YOLOv3 USING OPENCV DOCUMENTATION

What is YOLO? (2016 Research Paper)
https://arxiv.org/pdf/1506.02640v5.pdf

YOLO in action (DEMO video)
https://youtu.be/NM6lrxy0bxs

When you begin exploring the various methods of implementing YOLO, you might probably be overwhelmed by all the different methods we have whivh would easily get you all confused. So, here are the 3 most used and known frameworks compatible with YOLO and the advantages and disadvantages of each one of them:

•**Darknet** : it's the framework built from the developer of YOLO and made specifically for yolo.
Advantage: it's fast, it can work with GPU or CPU
Disadvantage: it olny works with Linux os

•**Darkflow:** it's the adaptation of darknet to Tensorflow (another deep leanring framework).
Advantage: it's fast, it can work with GPU or CPU, and it's also compatible with Linux, Windows and Mac.
Disadvantage: the installation it's really complex, especially on windows

•**Opencv:** also opencv has a deep learning framework that works with YOLO. Just make sure you have opencv 3.4.2 at least.
Advantage: it works without needing to install anything except opencv.
Disadvantage: it only works with CPU, so you can't get really high speed to process videos in real time.

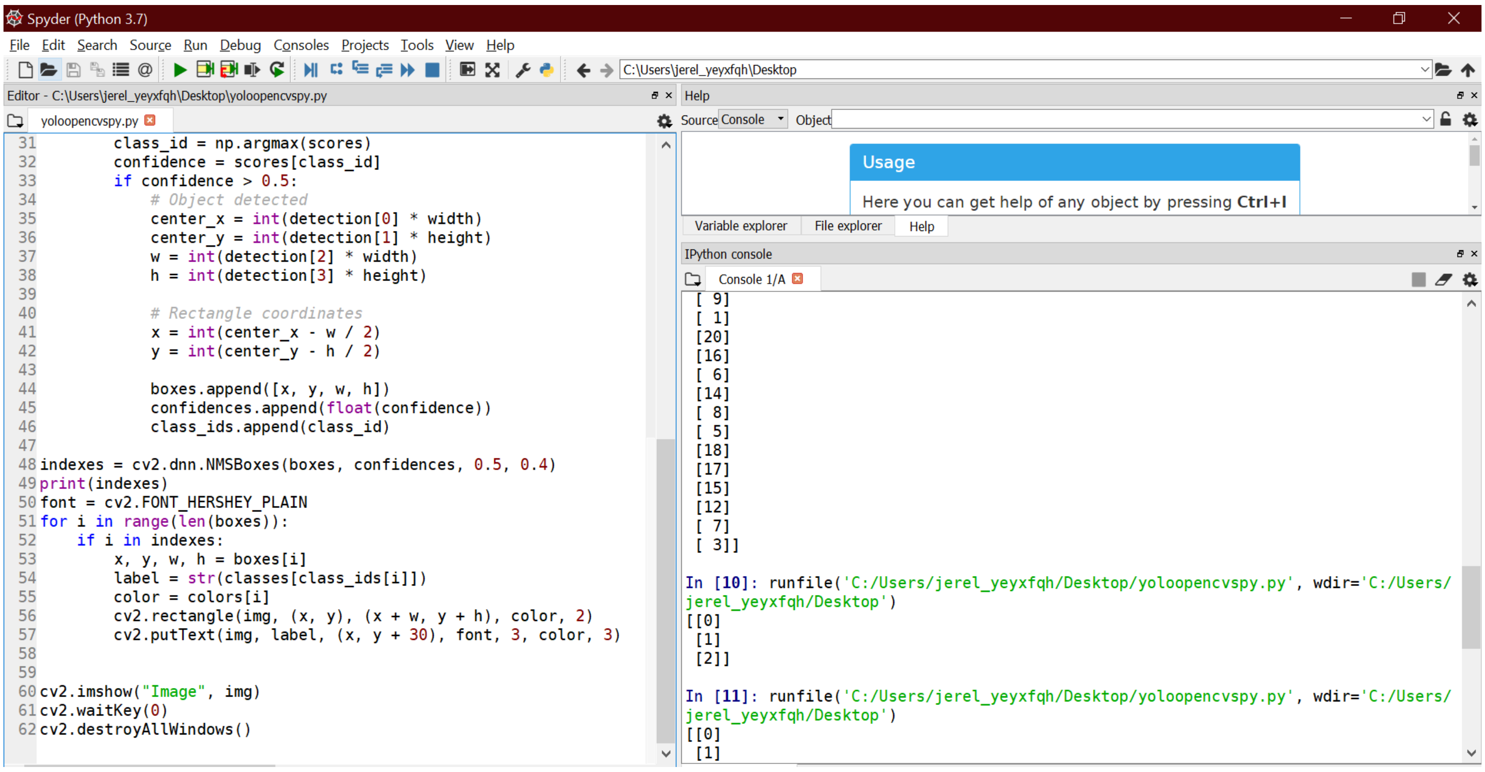
We will be focusing on the OpenCV implementation in this documentation.

PRE-REQUISITES

- We will be using a few classes from the libraries, opencv and numpy. So make sure you have that installed using the anaconda prompt.
- You need to make sure if Python3 is running on your system.
- What you'll need to import to run the algorithm
  To run the algorythm, we need three files:
  1. **Weight file:** it's the trained model, the core of the algorithm to detect the objects.
  2. **CFG file**: it's the configuration file, where there are all the settings of the algorythm.
  3. **Name files:** contains the name of the objects that the algorithm can detect.
- We will be using the GPU Google Collab, so all you need to have is a drive to work with.

IMPLEMENTATION

Get Anaconda Navigator running as we will be implementing the Algorithm on the GPU 'Spyder', which should look something like this


Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\jerel_yeyxfqh\Desktop
Editor - C:\Users\jerel_yeyxfqh\Desktop\yoloopencvspy.py
yoloopencvspy.py
Help
Source Console Object
Usage
Here you can get help of any object by pressing Ctrl+I
Variable explorer File explorer Help
IPython console
Console 1/A

```python
        class_id = np.argmax(scores)
        confidence = scores[class_id]
        if confidence > 0.5:
            # Object detected
            center_x = int(detection[0] * width)
            center_y = int(detection[1] * height)
            w = int(detection[2] * width)
            h = int(detection[3] * height)

            # Rectangle coordinates
            x = int(center_x - w / 2)
            y = int(center_y - h / 2)

            boxes.append([x, y, w, h])
            confidences.append(float(confidence))
            class_ids.append(class_id)

indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)
print(indexes)
font = cv2.FONT_HERSHEY_PLAIN
for i in range(len(boxes)):
    if i in indexes:
        x, y, w, h = boxes[i]
        label = str(classes[class_ids[i]])
        color = colors[i]
        cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)
        cv2.putText(img, label, (x, y + 30), font, 3, color, 3)


cv2.imshow("Image", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

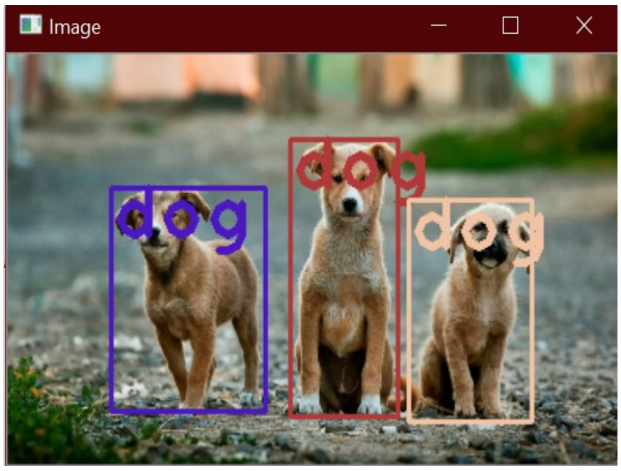
```
 [ 9]
 [ 1]
 [20]
 [16]
 [ 6]
 [14]
 [ 8]
 [ 5]
 [18]
 [17]
 [15]
 [12]
 [ 7]
 [ 3]]

In [10]: runfile('C:/Users/jerel_yeyxfqh/Desktop/yoloopencvspy.py', wdir='C:/Users/
jerel_yeyxfqh/Desktop')
[[0]
 [1]
 [2]]

In [11]: runfile('C:/Users/jerel_yeyxfqh/Desktop/yoloopencvspy.py', wdir='C:/Users/
jerel_yeyxfqh/Desktop')
[[0]
 [1]
```

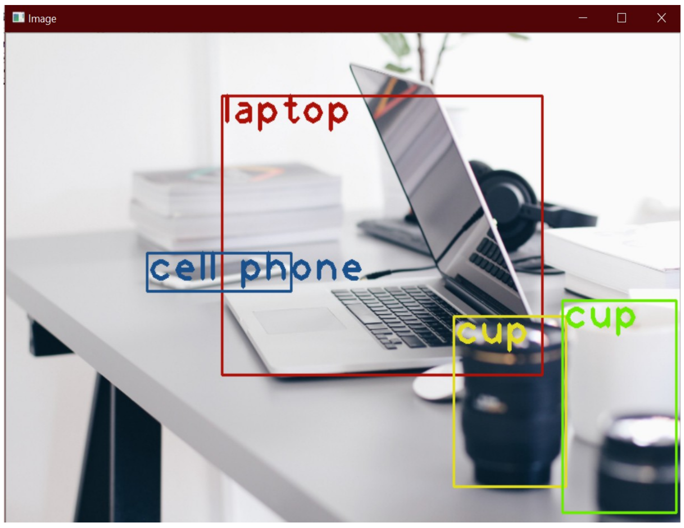
<u>Remember to fix the path to the files before running your code.</u>

SOURCE CODE

In the Python File

OUTPUT





MORE ABOUT YOLOv3? (A good medium post)
https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b