**OBJECT DETECTION**

This project is to create an ***Object Detector*** that has good balance between speed and accuracy and is able to run in real time while detecting multiple objects. The good thing about the project is that it does not require any third-party libraries other than ***OpenCV.***

**IDEA:**

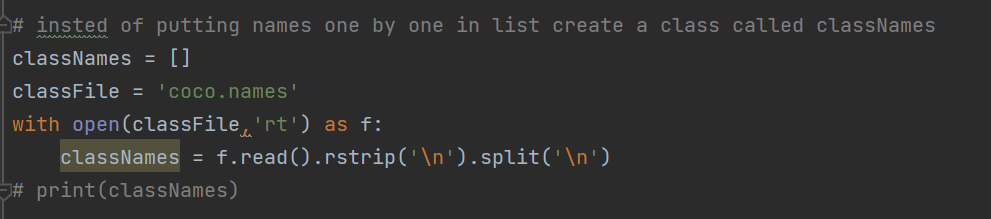
The idea behind this project is that we should be able to getup and running Object Detector as fast as possible without going into too much installation. It is a pre-trained model. If you want to use this code in a robot or a car just install in the machine and it should be ready for you to detect objects.



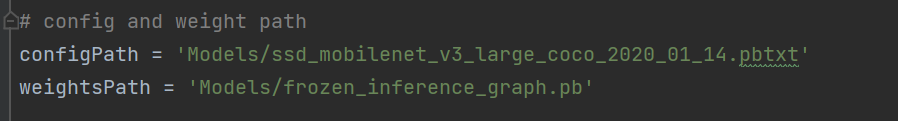
**EXPLANATION:**

Firstly Import the cv2, The **cv2** is a**cross-platform library** designed to solve all computer vision-related problems.

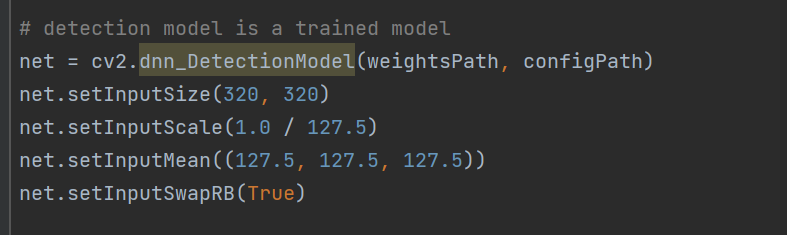
Then we need to read our image or use our video camera for live detection using **cv2.VideoCapture(0).** Then we define the cap set for the frame size of video camera.



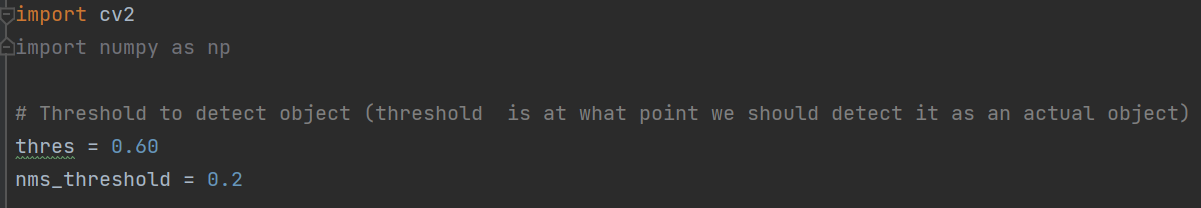
**Coco.names** are the classes that we can detect. Rather then putting the data set names one by one in list we import a class called **ClassNames** files which will arrange automatically rather than manually. It contains more than 90 objects that can be deteted at good accuracy rate.



Then we give a path called **configuration path** and **weight path**. This is the best and fast method right Now using **mobile net ssd**. These are Basically models.

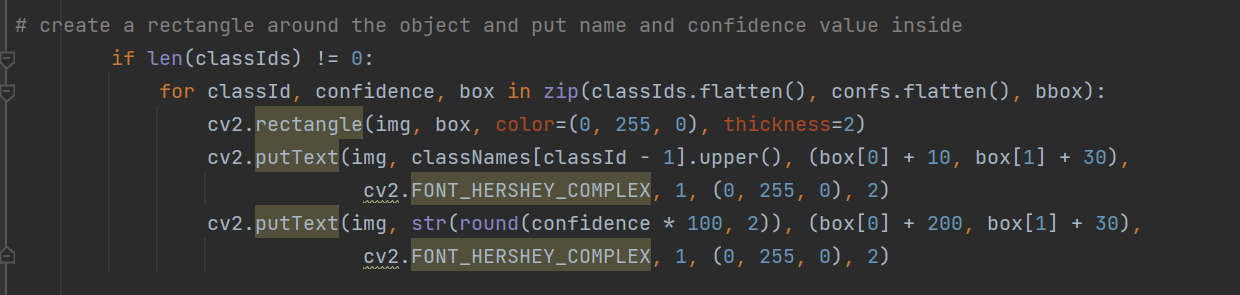


Create a model then called **cv2.dnn\_DetectionModel**. Open cv already provides us with a function that actually does all the processing for us we just have to import configuration path and weights path. Then we need to send our image to VideoCapture to the model then it will detect our objects.



**Threshold** is at what point we should detect it is an actual object. If its sure that it is **60% Object** then its good enough to detect object otherwise if its lower than 60% threshold it won’t detect the object.

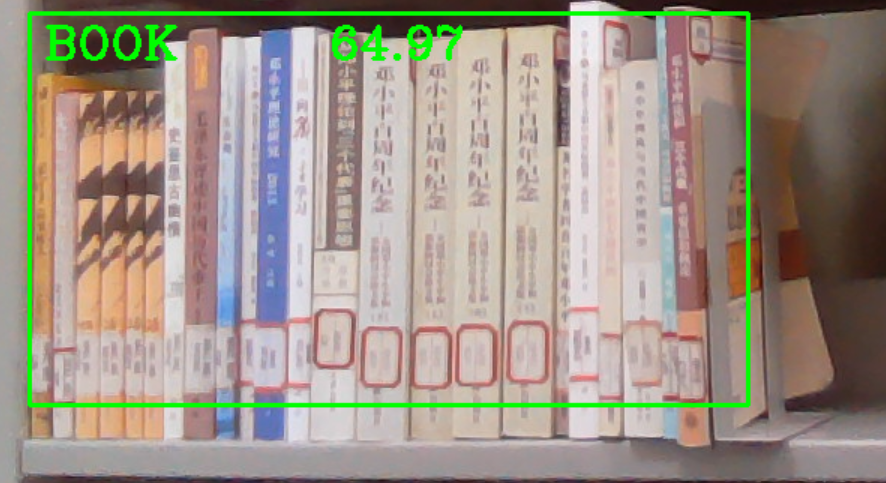
**bbox** is the bounding box around the detected object.



From this information we are going to create a **rectangle** around the detected object and also put a **name** of the object inside the rectangle to specify the name of the detected object and can also put the **threshold percentage** to let us know how much the model is sure that it’s right about the detected object.

Then we pass our code to **cv2.imshow** to run the camera and detect objects.

**RESULT:**



After running the code we can see that it has detect some objects in real time like a bottle, books, couch, mobile phone etc. Create a rectangle around the object and specify the name inside. You can see in image 1 it shows that our current model is almost **76%** sure that it is a bottle.