**Object Detection Using OpenCV Yolo**

You only look once (YOLO) is a state-of-the-art, real-time object detection system.YOLOv3 is extremely fast and accurate. Moreover, you can easily tradeoff between speed and accuracy simply by changing the size of the model, no retraining required!

### **How It Works**

Prior detection systems repurpose classifiers or localizers to perform detection. They apply the model to an image at multiple locations and scales. High scoring regions of the image are considered detections.

We use a totally different approach. We apply a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

## **Requirements**

-->Python

-->OpenCV

-->Numpy

-->yolov3.cfg

-->yolov3.txt

-->yolov3.weights

**Steps Involved in Detector:**

--> Firstly,store the classes that yolov3.txt had in the classes file.These classes are used to predict what kind of object is detected in the image. These classes are used by the clasifier to predict the classes of the object that belongs.

-->next read class names from text file into the classes .

--> generate different colors for different classes using np.random.uniform().

--> get\_output\_layers() is a function to get the output layer names in the architecture.

--> draw\_bounding\_box() is a function to draw bounding box on the detected object with class name.

-->read pre-trained model and config file using cv2.dnn.readNet().

--> create input blob.

blob = cv2.dnn.blobFromImage()

--> set input blob for the network.

net.setInput(blob)

--> run inference through the network and gather predictions from output layers.

-->for each detetion from each output layer get the confidence, class id, bounding box params and ignore weak detections (confidence < 0.5) then apply non-max suppression.

--> after non-max suppression and draw bounding box and display output image

#### **Advantages:**

Our model has several advantages over classifier-based systems. It looks at the whole image at test time so its predictions are informed by global context in the image. It also makes predictions with a single network evaluation unlike systems like R-CNN which require thousands for a single image. This makes it extremely fast, more than 1000x faster than R-CNN and 100x faster than Fast R-CNN. See our paper for more details on the full system.