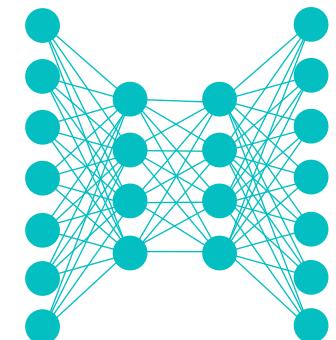


# Lecture Notes for **Neural Networks** **and Machine Learning**



Fully Convolutional Learning II:  
Object Detection



# Object Detection with YOLO



**Joseph Redmon** @pjreddie • Feb 20, 2020



"We shouldn't have to think about the societal impact of our work because it's hard and other people can do it for us" is a really bad argument.



**Roger Grosse** @RogerGrosse

Replying to @kevin\_zakkia and @hardmaru

To be clear, I don't think this is a positive step. Societal impacts of AI is a tough field, and there are researchers and organizations that study it professionally. Most authors do not have expertise in the area and won't do good enough scholarship to say something meaningful.



**Joseph Redmon**  
@pjreddie

I stopped doing CV research because I saw the impact my work was having. I loved the work but the military applications and privacy concerns eventually became impossible to ignore.



**Roger Grosse** @RogerGrosse

Replying to @skoularidou

What's an example of a situation where you think someone should decide not to submit their paper due to Broader Impacts reasons?

10:09 AM · Feb 20, 2020



related, but different approach

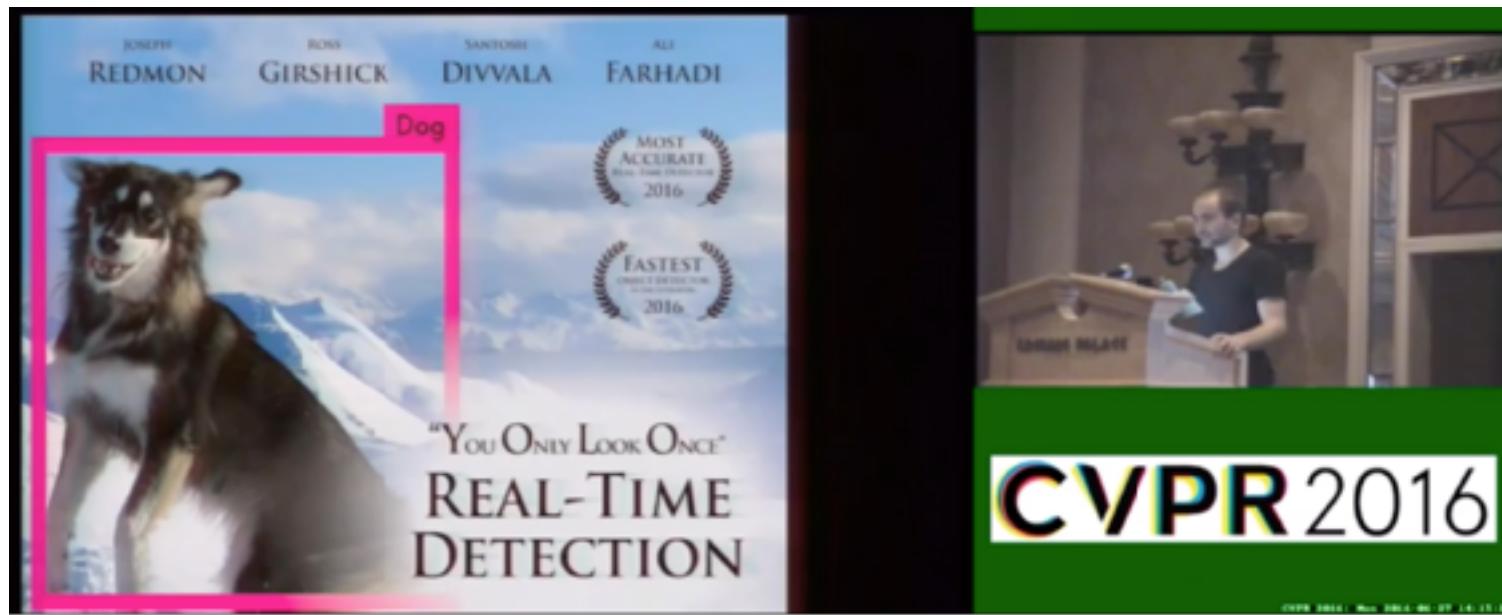


• e researchers started working on a



# 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, plus he is from the school where I did graduate school... so he is clearly superior by grace of proximity...



# 2018: Everybody has a Gimmick



## YOLO9000: Better, Faster, Stronger

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



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

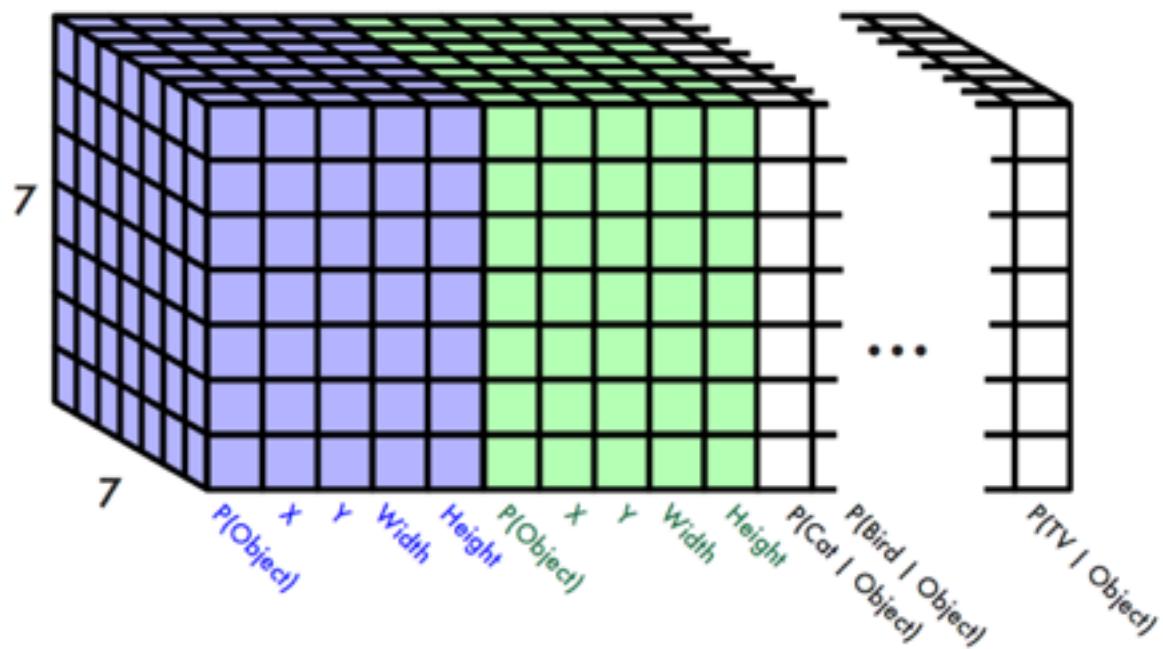


# The YOLO Output Tensor



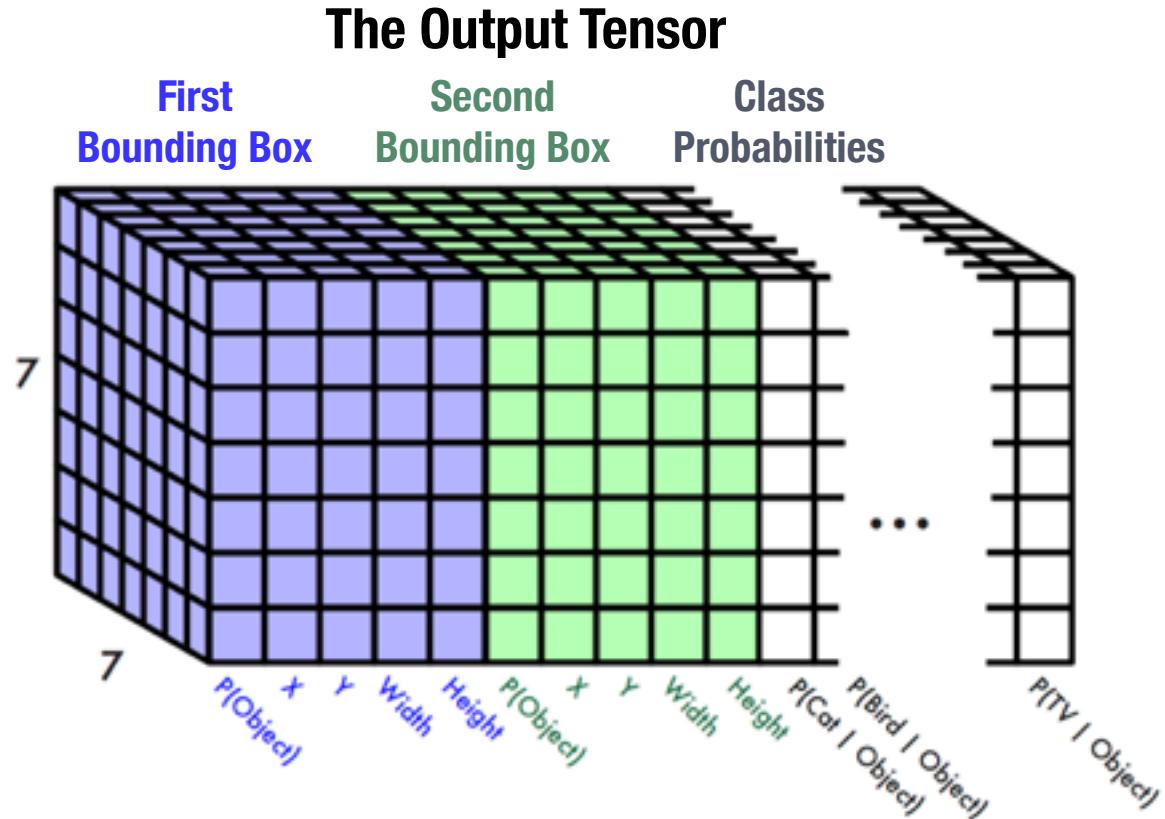
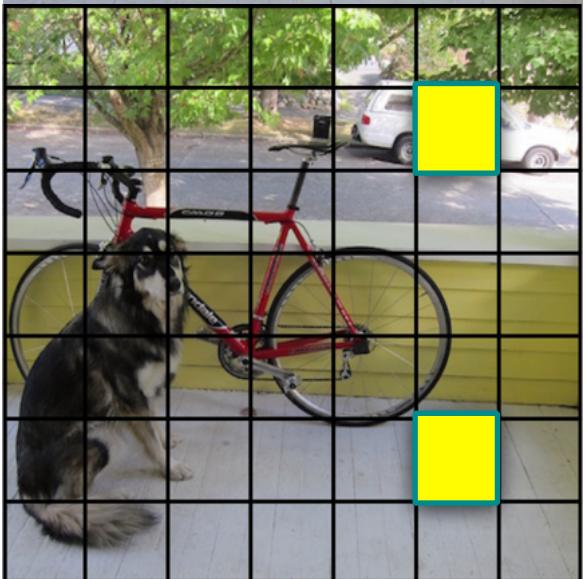
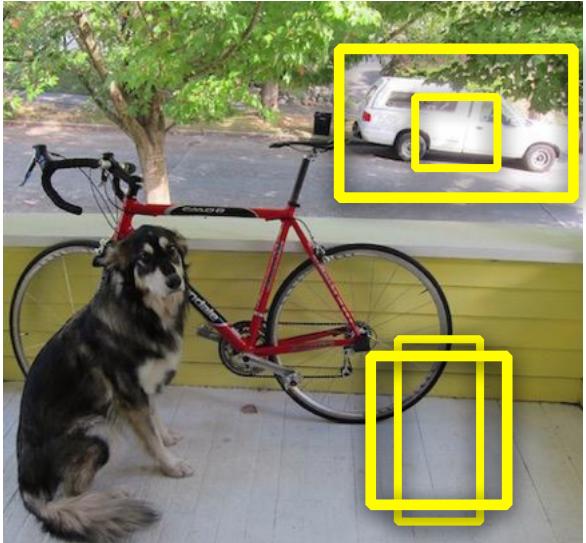
**The Output Tensor**

First Bounding Box      Second Bounding Box      Class Probabilities



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

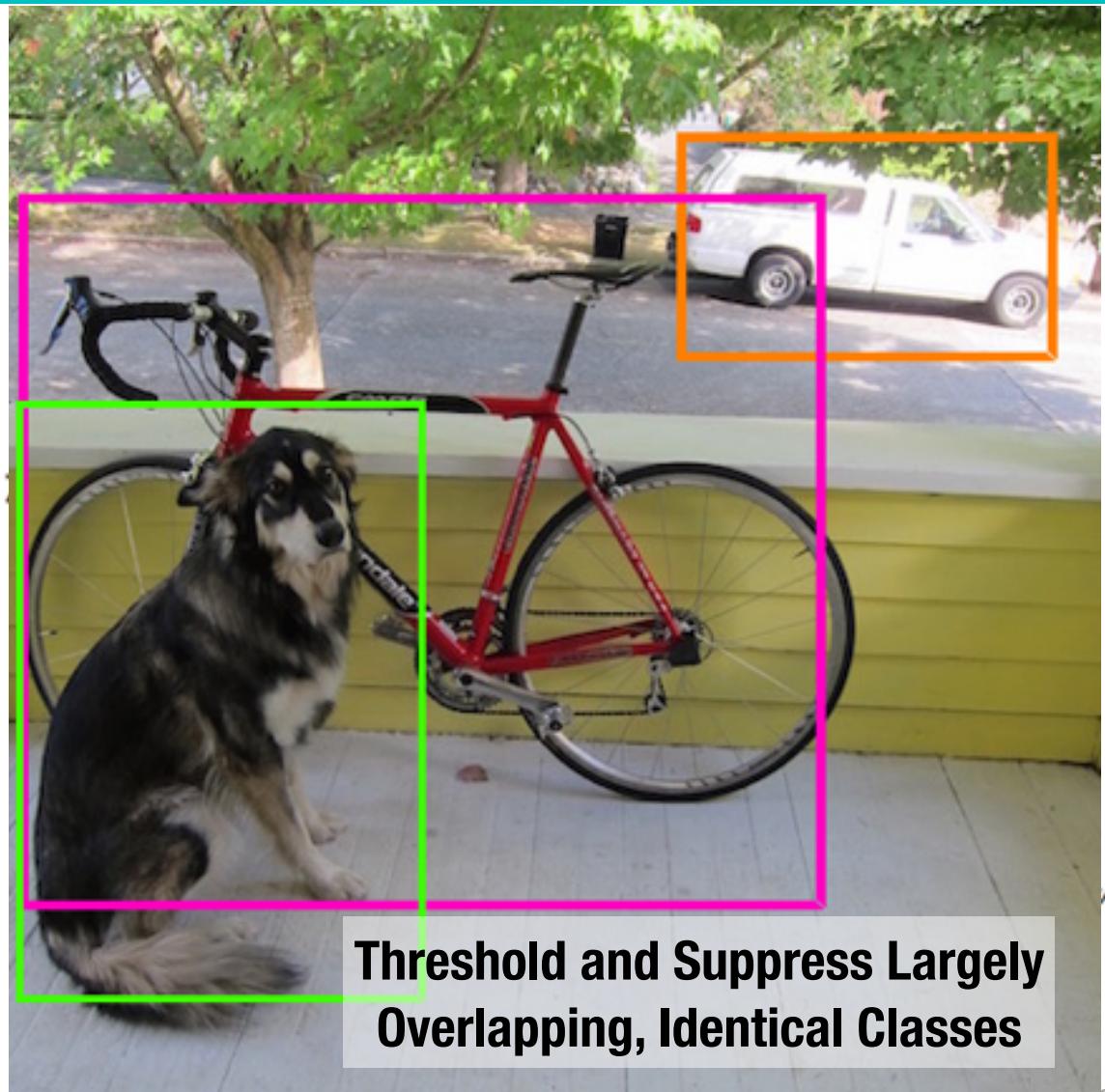
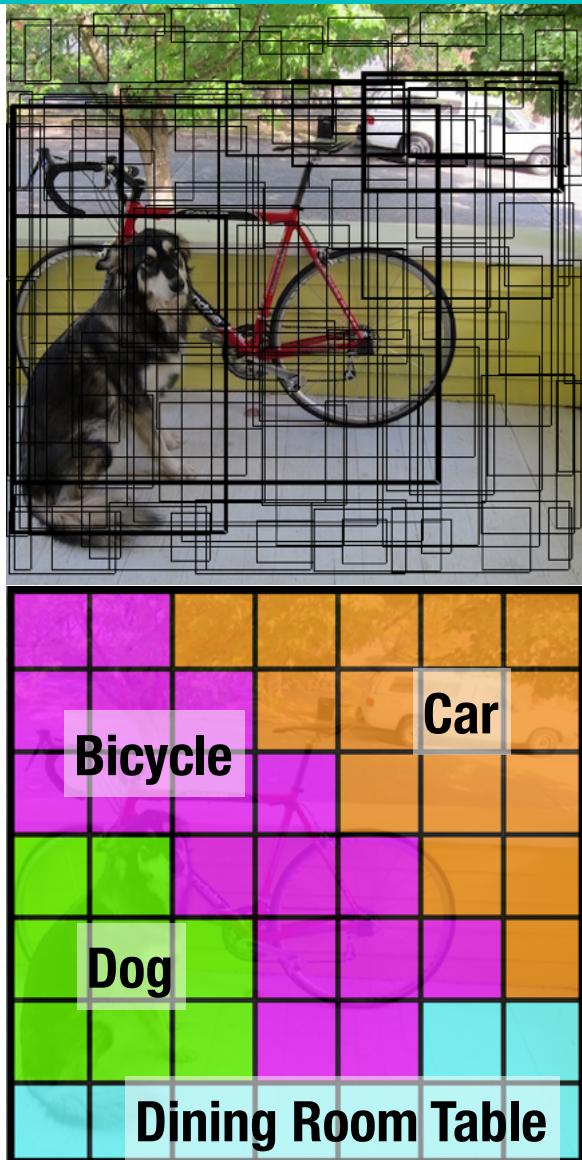
# The YOLO Output Tensor



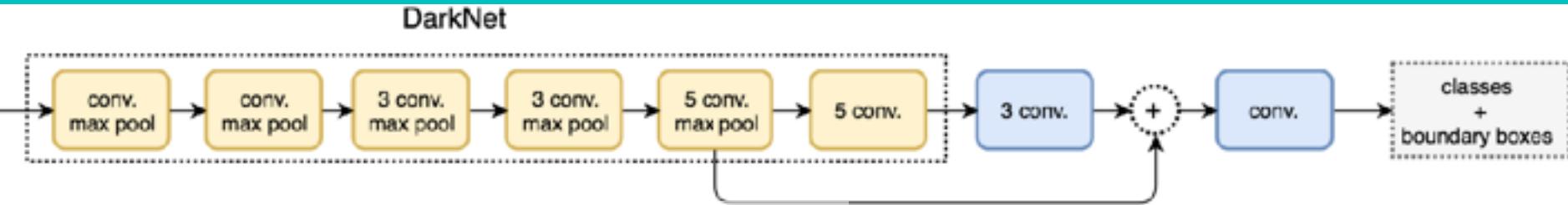
Redmon and Farhadi, YOLO9000: Better, Faster, Stronger, 2016,  
December 25 — Merry Christmas?



# The YOLO Output Tensor



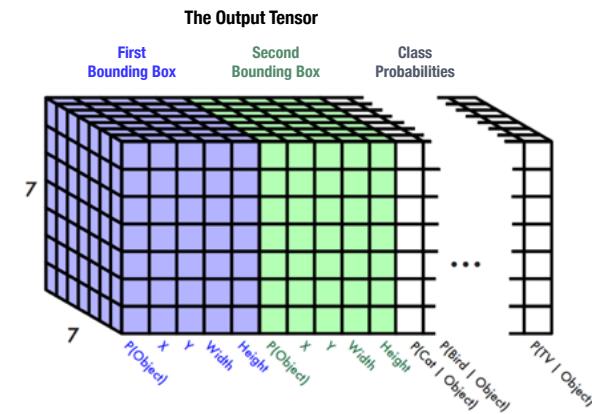
# The YOLO Architecture



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

	pjreddie guys one of my beehives died :-(
	guys one of my beehives died :-(
	SELU activation and yolo openimages
	GUYS I THINK MAYBE IT WAS BROKEN ON OPENCV IDK
	YO DAWG, I HEARD YOU LIKE LICENSES
	generate own license, totally legal :verified:

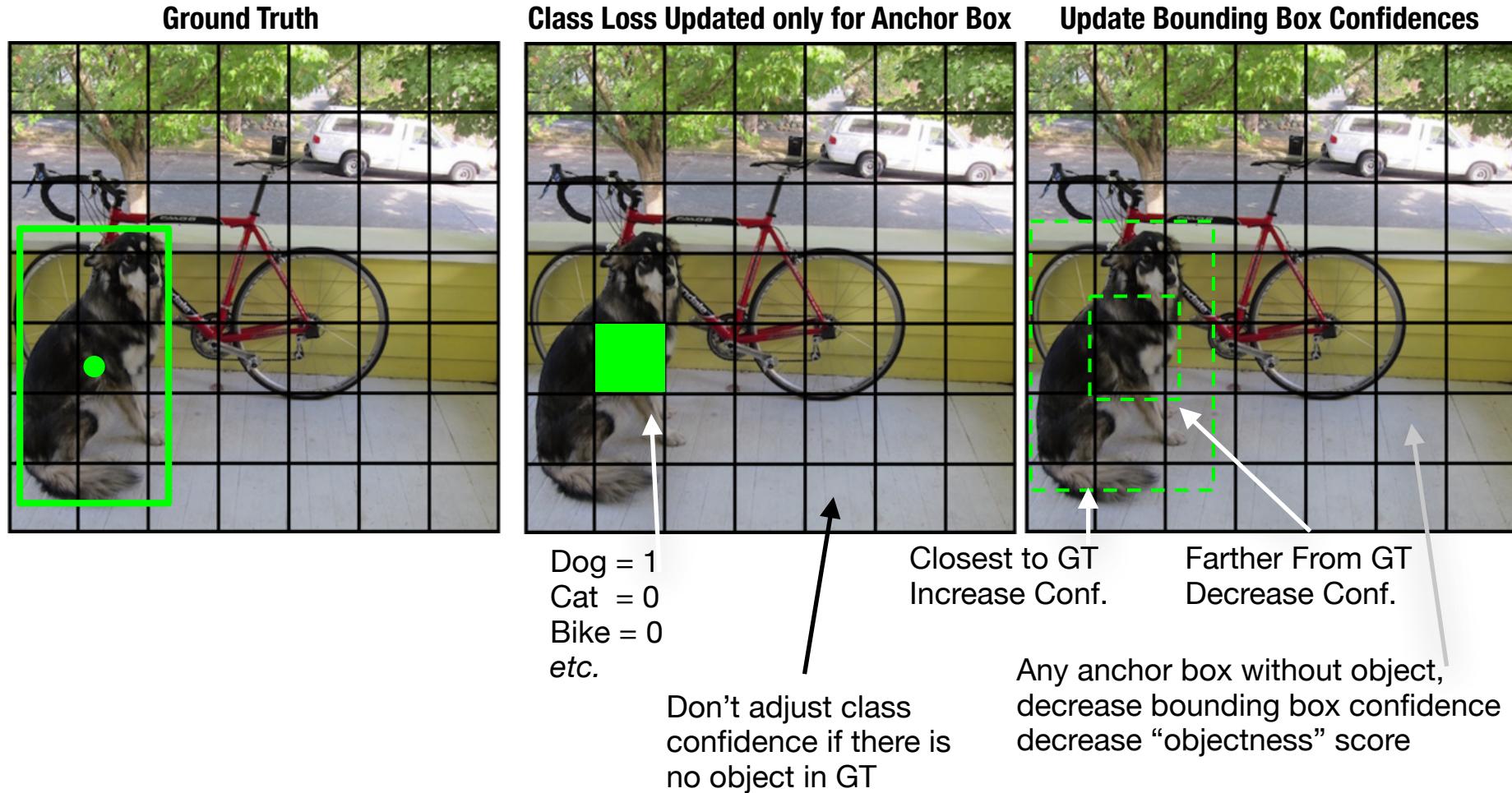
**Last layers:  
Trained on images  
with bounding boxes**



[https://medium.com/@jonathan\\_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088](https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088)

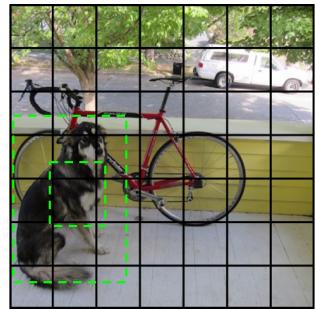


# Training the YOLO Architecture

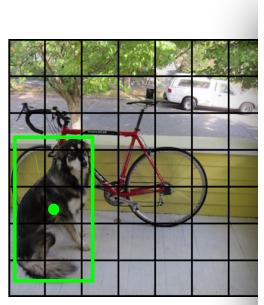


# The YOLO Loss Function

## Update Bounding Box



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



$S \times S$  cells,  $i^{\text{th}}$  cell

$B$  boxes per cell,  $j^{\text{th}}$  box

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

$\hat{C}$  is confidence per box

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

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

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

## Class Loss



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

[https://medium.com/@jonathan\\_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088](https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088)

Localization Loss

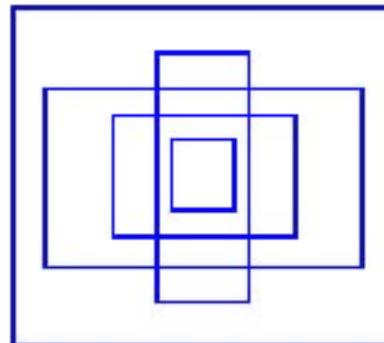
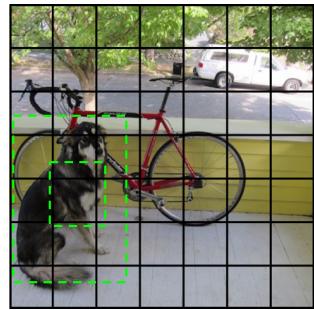
Object Detection Loss

Classification Loss

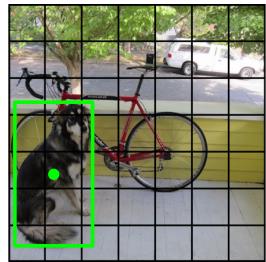


# Updated YOLO Localization (v4)

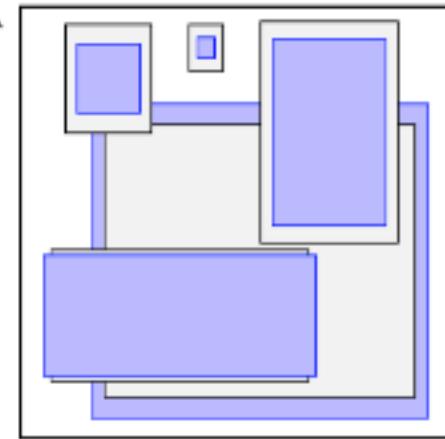
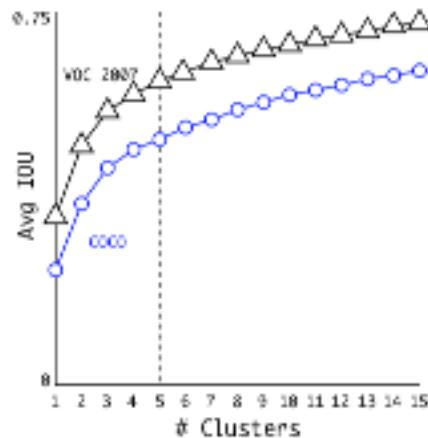
## Update Bounding Box



- 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



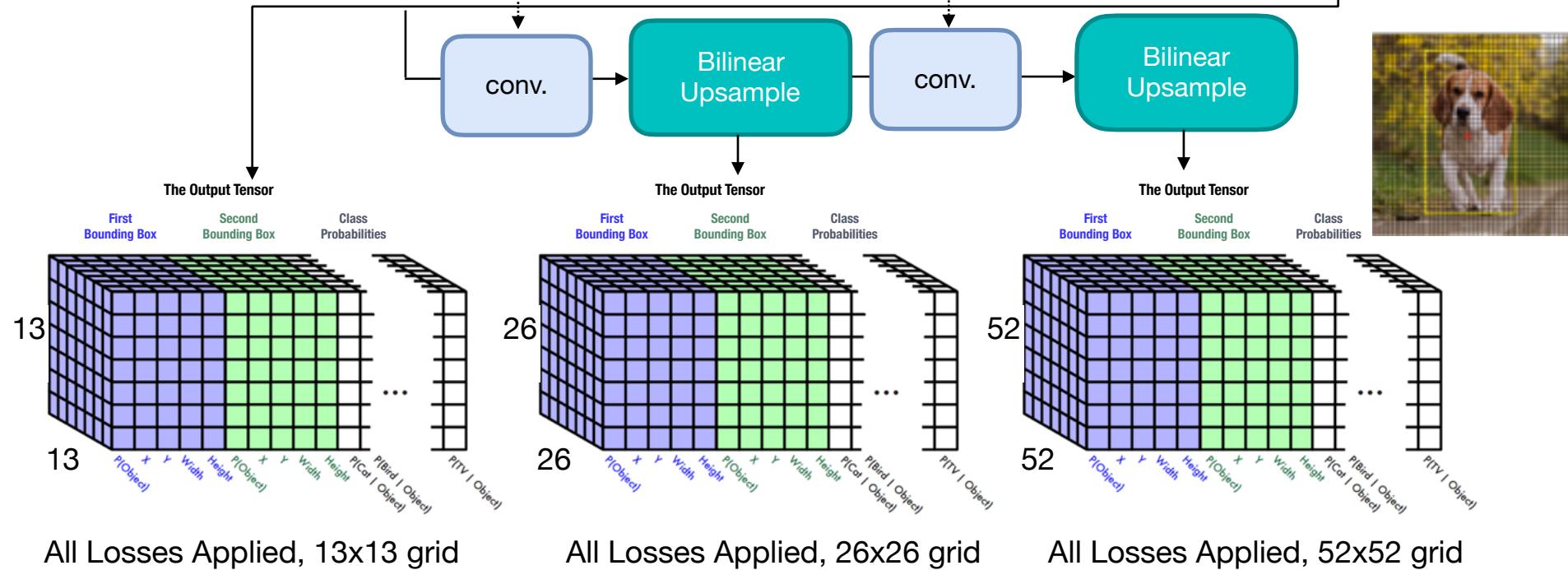
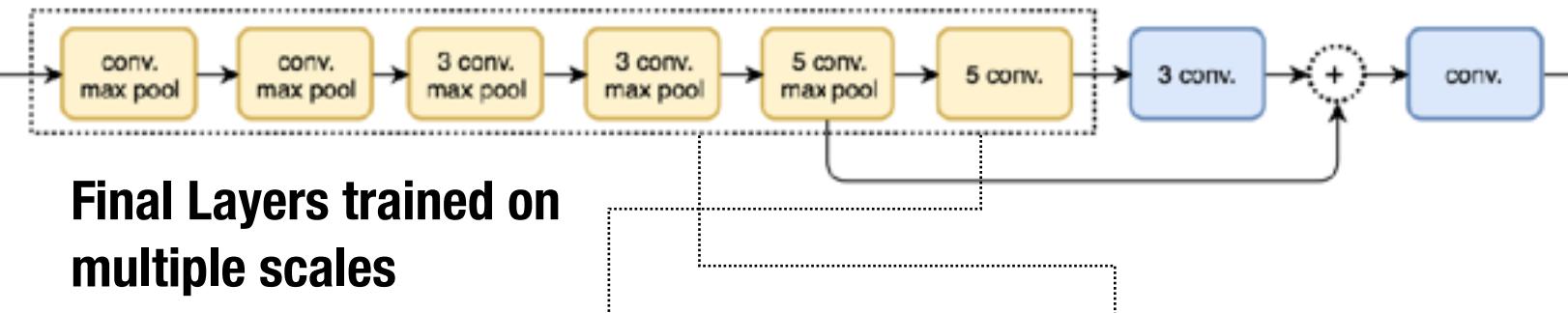
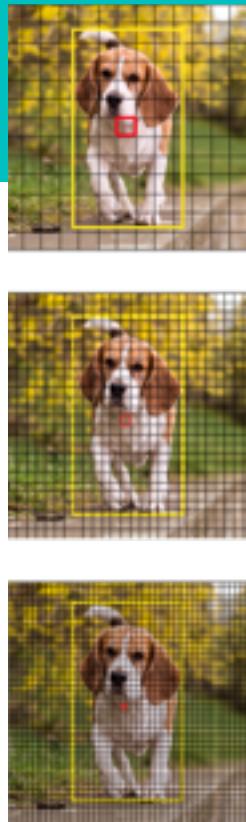
$$\begin{aligned} & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left( C_i - \hat{C}_{ij} \right)^2 \\ & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} \left( C_i - \hat{C}_{ij} \right)^2 + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} \left( p_i(c) - \hat{p}_i(c) \right)^2 \end{aligned}$$

[https://medium.com/@jonathan\\_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088](https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088)

Classification Loss



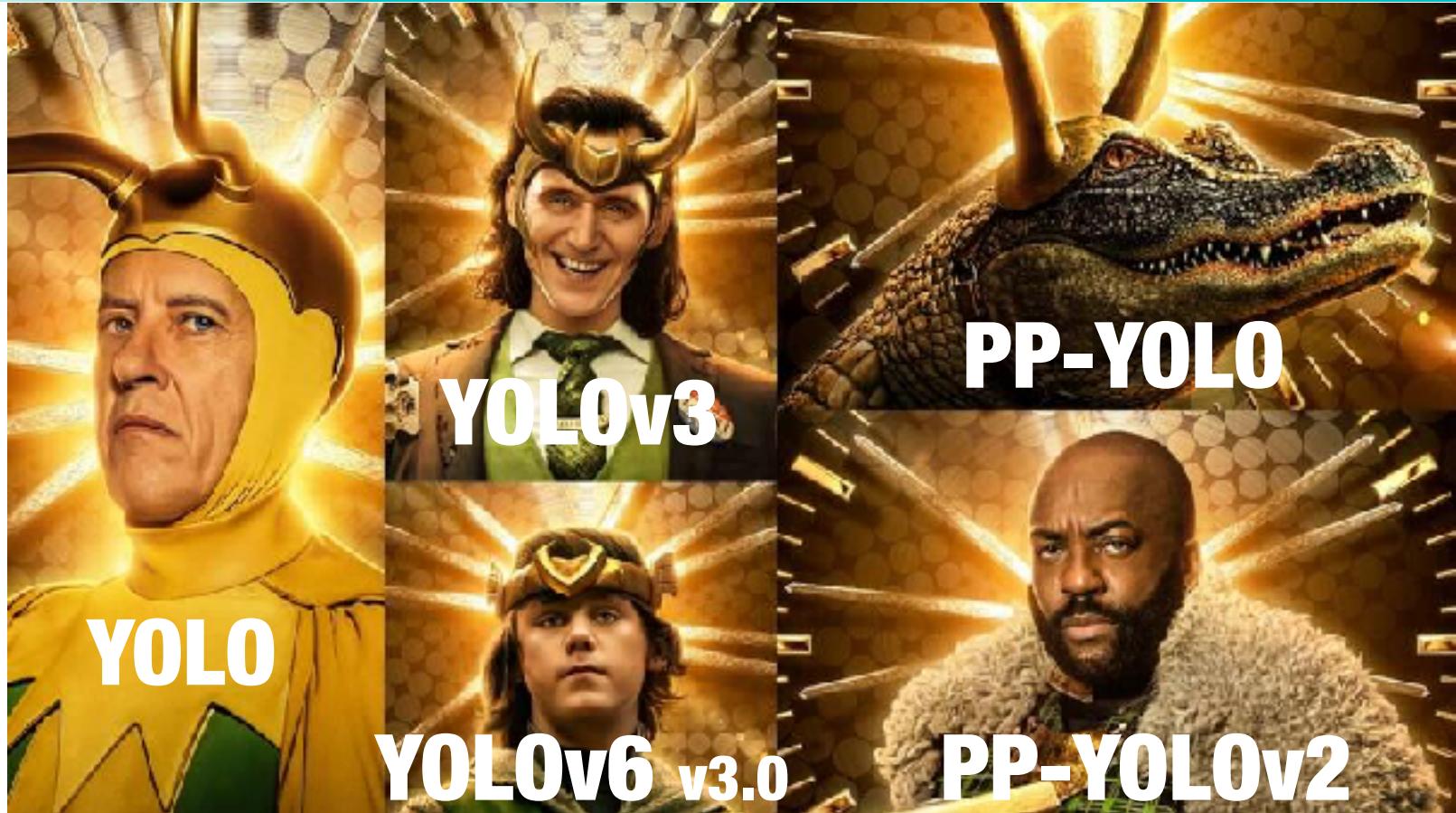
# The YOLOv3 Architecture



[https://medium.com/@jonathan\\_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088](https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088)

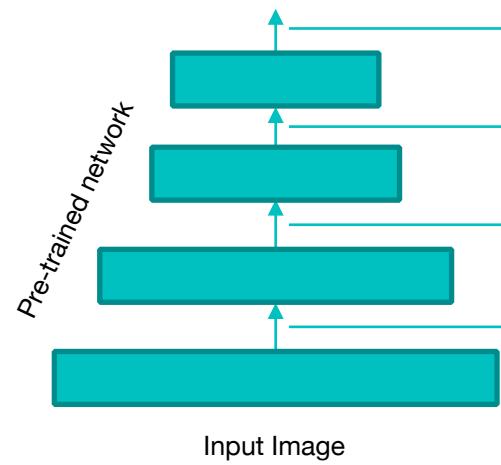


# More YOLO Variants

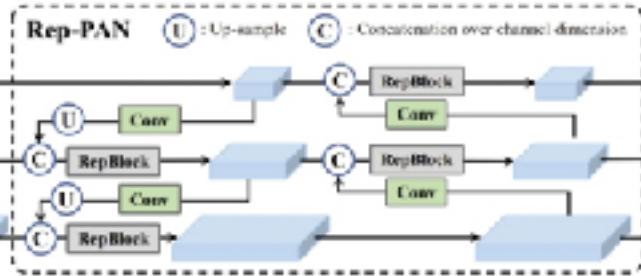


# YOLO Structures, In General

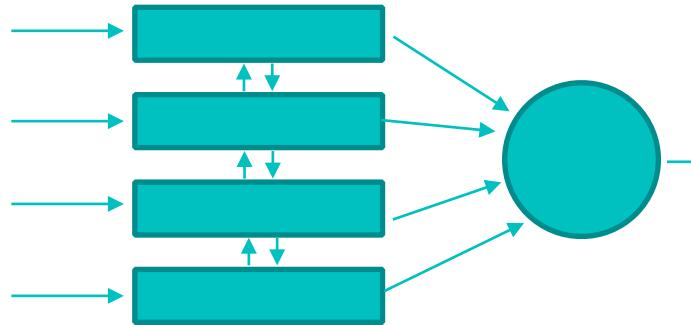
## Backbone



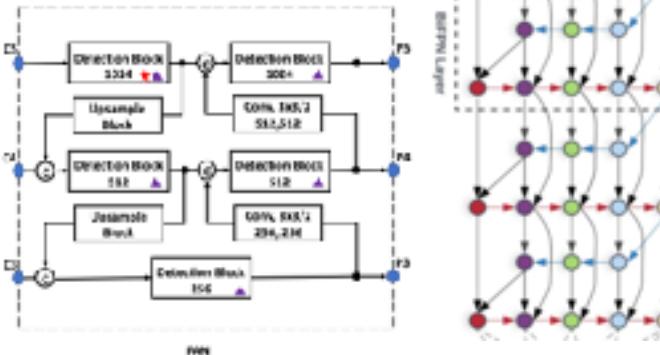
- YOLOv1-v4: Dark/ResNet
- Yv5: EfficientNet
- Yv6: EfficientNet-L2
- PP-Yv1-v2: ResNet50-vd
- Yv6 v3.0: RepVGG 😊



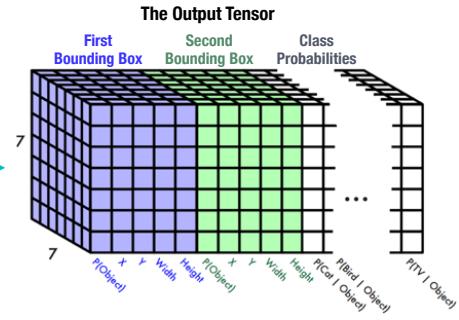
## Neck



- 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



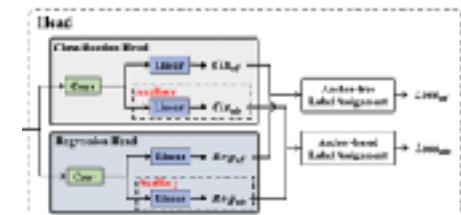
## Head(s)



- Yv1: single head
- Yv3: Multi-resolution heads
- Yv4: Clustered anchor boxes and new gradient penalty
- PP-Yv2: IoU prediction loss
- Yv6 v3.0: Dense Anchor Boxes (anchors as aux. task)

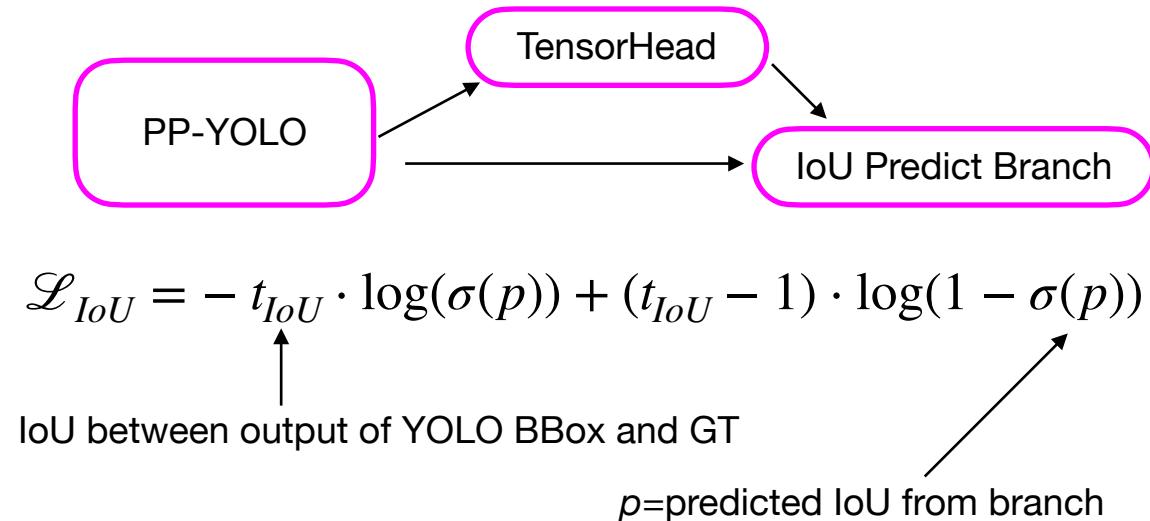
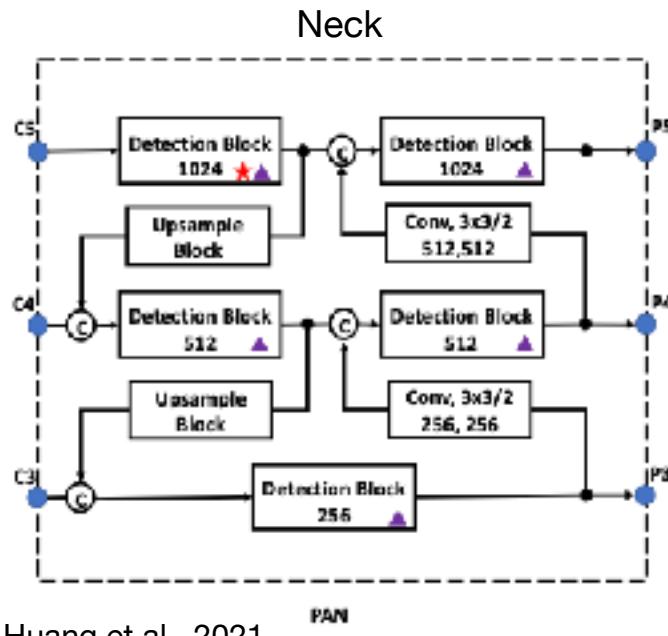
IoU Prediction

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



# 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



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

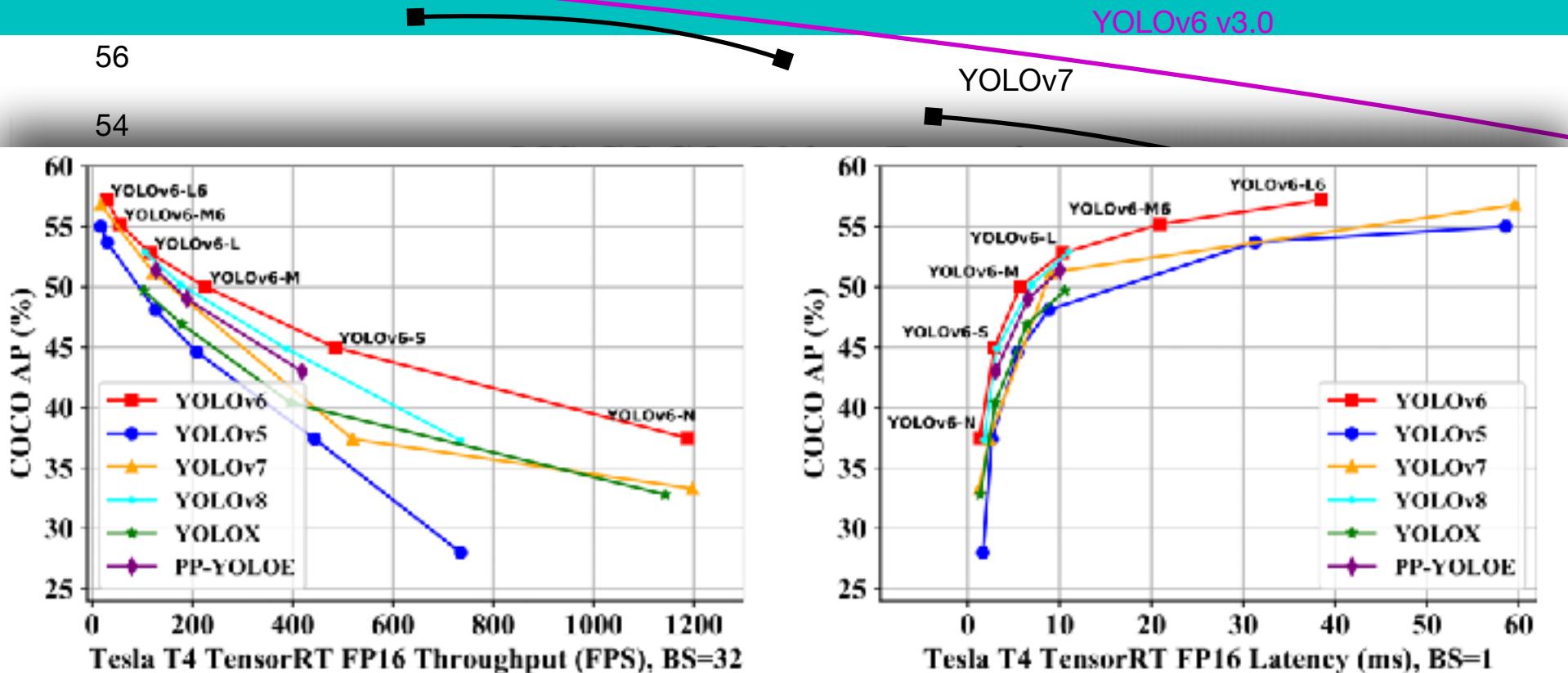
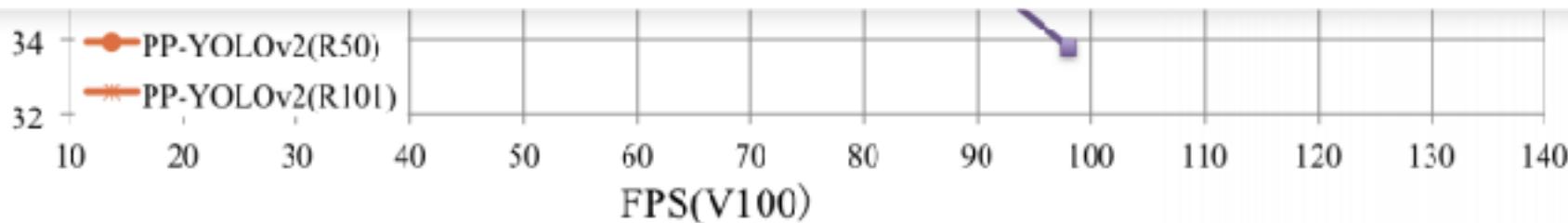


Figure 1: Comparison of state-of-the-art efficient object detectors. Both latency and throughput (at a batch size of 32) are given for a handy reference. All models are test with TensorRT 7.



# A closing thought from YOLOv3 Report

## Abstract

*We present some updates to YOLO! We made a bunch of little design changes to make it better. We also trained this new network that's pretty swell. It's a little bigger than last time but more accurate. It's still fast though, don't worry. At  $320 \times 320$  YOLOv3 runs in 22 ms at 28.2 mAP, as accurate as SSD but three times faster. When we look at the old .5 IOU mAP detection metric YOLOv3 is quite good. It achieves 57.9 AP<sub>50</sub> in 51 ms on a Titan X, compared to 57.5 AP<sub>50</sub> in 198 ms by RetinaNet, similar performance but 3.8× faster. As always, all the code is online at <https://pjreddie.com/yolo/>.*

But maybe a better question is: “What are we going to do with these detectors now that we have them?” A lot of the people doing this research are at Google and Facebook. I guess at least we know the technology is in good hands and definitely won’t be used to harvest your personal information and sell it to.... wait, you’re saying that’s exactly what it will be used for?? Oh.

Well the other people heavily funding vision research are the military and they’ve never done anything horrible like killing lots of people with new technology oh wait....<sup>1</sup>

I have a lot of hope that most of the people using computer vision are just doing happy, good stuff with it, like counting the number of zebras in a national park [13], or tracking their cat as it wanders around their house [19]. But computer vision is already being put to questionable use and as researchers we have a responsibility to at least consider the harm our work might be doing and think of ways to mitigate it. We owe the world that much.

In closing, do not @ me. (Because I finally quit Twitter).

---

<sup>1</sup>The author is funded by the Office of Naval Research and Google.



# A closing thought from YOLOv3 Report

## The Rebuttal I wish I could Write:

Reviewer #2 AKA Dan Grossman (lol blinding who does that) insists that I point out here that our graphs have not one but two non-zero origins. You're absolutely right Dan, that's because it looks way better than admitting to ourselves that we're all just here battling over 2-3% mAP. But here are the requested graphs. I threw in one with FPS too because we look just like super good when we plot on FPS.

Reviewer #4 AKA JudasAdventus on Reddit writes "Entertaining read but the arguments against the MSCOCO metrics seem a bit weak". Well, I always knew you would be the one to turn on me Judas. You know how when you work on a project and it only comes out alright so you have to figure out some way to justify how what you did actually was pretty cool? I was basically trying to do that and I lashed out at the COCO metrics a little bit. But now that I've staked out this hill I may as well die on it.

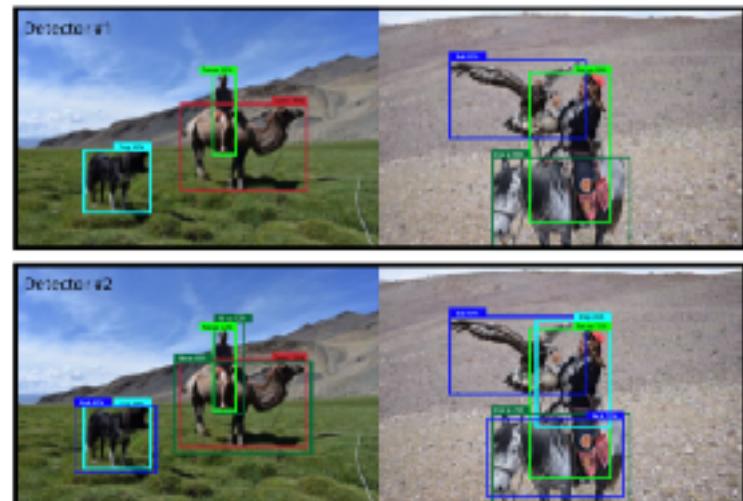
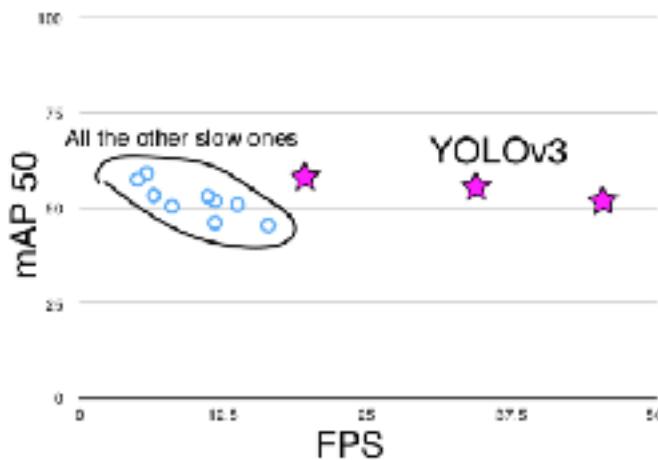


Figure 5. These two hypothetical detectors are perfect according to mAP over these two images. They are both perfect. Totally equal.

Now this is OBVIOUSLY an over-exaggeration of the problems with mAP but I guess my newly retconned point is that there are such obvious discrepancies between what people in the "real world" would care about and our current metrics that I think if we're going to come up with new metrics we should focus on these discrepancies. Also, like, it's already mean average precision, what do we even call the COCO metric, average mean average precision?

Here's a proposal, what people actually care about is given an image and a detector, how well will the detector find and classify objects in the image. What about getting rid of the per-class AP and just doing a global average precision? Or doing an AP calculation per image and averaging over that?

Boxes are stupid anyway though, I'm probably a true believer in masks except I can't get YOLO to learn them.



# YOLACT, one pass mask generation

## YOLACT Real-time Instance Segmentation

Daniel Bolya Chong Zhou Fanyi Xiao Yong Jae Lee

University of California, Davis

{dbolya, czhou, fyxiao, yongjaelee}@ucdavis.edu

### Abstract

We present a simple, fully-convolutional model for real-time instance segmentation that achieves 29.8 mAP on MS COCO at 33.5 fps evaluated on a single Titan Xp, which is significantly faster than any previous competitive approach. Moreover, we obtain this result after training on **only one GPU**. We accomplish this by breaking instance segmentation into two parallel subtasks: (1) generating a set of prototype masks and (2) predicting per-instance mask coefficients. Then we produce instance masks by linearly combining the prototypes with the mask coefficients. We find that because this process doesn't depend on repooling, this approach produces very high-quality masks and exhibits temporal stability for free. Furthermore, we analyze the emergent behavior of our prototypes and show they learn to localize instances on their own in a translation variant manner, despite being fully-convolutional. Finally, we also propose Fast NMS, a drop-in 12 ms faster replacement for standard NMS that only has a marginal performance penalty.

### 1. Introduction

*"Boxes are stupid anyway though. I'm probably a true believer in masks except I can't get YOLO to learn them."*

– Joseph Redmon, YOLOv3 [36]

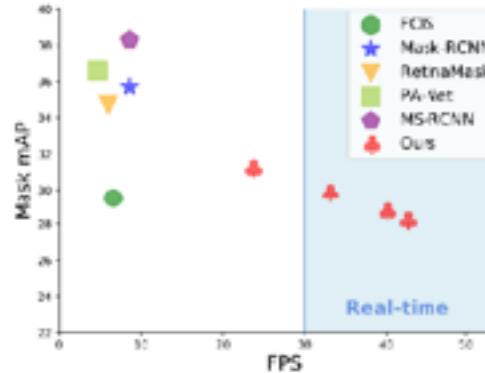


Figure 1: Speed-performance trade-off for various instance segmentation methods on COCO. To our knowledge, ours is the first *real-time* (above 30 FPS) approach with around 30 mask mAP on COCO test-dev.

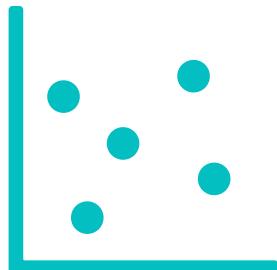
However, instance segmentation is hard—much harder than object detection. One-stage object detectors like SSD and YOLO are able to speed up existing two-stage detectors like Faster R-CNN by simply removing the second stage and making up for the lost performance in other ways. The same approach is not easily extendable, however, to instance segmentation. State-of-the-art two-stage

Bolya, Daniel, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. "YOLACT: real-time instance segmentation." In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9157-9166. October 2019.



# Lecture Notes for **Neural Networks** **and Machine Learning**

FCN Learning: Detection



**Next Time:**  
Instance Segmentation  
**Reading:** None

