Control Thymio Robot thru Hand Gestures

YOLO! Detect my hand gesture to control my Thymio robot.

By Ahmad Agbaryah

Abstract

With the help of Deep Learning, we all know that the field of Computer Vision has proliferated in the last decade. As a result, so many prevalent computers vision problems like image classification, object detection, and segmentation having real industrial use-case started to achieve accuracy like never. A new benchmark was set every year from 2012. And today, we will look at object detection from a practical perspective.

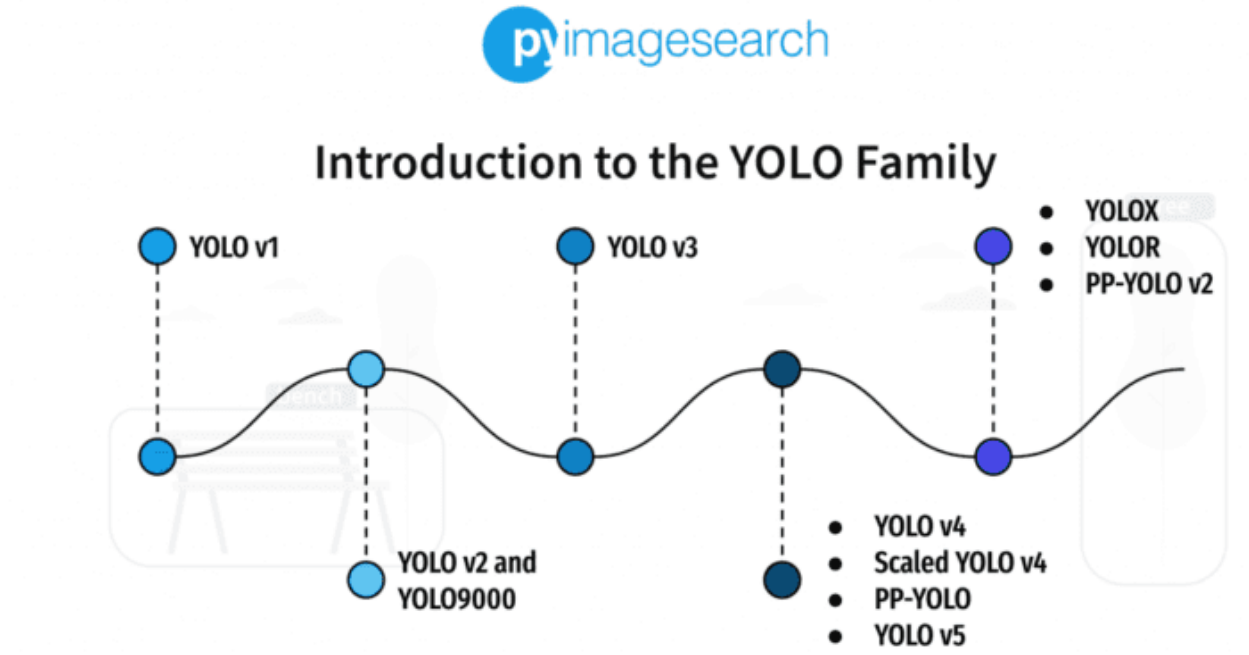
This proof of concept is combining a wireless Thymio II with a python software on computer to control the movements of the robot via hand gestures. Python software based on YOLOV5 is running on the computer to detect hand gestures via a camera. Once the python application recognizes new hand gesture, the python script decides what command to send to the Thymio robot. For example, hand gesture of one means "move forward", hand gesture of "five" means "stop", hand gesture of "fist" means "speed up", and so on.

Introduction

Object detection is one of the most crucial subjects in computer vision. Most computer vision problems involve detecting visual object categories like pedestrians, cars, buses – mainly for autonomous cars, faces, hand gestures – mainly for augmented reality and virtual reality use cases, etc. It is one such field that is widely used in both academia and real-world business use case in domains like video surveillance, healthcare, in-vehicle sensing, autonomous driving. And in this paper, we will demonstrate integration of hand gestures to control educational robot called Thymio. Let's see the main two components that composing this PoC.

YOLO V5

Many use cases, require good accuracy and real-time inference speed. Hence, choosing an Object Detector that fits the bill for both speed and accuracy becomes essential. YOLO (You Only Look Once) is a single-stage object detector introduced to achieve both goals (i.e., speed and accuracy). And today, we will give use case of integrating YOLOV5 with custom data to control educational robot called Thymio.



Yolov5 History

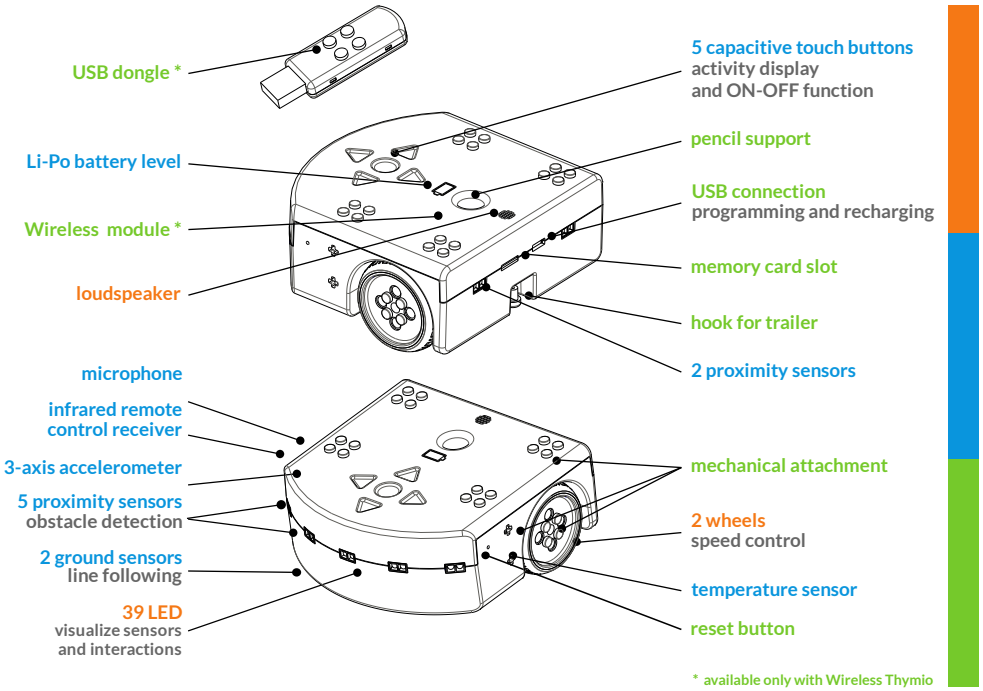
Before we start, a little story of the YOLOv5 birth controversy. YOLO was first created by Joseph Redmon in May 2016. That was a revolution in real-time object detection. Please notice that it was Joseph Redmon that came with this so-good name. The code has been updated with

associations with Ali Farhadi for the 3 first versions. From there, the pioneers give Alexey Bochkovskiy their benediction via Twitter to take the lead of YOLO institution. Joseph Redmon's tweet states that it isn't his anymore. Alexey Bochkovskiy introduced YOLOv4 in April 2020.

What Thymio is?

Thymio is an open-source educational robot designed by researchers from the EPFL, in collaboration with ECAL, and produced by Mobsya, a nonprofit association whose mission is to offer comprehensive, engaging STEAM journeys to learners of all ages. For more information, visit the official web site at: https://www.thymio.org/

What is Thymio composed of?[1]



Python

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today. A survey conducted by industry analyst firm RedMonk

[1] Image from: http://wiki.thymio.org/en:thymiospecifications

found that it was the second-most popular programming language among developers in 2021. (Coursera, 2022).

Python, one of the most popular programming languages in the world, has created everything from Netflix's recommendation algorithm to the software that controls self-driving cars. Python is a general-purpose language, which means it's designed to be used in a range of applications, including data science, software and web development, automation, and generally getting stuff done.

Demo

You may view our demo in YouTube Control Thymio by Hand Gesture (link: https://youtube.com/shorts/no3g03PY0nw)

Prepare Development environment

Thanks to Davies for his great online blog on Weights & Biases (Davies, 2022).

We will detail the steps to prepare the development environment to allow training and detecting hand gestures.

4.1 Python

4.1.1 Download Python

Visit https://www.python.org/downloads/ and pick 3.9 or 3.10 version for windows.

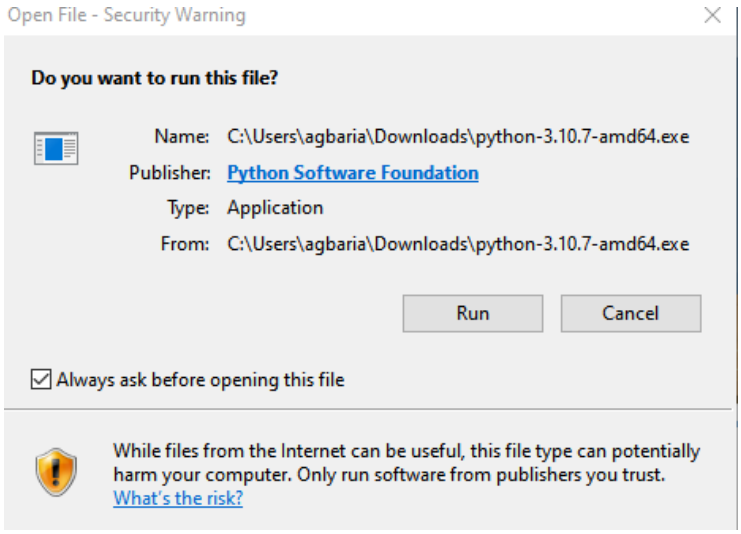
You may click the following direct link to download Python 3.10.7 version:

https://www.python.org/ftp/python/3.10.7/python-3.10.7-amd64.exe

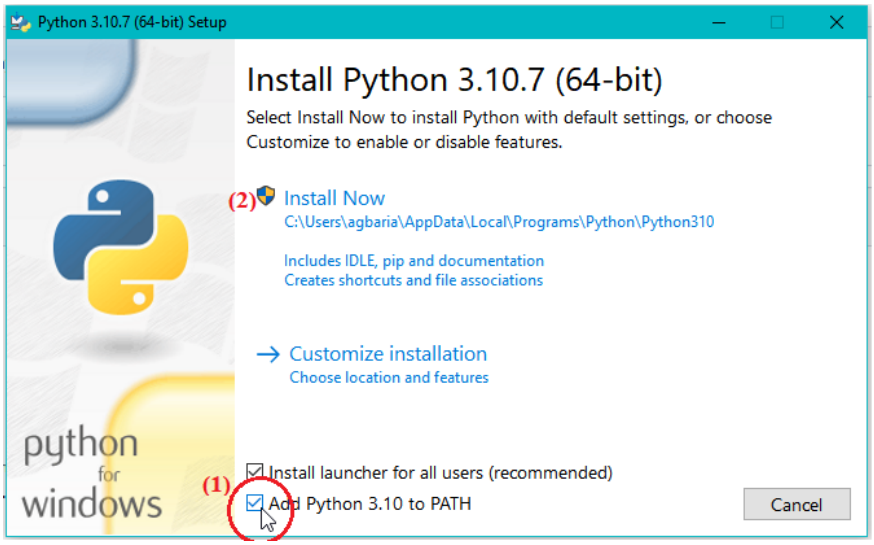
4.1.2 Install Python

Once the download process is done (from previous step), double click the downloaded file, and follow the steps below

1) Click the **Run** button



2) Make sure to click "Add Python … to PATH" option (see red circle in the screen shot) then click "**Install Now**"



Follow rest steps until install is done.

4.2 Thymio Suite

4.2.1 Download Thymio Suit

Click the link https://www.thymio.org/download-thymio-suite/ (If the current version does not support Python, then here is a direct link to beta version with Python support: https://github.com/Mobsya/aseba/releases/download/nightly/ThymioSuite-2.4.0-win64.exe)

4.2.2 Thymio Python documentation

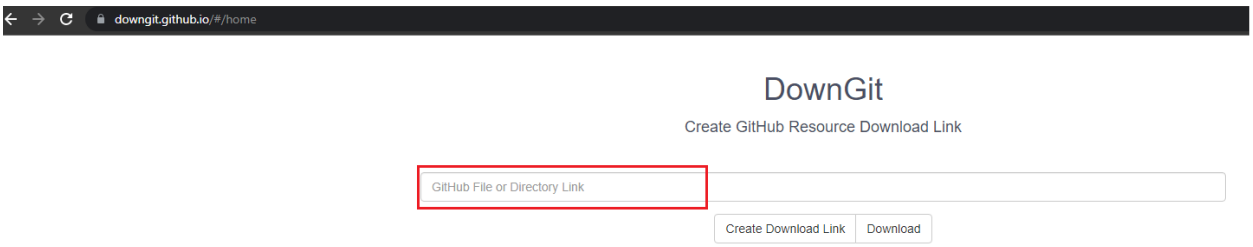
I found the following link helpful for my PoC: https://pypi.org/project/tdmclient/

4.3 Control Thymio via Hand Gesture (Quick Setup)

4.3.1 Setup – One Time

One way to prepare the environment quickly is to download the scrip file “prepare_env.bat” from github and then double click on it. How to do that?

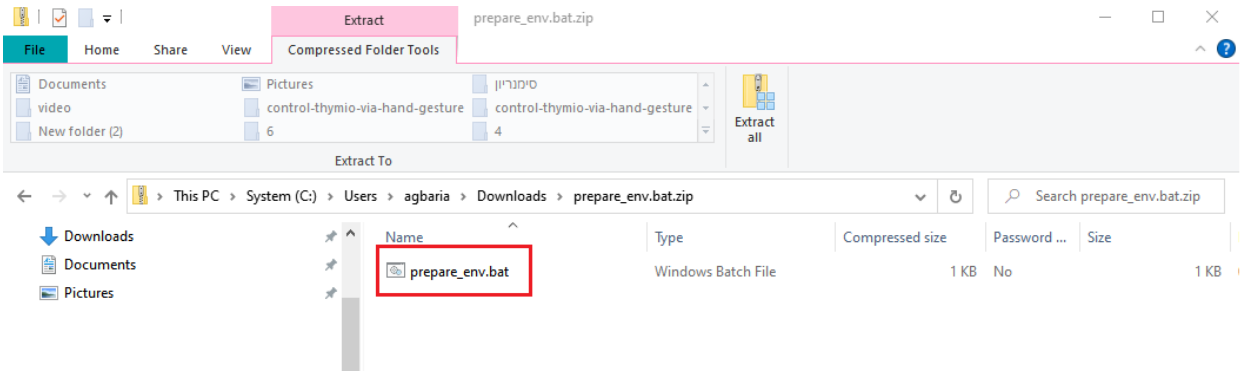
1) Visit DownGit in your browser, here is the webpage link: https://downgit.github.io/#/home



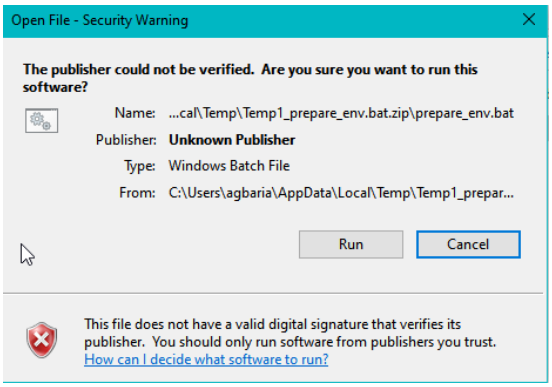
2) In the red area, type the script full path: https://github.com/ahmad081177/control-thymio-via-hand-gesture/blob/main/prepare_env.bat

3) Then click “Download” button. Wait for the download to finish.

4) Open the downloaded “prepare_env.bat.zip” (archived) file, and double click the “prepare_env.bat” file inside it. (See screen shot below)



5) If you received some prompt window, click the “Run” button



6) The script is going to download all needed packages to run the software…this may take some time.

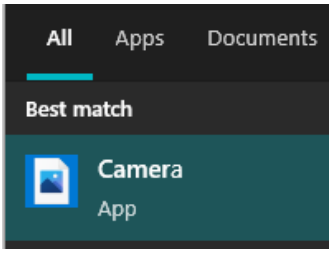
4.3.2 Run the PoC

To run the PoC, you need to make sure that:

1) Your camera is ready.

To check that, from the windows start menu (

) click the icon and type “camera”

click the camera app: 

and make sure the camera is working.

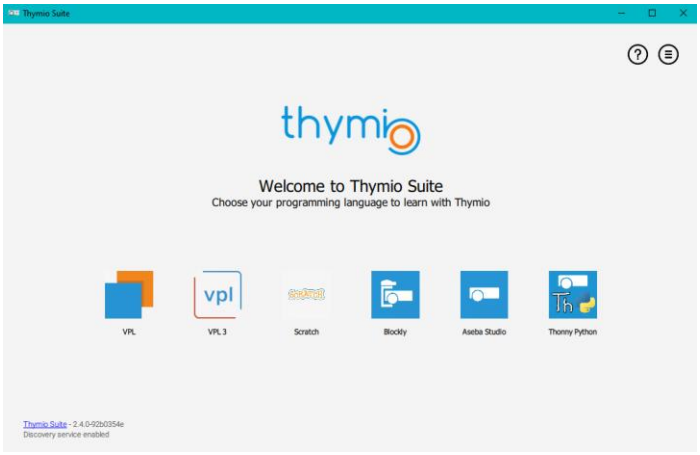
Close the camera app – since the PoC need the camera to be free, no app is locking it.

2) Thymio robot is turned on

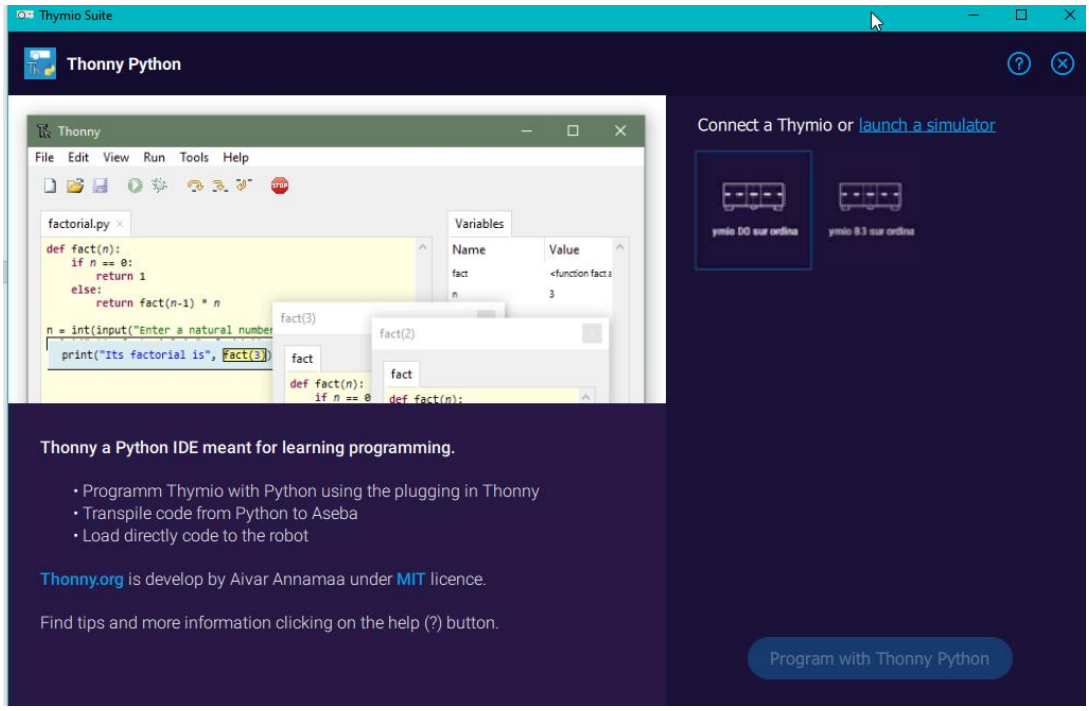
3) The wireless dongle is plugged into the computer

4) The computer recognizes your Thymio

4.1) double check that by opening the Thymio Suite (from your desktop)



4.2) Click the “Thonny Python” icon to open the Thonny Python listing the connected Thymio robots with your computer – see screen shot below



4.3.3 Run the PoC

To run the PoC, locate the script "run_detect.bat" under c:\workdir\thymio\yolov5 folder and double click on it.

Start moving your hand and enjoy controlling your robot.

4.4 In Details

This section is aimed to explain the detailed steps to recreate the PoC in your computer. We assume the reader has background in windows, command shell/prompt, window bat and some basic understanding of github. We also assume that the reader already has Python installed.

A common practice to work with Python is to create a virtual environment. A virtual environment is a Python environment such that the Python interpreter, libraries and scripts installed into it are isolated from those installed in other virtual environments, and (by default) any libraries installed in a "system" Python, i.e., one which is installed as part of your operating system. (Python documentation official site)

Remark: the following script assumes that we are going to create the environment software under c:\workdir\thymio folder. We will refer to it as WORKSPACE.

4.4.1 Create your Python Environment

Open a command prompt and run the following:

1) set WORKSPACE=c:\workdir\thymio

This command will create a temporary environment variable called *WORKSPACE* and its value is *c:\workdir\thymio*. This is useful for later references and shortcuts.

2) mkdir %WORKSPACE%

This command makes sure the *WORKSPACE* folders are created. i.e., it creates a *workdir* folder under *C:* drive, then it creates *thymio* subfolder under c*:\workdir* folder (just in case it is not created yet)

3) cd %WORKSPACE%

The command will change the current directory of your command prompt to the *WORKSPACE* folder

4) python -m venv %WORKSPACE%

This command tells Python to create new virtual environment under *WORKSPACE* folder. The virtual environment will be called as the last folder name, i.e., *thymio*.

5) %WORKSPACE%\Scripts\Activate.bat

This command will set the newly created virtual environment to be the active one. This leads any new package/library to be installed will be located under this environment.

6) git clone https://github.com/ultralytics/yolov5 yolov5

This command will ask git tools to clone the repository *https://github.com/ultralytics/yolov5* to a subfolder called: *yolov5* under the current working directory.

Git clone is a utility that downloads the entire repository (files and folders) from git into the desired target location. If you don't have it, don't worry, you may download any data from Git using DownGit website (https://downgit.github.io/#/home) and passing the repository/folder/file link to the site and it will download you a zipped file with all folders/files.

7) cd yolov5

Change the current directory of your command prompt to be *yolov5* subfolder.

8) pip install -r requirements.txt

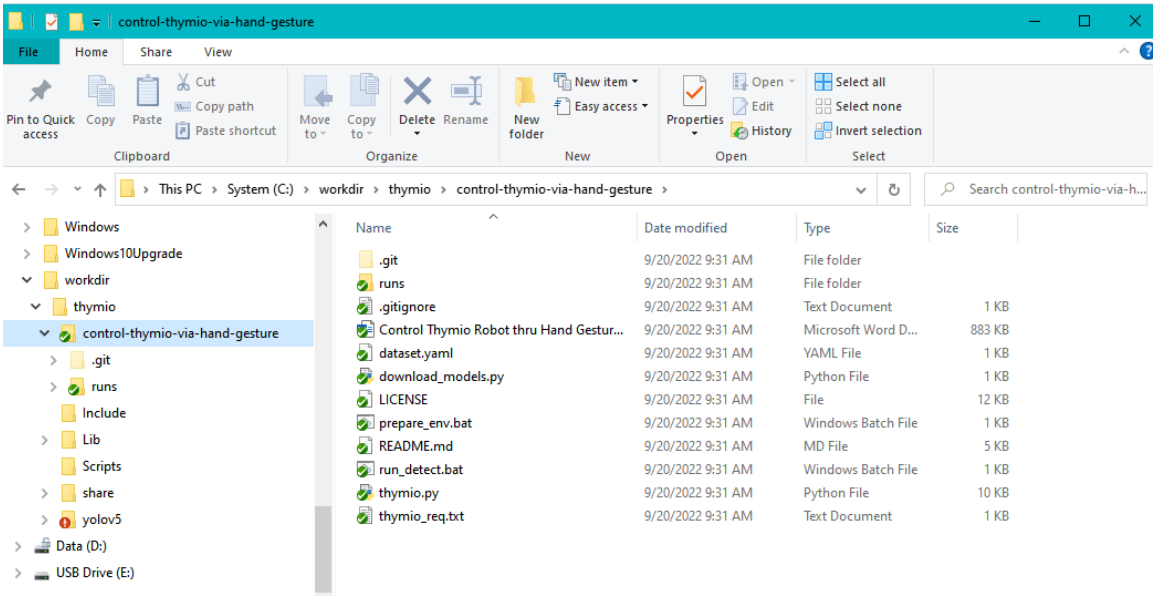
This command will ask Python's pip tool to install all the packages listed in the *requirements.txt* file to current active virtual environment. For instance, this will install PyTorch, numpy and other required Python libraries.

Python pip is a Package Installer for Python is the de facto and recommended package-management system written in Python and is used to install and manage software packages. It connects to an online repository of public packages, called the Python Package Index.

9) git clone https://github.com/ahmad081177/control-thymio-via-hand-gesture %WORKSPACE%\control-thymio-via-hand-gesture

This command will clone (using git) the *control-thymio-via-hand-gesture* repository to the *control-thymio-via-hand-gesture* subfolder of the *WORKSPACE* folder.

After this command, you should have the following folder's structure:



10) xcopy /E /Y /C %WORKSPACE%\control-thymio-via-hand-gesture* %WORKSPACE%\yolov5\.

This command copies the content of *control-thymio-via-hand-gesture* folder to *yolov5* folder.

Xcopy is a built-in windows utility that extend the default copy functionality. The parameters that are passed to the command line are:

/E - Copies all subdirectories, even if they are empty.

/Y - Suppresses prompting to confirm that you want to overwrite an existing destination file.

/C – Ignore errors

Then the two arguments are the source folder followed by the destination folder.

Remark: We pick to suppress error and prompting to make the setup working without any user interruption and to allow integration the script in background processes.

Note: the *control-thymio-via-hand-gesture* folder includes several files – will detail them later.

11) `cd %WORKSPACE%\yolov5`

Change current directory to the yolov5 – to run the main scripts later.

12) `pip install -r thymio_req.txt`

Ask python to install addition dependency libraries listed under *thymio_req.txt* file to allow integration between python and the Thymio robot.

4.4.2 Run the PoC

Again, make sure the Thymio robot is turned on and the wireless dongle is connected to your computer.

1) Open a command prompt

2) `cd c:\workdir\thymio\yolov5`

Change the current directory to the yolov5 subfolder under WORKAPCE folder.

3) `run_detect.bat`

Runs the PoC. This bat file will call the following python command:

python detect.py --weights runs/train/thymio_exp/weights/best.pt --img 416 --source 0 --nosave --max-det 1

it runs the *detect.py* file (located in the current directory) and passing to it several parameters:

--*weights runs/train/thymio_exp/weights/best.pt*: pass the pretrained model with all the weights to the detect script. Those weights were generated by us to extend YOLOV5 to recognize hand gestures

--*img 416*: tells YOLO's detector process to look for hand images with 416 pixels size (this is the same size of the customized hand gestures that we trained the model on)

--*source 0*: indicating the camera as the source of images for the detector.

--*nosave*: tells YOLO detector process not to save any images during the detect process. Saving during detecting is useful for debugging, not in production.

--*max-det 1*: tells the detector to detect only one hand in an image. i.e., if the user raises two hands, the detector will detect only one and ignore the other (it will recognize the one with high confidence)

4.4.3 control-thymio-via-hand-gesture repository

The repository *control-thymio-via-hand-gesture* (https://github.com/ahmad081177/control-thymio-via-hand-gesture) includes extension to YOLO v5 to allow controlling the Thymio II robot. The repository includes the following files:

1) *Control Thymio Robot thru Hand Gestures.docx*: documentation file about the PoC and how to run the PoC on your own environment.

2) *README.md*: Git README file, includes summary of the *Control Thymio Robot thru Hand Gestures.docx* file.

3) *LICENSE*: git license file

4) *dataset.yaml*: used in the training process, telling YOLO where the images of the train and test are located in addition to the class names that are used in the custom data.

5) *download_models.py*: Downloads some of the YOLO v5 pretrained models to be used in the training process.

6) *run_detect.bat*: bat script that runs the detect.py with predefined parameters (already explained in the previous section 4.4.2)

7) *detect.py:* YOLO v5 detector with modification to enable Thymio integration with YOLO v5 detector processes. The main changes are marked between the tags:

#Ahmad081177 – start

…

#Ahmad081177 – end

8) *thymio_req.txt*: includes the python library needed for Thymio integration

9) *thymio.py*: Python wrapper for calling Thymio robot using the tdmclient library

References:

https://towardsdatascience.com/yolov5-end-to-end-object-detector-project-on-custom-dataset-5d9cc2c95921

https://wandb.ai/onlineinference/YOLO/reports/YOLOv5-Object-Detection-on-Windows-Step-By-Step-Tutorial---VmlldzoxMDQwNzk4

https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python

Data:

The “Control Thymio Robot by Hand Gesture” demo was trained on hand gestures data by two phases. First, we used the “hand-gesture-recongition-yolo-v3” (link: https://www.kaggle.com/datasets/abdullahmujahidali/hand-gesture-recongition-yolo-v3?select=102.txt) from Kaggle website (https://www.kaggle.com/). Later, to improve the accuracy from one hand, and to add more classes (hand gesture types) from the other hand, we added more data to the dataset by creating our own pictures and labeling them using Label Image tool from python (https://github.com/heartexlabs/labelImg). The new dataset was uploaded to the public Kaggle dataset at the following link: https://www.kaggle.com/datasets/ahmadgabari/hand-gestures-new