Multi Object Detection Using

YOLO v1

Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi

Mrudula Y
50290843
Shishir Suvarna
50290573

List of topics

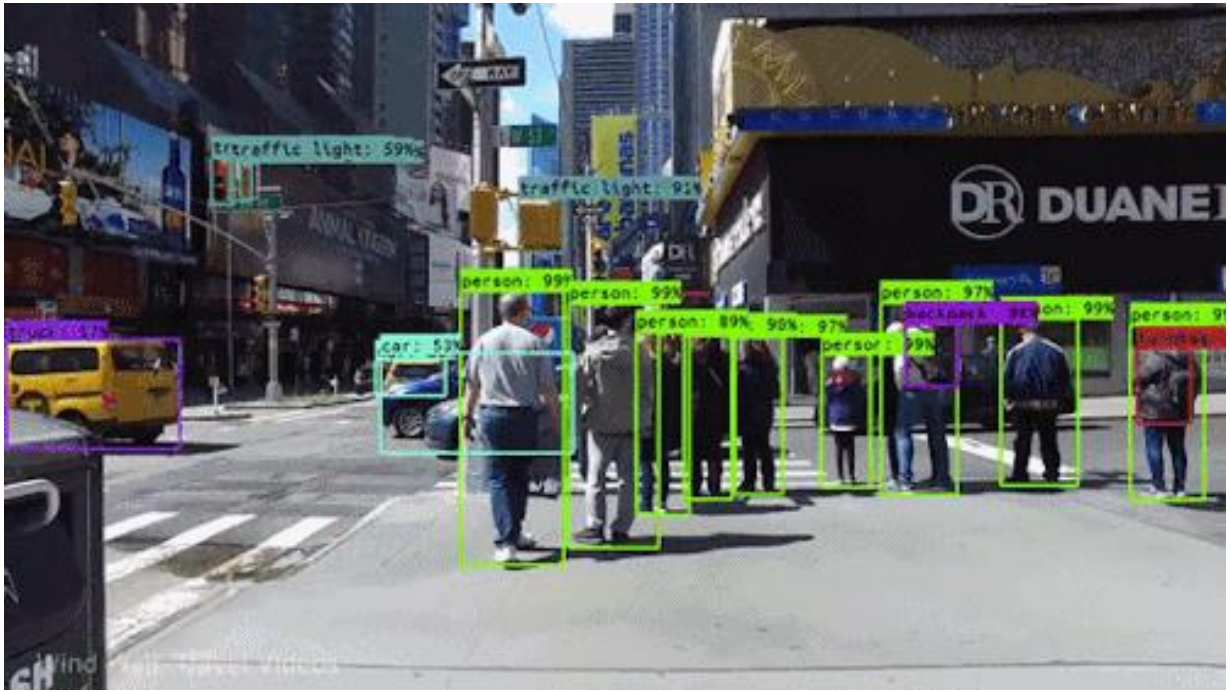
- Object detection
- Applications of Object Detection
- YOLO - Problem Statement
- YOLO - Motivation
- YOLO Algorithm
- YOLO - Network architecture
- Training the YOLO model
- Code Explanation
- Results
- Limitations
- Conclusions
- References

Object Detection

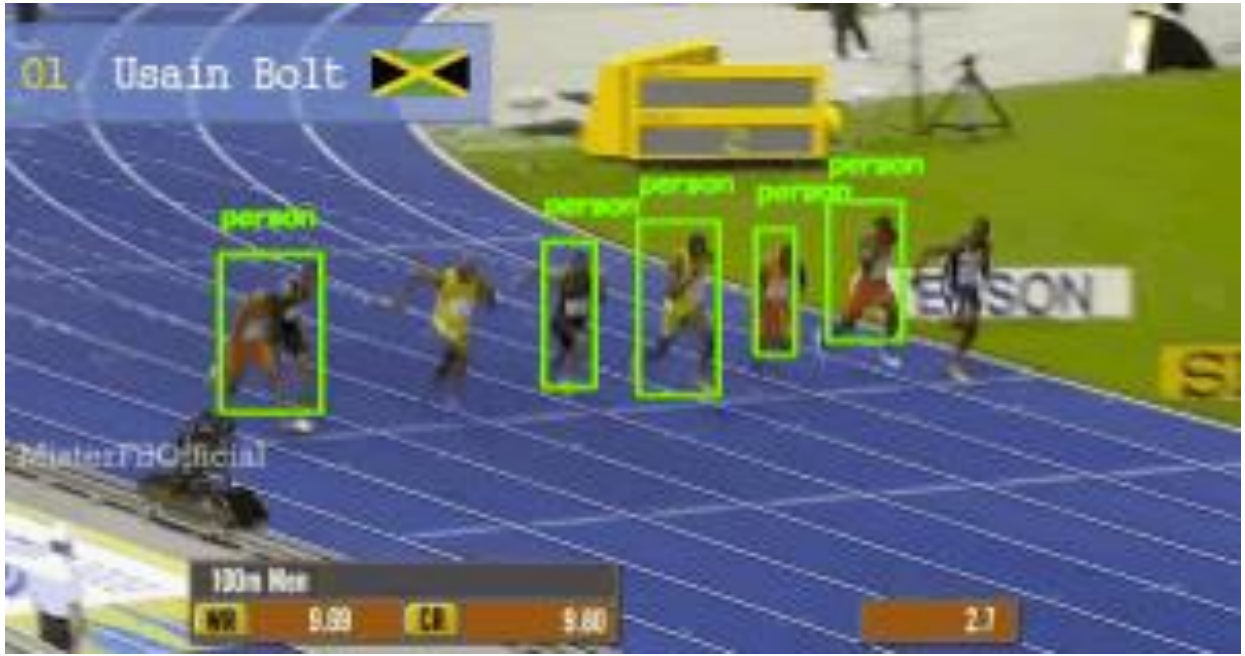
Involves detecting instances of objects from a particular class in an image.

Applications of Object Detection

Self-Driving Cars:



Object Tracking:



Optical Character Recognition

YOLO - You Only Look Once

Problem statement

YOLO is a new approach to object detection formulating it as a regression problem to spatially separated bounding boxes and associated class probabilities while having a unified single network architecture such that it reasons globally about the full image and all the objects in the image.

Motivation

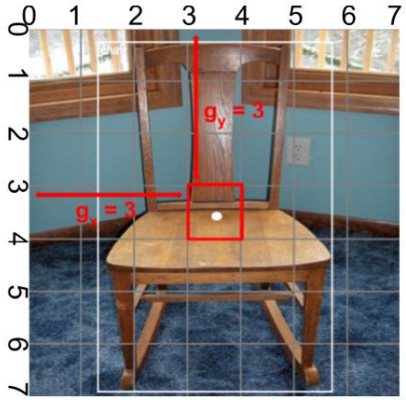
Current detection systems repurpose classifiers to perform detection

These complex pipelines are slow and hard to optimize since each individual component must be trained separately

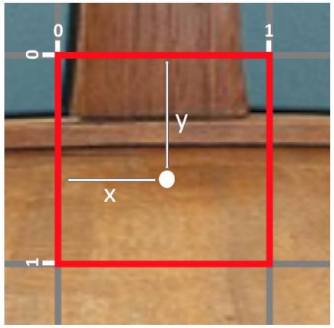
Current systems look at only a part of the image at one time. Hence, the global context of the image is not being captured by these systems

Bottomline: YOLO solves all the above problems by making predictions with a single network faster, which makes it suitable for real time detection

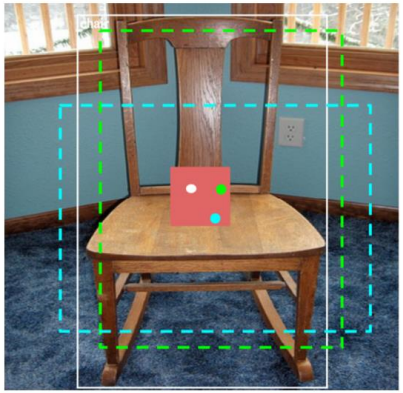
YOLO Algorithm



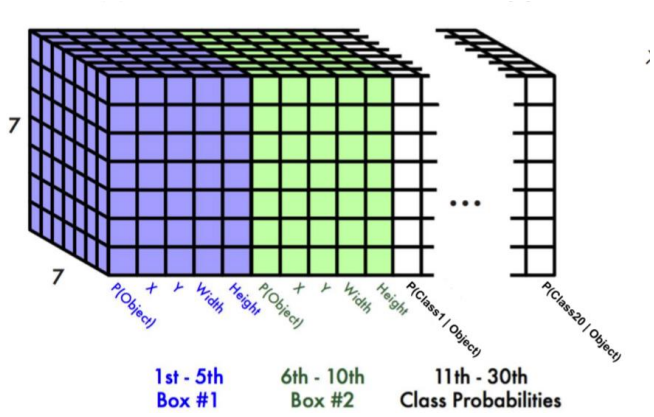
S=7

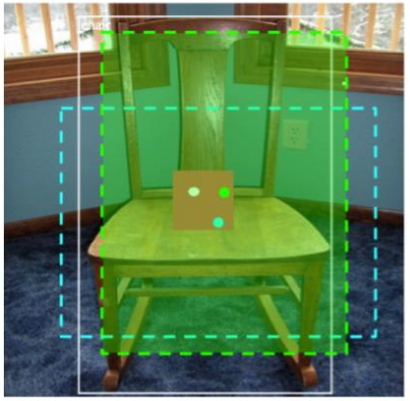


$$\hat{b}_1 = \underset{\hat{b} \in \{\hat{b}_1, \hat{b}_2\}}{\arg\max}\ IoU\left(b, \hat{b}\right)$$

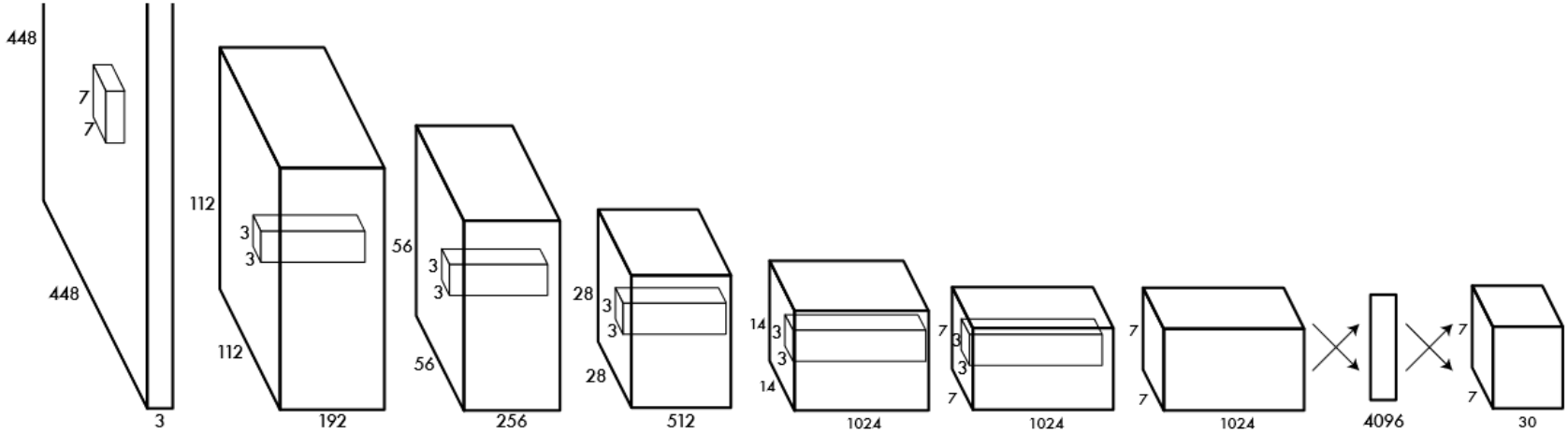


B = 2





YOLO - Network Architecture



Training the YOLO model

Step 1: Pre-Training

- Dataset: ImageNet 2012- 1000 class dataset.
- Network: 20-layer convolutional layer network followed by average-pooling layer followed by a fully connected layer.
- Accuracy: achieved 88% on the validation set.

Step 2: Training

Pascal VOC 2012 Dataset

20 classes. The train/val data has 11,530 images containing 27,450 ROI annotated objects and 6,929 segmentations

Loss Function

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{i=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2$$

Learning Rate

- Epoch 1-75: 10^{-2}
- Epoch 76-105: 10^{-3}
- Epoch 106-135: 10^{-4}

Hyperparameters/Parameters

No. of Epochs	135
Batch Size	64
Decay	0.0005
Momentum	0.9
λ_{coord}	5
λ_{noobj}	0.5

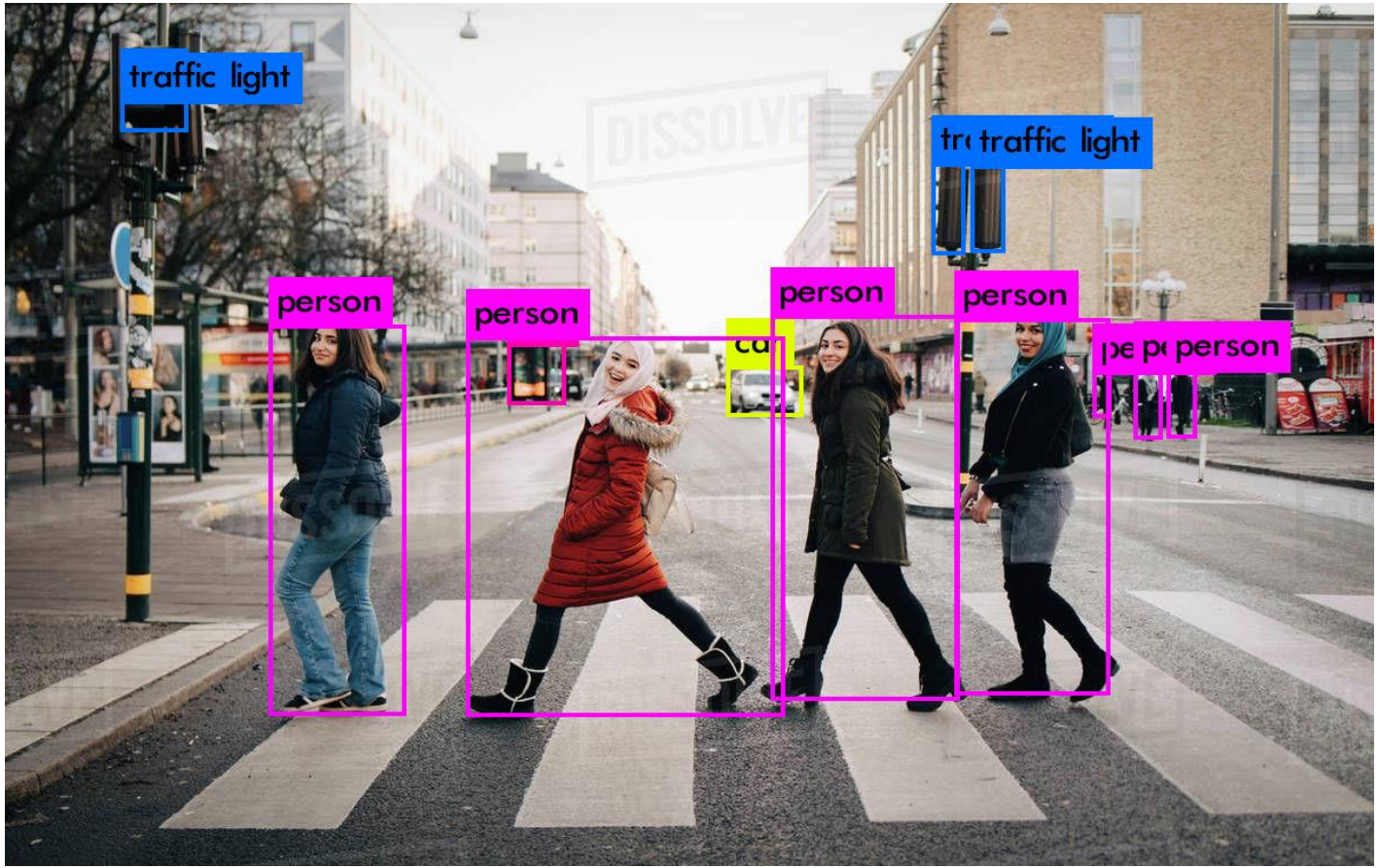
Code Explanation

Results

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [31]	2007	16.0	100
30Hz DPM [31]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [38]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[28]	2007+2012	73.2	7
Faster R-CNN ZF [28]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

Table 1: Real-Time Systems on PASCAL VOC 2007. Comparing the performance and speed of fast detectors. Fast YOLO is the fastest detector on record for PASCAL VOC detection and is still twice as accurate as any other real-time detector. YOLO is 10 mAP more accurate than the fast version while still well above real-time in speed.

Results


traffic light
tr traffic light
person
person
person
person
pe p person

Limitations:

- YOLO has spatial limits on how close the objects can be. This affects performance in that YOLO may miss out on certain objects.
- YOLO performance
- A small error in a large box generally doesn’t affect accuracy a lot but even a small error in a small box has a much greater effect on “intersection over union”. The main source of error is incorrect localizations.

Conclusion

- YOLO is an efficient model for object detection which is suitable for many real time applications and newer versions of YOLO improve the accuracy

References:

- https://arxiv.org/pdf/1506.02640v5.pdf
- https://pjreddie.com/darknet/yolo/
- https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006
- https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088
- https://medium.com/oracledevs/final-layers-and-loss-functions-of-single-stage-detectors-part-1-4abbfa9aa71c
- https://towardsdatascience.com/yolo-object-detection-in-matlab-start-to-finish-3f78ec80419d
- http://host.robots.ox.ac.uk/pascal/VOC/voc2012/VOCtrainval_11-May-2012.tar

Thanks!