**Research paper on YOLO algorithm**

YOLO (You Only Look Once) algorithm has emerged as a well-liked and structured solution for real-time object detection due to its ability to detect items in one operation through the neural network.

Concept of Object detection using YOLO Dividing the input image into a grid and every grid cell estimates some certain bounding boxes in fixed numbers along with respective class probabilities. And these boxes are responsible for detecting the object within the particular grid cell

Key features in the process of YOLO’s unified approach:

* Anchor Boxes
* Grid Division
* Prediction Generation
* Non-maximum Suppression

A simple neural network predicts class and bounding box probabilities directly from the images in just one set of evaluations.

For Object detection we define a 7 elemental vertical matrix that looks like this:

Untitled.png

Px: value that tells the presence of an object(0 or 1)

Bx: x-coordinate of the anchor point that detects the object

By: y-coordinate of the same

Bw: width of the bounding box

Bh: height of the bounding box

C1 and C2 are defined when there are multiple images or during image classification

Eg. C1 is animal presence and C2 is human presence

Eg. C1 is animal presence and C2 is human presence



For the first image: Px=0(no object)

and therefore the values of other parameters is not defined.



For the second image Px=1, C1=1, C2=0 and so on for human image

But for grids with more than a single object we expand the matrix dimensions to 7 times the number of objects

Before YOLO V2 , traditional object detection methods was used like FULLY CONNECTED LAYERS. These were able to find the coordinates of the bounding box for detected objects, but failed for small objects and was not able to provide enough info to make accurate predictions

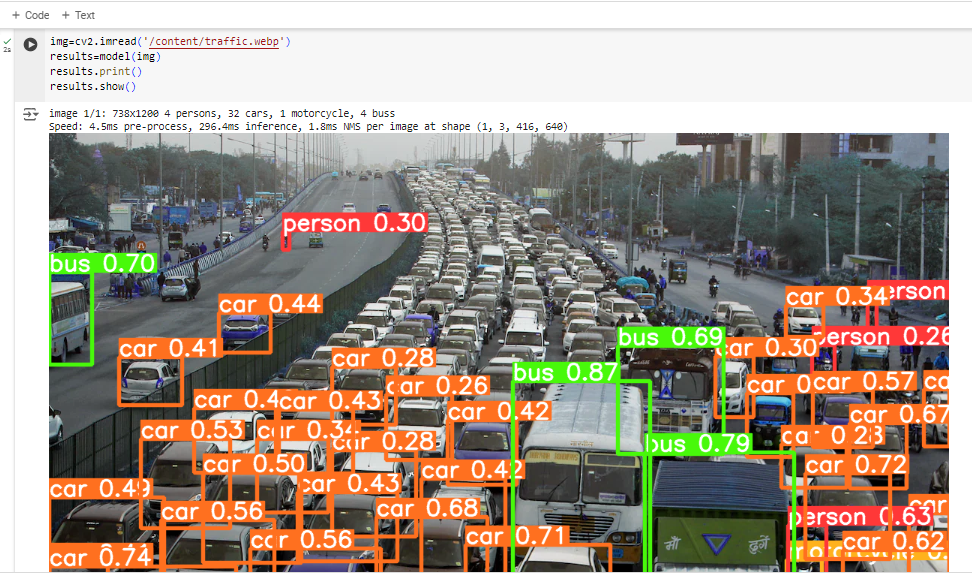
The concept of ANCHOR BOXES solved this problem. YOLOv2 uses pre-defined boxes of various sizes and aspect ratios at different locations in the image. This approach allows the model to handle small objects more effectively. By using pre-defined anchor boxes that might already be close in size to small objects, the model only needs to predict smaller adjustments (offsets) to get accurate bounding boxes. It uses higher resolution images as compared to yolov1 and even Region based CNNs etc

Imagine a tailor with pre-cut fabric pieces (anchor boxes) of different sizes. Instead of guessing the exact measurements for a shirt from scratch (fully connected layers), the tailor adjusts these pre-cut pieces (predicts offsets) to fit the customer perfectly (accurate bounding boxes) - especially useful for smaller sizes (small objects). This approach is generally faster (efficient) and works better for a wider range of sizes (multi-scale training).

YOLOV3 uses the concept of DARKNET

It is a open source neural network framework on which YOLOV3 works and is suitable for real-time object detectio through the use of GPUs

The below image shows how I tried out the ultralytics yolov5s model to detect cars of different shapes. And how the size of the bounding boxes changes(concept of Anchor Boxes)

Link for the entire code for YOLO: [yolo\_model.ipynb](https://colab.research.google.com/drive/1CimNaoQrv4p2O_Kxog4trJ9ohbRAS3Mr#scrollTo=xTa6W_omS3rI)