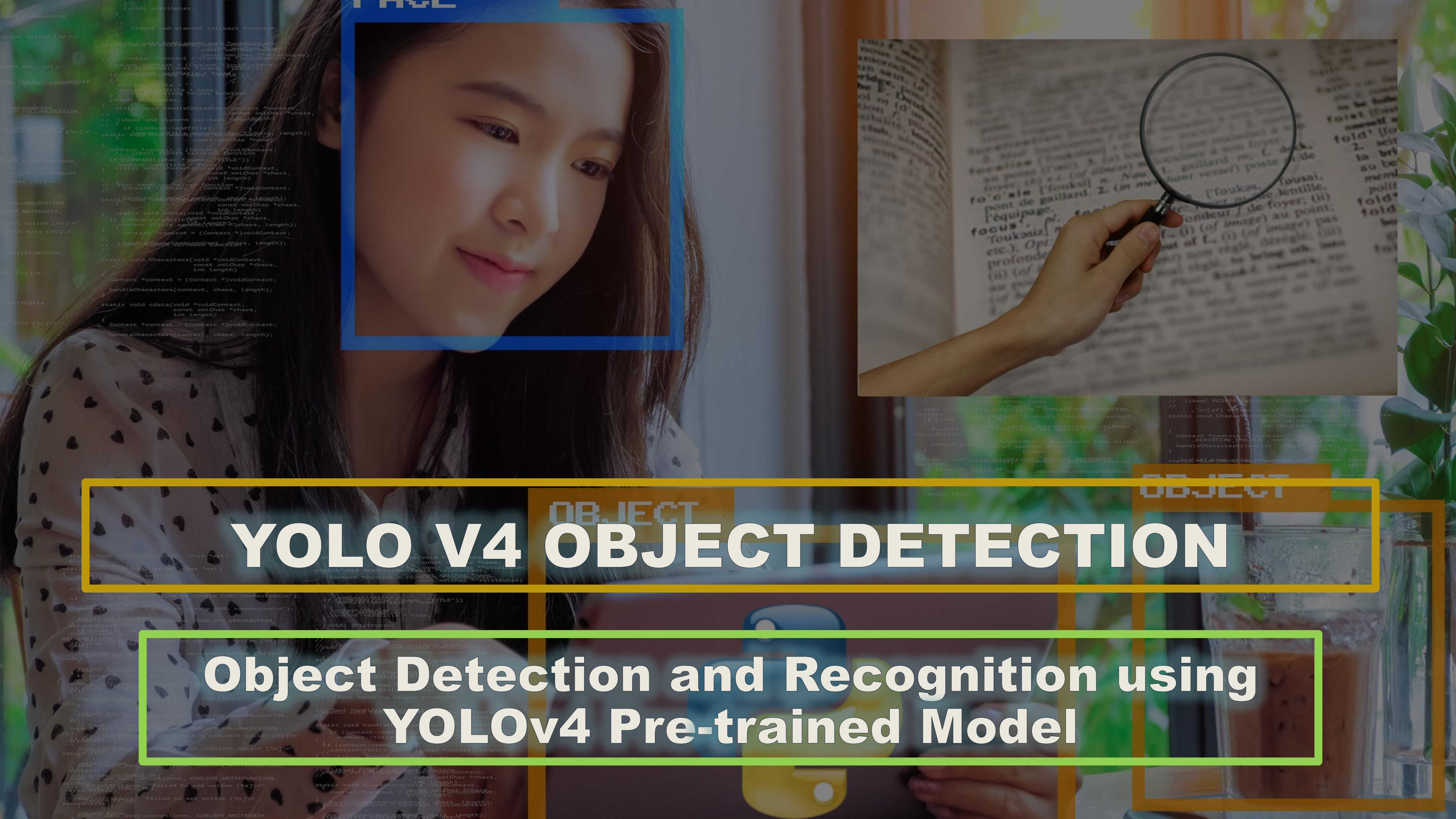


YOLO V4 OBJECT DETECTION

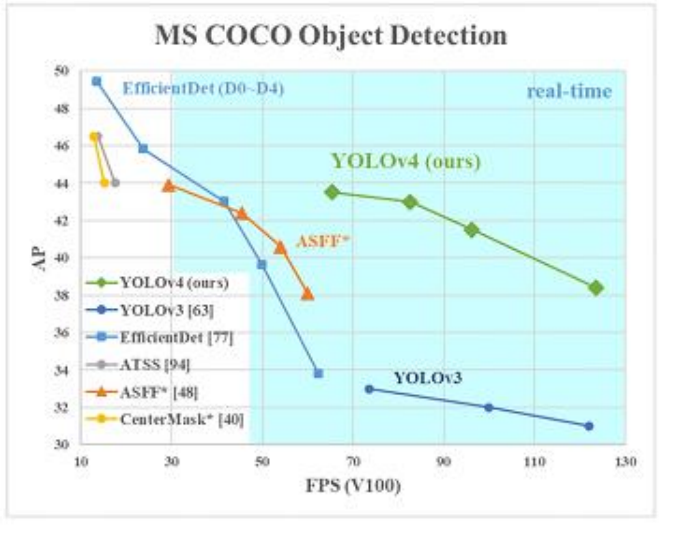
Object Detection and Recognition using YOLOv4 Pre-trained Model

ABOUT YOLO V4

- The actual author of the YOLO Object detector is Joseph Chet Redmon. (The first author of Darknet)

- In Feb 2020, Joseph Redmon announced he was leaving the field of computer vision.

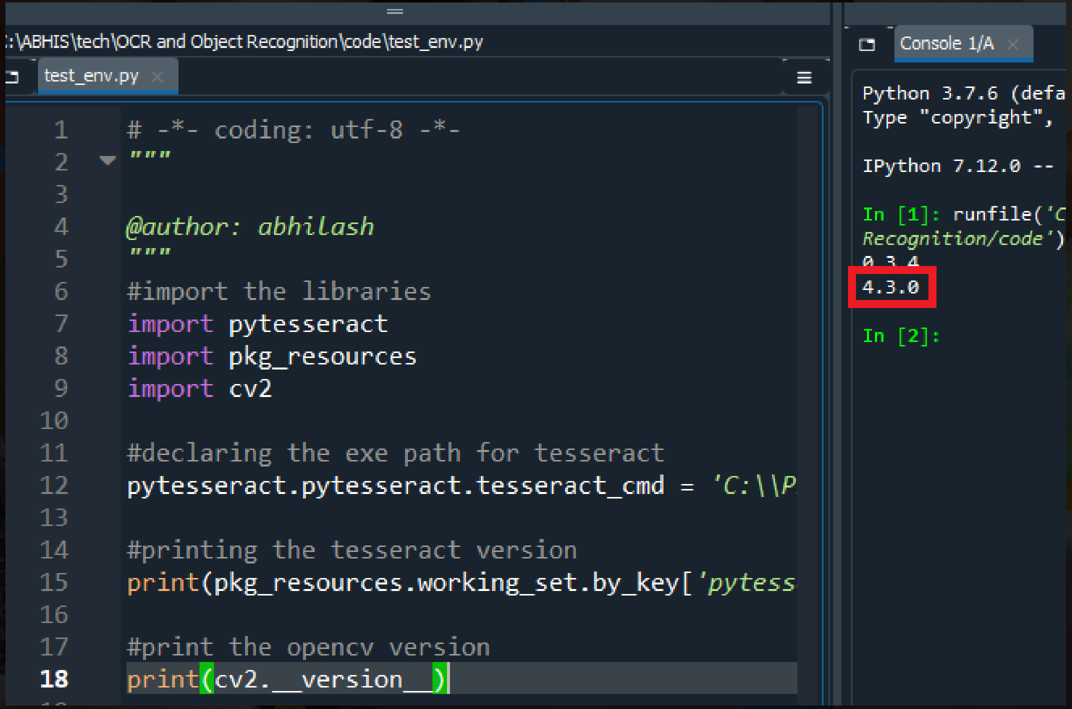
- Later YOLO v4 , a more fast and efficient version of YOLO v3 was released in April 2020 by three authors: Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao.



- YOLO v4 is officially maintained in in Alexey’s git repository

CHECKING OPENCV VERSION

- OpenCV needs to be updated to latest version (4.4 or above) for YOLOv4
- Verify the version. Uninstall current OpenCV <u>if it's a lower version</u>



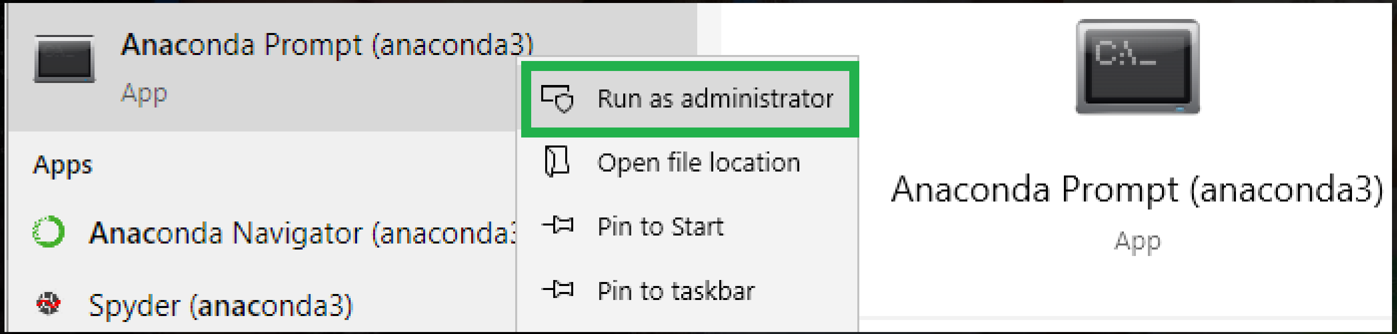
```python
# -*- coding: utf-8 -*-
"""

@author: abhilash
"""
#import the libraries
import pytesseract
import pkg_resources
import cv2

#declaring the exe path for tesseract
pytesseract.pytesseract.tesseract_cmd = 'C:\\P

#printing the tesseract version
print(pkg_resources.working_set.by_key['pytess

#print the opencv version
print(cv2.__version__)
```

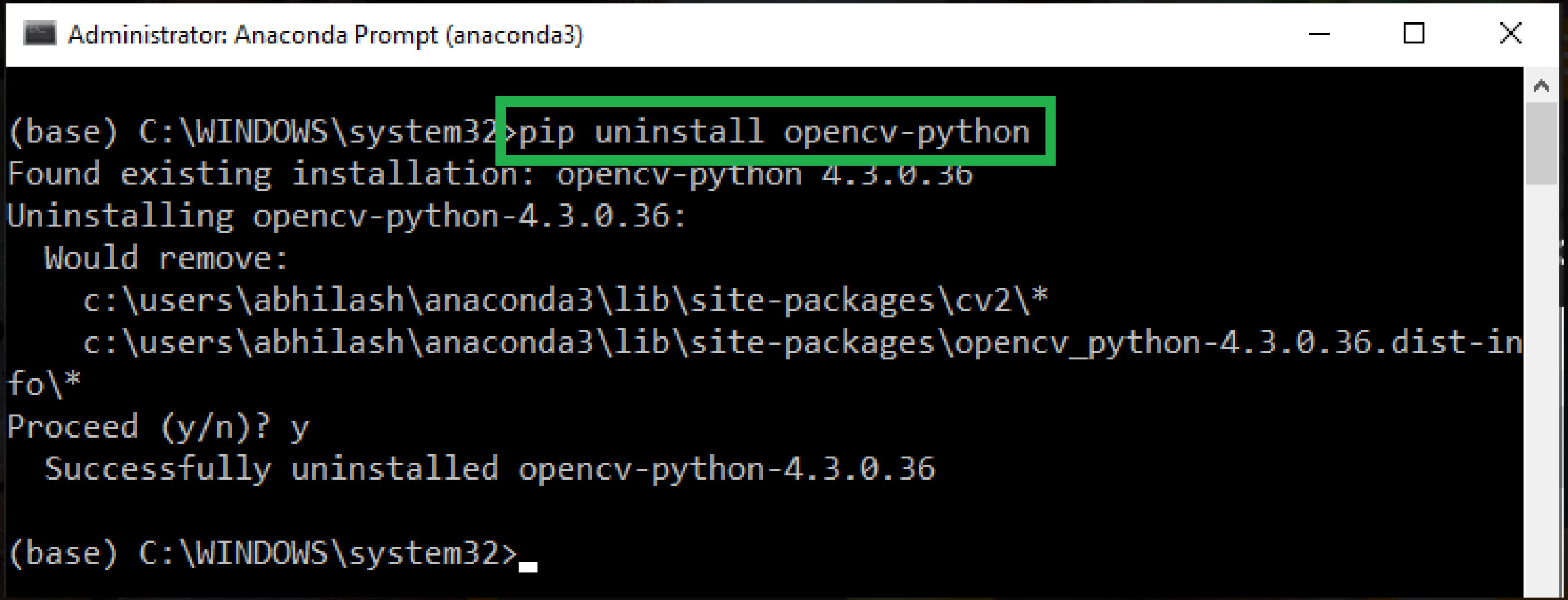
```
Python 3.7.6 (defa
Type "copyright",

IPython 7.12.0 --

In [1]: runfile('C
Recognition/code')
0.3.4
4.3.0

In [2]:
```

- Skip the 'OpenCV updating' part if the OpenCV <u>version is 4.4 or above</u> and proceed to 'Downloading YOLOv4 weights & config' section

UPDATING OPENCV VERSION

- If the version is below 4.4:
- Close all Spyder or Anaconda Navigator windows and run uninstall command

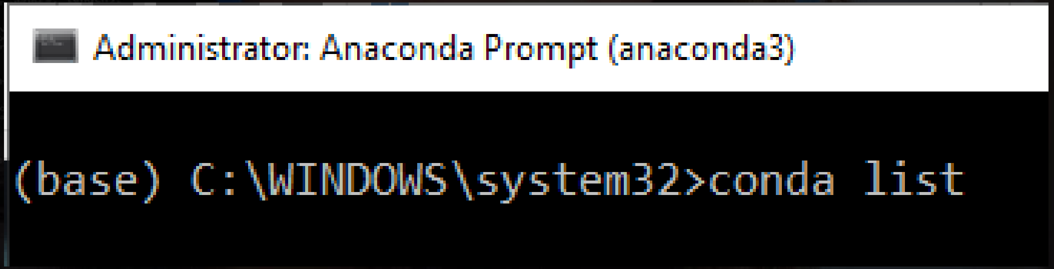




```
(base) C:\WINDOWS\system32>pip uninstall opencv-python
Found existing installation: opencv-python 4.3.0.36
Uninstalling opencv-python-4.3.0.36:
  Would remove:
    c:\users\abhilash\anaconda3\lib\site-packages\cv2\*
    c:\users\abhilash\anaconda3\lib\site-packages\opencv_python-4.3.0.36.dist-in
fo\*
Proceed (y/n)? y
  Successfully uninstalled opencv-python-4.3.0.36

(base) C:\WINDOWS\system32>_
```

UPDATING OPENCV VERSION

- And check if the OpenCV version in the conda package list is an updated (4.4 or above) version

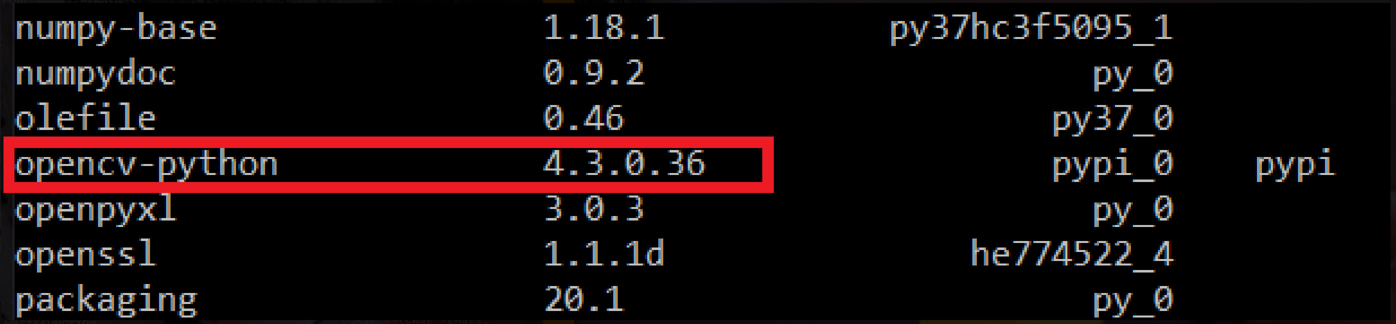


Administrator: Anaconda Prompt (anaconda3)

```
(base) C:\WINDOWS\system32>conda list
```

- If YES, proceed with installation using ‘pip install opencv-python’

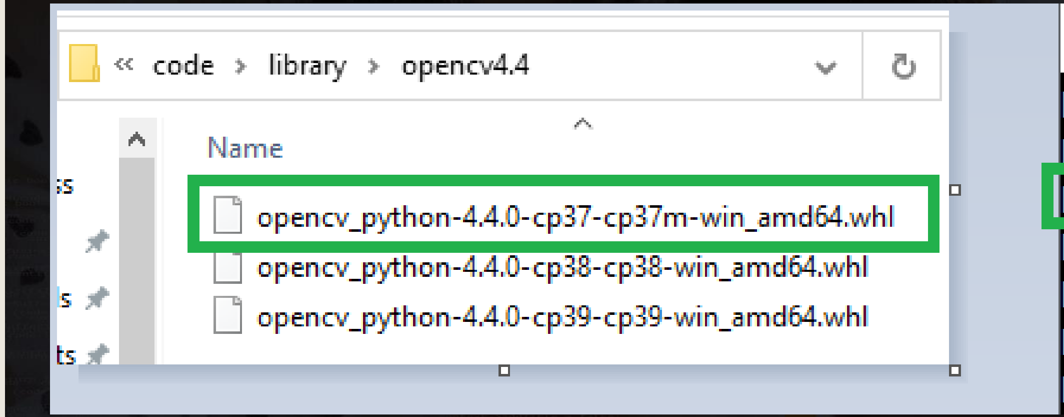
- If NOT, proceed with a manual installation using a wheel file



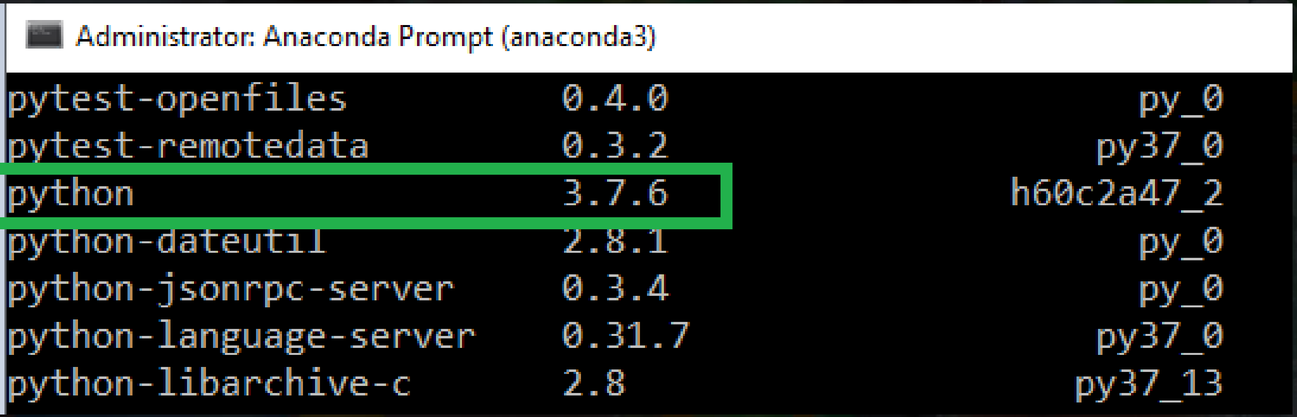
```
numpy-base                1.18.1           py37hc3f5095_1
numpydoc                  0.9.2                      py_0
olefile                   0.46                     py37_0
opencv-python             4.3.0.36                 pypi_0    pypi
openpyxl                  3.0.3                      py_0
openssl                   1.1.1d               he774522_4
packaging                 20.1                       py_0
```

MANUAL OPENCV SETUP USING .WHL FILE

- Already downloaded and verified the OpenCV 4.4 version wheel file for Python 3.7 (or) 3.8 (or) 3.9 from https://www.lfd.uci.edu/~gohlke/pythonlibs/#opencv
- Downloaded into 'code/library' folder
- Find the appropriate version



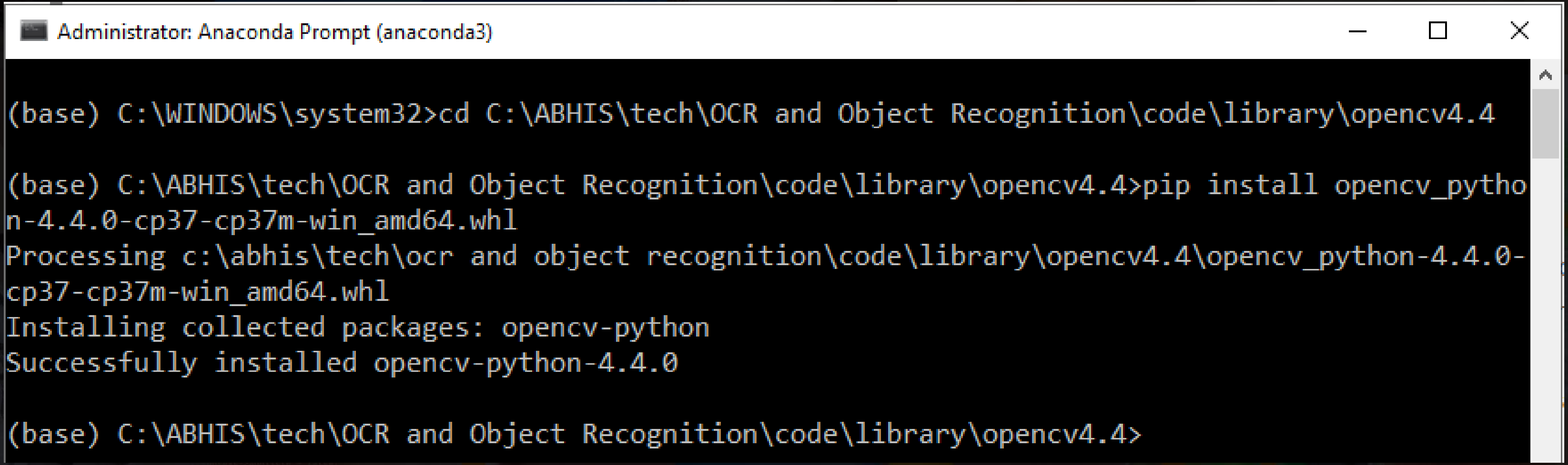
opencv_python-4.4.0-cp37-cp37m-win_amd64.whl
opencv_python-4.4.0-cp38-cp38-win_amd64.whl
opencv_python-4.4.0-cp39-cp39-win_amd64.whl



```
pytest-openfiles          0.4.0                      py_0
pytest-remotedata         0.3.2                    py37_0
python                    3.7.6               h60c2a47_2
python-dateutil           2.8.1                      py_0
python-jsonrpc-server     0.3.4                      py_0
python-language-server    0.31.7                   py37_0
python-libarchive-c       2.8                     py37_13
```

MANUAL OPENCV SETUP USING .WHL FILE

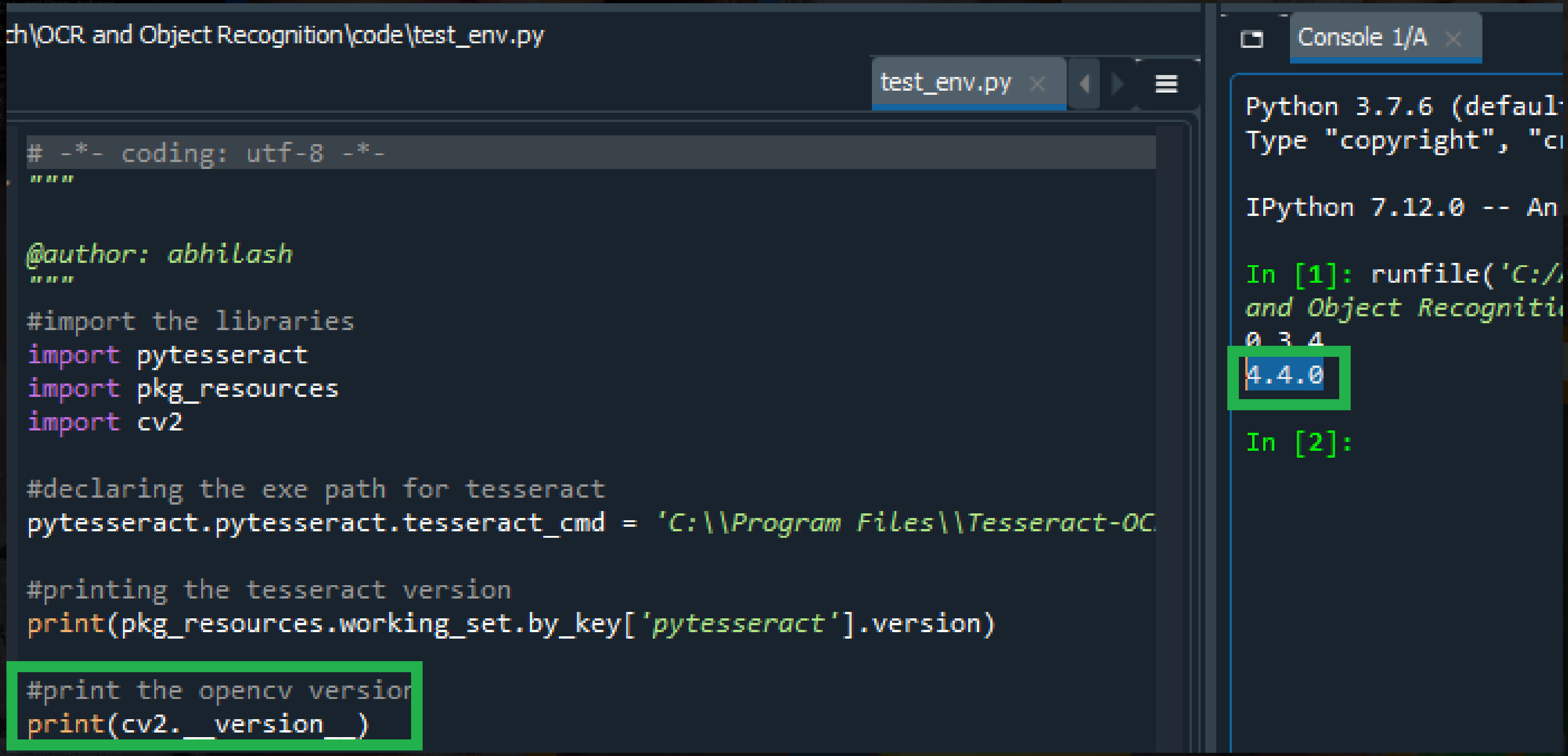
- Then go inside that folder using ‘cd’ command
- And using pip install , install opencv-python



```
(base) C:\WINDOWS\system32>cd C:\ABHIS\tech\OCR and Object Recognition\code\library\opencv4.4

(base) C:\ABHIS\tech\OCR and Object Recognition\code\library\opencv4.4>pip install opencv_pytho
n-4.4.0-cp37-cp37m-win_amd64.whl
Processing c:\abhis\tech\ocr and object recognition\code\library\opencv4.4\opencv_python-4.4.0-
cp37-cp37m-win_amd64.whl
Installing collected packages: opencv-python
Successfully installed opencv-python-4.4.0

(base) C:\ABHIS\tech\OCR and Object Recognition\code\library\opencv4.4>
```

VERIFY OPENCV VERSION

- Now, open Spyder and Verify the OpenCV version

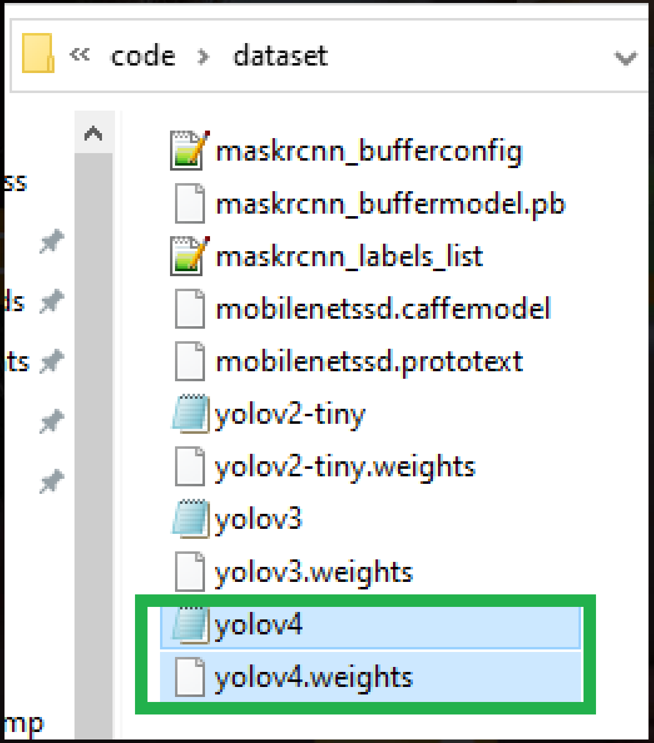


```
# -*- coding: utf-8 -*-
"""

@author: abhilash
"""
#import the libraries
import pytesseract
import pkg_resources
import cv2

#declaring the exe path for tesseract
pytesseract.pytesseract.tesseract_cmd = 'C:\\Program Files\\Tesseract-OC

#printing the tesseract version
print(pkg_resources.working_set.by_key['pytesseract'].version)

#print the opencv version
print(cv2.__version__)
```

```
Python 3.7.6 (defaul
Type "copyright", "c

IPython 7.12.0 -- An

In [1]: runfile('C:/
and Object Recogniti
0.3.4
4.4.0

In [2]:
```

DOWNLOADING YOLOV4 WEIGHTS & CONFIG

- Already downloaded both version-4 weights and config into /dataset folder from the github https://github.com/AlexeyAB/darknet
- Tested and verified.

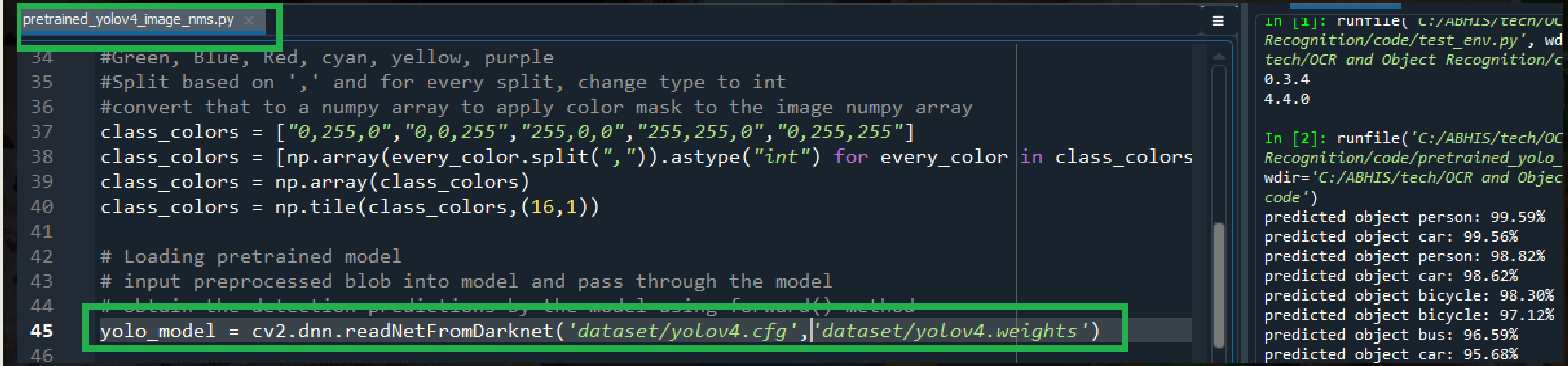


maskrcnn_bufferconfig
maskrcnn_buffermodel.pb
maskrcnn_labels_list
mobilenetssd.caffemodel
mobilenetssd.prototext
yolov2-tiny
yolov2-tiny.weights
yolov3
yolov3.weights
yolov4
yolov4.weights

COCO
Common Objects in Context

SAVE A COPY AND MAKE CHANGES IN THE CODE FOR THE FILE PATHS

- Do the same change for image, real-time and pre-saved video exercises



```python
#Green, Blue, Red, cyan, yellow, purple
#Split based on ',' and for every split, change type to int
#convert that to a numpy array to apply color mask to the image numpy array
class_colors = ["0,255,0","0,0,255","255,0,0","255,255,0","0,255,255"]
class_colors = [np.array(every_color.split(",")).astype("int") for every_color in class_colors
class_colors = np.array(class_colors)
class_colors = np.tile(class_colors,(16,1))

# Loading pretrained model
# input preprocessed blob into model and pass through the model
# obtain the detection predictions by the model using forward() method
yolo_model = cv2.dnn.readNetFromDarknet('dataset/yolov4.cfg','dataset/yolov4.weights')

```

```
In [1]: runfile('C:/ABHIS/tech/OC
Recognition/code/test_env.py', wd
tech/OCR and Object Recognition/c
0.3.4
4.4.0

In [2]: runfile('C:/ABHIS/tech/OC
Recognition/code/pretrained_yolo_
wdir='C:/ABHIS/tech/OCR and Objec
code')
predicted object person: 99.59%
predicted object car: 99.56%
predicted object person: 98.82%
predicted object car: 98.62%
predicted object bicycle: 98.30%
predicted object bicycle: 97.12%
predicted object bus: 96.59%
predicted object car: 95.68%
```