**MDP Image Recognition**

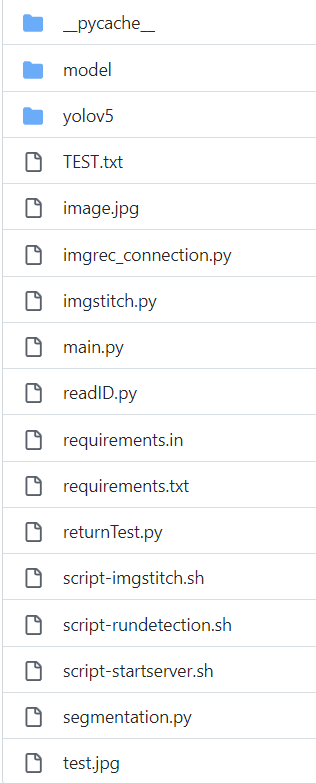
(Based on what I can remember)

Link to main reference:

<https://betterprogramming.pub/machine-learning-model-api-using-yolov5-with-fast-api-192f1290a982>

**File Explanation**

List of Files:



\_\_pycache\_\_ folder

* Never touch

model folder

* Put the trained model(best.pt) in here

yolov5 folder

* Never really touch also I think

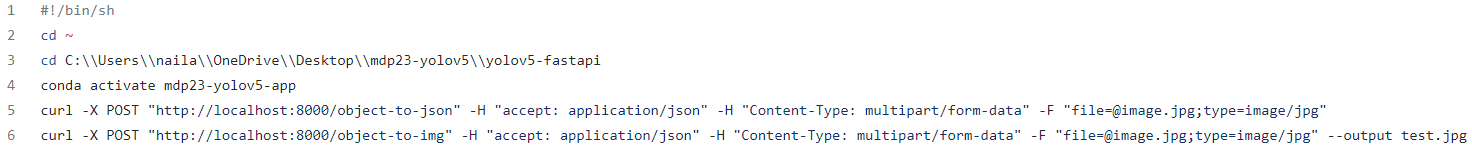
TEST.txt

* Contains the ID of the image detected

image.jpg

* Originally a Blank image – cause when my model starts up it runs the detection immediately, and to make sure it doesn’t detect its surroundings, image.jpg MUST be a blank image at the START
* Once after the first run, it will be replaced by the actual captured image by the rpi camera, so rmb to change it back to the blank image at the start of each run

script-rundetection.py



* Execute the curl command to do the detection and identify the symbol present in the image

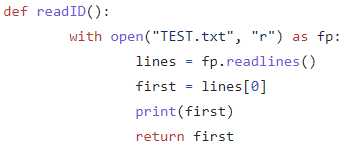
returnTest.py

Graphical user interface, text, application, email

Description automatically generated

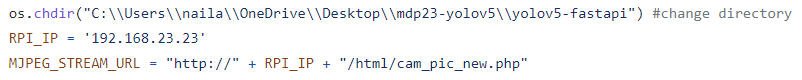
* Basically runs the script-rundetection.sh

readID.py



* Reads the number populated in the TEST.txt file, which indicates the ID of the symbol

imgrec\_connection.py



* Change to which ever directory you place this folder and your team’s rpi code

Graphical user interface, text, application

Description automatically generated

* Very dependent on how your rpi person connection implementation (sorry I don’t have it)
* But the general idea is when the robot signals that it has reached a block, it will send a message to the rpi, which will call imgrec to run its process. Then it will replace the image.jpg with the captured photo and run the detection to obtain the symbol ID. The image with the dectections will be saved as test.jpg

Graphical user interface, text, application, email

Description automatically generated

* Moves the image with detection to a folder to be stitched (sorry I forgot how only took the image with detection)

imgstitch.py

* Online code I found to do the stitching of images

main.py

* Don’t really rmb much about this file, think its connection from the model to the fast api

requirements.in/requirements.txt

* Required libraries to install in your environment

script-imgstitch.py

* Script to call python to do image stitching

scripy-startserver.py

* Script to run the server (its commented out in returnTest.py, think need to uncomment)

segmentation.py

* Don’t think I touch this code haha

test.jpg

* The image with detections when the rpi camera calls for img rec