SUMMARY ON YOLO ALGORITHM

Gaurav Singh
22b0668

YOLO (YOU ONLY LOOK ONCE)

I read the research paper and below is the summary , I also did some research to find how the algorithm works , also below are the handwritten notes I made about how YOLO algorithm works. I also ran YOLOv5 on my computer I have attached the link to the results.

Summary and Analysis of Papers on YOLO and Vehicle Classification

Manually counting and classifying vehicular traffic is very time-consuming and require a lot of effort plus they are not really suitable for modern intelligent transport systems. YOLO algorithm uses Neural Network based approach to **detect** and **classify objects**. YOLO structure consists of 24 convolution layers , followed by two fully connected layers. YOLO predicts multiple bounding boxes per grid cell but the box with highest Intersection Over Union (IOU) is selected (IOU>0.5) , which is known as non maxima suppression. YOLO V2 addresses the inaccuracies in positioning and the lower recall rate found in YOLO by focusing on these two key aspects. Instead of making the network deeper or broader, it simplifies the network to enhance performance, making it both more accurate and faster.

YOLO can detect objects in images and videos quickly and in real time too. Over the years, several versions of YOLO have been developed , in the paper upto version 5 is given, the latest version of YOLO the version 10 (YOLOv10) just came about 2 weeks ago . **The differentiation in YOLO models is their model size, accuracy, functionality and how fast the algorithm is and it keeps getting better with new versions .**

YOLO is very effective for various object detection tasks, including vehicle classification. Key advantages of using it include:

Speed - YOLO can process images in real-time, which is crucial for continuous video streams in intelligent transport systems.

Accuracy – previous versions had some issues with small or overlapping objects , newer versions have good accuracy.

Simplicity – It needs single input , making it fast and easy to use

Insights from Comparison

Each version of YOLO introduced new techniques to fix the problems of the previous versions.

Version	What's Better	Key Features
YOLOv1	It's fast but not very accurate	Detects objects in one go
YOLOv2	More accurate and finds more objects and easier to use in comparison to v1	Can detect over 9000 different objects.
YOLOv3	better at finding small objects, uses a deeper network	Detects objects at three different scales.
YOLOv4	faster and more accurate , uses new techniques like CSP and mosaic data.	Good for real-time detection.
YOLOv5	easier to use with different model sizes , more precise object detection.	User- frienldy , pre-trained models available
YOLOv6	Even faster and more accurate, better with small objects.	Balanced for speed and accuracy , works well on different hardware.
YOLOv7	Improved accuracy and speed , better for real-world use.	High accuracy in real-time

YOLOv2 introduced **batch normalization** to **standardize input layers**, accelerating training and boosting mean Average Precision (mAP) by approximately 2%. It also adopted a high-resolution classifier strategy, initially training on smaller images and gradually transitioning to larger ones during detection, thereby improving adaptability and classification accuracy. Additionally, YOLOv2 integrated fine features to connect different-sized feature maps, crucial for detecting both large and small objects effectively**.** **Multi-scale training** further optimized performance across different image resolutions.

In YOLOv3, the algorithm advanced with **multi-scale feature maps** using three scales to better detect objects of varying sizes (13x13, 26x26, and 52x52). The adoption of **Darknet-53**, a deeper residual network architecture, enhanced feature extraction capabilities compared to previous versions, significantly improving detection accuracy.

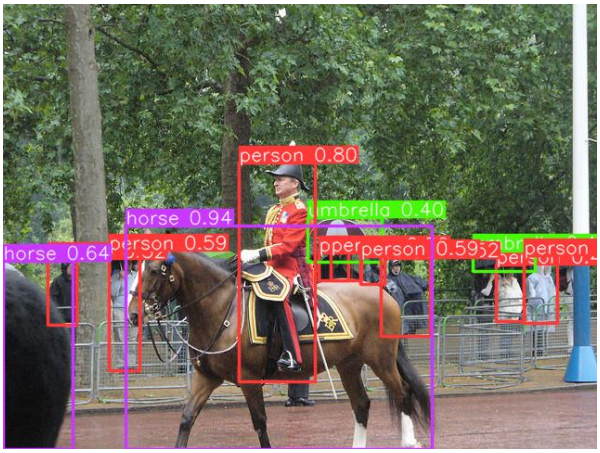
YOLOv4 represents a further evolution, integrating advanced architectures like **CSPDarknet53, SPP, PAN**, and innovations from YOLOv3. These integrations enabled YOLOv4 to achieve superior efficiency and accuracy in real-time object detection tasks. State-of-the-art techniques such as Cross-Stage Partial Connections (CBN) and the Complete Intersection over Union (CIOU) loss function were also implemented, greatly enhancing training efficiency and the precision of object localization. Adjustments in

how anchor points are utilized further improved the algorithm's ability to handle data imbalances and accurately detect objects.

Overall, these advancements in the YOLO algorithm have made it a leading choice for applications requiring high-performance object detection capabilities with real-time processing demands.

The YOLO series, culminating in **YOLOv5**, has evolved to prioritize ease of use with the PyTorch framework, efficient model training, and enhanced object detection capabilities, especially for small objects. Despite debates over innovation, YOLOv5 offers flexible model sizes and advanced data augmentation, maintaining its position at the forefront of real-time object detection technology.

RESULTS WHEN I RAN YOLOv5 ON MY COMPUTER



TRAFFIC VIDEO LINK

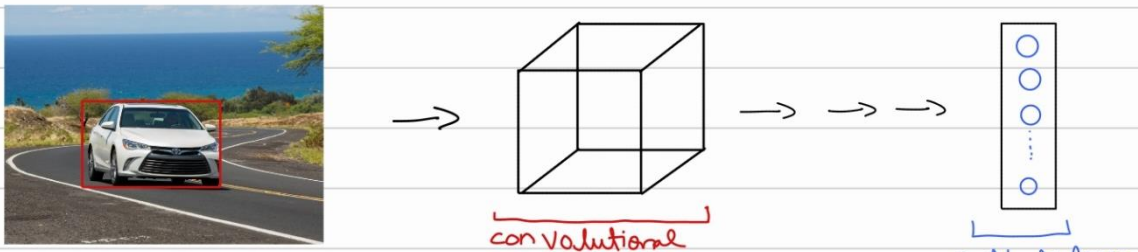
YOLO - YOU ONLY LOOK ONCE

"The differentation in YOLO Models is their model's small size and fast calculation."

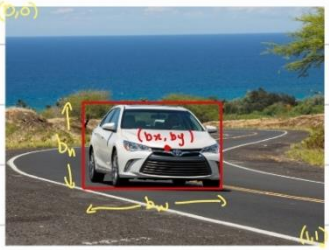
Classification vs Localization vs Detection

"Image Classification" → "Localization" → "Object Detection"

Car | Bounding Box | Object detection have multiple objects eg. Person, motorcycle, car



A Convolution converts all the pixels in its receptive field into a single value.



(bx, by) → mid point of bounding box

b_h: height of box — eg. a % of image dimension

b_w: width of box — eg. b% of image dimension

b_x: x coordinate of midpoint

b_y: y coordinate of mid point

P_c: whether objects are present or not, it takes 0 or 1.

c_1: car

c_2: motorcycle

c_3: person

⇒ If there is no object in image

$$y_0 = \begin{bmatrix} 0 \\ x \\ x \\ x \\ x \\ x \\ x \\ x \end{bmatrix}$$

$$y = \begin{bmatrix} P_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ \end{bmatrix}$$

whether an object present/not 0 or 1

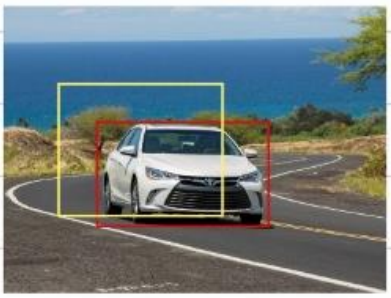
$$y_{car} = \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \cdot 96 \end{bmatrix}$$

Loss function

$$\text{Loss} = \begin{cases} (y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2 + \cdots (y_8 - \hat{y}_8)^2 & \text{if } y_1 = 1 \\ (y_1 - \hat{y}_1)^2 & \text{if } y_1 = 0 \end{cases}$$

YOLO has 19 x 19 grid

Intersection over Union → One of the evaluating factors which determines how well is our bound ary box is predicted.

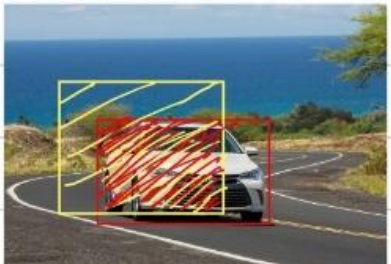


Red box → Correct box

Yellow box → box predicted by our algorithm

To evaluate how good our algorithm did we use IOU.

$$\text{IOU} = \frac{\text{Area of intersection}}{\text{Area of Union}}$$

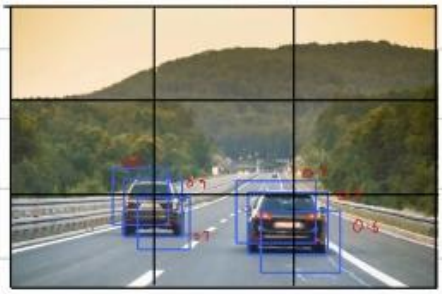


IOU ≥ 0.5 (good prediction)

IOU ≤ 0.5 (bad prediction)

Non Max Suppression

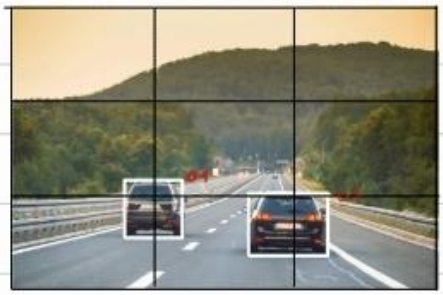
: It is possible than one single object can have multiple detection / boxes. Each box will think that, that is the central point of this car



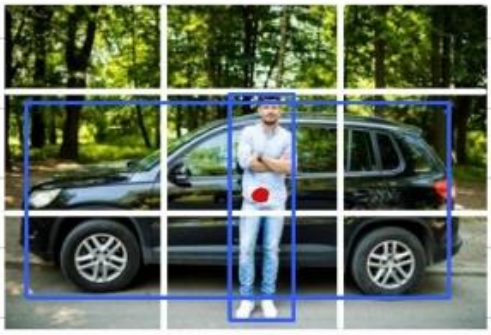
0.9 is the P_c value

To handle the above problem we use Non Max Suppression.

So Non Max Suppression will take the highest P_c value box. Then it takes the "IOU" of the max P_c value box with the other boxes. If IOU > 0.5 we can eliminate other boxes.



Anchor Boxes → Used when one cell has the midpoint of both the objects



● → Mid point

The central grid cell will have 2 P_c values for y.

Anchor Box 1

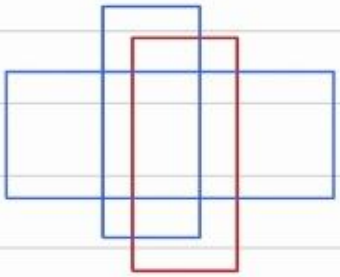
Anchor Box 2

So

$$y = \begin{bmatrix} P_{c_1} \\ b_{x_1} \\ b_{y_1} \\ b_{w_1} \\ b_{h_1} \\ c_1 \\ c_2 \\ c_3 \\ P_{c_2} \\ b_{x_2} \\ b_{y_2} \\ b_{w_2} \\ b_{h_2} \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

Anchor box 1

Anchor Box 2



IOU

Whichever cells gets the higher IOU that anchor box will be considered for evaluation.

If both car & person are present

$$y = \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 0 \\ 1 \\ 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

If only person is present.

$$y_2 = \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 0 \\ 1 \\ 0 \\ x \\ x \\ x \\ x \\ x \\ x \\ x \end{bmatrix}$$

So This will be $3 \times 3 \times 8 \times 2$

Limitation :- When there is same anchor box for two object with same centre.