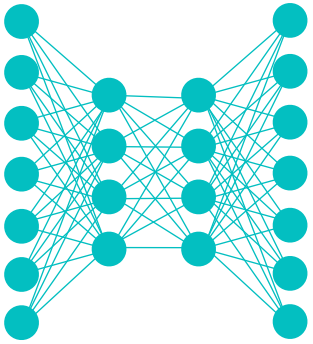
Lecture Notes for

Neural Networks and Machine Learning

Fully Convolutional Learning II: Object Detection



Object Detection with YOLO

- e researchers started working on a related, but different approach

Don’t want to listen to me explain it?

- Check out Joseph Redmon’s Talk at CVPR 2016:
  - https://www.youtube.com/watch?v=NM6lrxy0bxs
  - This is how you give a Technical presentation, mastery of presentation with technical depth

2018: Everybody has a Gimmick

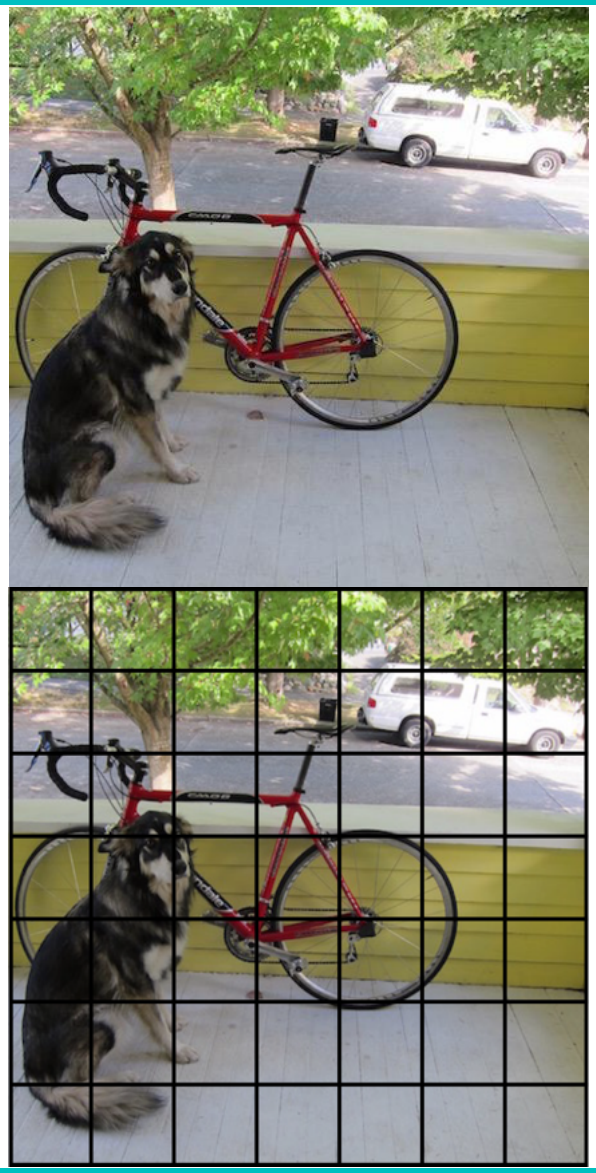
YOLO9000:
Better, Faster, Stronger

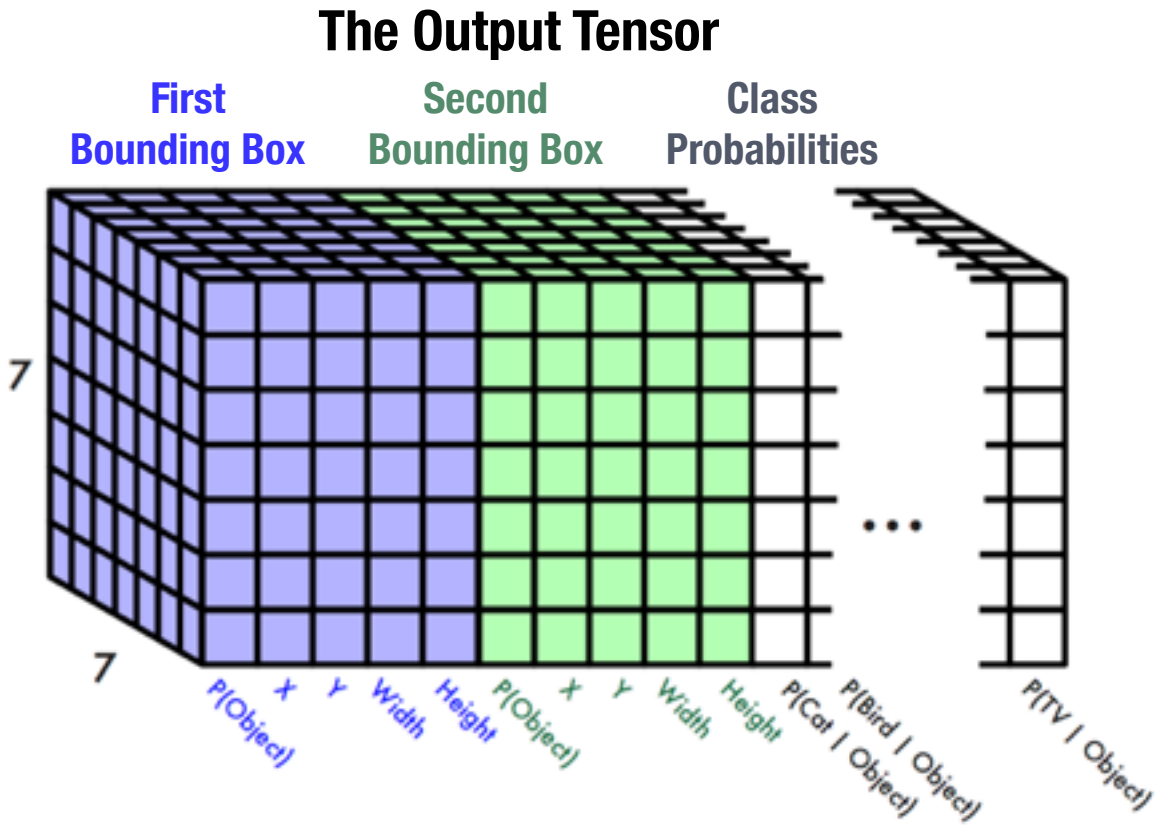
Joseph Redmon*†, Ali Farhadi*†
University of Washington*, Allen Institute for AI†
http://pjreddie.com/yolo9000/

- YOLO ~40-60 FPS
- Slightly more Accurate than **Faster R-CNN**

Redmon and Farhadi, YOLO9000: Better, Faster, Stronger, 2016, December 25 — *Merry Christmas?*

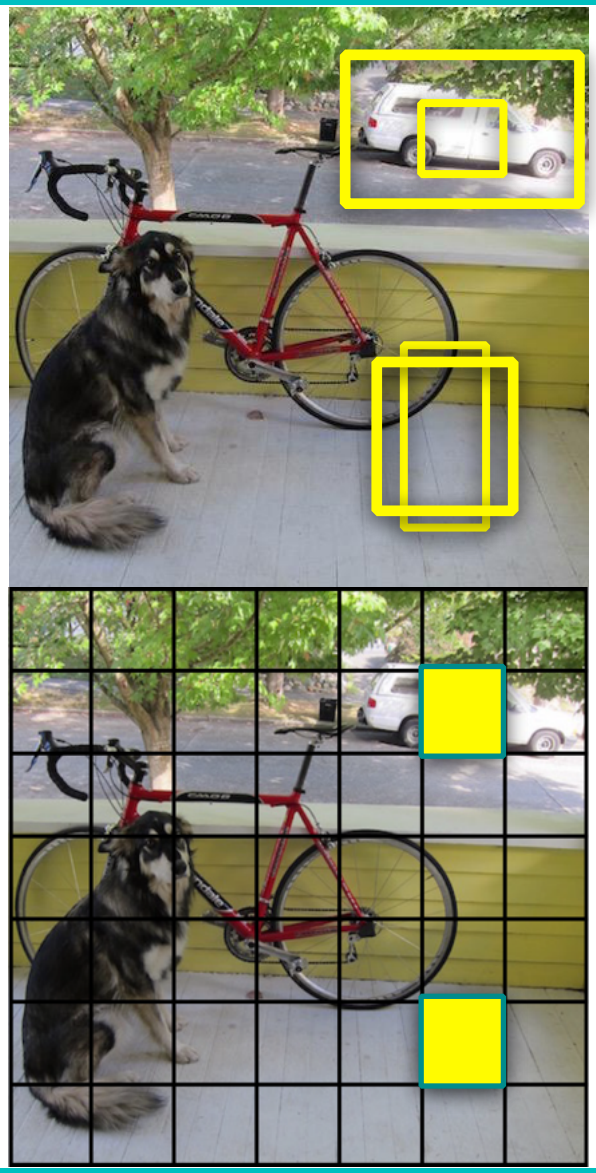
The YOLO Output Tensor

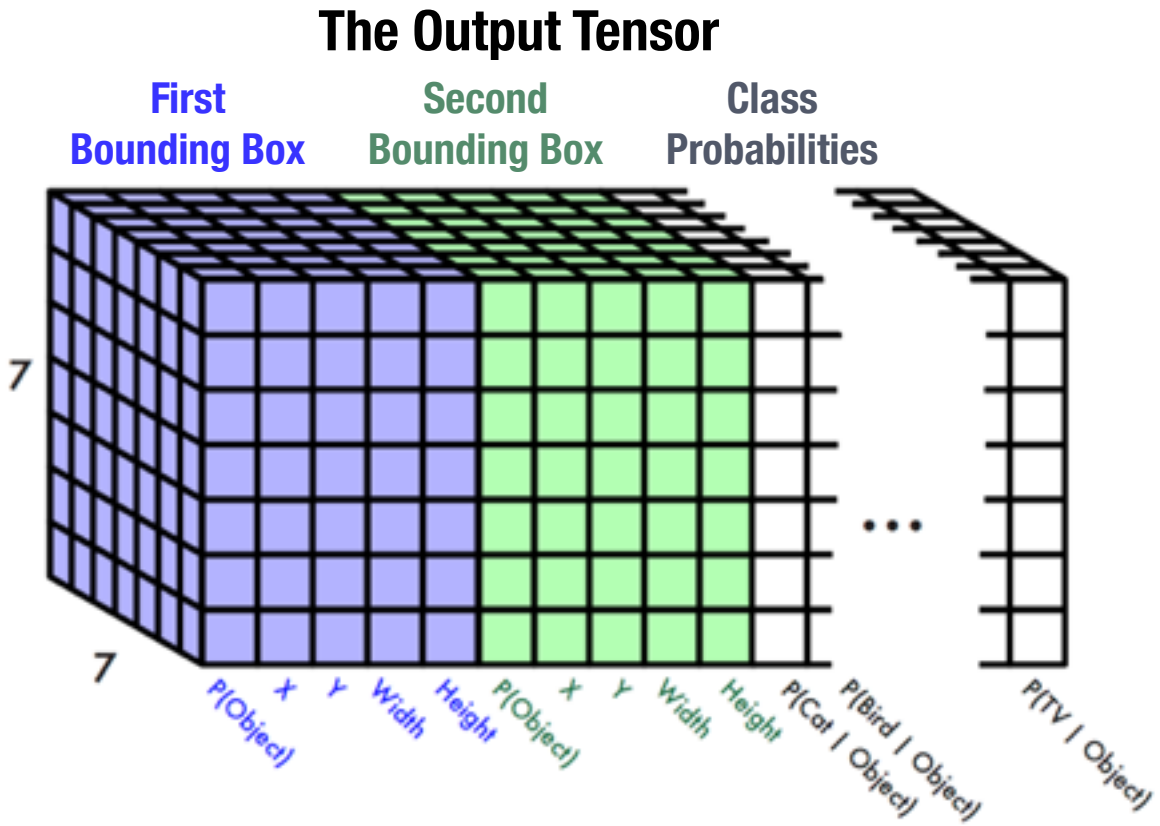




Redmon and Farhadi, YOLO9000: Better, Faster, Stronger, 2016, December 25 — *Merry Christmas?*

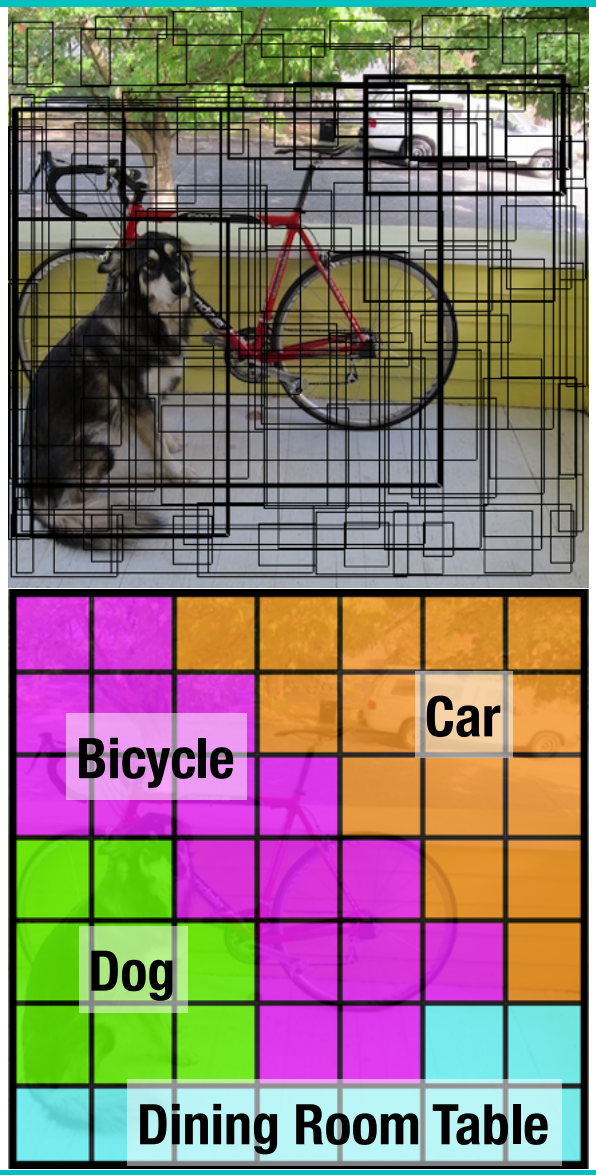
The YOLO Output Tensor

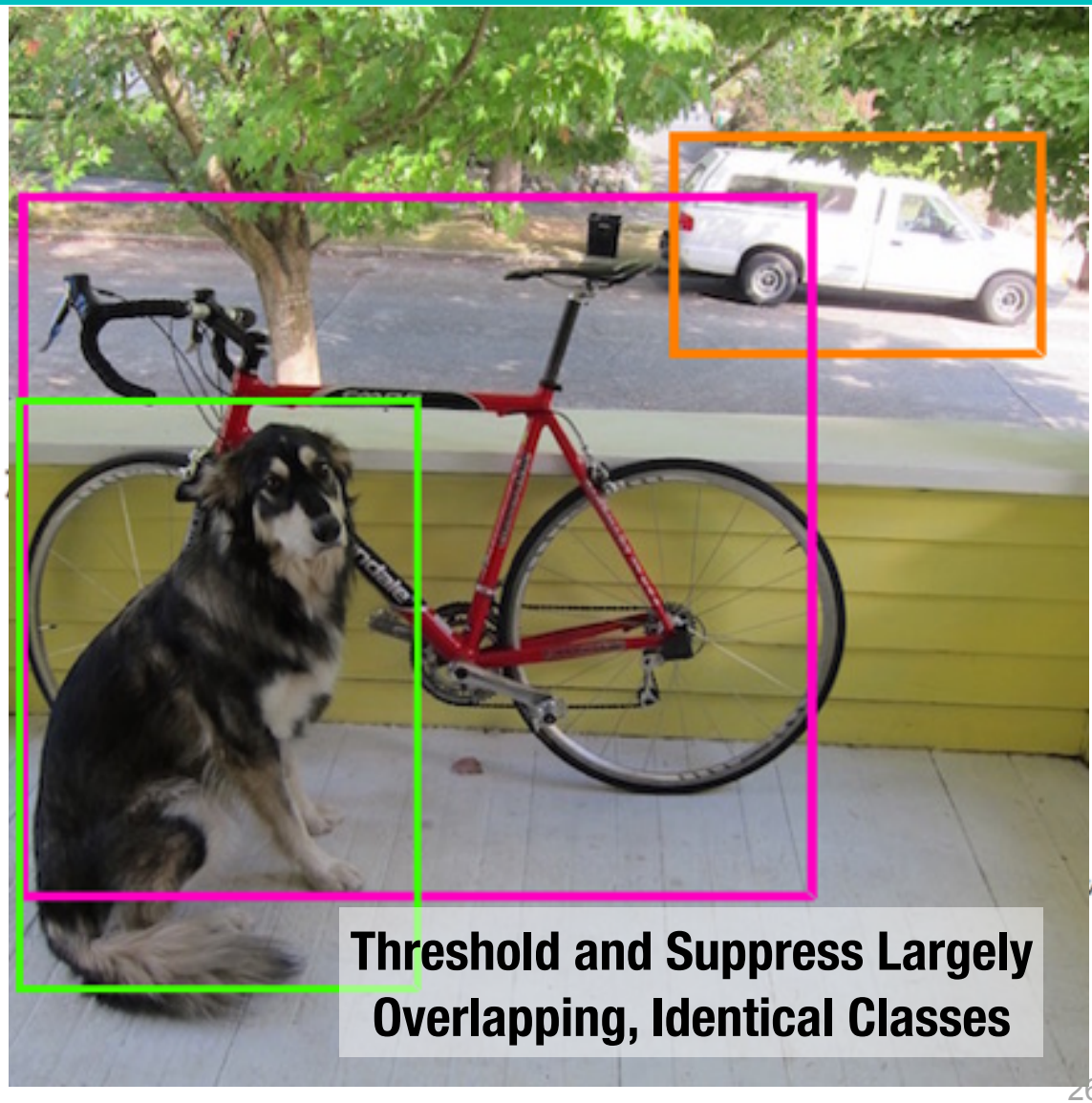




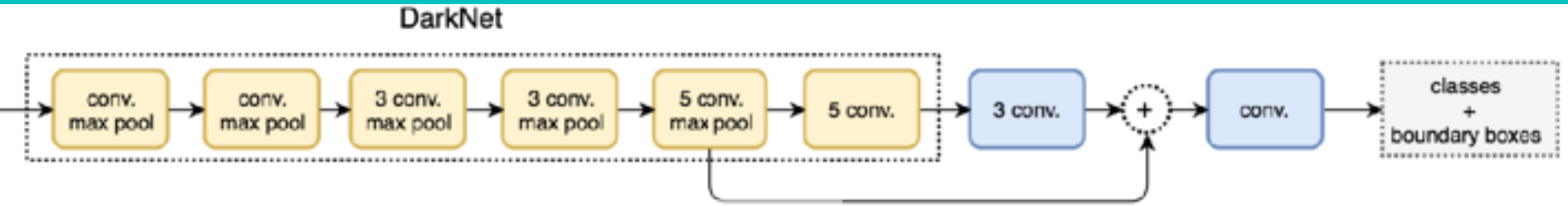
Redmon and Farhadi, YOLO9000: Better, Faster, Stronger, 2016, December 25 — *Merry Christmas?*

The YOLO Output Tensor



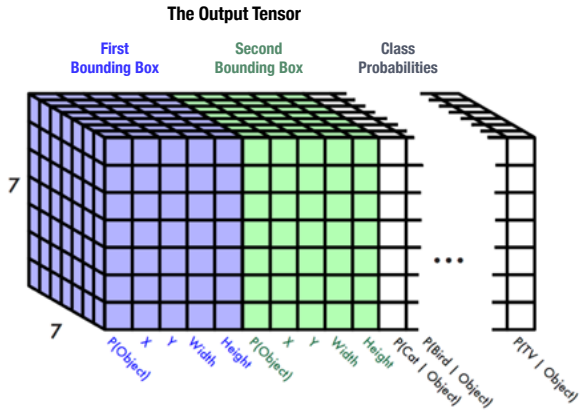


The YOLO Architecture



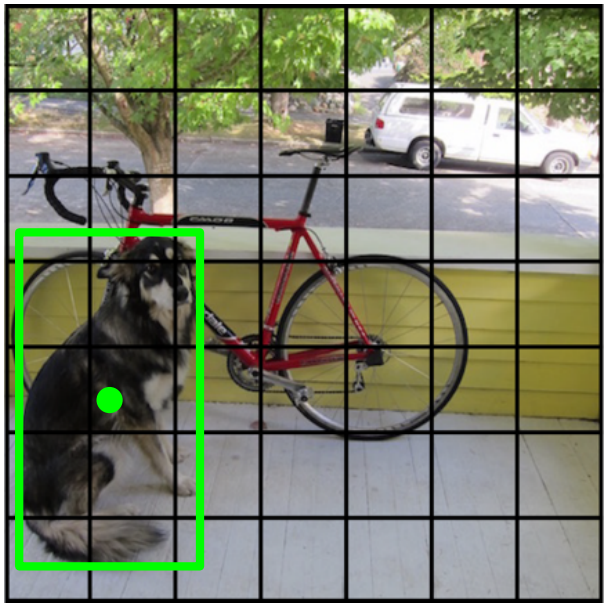
Trained from Traditional Image Dataset.
Architecture usually: DarkNet on ImageNet

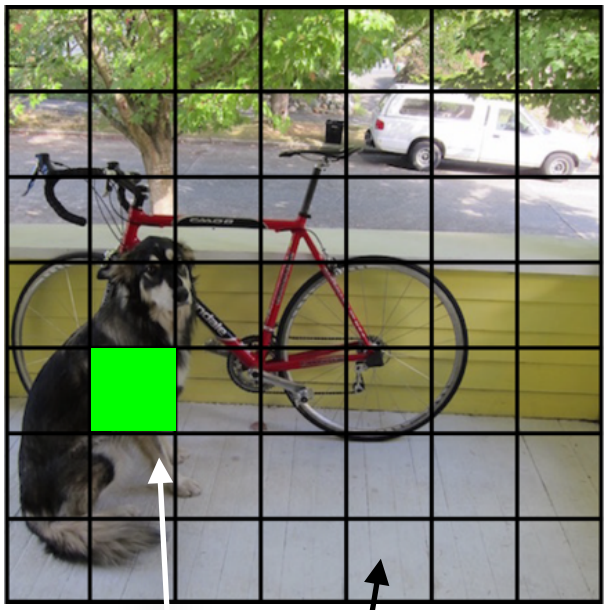
Last layers:
Trained on images
with bounding boxes

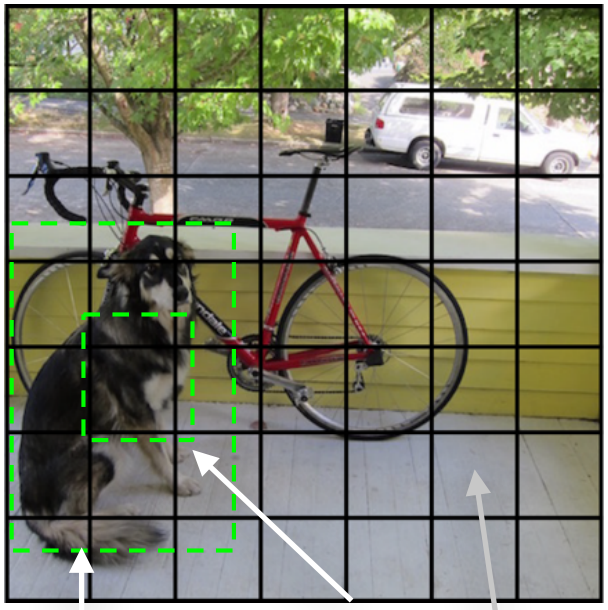


https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088

Training the YOLO Architecture







Dog = 1
Cat = 0
Bike = 0
etc.

Don't adjust class confidence if there is no object in GT

Closest to GT
Increase Conf.

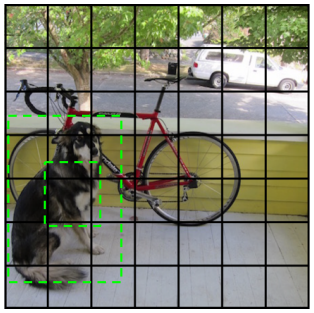
Farther From GT
Decrease Conf.

Any anchor box without object, decrease bounding box confidence decrease "objectness" score

Redmon and Farhadi, YOLO9000: Better, Faster, Stronger, 2016, December 25 — *Merry Christmas?*

The YOLO Loss Function

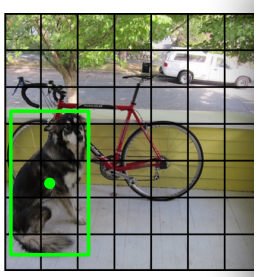
Update Bounding Box



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

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

Localization Loss



SxS cells, i^{th} cell

B boxes per cell, jth box

$\mathbb{1}^{\text{obj}}$ indicator function, from GT

$\hat{C}$ is confidence per box

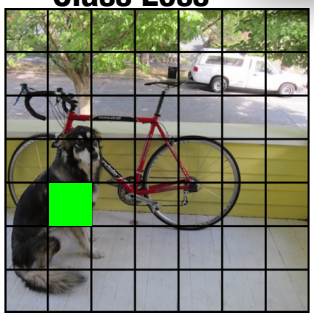
$\hat{p}(c)$ softmax output, per class

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

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

Object Detection Loss

Class Loss



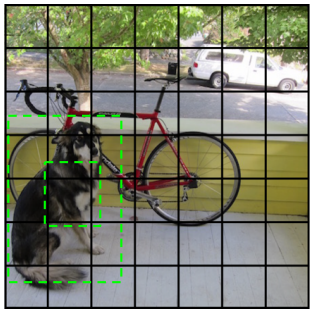
$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2$$

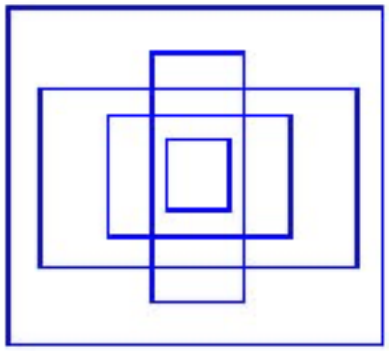
Classification Loss

https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088

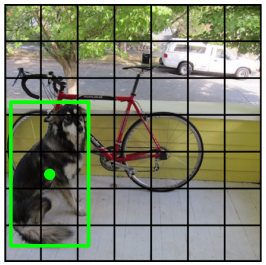
Updated YOLO Localization (v4)

Update Bounding Box

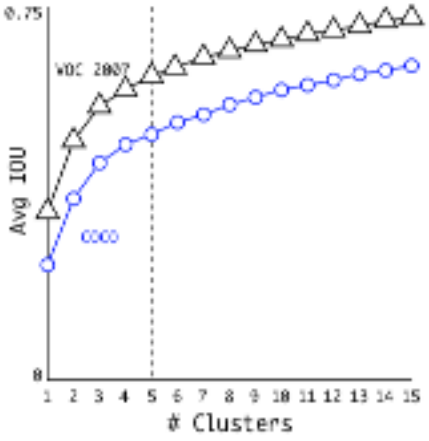


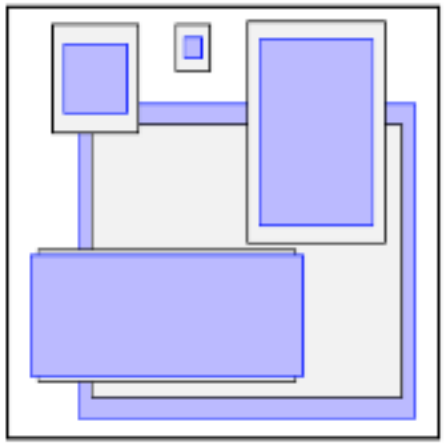


- Define 5 pre-defined box shapes (based on data)
- Regress x, y offset from cell center
- Bound x and y to the bounds of the cell
- And s scaling param of predefined shape

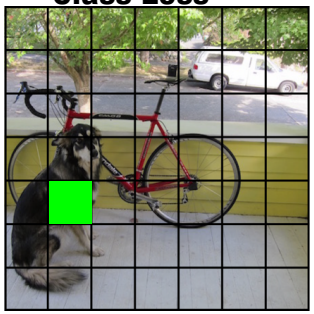


"Good" Shape Priors Found via Clustering





Class Loss



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

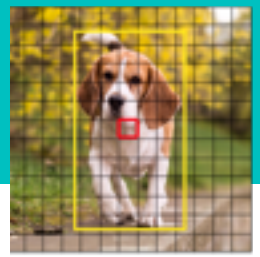
$$+\lambda_{\text{noobj}}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{\text{noobj}}\left(C_i-\hat{C}_{ij}\right)^2$$

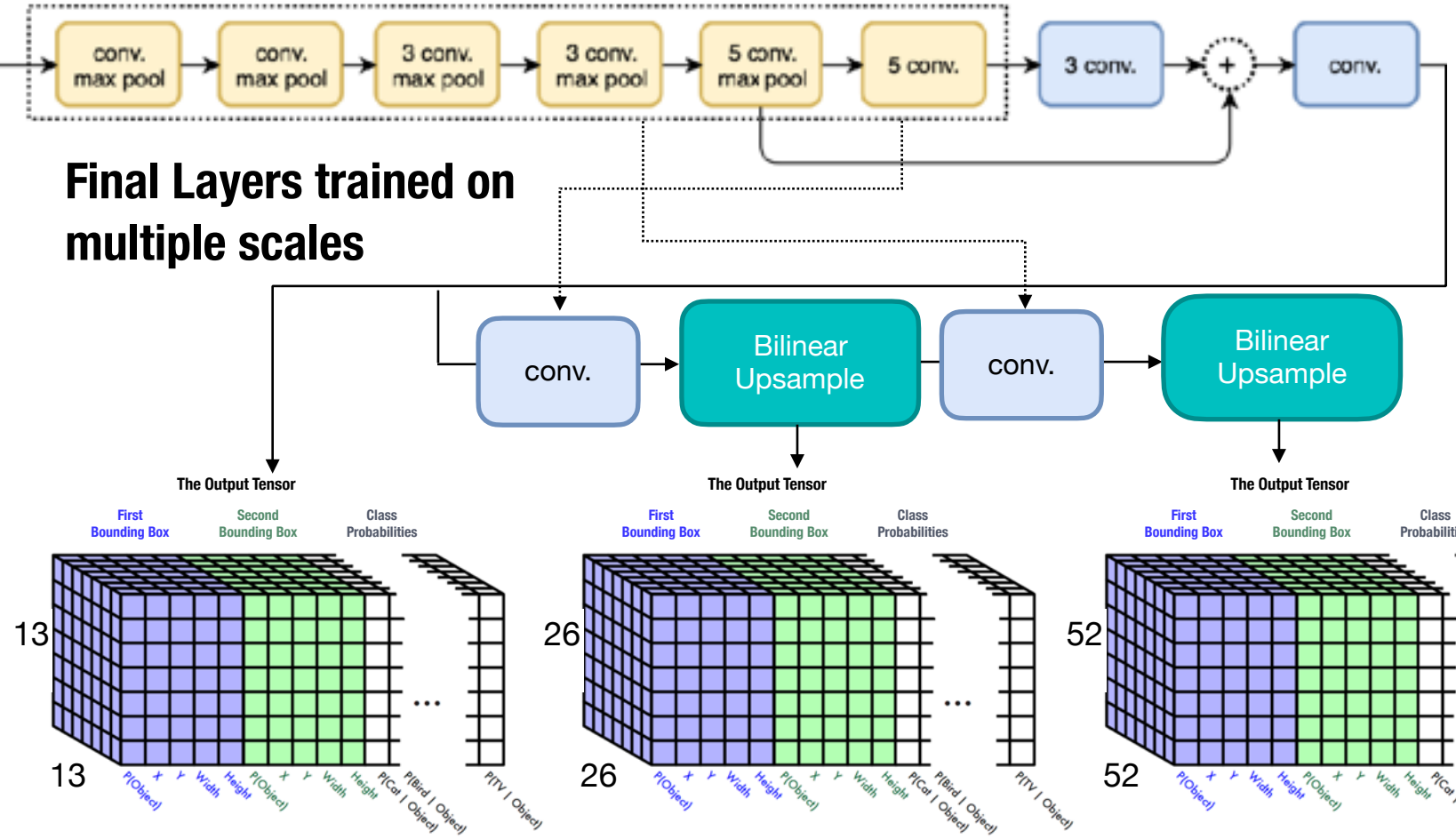
$$+\sum_{i=0}^{S^2}\mathbb{1}_{i}^{\text{obj}}\sum_{c\in\text{classes}}\left(p_i(c)-\hat{p}_i(c)\right)^2$$

Classification Loss

https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088

The YOLOv3 Architecture





All Losses Applied, 13x13 grid

All Losses Applied, 26x26 grid

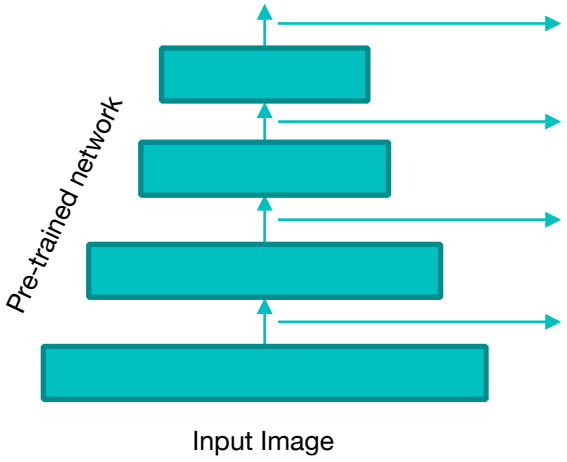
All Losses Applied, 52x52 grid

https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088

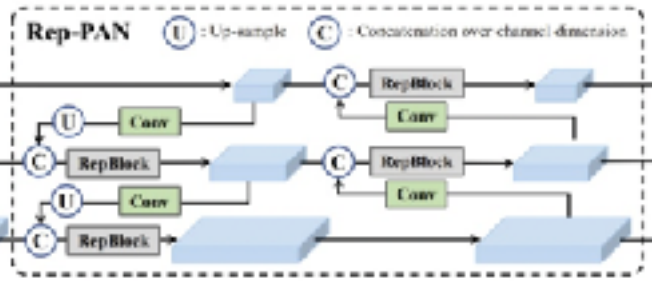
More YOLO Variants

YOLO Structures, In General

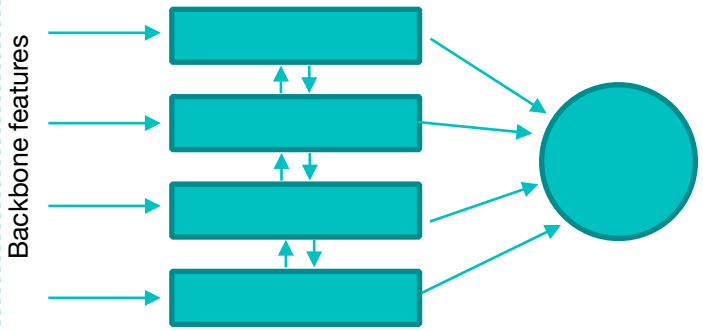
Backbone



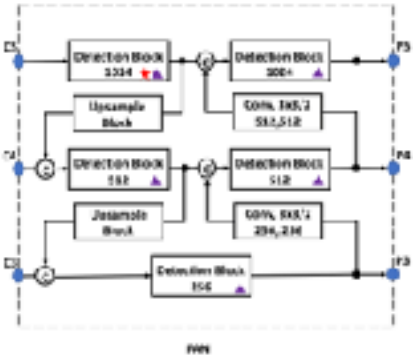
- YOLOv1-v4: Dark/ResNet
- Yv5: EfficientNet
- Yv6: EfficientNet-L2
- PP-Yv1-v2: ResNet50-vd
- Yv6 v3.0: RepVGG 🙄

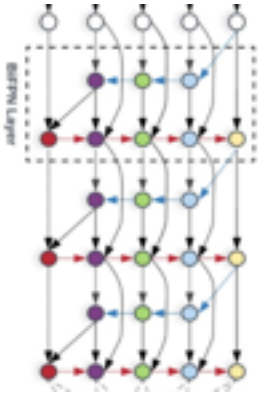


Neck

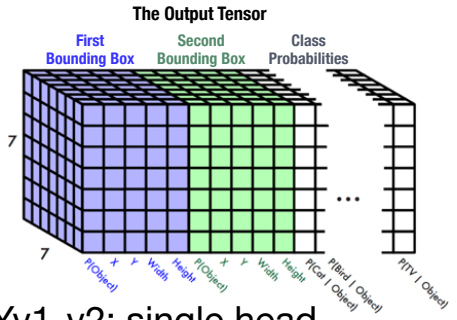


- Yv1-v2: Transfer Learning, Single Layer
- Yv3: Upsampling, Multiple Layers
- Yv4: Cross Stage Partial layers (CSPNet). Use multiple layers from backbone with weighted concat
- PP-Y: Path Aggregation Network
- PP-Yv2: Multi-scale PAN

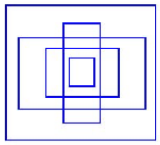




Head(s)

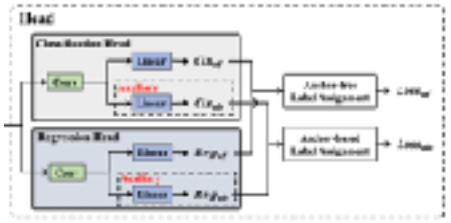


- Yv1-v2: single head
- Yv3: Multi-resolution heads
- Yv4: Clustered anchor boxes and new gradient penalty
- PP-Yv2: IoU prediction loss
- Yv6-v8: Dense Anchor Boxes (anchors as aux. task)
- Yv5-8: Mosaic Augmentation

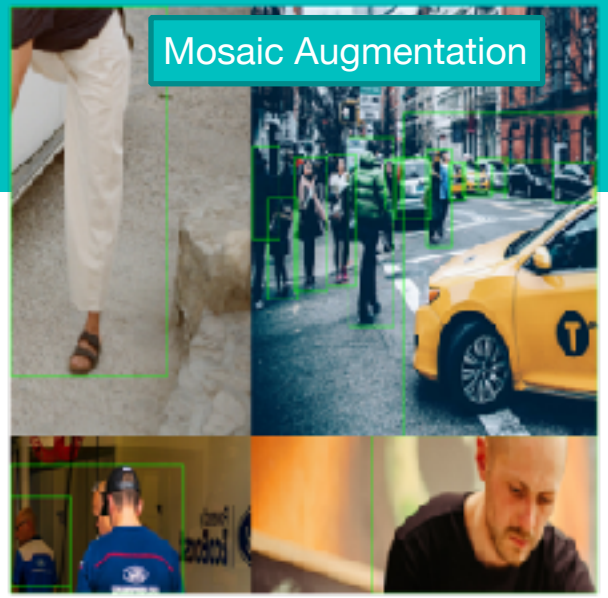


$$\mathcal{L}_{IoU} = \begin{matrix} -t_{IoU} \cdot \log(\sigma(p)) + \\ (t_{IoU} - 1) \cdot \log(1 - \sigma(p)) \end{matrix}$$

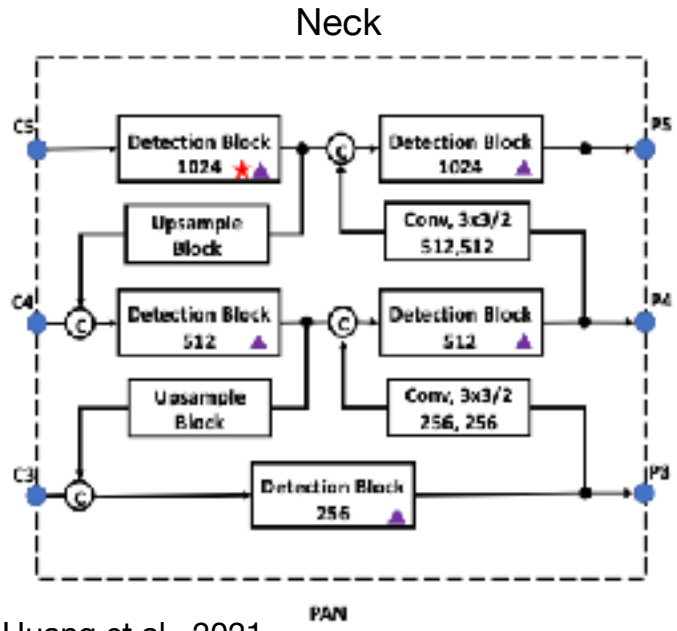
IoU Prediction



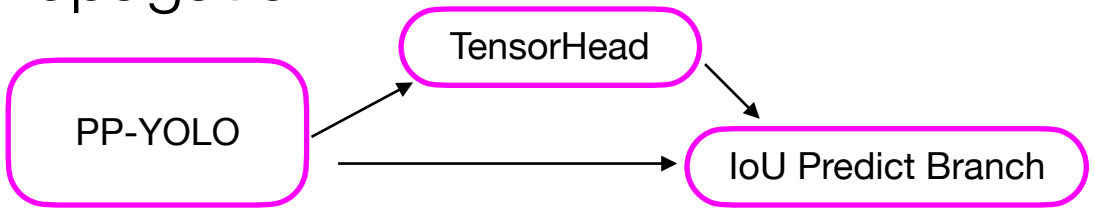
One Example: PP-YOLOv2



- Path Aggregation Network (PAN) for multi-scale processing
- Varied Input Sized Images and "Mish" activation
- "IoU Aware Branch" that uses the IoU prediction in the back propagation



Huang et al., 2021



$$\mathcal{L}_{IoU} = -t_{IoU} \cdot \log(\sigma(p)) + (t_{IoU} - 1) \cdot \log(1 - \sigma(p))$$

IoU between output of YOLO BBox and GT

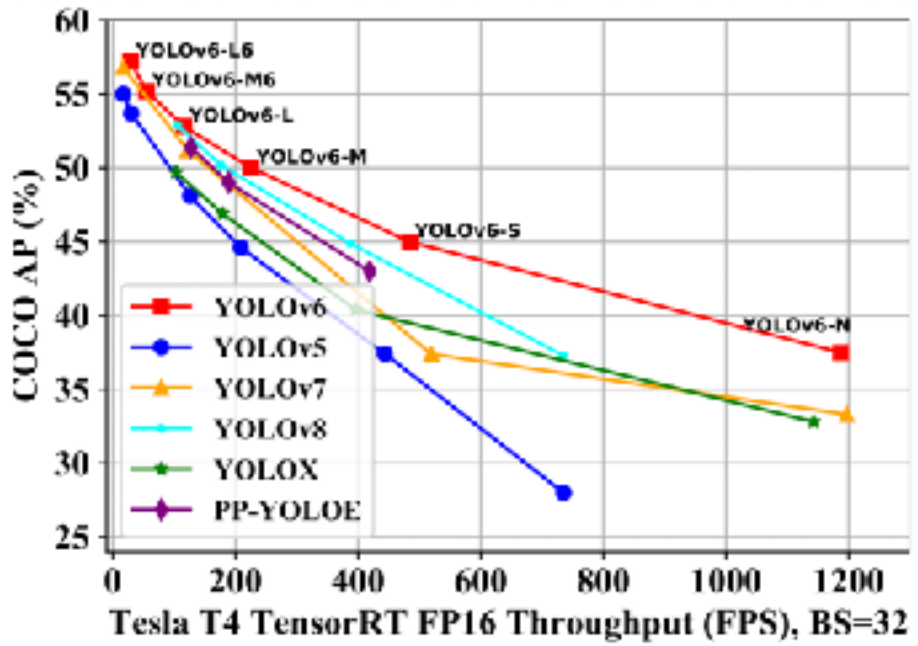
p=predicted IoU from branch

Main idea is to predict the IoU in a separate multi-task branch, which makes the network better at finding the correct bounding boxes

PP-YOLOv2: Results

YOLOv6 v3.0 + YOLOv8

YOLOv7



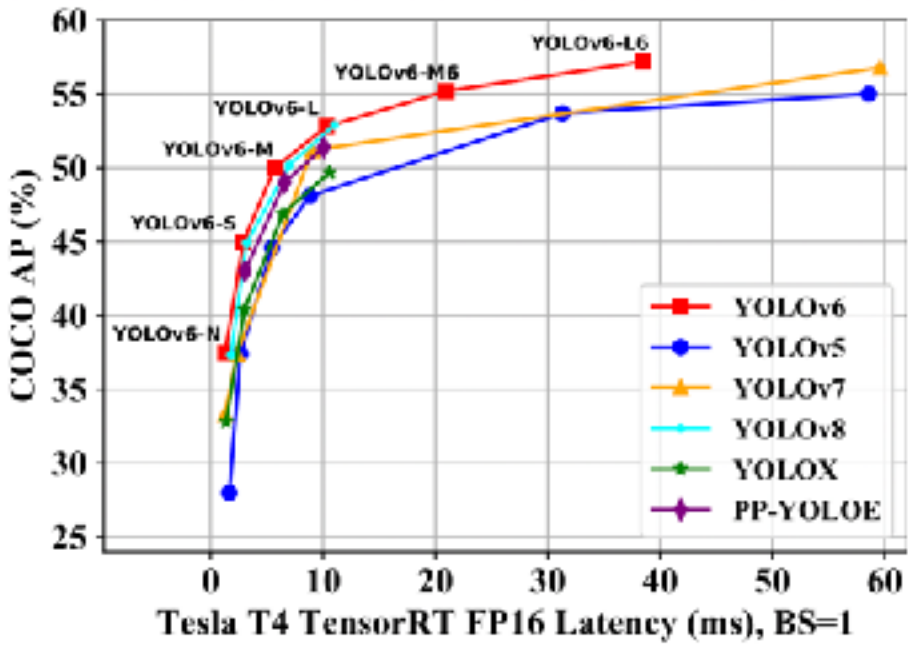


Figure 1: Comparison of state-of-the-art efficient object detectors. Both latency and throughput (at a batch size of 32) are given for a handy reference. All models are test with TensorRT 7.