

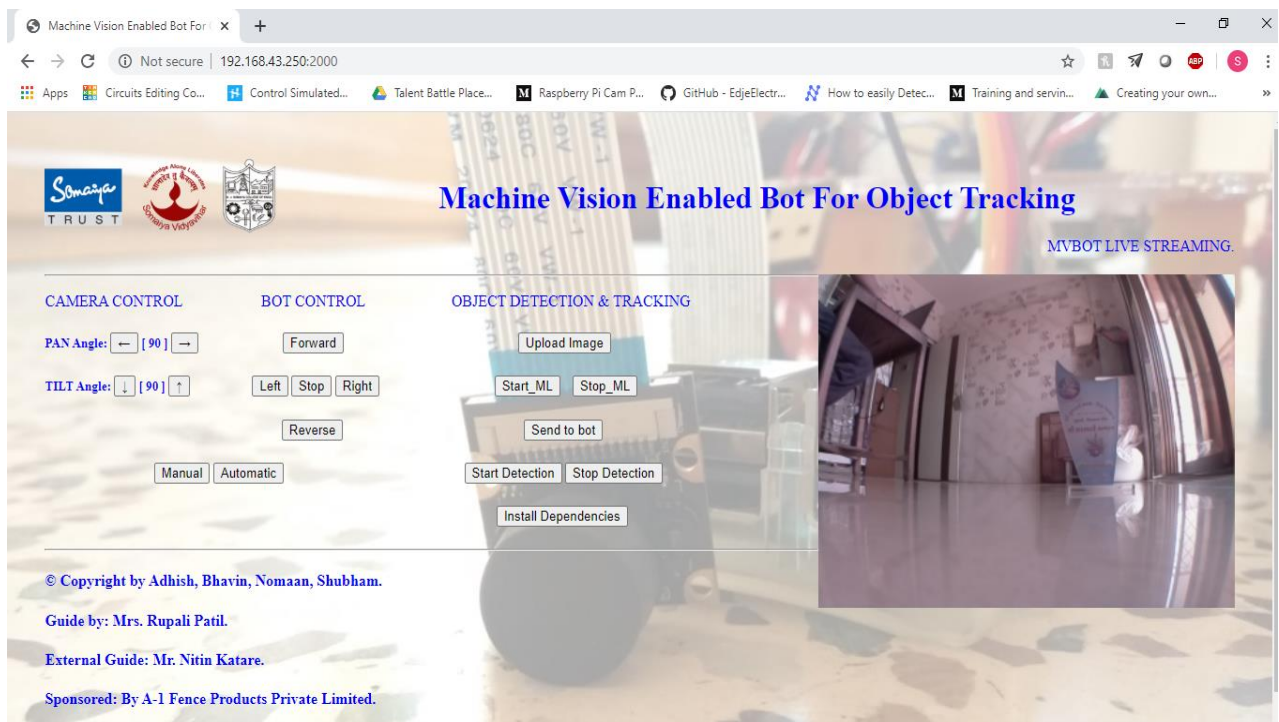
*****Machine Vision Bot for Object Detection and Tracking*****

Pre-requisites:

- Python3 should be installed in windows computer.
- -"KJSCE_Object_Detection" must be in Desktop with "Desktop" named as "Desktop" and not any other name as it will give path error.
- -In KJSCE_Object_Detection folder open api.py file and edit in line "app.run(host='192.168.43.97', port=5001, debug=True, threaded=True)" replace "host='192.168.43.97'" with 'your pc's IP address'
- In KJSCE_Object_Detection folder open api.py file and edit in line "app.run(host='192.168.43.97', port=5001, debug=True, threaded=True)" replace (host='192.168.43.97' with 'your pc's IP address').
- In Raspberry pi, open appCamPanTilt.py in "/home/pi/tensorflow1/models-master/research/object_detection" folder and edit ('192.168.43.97' with 'your pc's IP address') and ('192.168.43.250' with 'your pi's IP address') wherever it is.

Need to Run 2 files:

- 1)"python api.py" (in directory "KJSCE_Object_Detection") in command line of windows.
- 2)"sudo python3 appCamPanTilt.py" (in directory /home/pi/ tensorflow1/models-master/research/object_detection) in command line of Raspberry pi.
- 3)After running both the files, go to your pc's chrome browser and type 'your pi's IP address with port no'
- 4)A Webpage will be displayed (as shown below).



After the Webpage page is loaded:

1) Click on "Install Dependencies" button, if you are using it for the first time or else skip this step.

Use - It will install all the required libraries need for learning process and its steps.

Note – After doing this steps need to restart the server again (need to rerun api.py file).

2) Need to set path (as shown in image below):

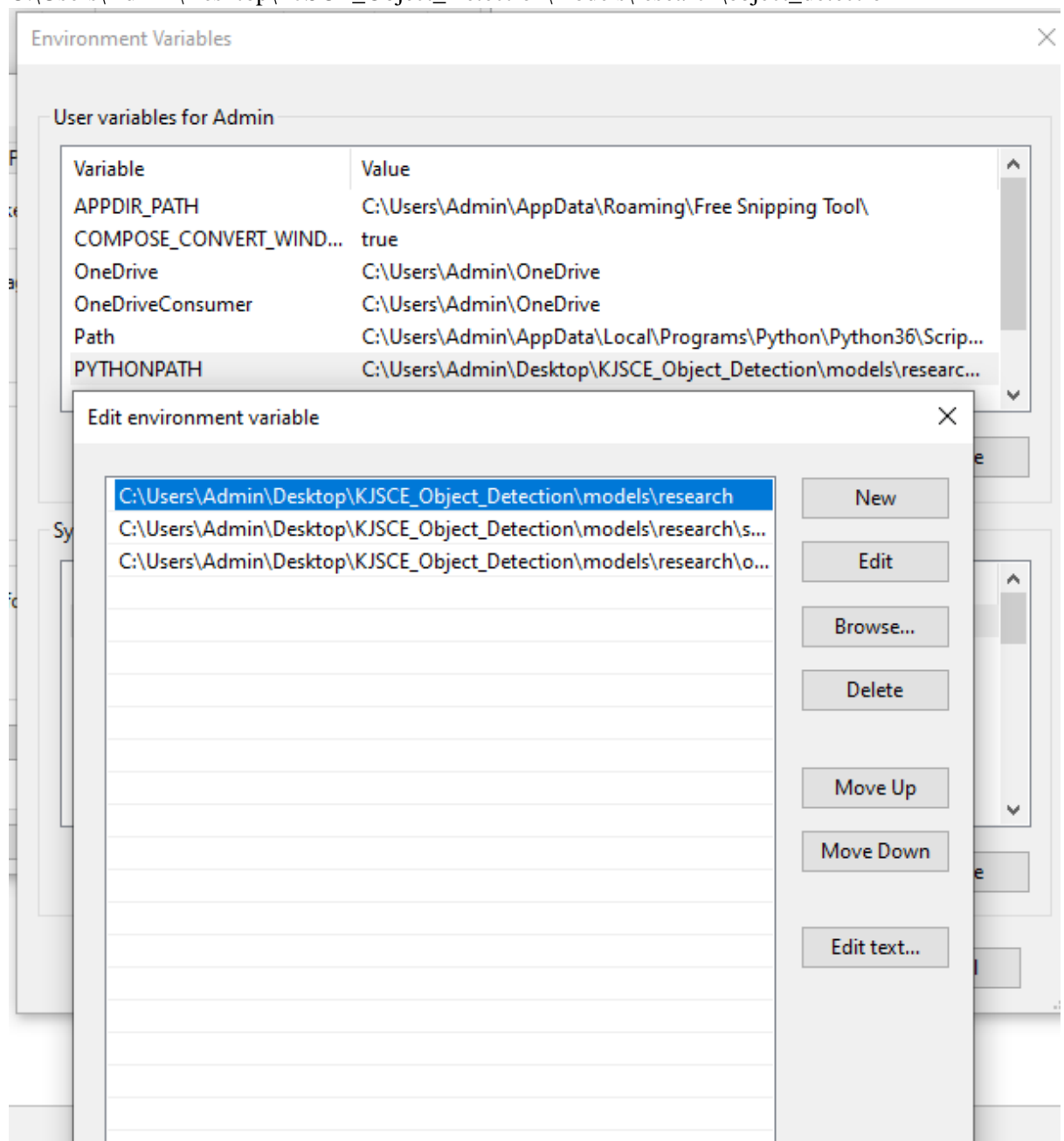
- Go to Environment variable.
- Click on New

Put variable name = PYTHONPATH

Put variable value = C:\Users\Admin\Desktop\KJSCE_Object_Detection\models\research\slim

C:\Users\Admin\Desktop\KJSCE_Object_Detection\models\research

C:\Users\Admin\Desktop\KJSCE_Object_Detection\models\research\object_detection



Guidelines for running the project

3) Open Command prompt and run the following commands separately:

```
cd C:\Users\Admin\Desktop\KJSCE_Object_Detection\models\research
```

```
python setup.py build
```

```
python setup.py install
```

Note: If you did not understand any of the above 3 steps or having difficulty in these steps, kindly refer to the link given below in references.

CONGRATULATIONS!! You are done with your installations

How to Use the webpage for the object detection and tracking:

There are 2 options given 1) Manual 2) Automatic

1) If you click on Manual button,

Buttons : “upload image” “Start ML” “Stop ML” “send_to_bot” “Start detection” “Stop Detection” “install dependencies” would be disabled.

2) If you click on Automatic button,

Buttons that are not mention above would be disabled.

For Automatic Detection:

Pre-requistics: Need to have images of the object you want to detection. Keep all those images in the “C:\Users\Admin\Desktop\KJSCE_Object_Detection\Img Files” folder.

Note: To check the status of button working or not and what is going, need to look into the cmd where api.py is running.

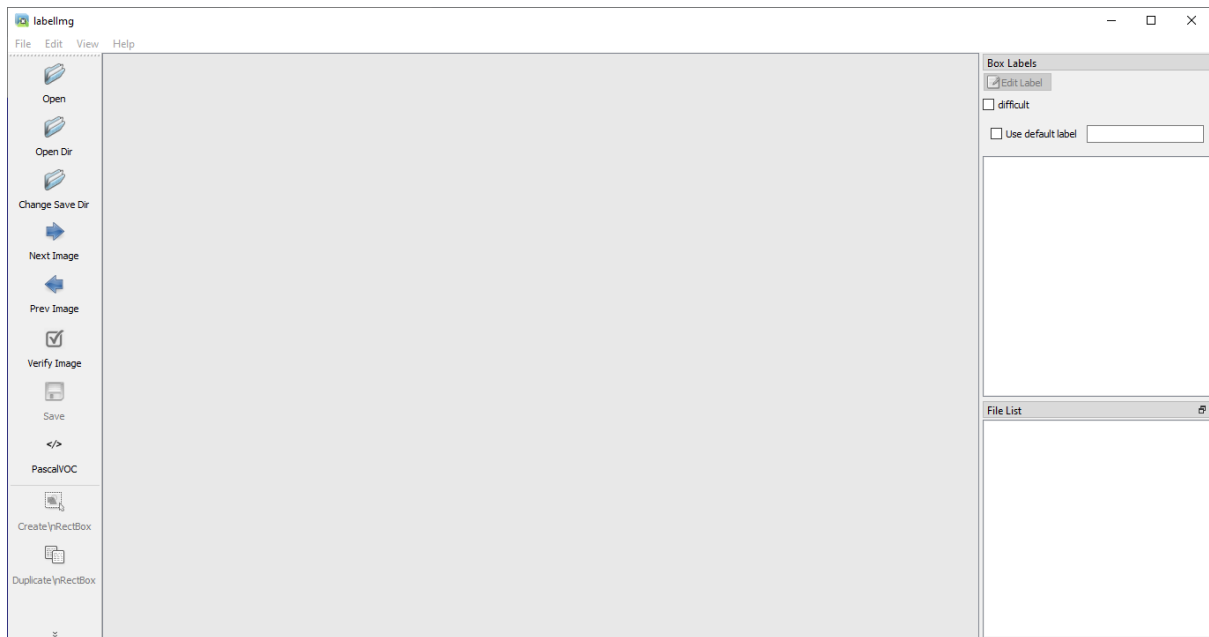
➤ On Clicking 1st Button “upload image”-

After 2 seconds a software LabelImg will open. Go to “open dir” and select “Img folder” which contains the image file.

Select “change save dir” as “XML Files” in KJSCE_Object_Detection folder.

Select each image and draw bounding box across the object in the image you want to detect. (Note: Keep Format as PascelVOC).

Guidelines for running the project



➤ On Clicking 2nd Button “Start_ML”

Two files will run after one and other which are “start_learning_process.py” and “test_with_code_v1.bat”

start_learning process will arrange all the images and xml files, delete previously run and saved files, make multiple copies for more accurate training, separating img and xml files into test and train folder and other basic such stuffs.

Test_with_code_v1.bat will actually start the learning process by generating .csv files, train.record, test.record, etc.

This training process will run indefinitely until Stop_ML button is not press.

➤ On clicking 3rd Button “Stop_ML”

It will run a python file “stop_learning_process.py”. It will stop the learning process and will restart the server again. (Server needs to restart again because to stop learning process, we need to press “ctrl+c” due to which server also gets terminated while stopping the learning process.

➤ On Clicking 4th Button “Send_to_bot”

It will run 3 python files “generate_record.py” “delete_from_dropbox.py” “upload_to_dropbox.py”

1st file will generate “frozen_inference_graph.pb” which needs to be send to Rpi on bot with which it will start it object detection procedure.(**Note:** If it is struck then click “enter” 2/3 times)

2nd file will check wheather any previous instance of “frozen_inference_graph.pb” is present on dropbox or not , if present it will delete it.(**Note:** If file is not present in dropbox, it will throw exception error, but ignore it.)

Guidelines for running the project

3rd file will upload the generate “frozen_inference_graph.pb” to dropbox.

➤ On Clicking 5th Button “Start_detection”

It will run 2 python files on raspberry pi on the bot.

1st file will delete the local “frozen_inference_graph.pb” present and will download the latest “frozen_inference_graph.pb” on the rpi.

2nd file will start the object detection and on detection or recognition of the object, it would start tracking it.

Note: It will Take approximately 1-2min to run.

➤ On Clicking 6th Button “Stop_Detection”

It will run “stop_ml.py” file which will stop object detection on raspberrypi(bot) and will restart the server again.

For Manual Detection:

Pan angle: Left and Right Moment of Camera.

Tilt angle: Up and Down Moment of Camera.

References: <https://gilberttanner.com/blog/installing-the-tensorflow-object-detection-api>

<https://gilberttanner.com/blog/creating-your-own-objectdetector>