Open Interactive Window
Manage Python Packages...
Install from requirements.txt
Generate requirements.txt
Scope to This
New Solution Explorer View
Open Folder in File Explorer
Open Command Prompt Here...
Copy Full Path
Properties
No issues found
Output
Show output from:
This item does not support previewing
Add to Source Control


Python Environments
Add Environment...
Python 3.9 (64-bit)
Python Software Foundation
Packages (PyPI)
Search PyPI and installed packages
numpy (1.23.3)
opencv-contrib-python (4.6.0.66)
pip (22.2.2)
setuptools (65.3.0)
Python 3.8 (64-bit) Interactive 3
Environment: Python 3.8 (64-bit)
Ready
Add to Source Control

```
Python 3.8 (64-bit) interactive window [PTVS 16.11.21196.2-16.0]
Type $help for a list of commands.
>>>
```

5. Download Code (https://github.com/DSLeong/PoseDetectorPi)
   a. Via the Releases (https://github.com/DSLeong/PoseDetectorPi/releases)


Releases · DSLeong/PoseDetecto
https://github.com/DSLeong/PoseDetectorPi/releases
Releases Tags
Draft a new release
Find a release

2 days ago
DSLeong
v0.92_4-10-20...
ffe3d67
Compare

v0.92 - 4/10/2022 Latest

A near complete Pose Detector

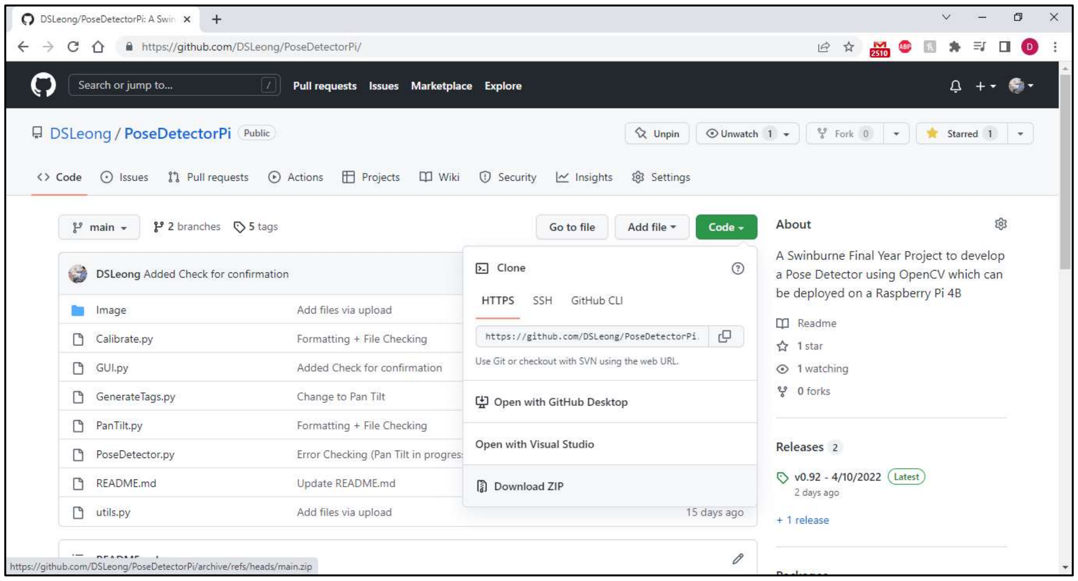
Functions

- Camera Selection
- Camera Calibration
- Generate ArUco TAg
- Detect Pose
- Follow Pose (Pan Tilt NOT WORKING)

Assets 2

Source code (zip)	3 hours ago
Source code (tar.gz)	3 hours ago

   b. Experimentally via the code button in GitHub


DSLeong/PoseDetectorPi: A Swin
https://github.com/DSLeong/PoseDetectorPi/
Search or jump to... Pull requests Issues Marketplace Explore
DSLeong / PoseDetectorPi Public
Unpin Unwatch 1 Fork 0 Starred 1
Code Issues Pull requests Actions Projects Wiki Security Insights Settings
main 2 branches 5 tags Go to file Add file Code

DSLeong Added Check for confirmation

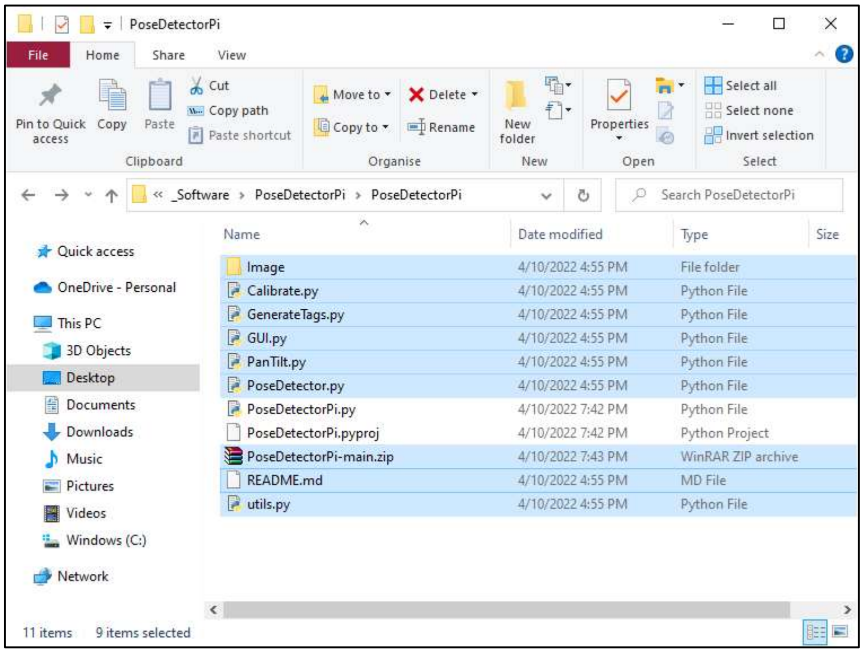
Image	Add files via upload
Calibrate.py	Formatting + File Checking
GUI.py	Added Check for confirmation
GenerateTags.py	Change to Pan Tilt
PanTilt.py	Formatting + File Checking
PoseDetector.py	Error Checking (Pan Tilt in progres
README.md	Update README.md
utils.py	Add files via upload	15 days ago

Clone
HTTPS SSH GitHub CLI
https://github.com/DSLeong/PoseDetectorPi.
Use Git or checkout with SVN using the web URL.
Open with GitHub Desktop
Open with Visual Studio
Download ZIP

About
A Swinburne Final Year Project to develop a Pose Detector using OpenCV which can be deployed on a Raspberry Pi 4B
Readme
1 star
1 watching
0 forks
Releases 2
v0.92 - 4/10/2022 Latest
2 days ago
+ 1 release

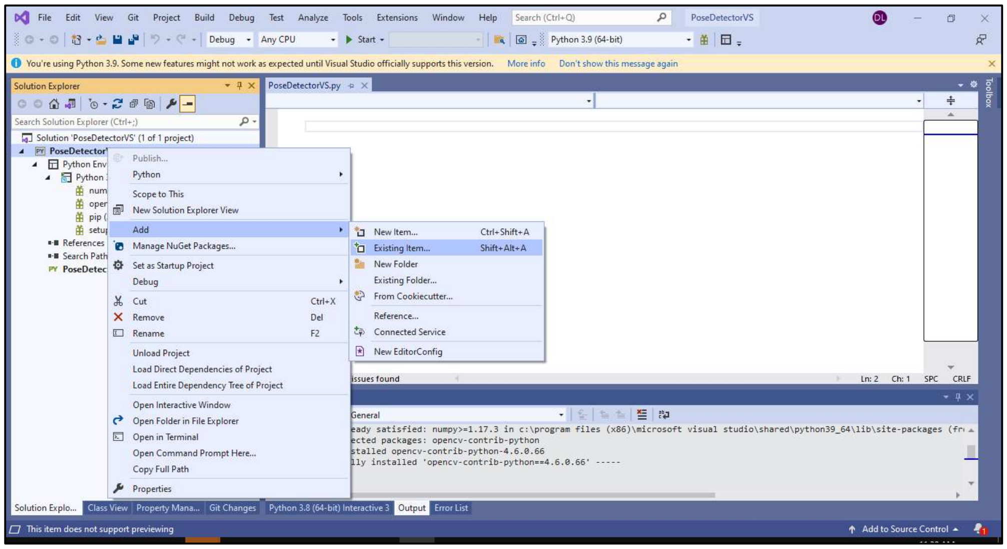
https://github.com/DSLeong/PoseDetectorPi/archive/refs/heads/main.zip

6. Extract Code into your VS project Directory



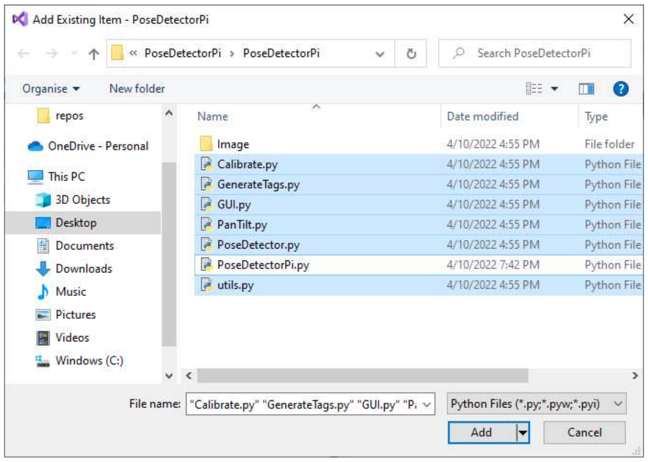
Name	Date modified	Type	Size
Image	4/10/2022 4:55 PM	File folder	
Calibrate.py	4/10/2022 4:55 PM	Python File	
GenerateTags.py	4/10/2022 4:55 PM	Python File	
GUI.py	4/10/2022 4:55 PM	Python File	
PanTilt.py	4/10/2022 4:55 PM	Python File	
PoseDetector.py	4/10/2022 4:55 PM	Python File	
PoseDetectorPi.py	4/10/2022 7:42 PM	Python File	
PoseDetectorPi.pyproj	4/10/2022 7:42 PM	Python Project	
PoseDetectorPi-main.zip	4/10/2022 7:43 PM	WinRAR ZIP archive	
README.md	4/10/2022 4:55 PM	MD File	
utils.py	4/10/2022 4:55 PM	Python File	

7. Add Python Files to Current Project.



You're using Python 3.9. Some new features might not work as expected until Visual Studio officially supports this version. More info Don't show this message again

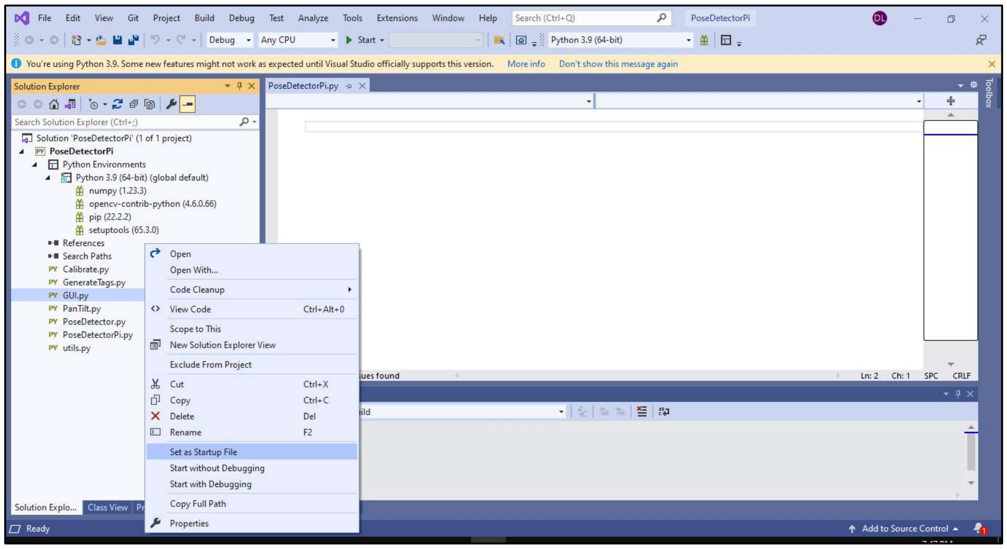
```
eady satisfied: numpy>=1.17.3 in c:\program files (x86)\microsoft visual studio\shared\python39_64\lib\site-packages (fr
ected packages: opencv-contrib-python
stalled opencv-contrib-python-4.6.0.66
lly installed 'opencv-contrib-python==4.6.0.66' -----
```



Name	Date modified	Type
Image	4/10/2022 4:55 PM	File folder
Calibrate.py	4/10/2022 4:55 PM	Python File
GenerateTags.py	4/10/2022 4:55 PM	Python File
GUI.py	4/10/2022 4:55 PM	Python File
PanTilt.py	4/10/2022 4:55 PM	Python File
PoseDetector.py	4/10/2022 4:55 PM	Python File
PoseDetectorPi.py	4/10/2022 7:42 PM	Python File
utils.py	4/10/2022 4:55 PM	Python File

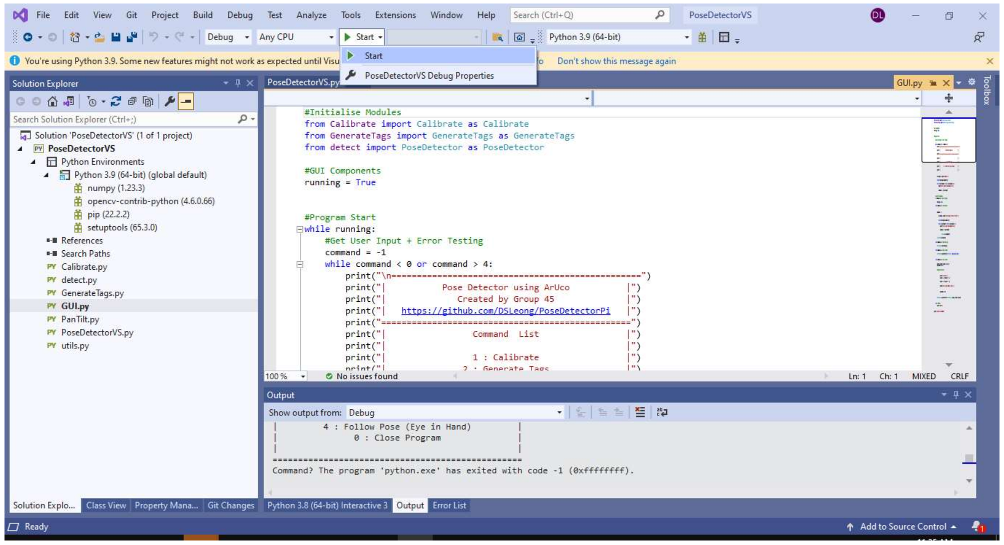
File name: "Calibrate.py" "GenerateTags.py" "GUI.py" "P
Python Files (*.py;*.pyw;*.pyi)

8. Set GUI.py as 'Set as Startup File'



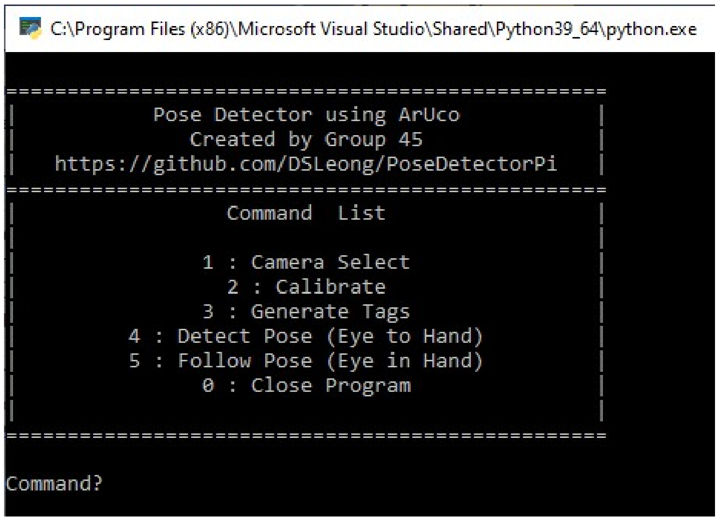
You're using Python 3.9. Some new features might not work as expected until Visual Studio officially supports this version. More info Don't show this message again

9. Run the Project.



```
#Initialise Modules
from Calibrate import Calibrate as Calibrate
from GenerateTags import GenerateTags as GenerateTags
from detect import PoseDetector as PoseDetector

#GUI Components
running = True


#Program Start
while running:
    #Get User Input + Error Testing
    command = -1
    while command < 0 or command > 4:
        print("\n==================================================")
        print("|            Pose Detector using ArUco           |")
        print("|               Created by Group 45              |")
        print("|   https://github.com/DSLeong/PoseDetectorPi    |")
        print("==================================================")
        print("|                  Command  List                 |")
        print("|                                                |")
        print("|                  1 : Calibrate                 |")
```

```
|        4 : Follow Pose (Eye in Hand)            |
|              0 : Close Program                  |
|                                                 |
==================================================
Command? The program 'python.exe' has exited with code -1 (0xffffffff).
```


C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe

```
==================================================
|            Pose Detector using ArUco           |
|               Created by Group 45              |
|   https://github.com/DSLeong/PoseDetectorPi    |
==================================================
|                  Command  List                 |
|                                                |
|               1 : Camera Select                |
|                 2 : Calibrate                  |
|               3 : Generate Tags                |
|         4 : Detect Pose (Eye to Hand)          |
|         5 : Follow Pose (Eye in Hand)          |
|               0 : Close Program                |
|                                                |
==================================================

Command?
```

Raspberry Pi

Material Needed

- Pose Detector Code (via https://github.com/DSLeong/PoseDetectorPi)
- OpenCV (Version >3.0)
- Pan-Tilt HAT (Optional)

Installing Modules

OpenCV

Video reference (near identical): https://www.youtube.com/watch?v=QzVYnG-WaM4
Within Terminal:

[Update/upgrade Raspberry Pi]
```
sudo apt-get update && sudo apt-get upgrade
```

[Check Python version (May need to install a Python 3 version)]
```
python -V
python3 -V
```

[System Packages & Upgrades]
```
sudo apt install -y build-essential cmake pkg-config libjpeg-dev
libtiff5-dev libjasper-dev libpng-dev libavcodec-dev libavformat-
dev libswscale-dev libv4l-dev libxvidcore-dev libx264-dev
libfontconfig1-dev libcairo2-dev libgdk-pixbuf2.0-dev libpango1.0-
dev libgtk2.0-dev libgtk-3-dev libatlas-base-dev gfortran libhdf5-
dev libhdf5-serial-dev libhdf5-103 libqt5gui5 libqt5webkit5
libqt5test5 python3-pyqt5 python3-dev

pip install --upgrade pip setuptools wheel numpy
```

[Enable Legacy Camera Support (Raspi 4)]
```
sudo raspi-config
 Interface Options
 Legacy Camera Support – Enable
```

[Using PiCamera]
```
pip install "picamera[array]"
```

[Enable Camera (Raspi 3)]
```
Open Raspberry Pi Configuration.
Select Interface tab.
Enable Camera.
```

[Installing OpenCV (NOTE – Takes about 2 hours to install)]
```
pip install opencv-contrib-python
```

[Checking if installed]
```
python
import cv2
cv2.__version__
```

Pan Tilt (OPTIONAL)

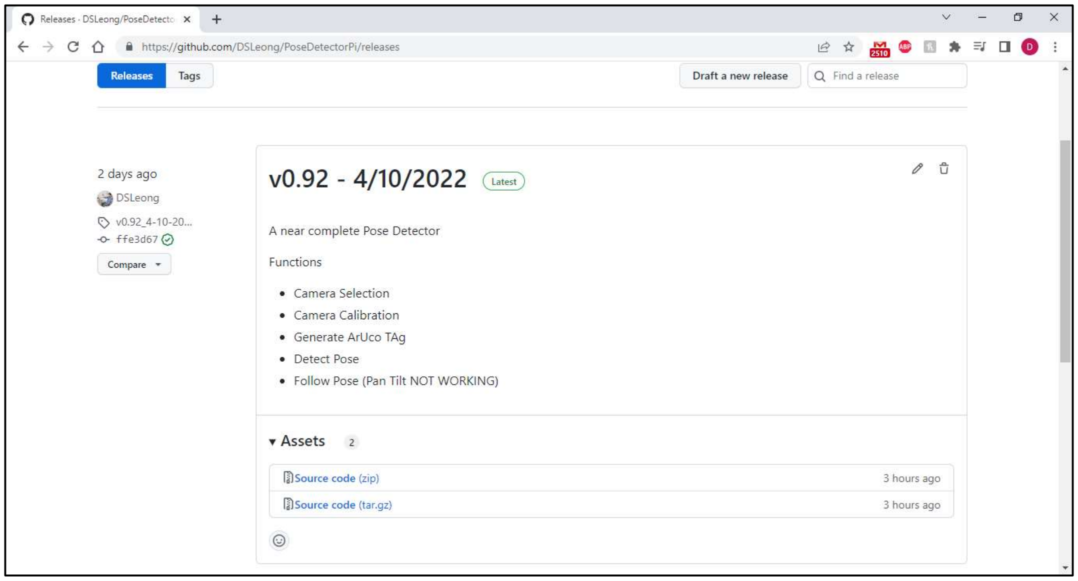
Video reference: https://www.youtube.com/watch?v=Dc9AEFw0hww
Please follow Video above or use the Guide below:

[Update/Upgrade Raspberry Pi]
```
sudo apt-get update && sudo apt-get upgrade
```

[Installing PanTiltHAT]
```
curl https://get.pimoroni.com/pantilthat | bash
```

Installation Guide

1. Download Code (https://github.com/DSLeong/PoseDetectorPi)
   a. Via the Releases (https://github.com/DSLeong/PoseDetectorPi/releases)


Releases · DSLeong/PoseDetecto
https://github.com/DSLeong/PoseDetectorPi/releases
Releases
Tags
Draft a new release
Find a release
2 days ago
DSLeong
v0.92_4-10-20...
ffe3d67
Compare
Latest
Assets 2
Source code (zip) 3 hours ago
Source code (tar.gz) 3 hours ago

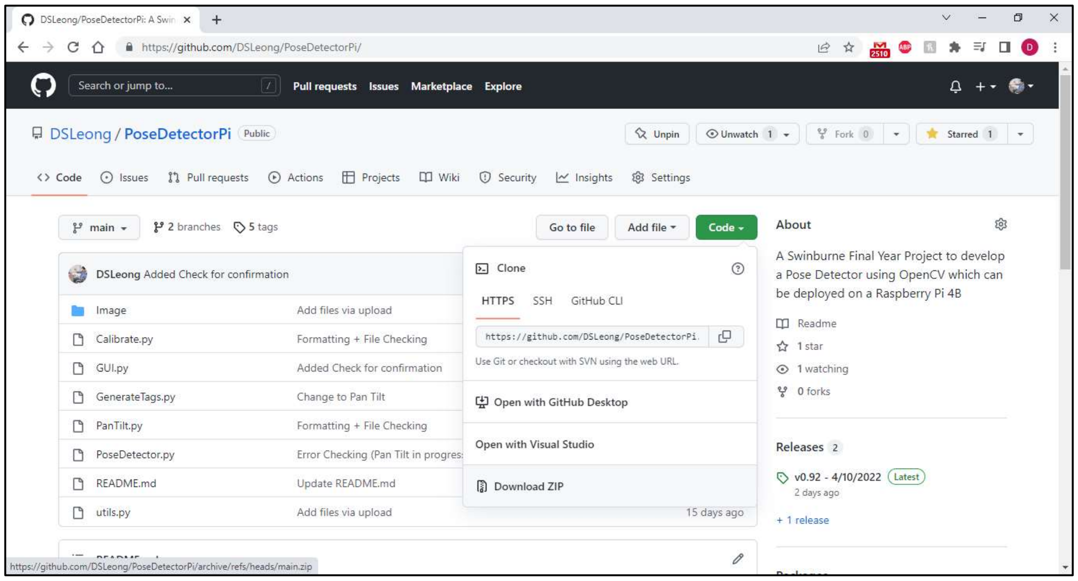
v0.92 - 4/10/2022

A near complete Pose Detector

Functions

- Camera Selection
- Camera Calibration
- Generate ArUco TAg
- Detect Pose
- Follow Pose (Pan Tilt NOT WORKING)

   b. Experimentally via the code button in GitHub



DSLeong Added Check for confirmation

Image	Add files via upload
Calibrate.py	Formatting + File Checking
GUI.py	Added Check for confirmation
GenerateTags.py	Change to Pan Tilt
PanTilt.py	Formatting + File Checking
PoseDetector.py	Error Checking (Pan Tilt in progres
README.md	Update README.md
utils.py	Add files via upload

15 days ago

A Swinburne Final Year Project to develop a Pose Detector using OpenCV which can be deployed on a Raspberry Pi 4B

2. Extract to a location such as /home/pi/
3. In terminal:

```
[Change Directory to location of extracted code]
cd /home/pi/PoseDetector
```

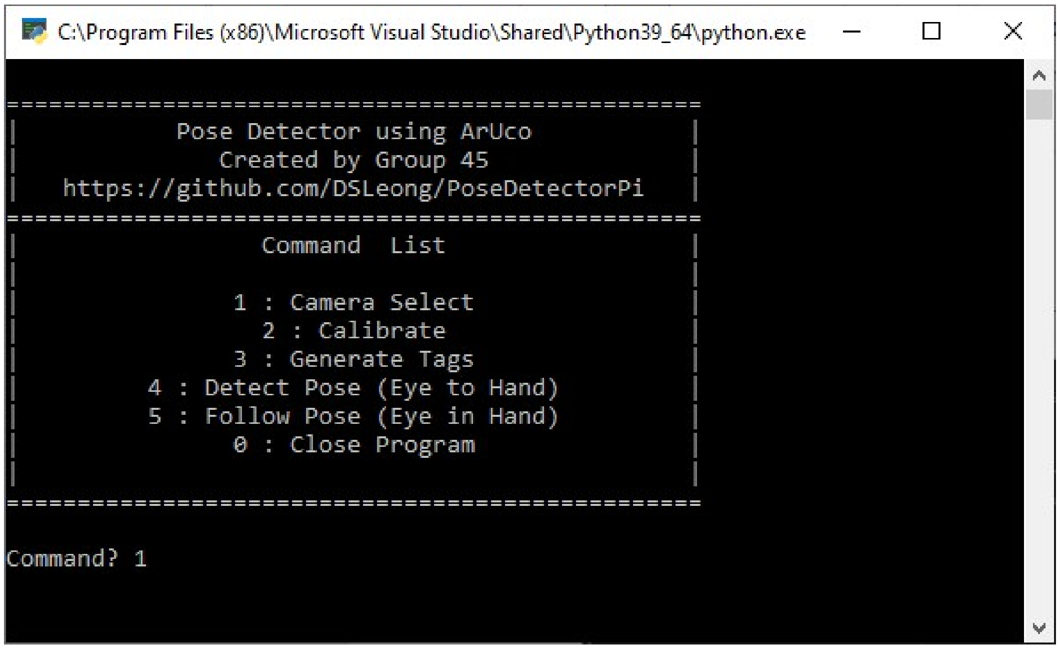
```
[Running the Software]
python GUI.py
or
python3 GUI.py
```

Program

Images are taken with Visual Studio; however, all the steps should be the same on a Raspberry Pi.

Configuring Camera

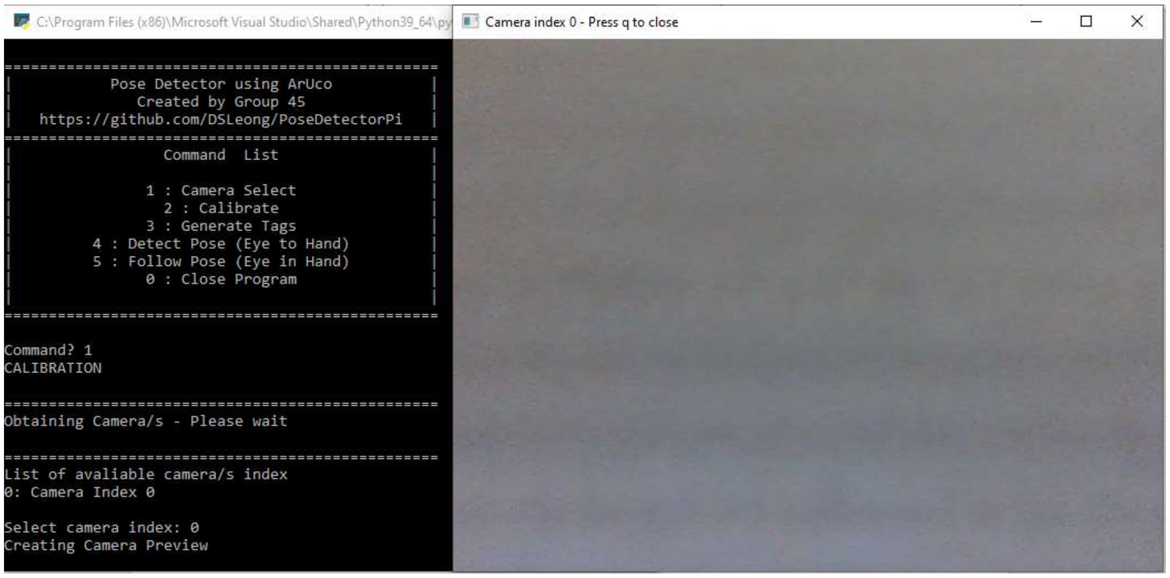
1. Run Program and Input ‘1’ for Camera Select



C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe

```
================================================
|           Pose Detector using ArUco          |
|              Created by Group 45             |
|    https://github.com/DSLeong/PoseDetectorPi  |
================================================
|                 Command  List                |
|                                              |
|               1 : Camera Select              |
|                 2 : Calibrate                |
|               3 : Generate Tags              |
|         4 : Detect Pose (Eye to Hand)        |
|         5 : Follow Pose (Eye in Hand)        |
|               0 : Close Program              |
|                                              |
================================================

Command? 1
```

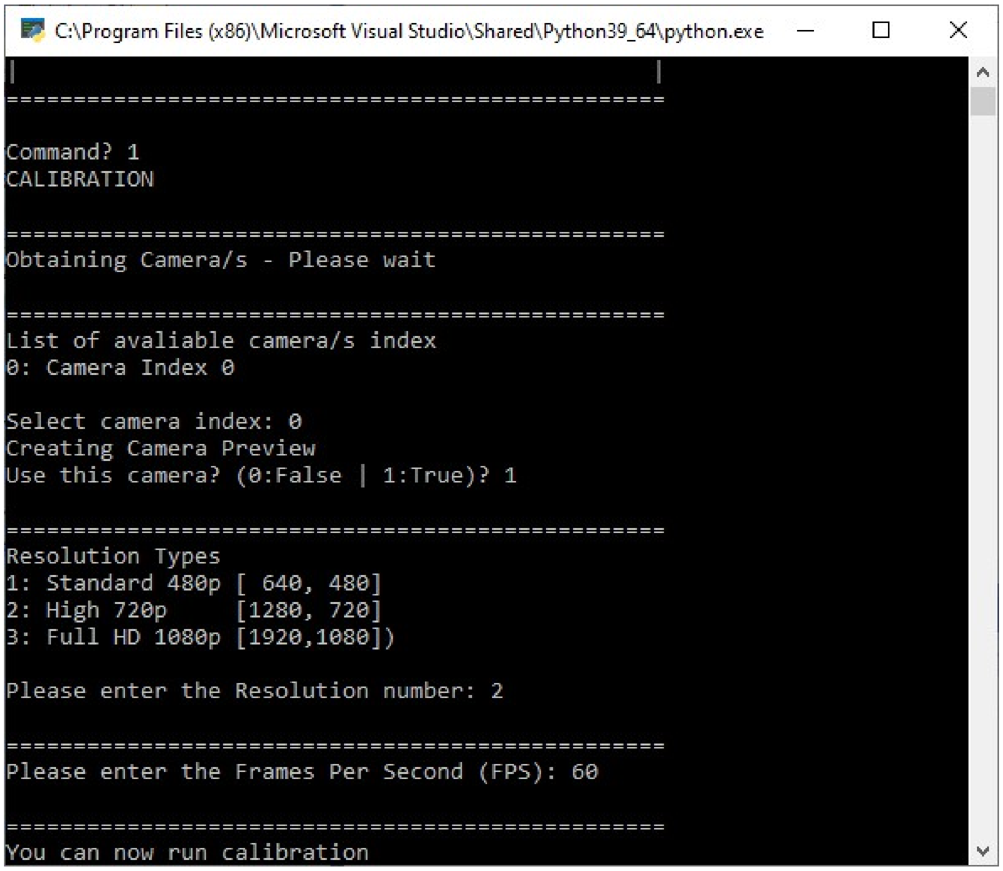
2. Select your desired Camera, it will also Preview the selected camera.



C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\py

```
================================================
|           Pose Detector using ArUco          |
|              Created by Group 45             |
|    https://github.com/DSLeong/PoseDetectorPi  |
================================================
|                 Command  List                |
|                                              |
|               1 : Camera Select              |
|                 2 : Calibrate                |
|               3 : Generate Tags              |
|         4 : Detect Pose (Eye to Hand)        |
|         5 : Follow Pose (Eye in Hand)        |
|               0 : Close Program              |
|                                              |
================================================

Command? 1
CALIBRATION

================================================
Obtaining Camera/s - Please wait

================================================
List of avaliable camera/s index
0: Camera Index 0

Select camera index: 0
Creating Camera Preview
```

3. Configure your camera settings



C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe

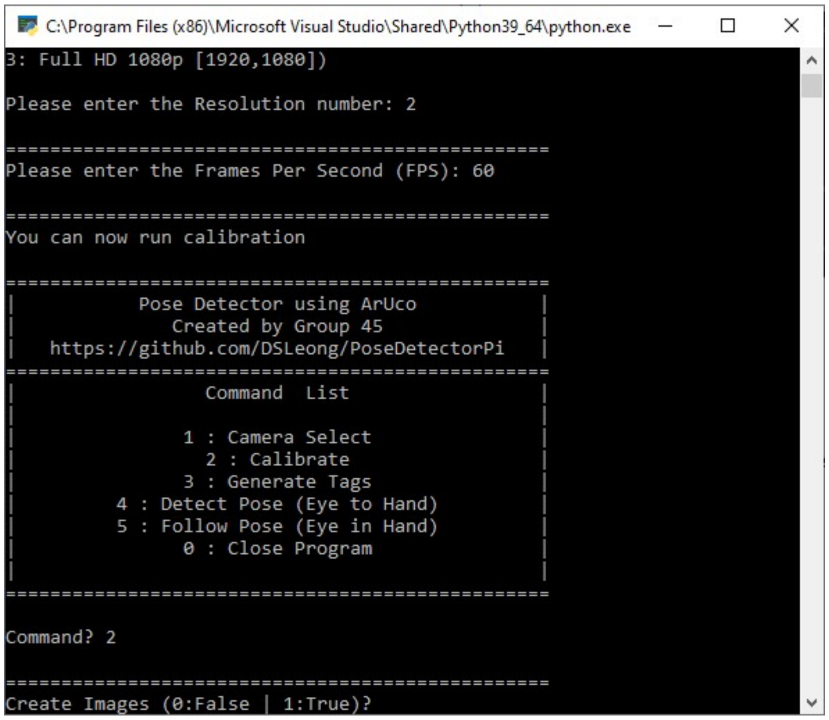
```
|                                                |
=================================================

Command? 1
CALIBRATION

=================================================
Obtaining Camera/s - Please wait

=================================================
List of avaliable camera/s index
0: Camera Index 0

Select camera index: 0
Creating Camera Preview
Use this camera? (0:False | 1:True)? 1

=================================================
Resolution Types
1: Standard 480p [ 640, 480]
2: High 720p     [1280, 720]
3: Full HD 1080p [1920,1080])

Please enter the Resolution number: 2

=================================================
Please enter the Frames Per Second (FPS): 60

=================================================
You can now run calibration
```

Calibration of Camera

Material Need:

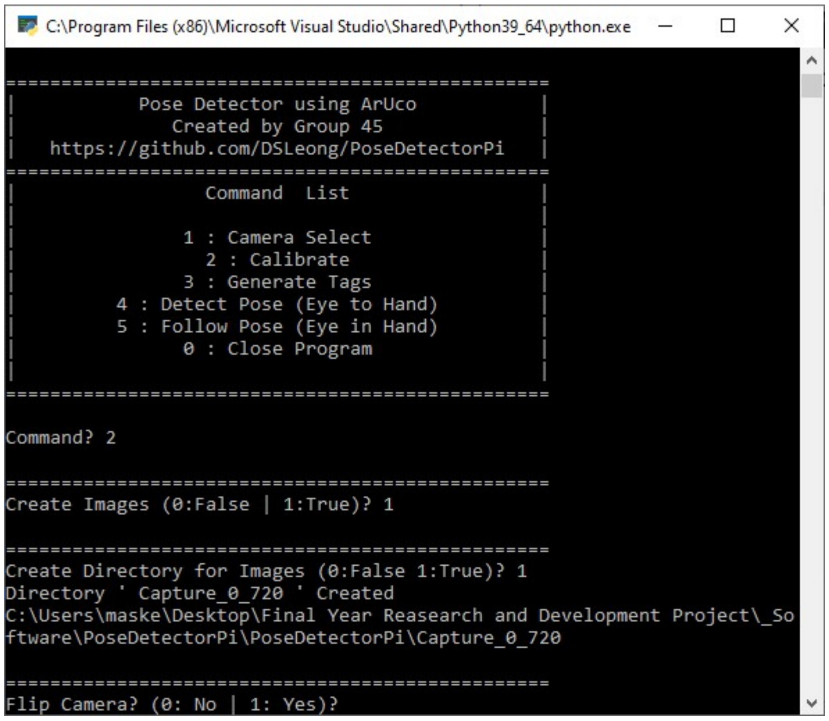
- Checkerboard print out (Attached at end)

1. Run the Camera Select first, then input '2' and Enter



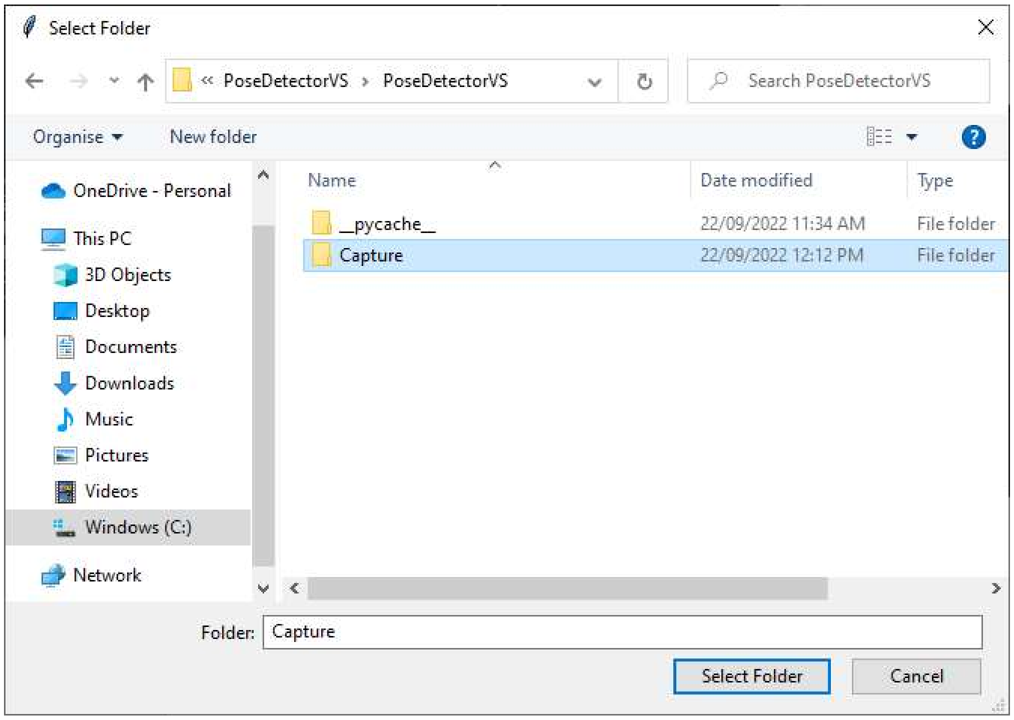
```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe
3: Full HD 1080p [1920,1080])

Please enter the Resolution number: 2

================================================
Please enter the Frames Per Second (FPS): 60

================================================
You can now run calibration

================================================
|           Pose Detector using ArUco          |
|              Created by Group 45             |
|   https://github.com/DSLeong/PoseDetectorPi  |
================================================
|                 Command  List                |
|                                              |
|               1 : Camera Select              |
|                 2 : Calibrate                |
|               3 : Generate Tags              |
|         4 : Detect Pose (Eye to Hand)        |
|         5 : Follow Pose (Eye in Hand)        |
|               0 : Close Program              |
|                                              |
================================================

Command? 2

================================================
Create Images (0:False | 1:True)?
```

2. (Optional but Recommended) Creating Images for Calibration
   a. Input '1' and Enter, then again. This will create a folder within your directory.



```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe

================================================
|           Pose Detector using ArUco          |
|              Created by Group 45             |
|   https://github.com/DSLeong/PoseDetectorPi  |
================================================
|                 Command  List                |
|                                              |
|               1 : Camera Select              |
|                 2 : Calibrate                |
|               3 : Generate Tags              |
|         4 : Detect Pose (Eye to Hand)        |
|         5 : Follow Pose (Eye in Hand)        |
|               0 : Close Program              |
|                                              |
================================================

Command? 2

================================================
Create Images (0:False | 1:True)? 1

================================================
Create Directory for Images (0:False 1:True)? 1
Directory ' Capture_0_720 ' Created
C:\Users\maske\Desktop\Final Year Reasearch and Development Project\_So
ftware\PoseDetectorPi\PoseDetectorPi\Capture_0_720

================================================
Flip Camera? (0: No | 1: Yes)?
```

b. Or select a folder



Name	Date modified	Type
__pycache__	22/09/2022 11:34 AM	File folder
Capture	22/09/2022 12:12 PM	File folder

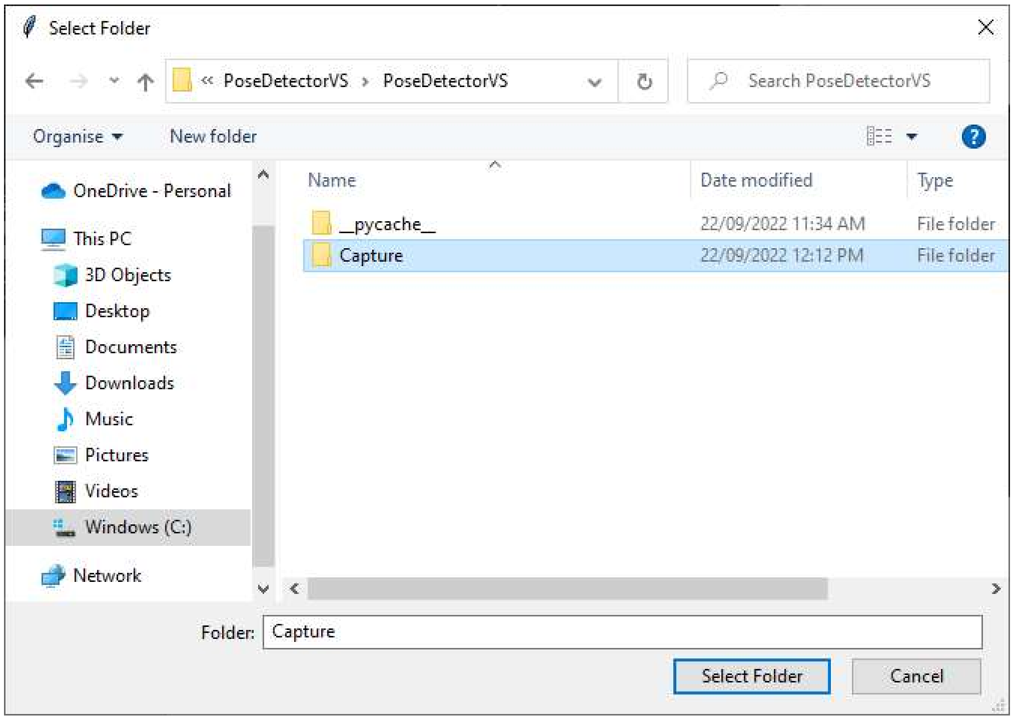
Folder: Capture

c. You can also flip your camera

d. A camera preview is opened, Present the Checkerboard print in various positions. Press ‘q’ on preview to stop. (More images = More processing)

e. Images are saved within the folder.

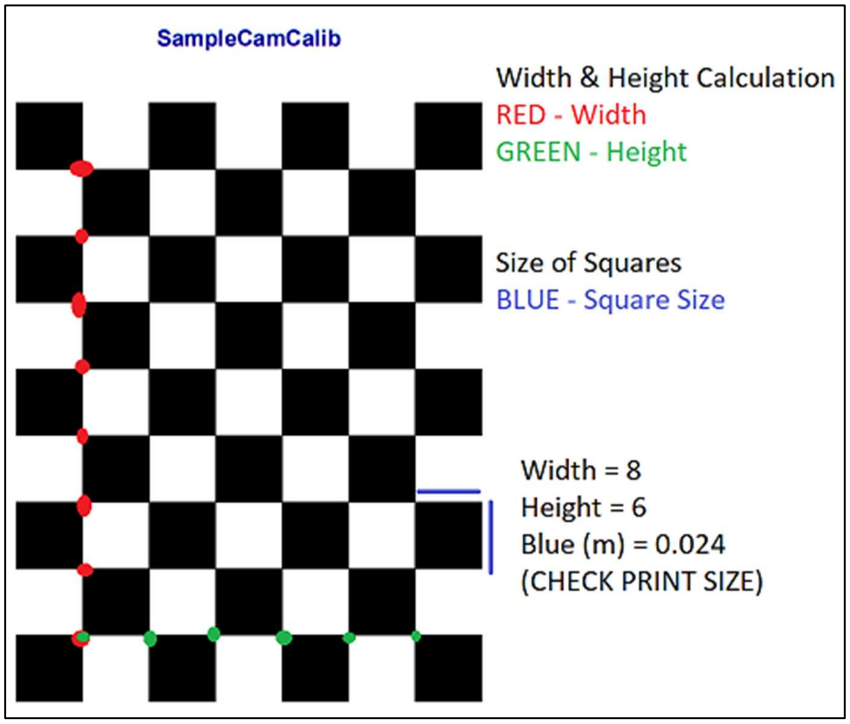
3. Enter and select the folder where the images that contain the checkerboard print



Name	Date modified	Type
__pycache__	22/09/2022 11:34 AM	File folder
Capture	22/09/2022 12:12 PM	File folder

Folder: Capture

4. With provided Checkerboard print specifications (attached at end/Part of GitHub)
    a. Width = 8 (or 6)
    b. Height = 6 (or 8)
    c. Size of Square = 24 mm (double check width/height of square due to printing)

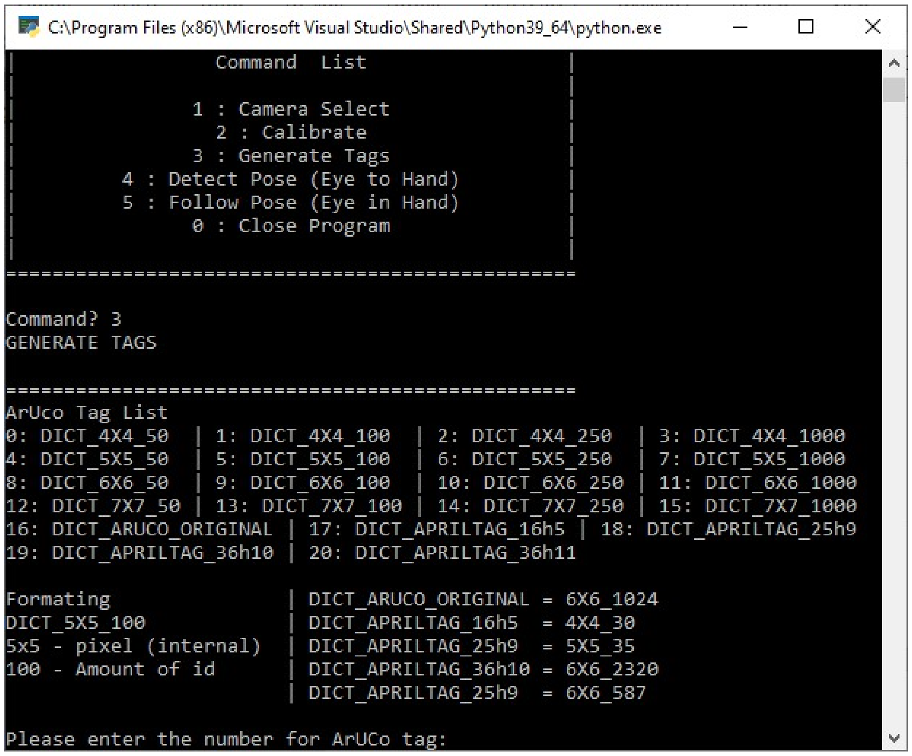


5. Once entered, images are previewed with calibration. Wait for program to process matrix for estimation. (more images = more processing)

Generate Markers

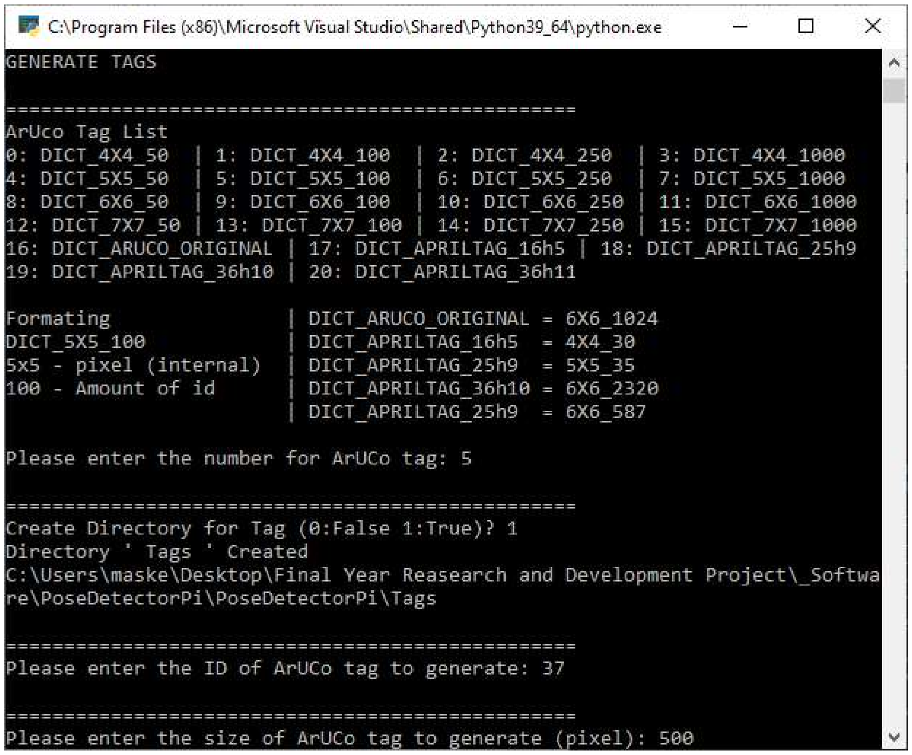
There is also a website which does the same (https://chev.me/arucogen/)

1. Input '3' and Enter. A List of ArUco Tags that can be used are displayed



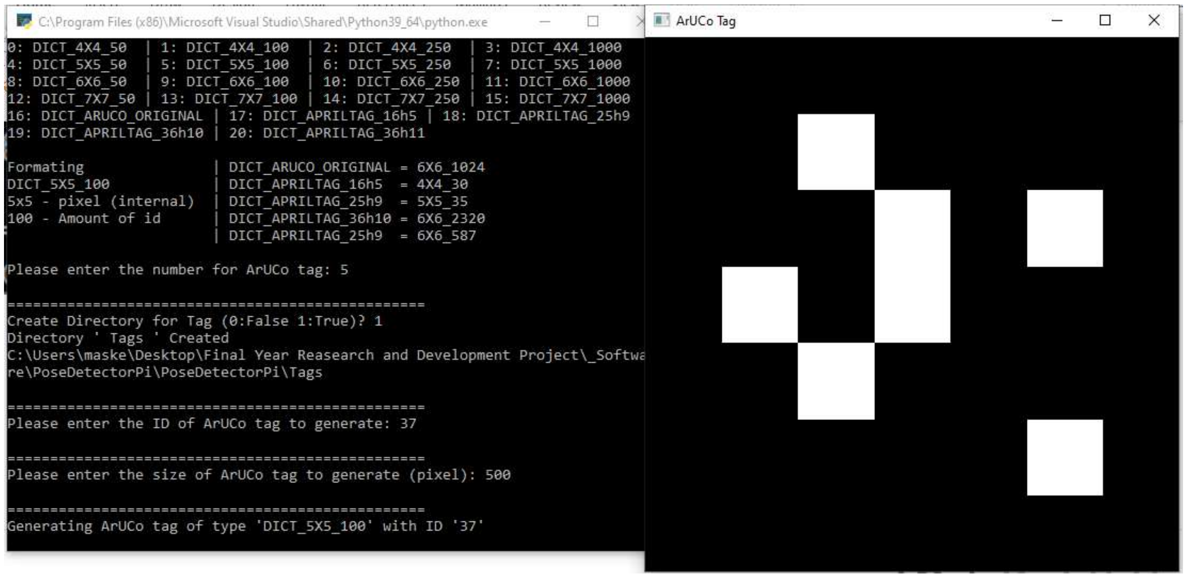
```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe
|                 Command  List                 |
|                                               |
|               1 : Camera Select               |
|                 2 : Calibrate                 |
|               3 : Generate Tags               |
|         4 : Detect Pose (Eye to Hand)         |
|         5 : Follow Pose (Eye in Hand)         |
|               0 : Close Program               |
|                                               |
=================================================

Command? 3
GENERATE TAGS

=================================================
ArUco Tag List
0: DICT_4X4_50  | 1: DICT_4X4_100  | 2: DICT_4X4_250  | 3: DICT_4X4_1000
4: DICT_5X5_50  | 5: DICT_5X5_100  | 6: DICT_5X5_250  | 7: DICT_5X5_1000
8: DICT_6X6_50  | 9: DICT_6X6_100  | 10: DICT_6X6_250 | 11: DICT_6X6_1000
12: DICT_7X7_50 | 13: DICT_7X7_100 | 14: DICT_7X7_250 | 15: DICT_7X7_1000
16: DICT_ARUCO_ORIGINAL | 17: DICT_APRILTAG_16h5 | 18: DICT_APRILTAG_25h9
19: DICT_APRILTAG_36h10 | 20: DICT_APRILTAG_36h11

Formating                | DICT_ARUCO_ORIGINAL = 6X6_1024
DICT_5X5_100             | DICT_APRILTAG_16h5  = 4X4_30
5x5 - pixel (internal)   | DICT_APRILTAG_25h9  = 5X5_35
100 - Amount of id       | DICT_APRILTAG_36h10 = 6X6_2320
                         | DICT_APRILTAG_25h9  = 6X6_587

Please enter the number for ArUCo tag:
```

2. For our case we will use 5 – DICT_5X5_100, id of 37 and size of 500.



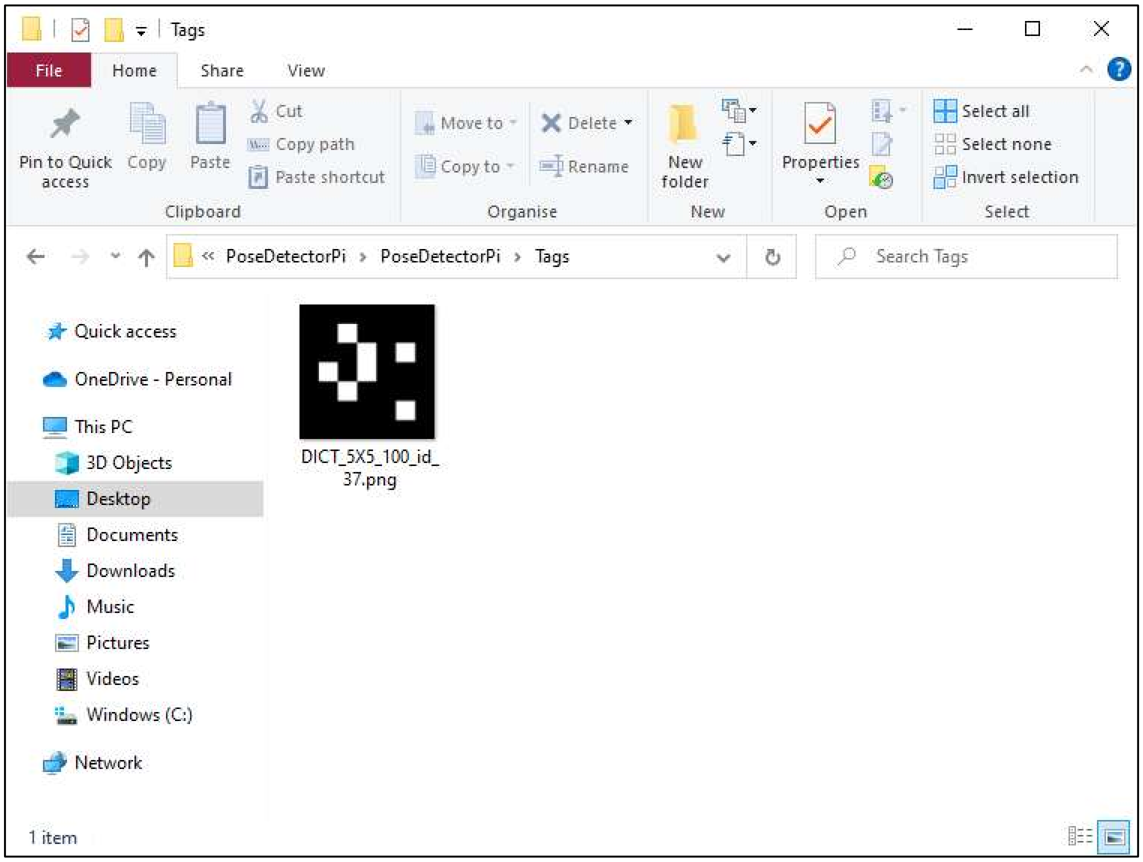
```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe
GENERATE TAGS

=================================================
ArUco Tag List
0: DICT_4X4_50  | 1: DICT_4X4_100  | 2: DICT_4X4_250  | 3: DICT_4X4_1000
4: DICT_5X5_50  | 5: DICT_5X5_100  | 6: DICT_5X5_250  | 7: DICT_5X5_1000
8: DICT_6X6_50  | 9: DICT_6X6_100  | 10: DICT_6X6_250 | 11: DICT_6X6_1000
12: DICT_7X7_50 | 13: DICT_7X7_100 | 14: DICT_7X7_250 | 15: DICT_7X7_1000
16: DICT_ARUCO_ORIGINAL | 17: DICT_APRILTAG_16h5 | 18: DICT_APRILTAG_25h9
19: DICT_APRILTAG_36h10 | 20: DICT_APRILTAG_36h11

Formating                | DICT_ARUCO_ORIGINAL = 6X6_1024
DICT_5X5_100             | DICT_APRILTAG_16h5  = 4X4_30
5x5 - pixel (internal)   | DICT_APRILTAG_25h9  = 5X5_35
100 - Amount of id       | DICT_APRILTAG_36h10 = 6X6_2320
                         | DICT_APRILTAG_25h9  = 6X6_587

Please enter the number for ArUCo tag: 5

=================================================
Create Directory for Tag (0:False 1:True)? 1
Directory ' Tags ' Created
C:\Users\maske\Desktop\Final Year Reasearch and Development Project\_Softwa
re\PoseDetectorPi\PoseDetectorPi\Tags

=================================================
Please enter the ID of ArUCo tag to generate: 37

=================================================
Please enter the size of ArUCo tag to generate (pixel): 500
```

3. A Display of the ArUco Tag will be displayed once complete.


C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe
ArUCo Tag

```
0: DICT_4X4_50  | 1: DICT_4X4_100  | 2: DICT_4X4_250  | 3: DICT_4X4_1000
4: DICT_5X5_50  | 5: DICT_5X5_100  | 6: DICT_5X5_250  | 7: DICT_5X5_1000
8: DICT_6X6_50  | 9: DICT_6X6_100  | 10: DICT_6X6_250 | 11: DICT_6X6_1000
12: DICT_7X7_50 | 13: DICT_7X7_100 | 14: DICT_7X7_250 | 15: DICT_7X7_1000
16: DICT_ARUCO_ORIGINAL | 17: DICT_APRILTAG_16h5 | 18: DICT_APRILTAG_25h9
19: DICT_APRILTAG_36h10 | 20: DICT_APRILTAG_36h11

Formating                | DICT_ARUCO_ORIGINAL = 6X6_1024
DICT_5X5_100             | DICT_APRILTAG_16h5  = 4X4_30
5x5 - pixel (internal)   | DICT_APRILTAG_25h9  = 5X5_35
100 - Amount of id       | DICT_APRILTAG_36h10 = 6X6_2320
                         | DICT_APRILTAG_25h9  = 6X6_587

Please enter the number for ArUCo tag: 5

================================================
Create Directory for Tag (0:False 1:True)? 1
Directory ' Tags ' Created
C:\Users\maske\Desktop\Final Year Reasearch and Development Project\_Softwa
re\PoseDetectorPi\PoseDetectorPi\Tags

================================================
Please enter the ID of ArUCo tag to generate: 37

================================================
Please enter the size of ArUCo tag to generate (pixel): 500

================================================
Generating ArUCo tag of type 'DICT_5X5_100' with ID '37'
```

4. This Tag is also saved within the folder created to be used for printing

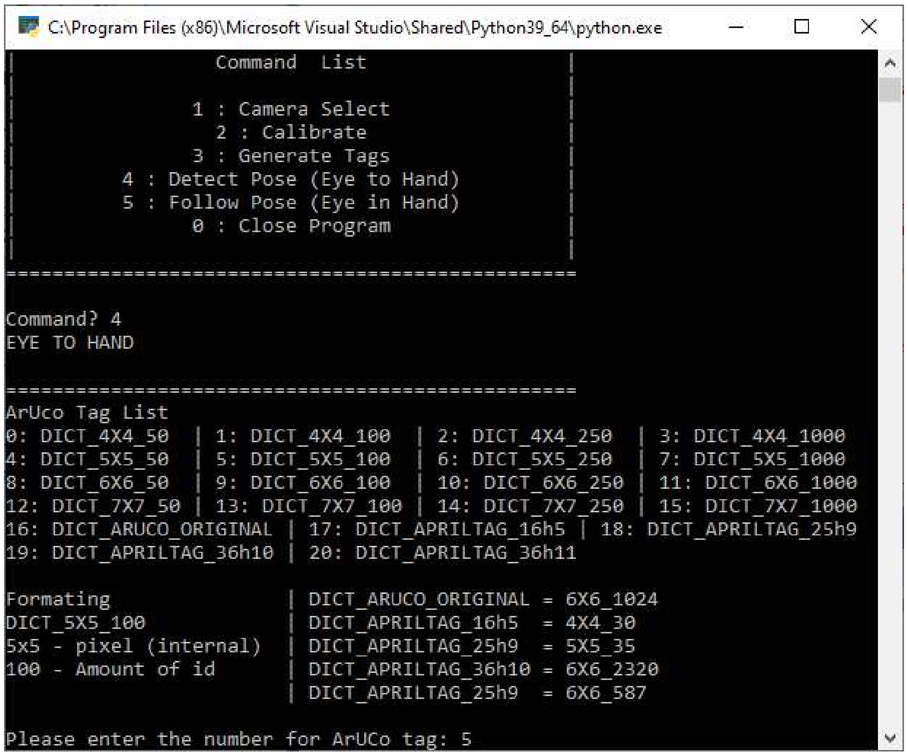


Detect and Follow Pose

Material Need:

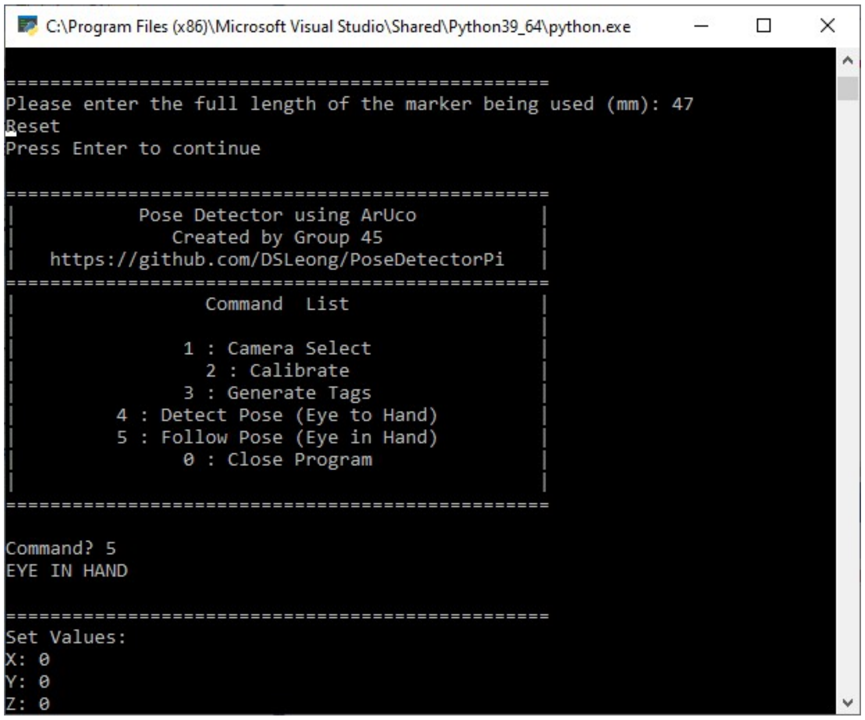
- Checkerboard print out (Attached at end)

1. Run the Camera Select and Calibration first
    a. For Detecting, input '4' and Enter, then continue



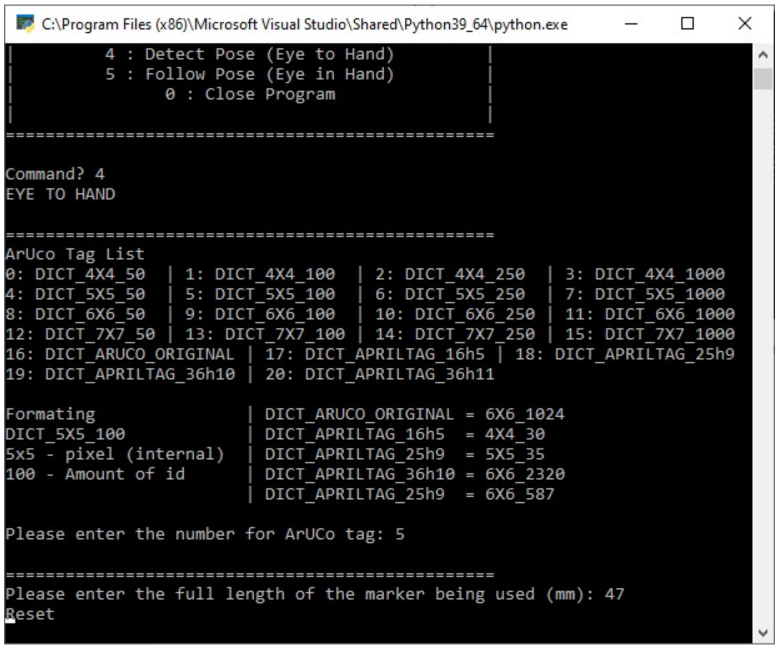
```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe
|                 Command  List                 |
|                                               |
|               1 : Camera Select               |
|                 2 : Calibrate                 |
|               3 : Generate Tags               |
|         4 : Detect Pose (Eye to Hand)         |
|         5 : Follow Pose (Eye in Hand)         |
|               0 : Close Program               |
|                                               |
=================================================

Command? 4
EYE TO HAND

=================================================
ArUco Tag List
0: DICT_4X4_50  | 1: DICT_4X4_100  | 2: DICT_4X4_250  | 3: DICT_4X4_1000
4: DICT_5X5_50  | 5: DICT_5X5_100  | 6: DICT_5X5_250  | 7: DICT_5X5_1000
8: DICT_6X6_50  | 9: DICT_6X6_100  | 10: DICT_6X6_250 | 11: DICT_6X6_1000
12: DICT_7X7_50 | 13: DICT_7X7_100 | 14: DICT_7X7_250 | 15: DICT_7X7_1000
16: DICT_ARUCO_ORIGINAL | 17: DICT_APRILTAG_16h5 | 18: DICT_APRILTAG_25h9
19: DICT_APRILTAG_36h10 | 20: DICT_APRILTAG_36h11

Formating               | DICT_ARUCO_ORIGINAL = 6X6_1024
DICT_5X5_100            | DICT_APRILTAG_16h5  = 4X4_30
5x5 - pixel (internal)  | DICT_APRILTAG_25h9  = 5X5_35
100 - Amount of id      | DICT_APRILTAG_36h10 = 6X6_2320
                        | DICT_APRILTAG_25h9  = 6X6_587

Please enter the number for ArUCo tag: 5
```

    b. For Following, input '5' and Enter. Set the Parameters



```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe

=================================================
Please enter the full length of the marker being used (mm): 47
Reset
Press Enter to continue

=================================================
|            Pose Detector using ArUco          |
|              Created by Group 45              |
|   https://github.com/DSLeong/PoseDetectorPi   |
=================================================
|                 Command  List                 |
|                                               |
|               1 : Camera Select               |
|                 2 : Calibrate                 |
|               3 : Generate Tags               |
|         4 : Detect Pose (Eye to Hand)         |
|         5 : Follow Pose (Eye in Hand)         |
|               0 : Close Program               |
|                                               |
=================================================

Command? 5
EYE IN HAND

=================================================
Set Values:
X: 0
Y: 0
Z: 0
```

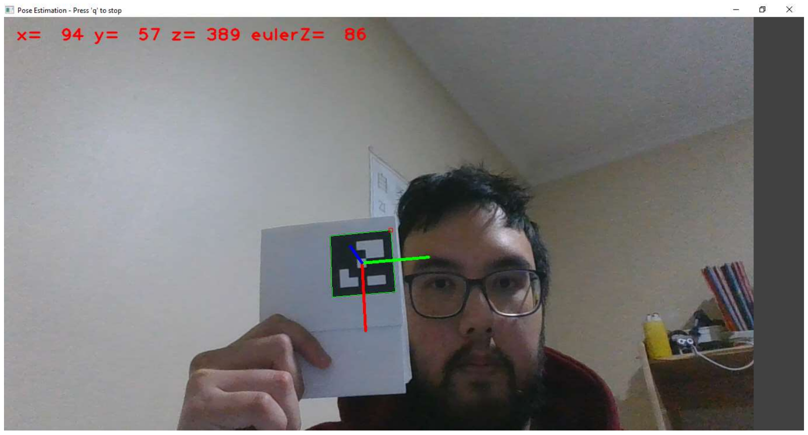
2. Set Parameters. In this case Tag as 5 – DICT_5X5_100 and size of 47mm

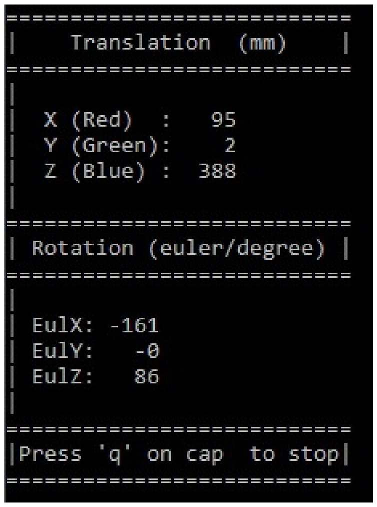


C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe

```
|         4 : Detect Pose (Eye to Hand)          |
|         5 : Follow Pose (Eye in Hand)          |
|               0 : Close Program                |
|                                                |
=================================================

Command? 4
EYE TO HAND

=================================================
ArUco Tag List
0: DICT_4X4_50  | 1: DICT_4X4_100  | 2: DICT_4X4_250  | 3: DICT_4X4_1000
4: DICT_5X5_50  | 5: DICT_5X5_100  | 6: DICT_5X5_250  | 7: DICT_5X5_1000
8: DICT_6X6_50  | 9: DICT_6X6_100  | 10: DICT_6X6_250 | 11: DICT_6X6_1000
12: DICT_7X7_50 | 13: DICT_7X7_100 | 14: DICT_7X7_250 | 15: DICT_7X7_1000
16: DICT_ARUCO_ORIGINAL | 17: DICT_APRILTAG_16h5 | 18: DICT_APRILTAG_25h9
19: DICT_APRILTAG_36h10 | 20: DICT_APRILTAG_36h11

Formating                | DICT_ARUCO_ORIGINAL = 6X6_1024
DICT_5X5_100             | DICT_APRILTAG_16h5  = 4X4_30
5x5 - pixel (internal)   | DICT_APRILTAG_25h9  = 5X5_35
100 - Amount of id       | DICT_APRILTAG_36h10 = 6X6_2320
                         | DICT_APRILTAG_25h9  = 6X6_587

Please enter the number for ArUCo tag: 5

=================================================
Please enter the full length of the marker being used (mm): 47
Reset
```

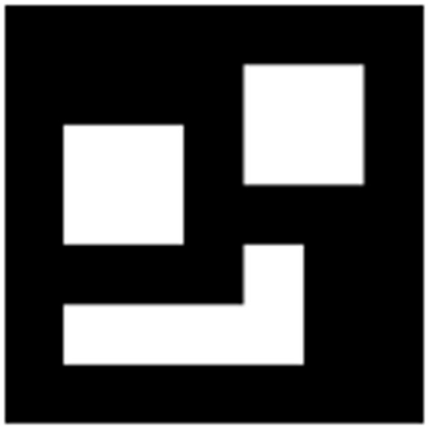
3. A Preview of your video appears. Once a marker is detected within your feed, the full pose values are displayed within the terminal and partially within video.

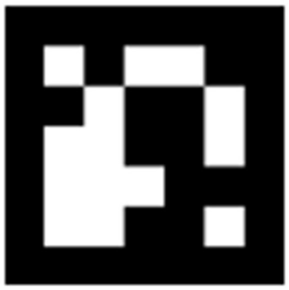

Pose Estimation - Press 'q' to stop
x= 94 y= 57 z= 389 eulerZ= 86

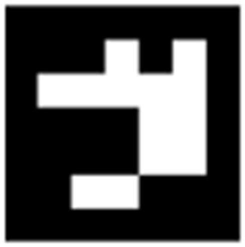


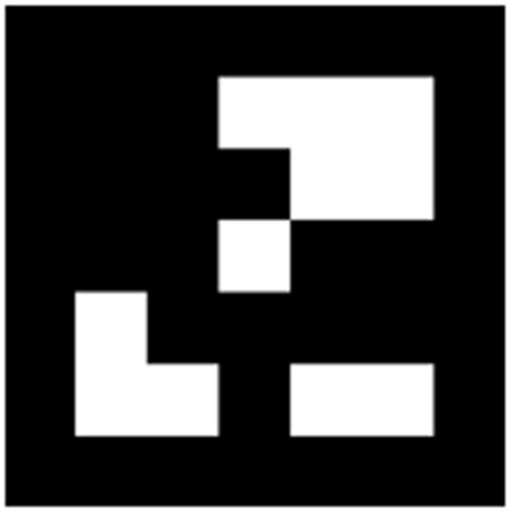
```
===========================
|    Translation  (mm)    |
===========================
|
|  X (Red)  :   95
|  Y (Green):    2
|  Z (Blue) :  388
|
===========================
| Rotation (euler/degree) |
===========================
|
| EulX: -161
| EulY:   -0
| EulZ:   86
|
===========================
|Press 'q' on cap  to stop|
===========================
```

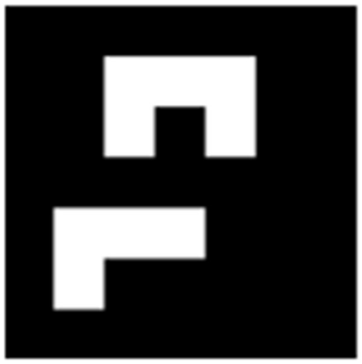
Marker (DICT_5X5_100)

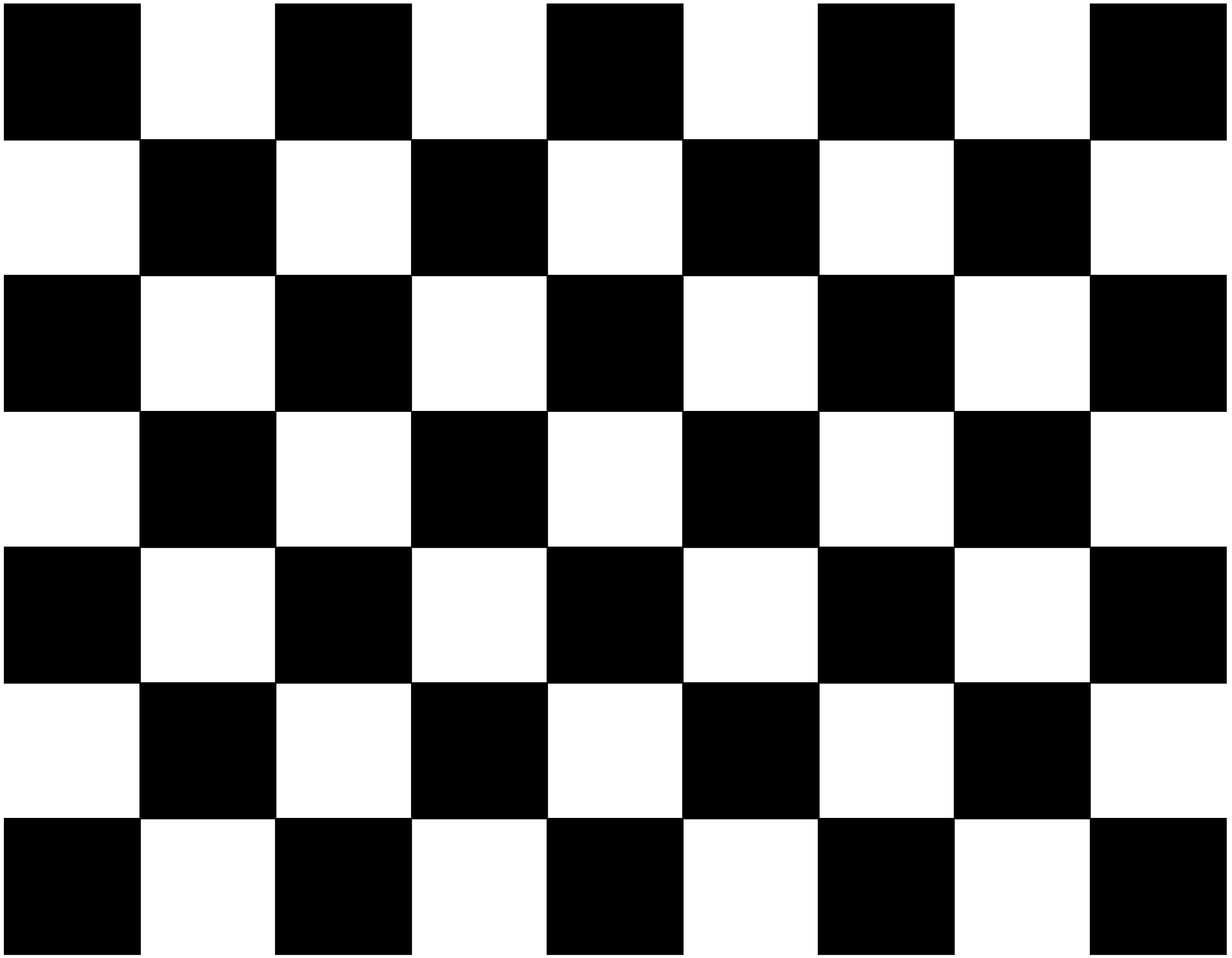












SampleCamCalib