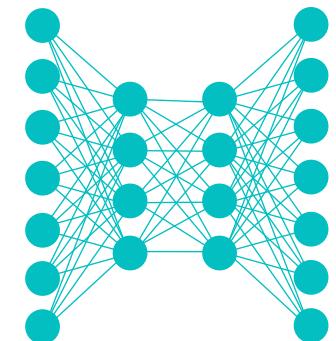


Lecture Notes for **Neural Networks** **and Machine Learning**



Fully Convolutional Learning
Object Detection

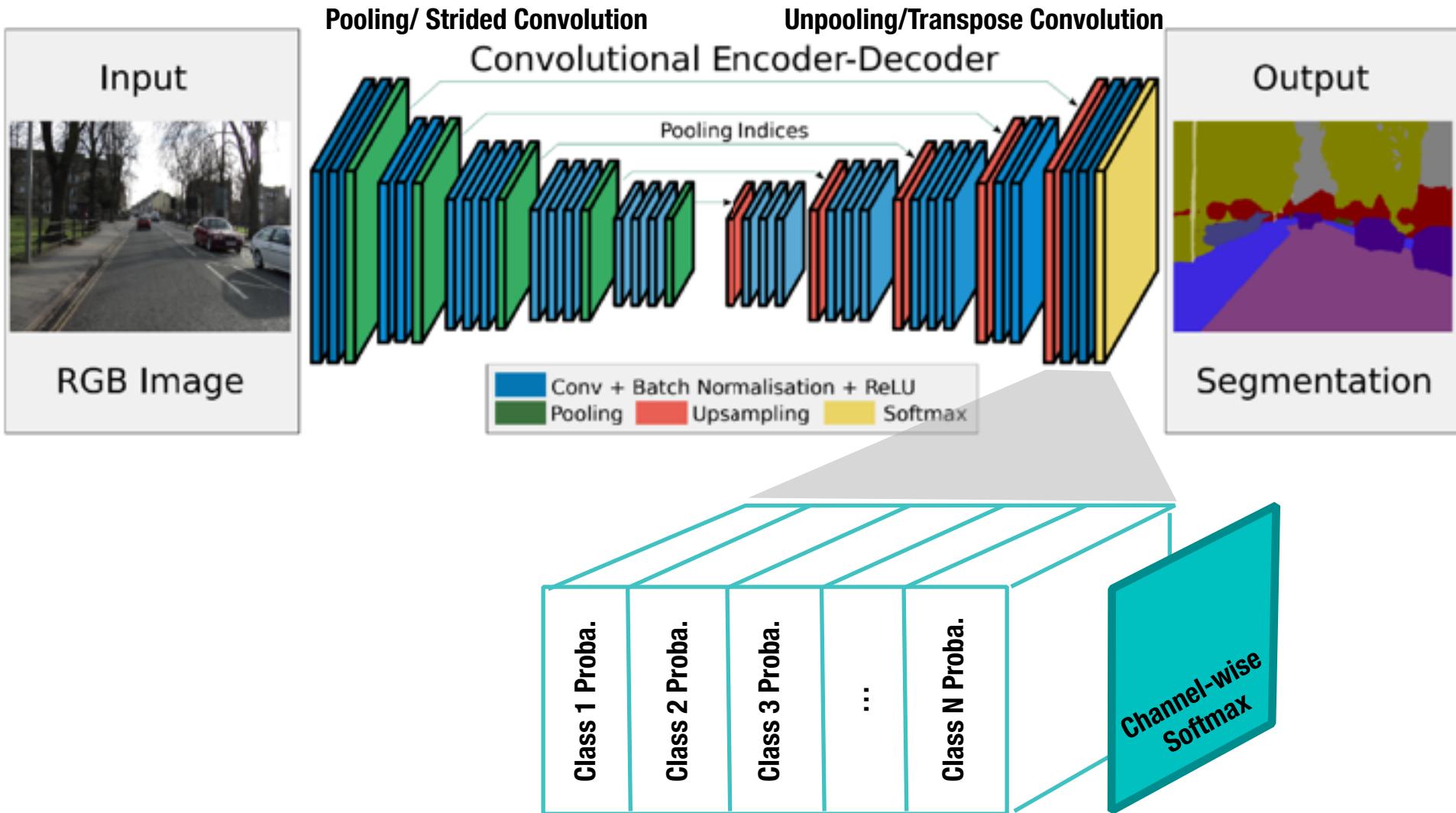


Logistics and Agenda

- Logistics
 - Next lab due soon
 - Lab grading updates
- Agenda
 - Full Convolutional Architectures
 - ◆ Semantic Segmentation (last time):
 - DeepLabV3, GSCNN
 - ◆ Object Detection (this time):
 - RCNN, YOLO
 - ◆ Instance Segmentation (next time, probably):
 - Mask-RCNN, YOLACT

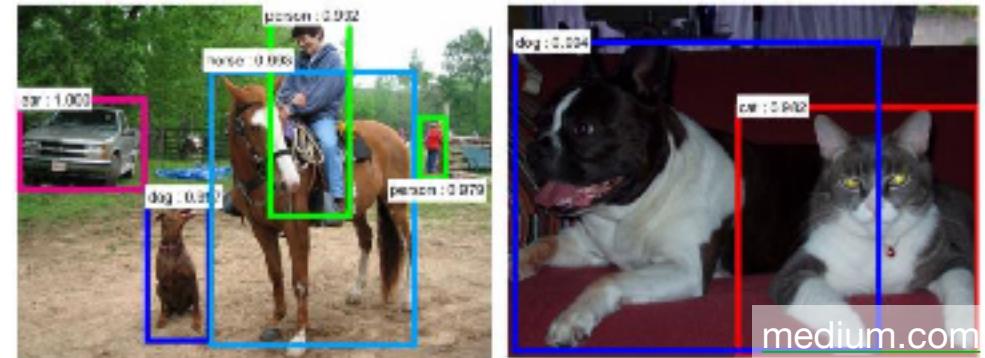


Last Time

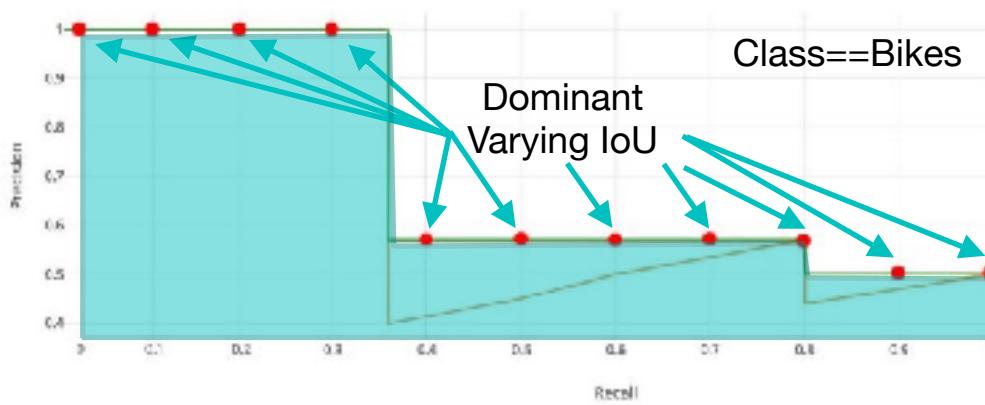
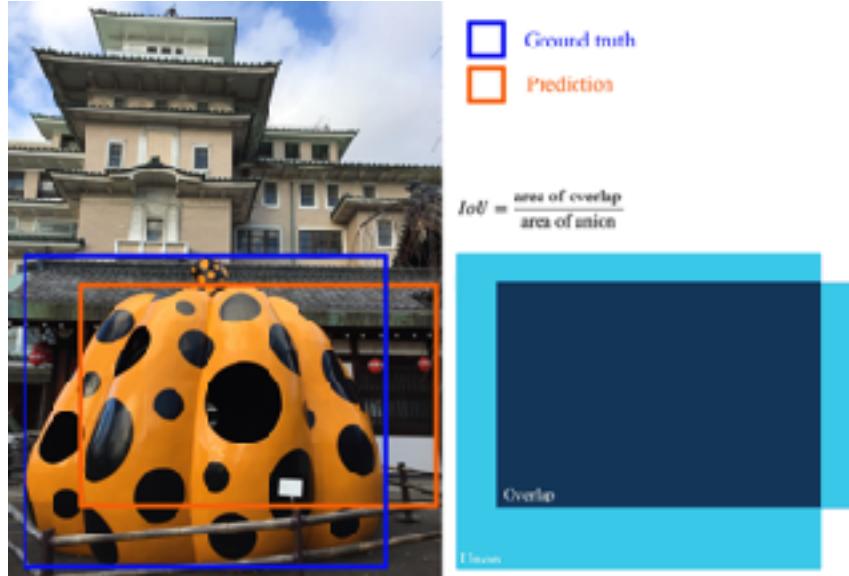


This time... Object Detection Methods

- Semantic segmentation is not great in terms of mIoU (some classes are at 50%)
- How to adapt these techniques to get bounding boxes, not semantic segmentations?
 - Could this be easier? More stable?
 - More consistent labeling?
 - Suitable for “higher risk” tracking applications?



First: Measuring Performance



- mAP($\text{IoU}=x\%$)
 - if $\text{IoU} > X\%$, correct
 - else not
 - Usually: 50% and 75%
 - Define precision for each class, take average
- mAP(%), sometimes just AP
 - Formulate precision/recall curve for a class at varying levels of IoU (50%-95%)
 - Calculate dominating points
 - Take area under curve
 - Take average area over all classes (macro or micro averaging can be done, usually macro)

<https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173>

5



Object Detection with RCNN

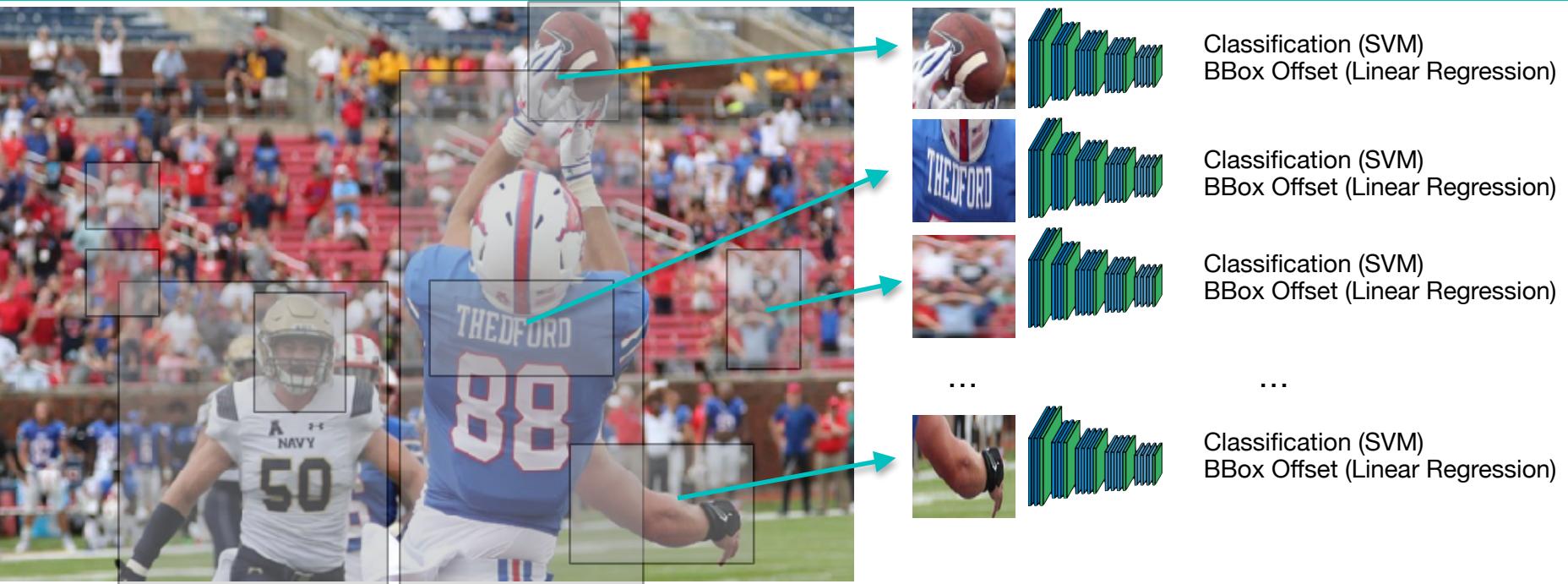


**A history in naming one network five different times
with five different papers
each time changing one thing about the architecture**

Research!



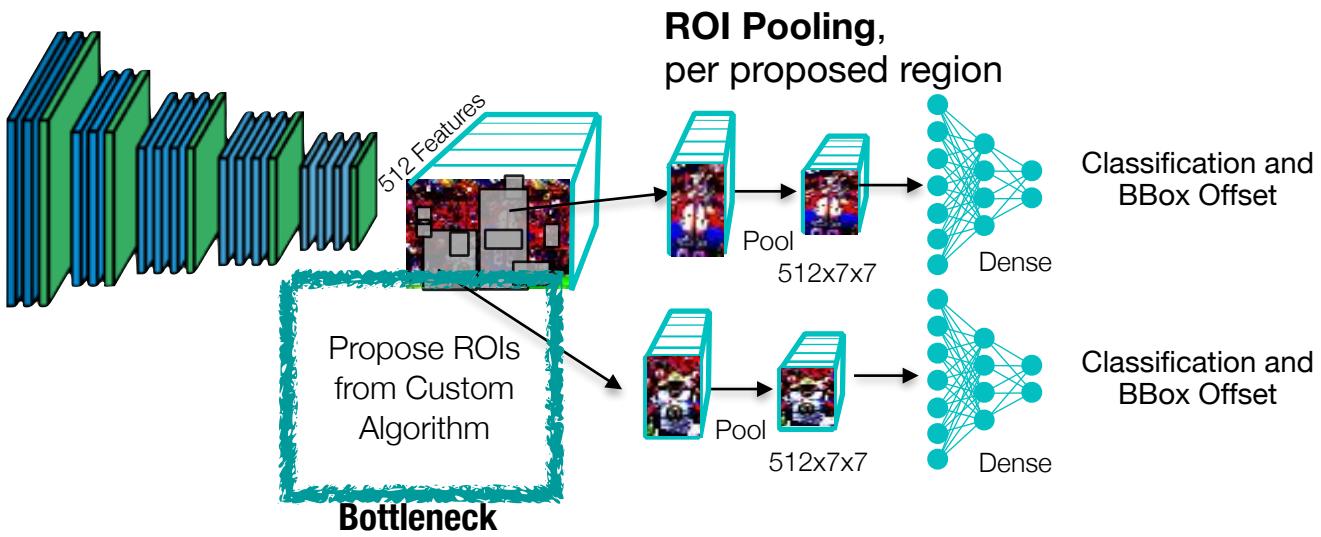
2014: R-CNN



- Too Slow to Be Useful
- SVM and BBox Regression Trained Separately
- Fine Tuned Existing ConvNet (for Warped Images)
- ~50 Seconds per Image when Deployed



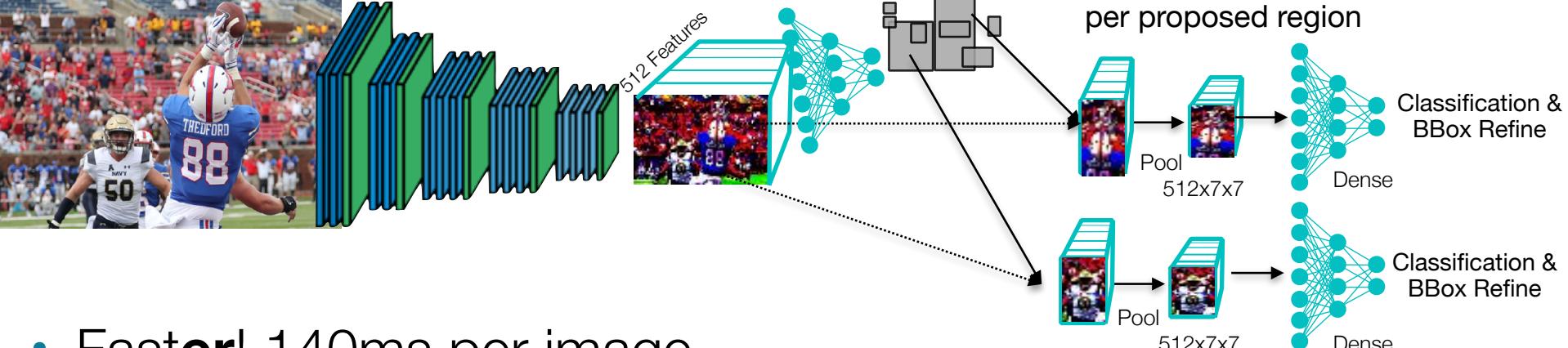
2015: Fast R-CNN



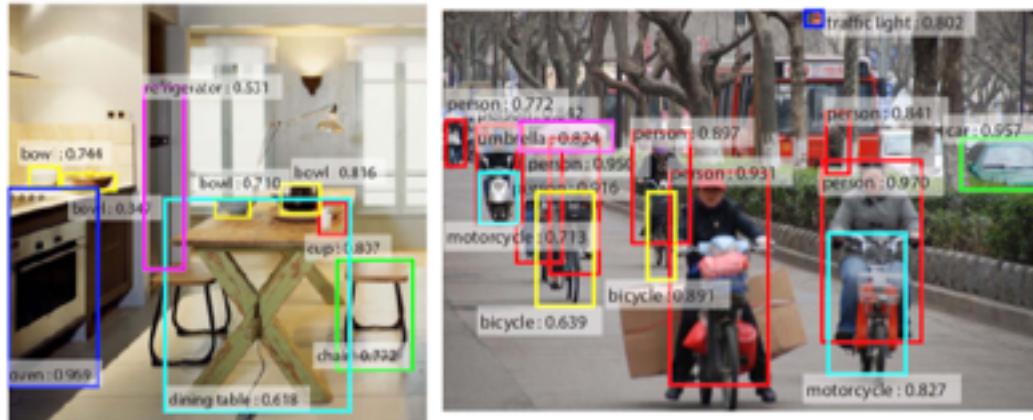
- Fast! 2.3 seconds per image (not ~50)
- But still not real time...



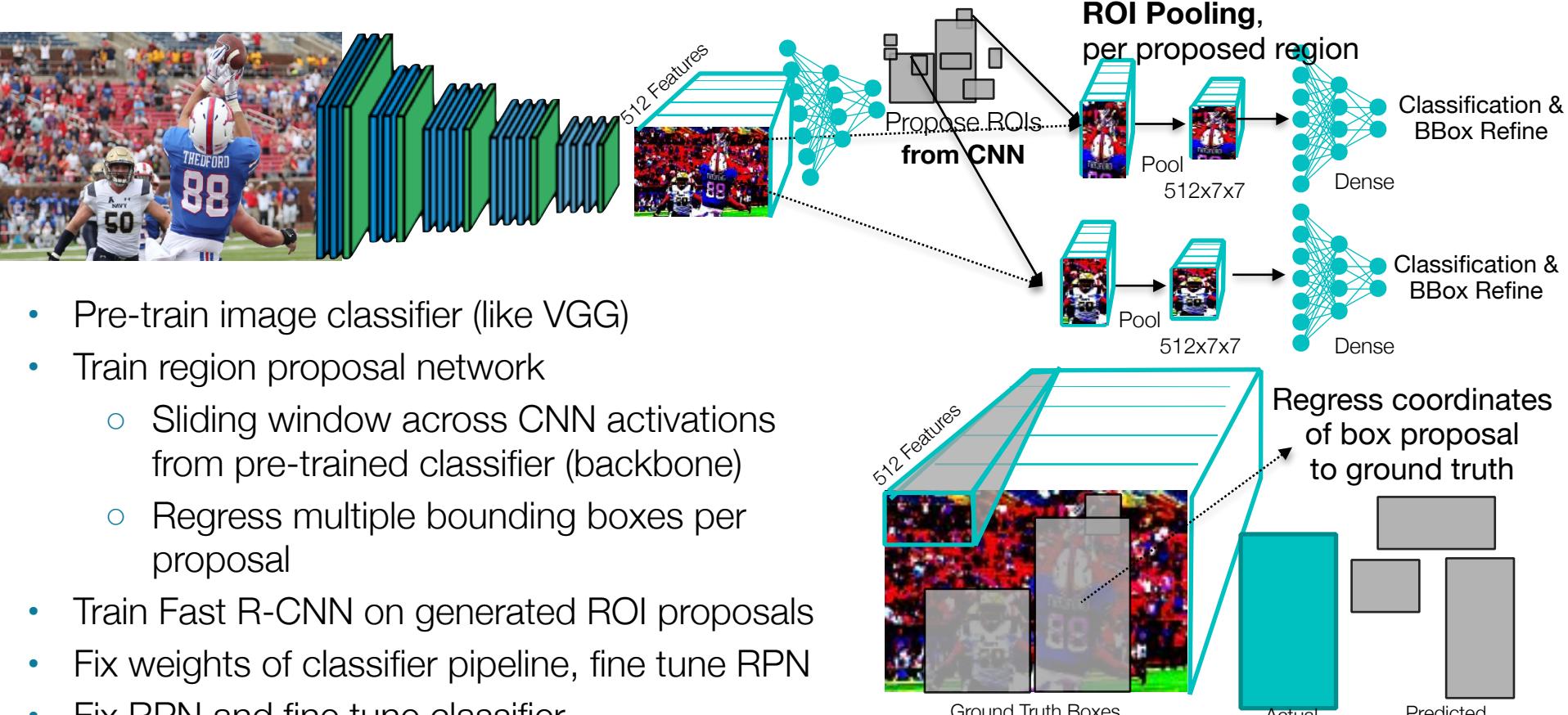
2015: Faster R-CNN



- **Faster!** 140ms per image (7 FPS)
- Highly Accurate



2015: Faster R-CNN, Training



$$l_{box} = \sum_i \mathbf{1}_i \left[(x - \hat{x}_i)^2 + (y - \hat{y}_i)^2 + (\log w - \log \hat{w}_i)^2 + (\log h - \log \hat{h}_i)^2 \right]$$

$$l_{class} = \sum_c CE(c, \hat{c})$$

Ren et al. Faster R-CNN, 2015, November

10



Object Detection with YOLO

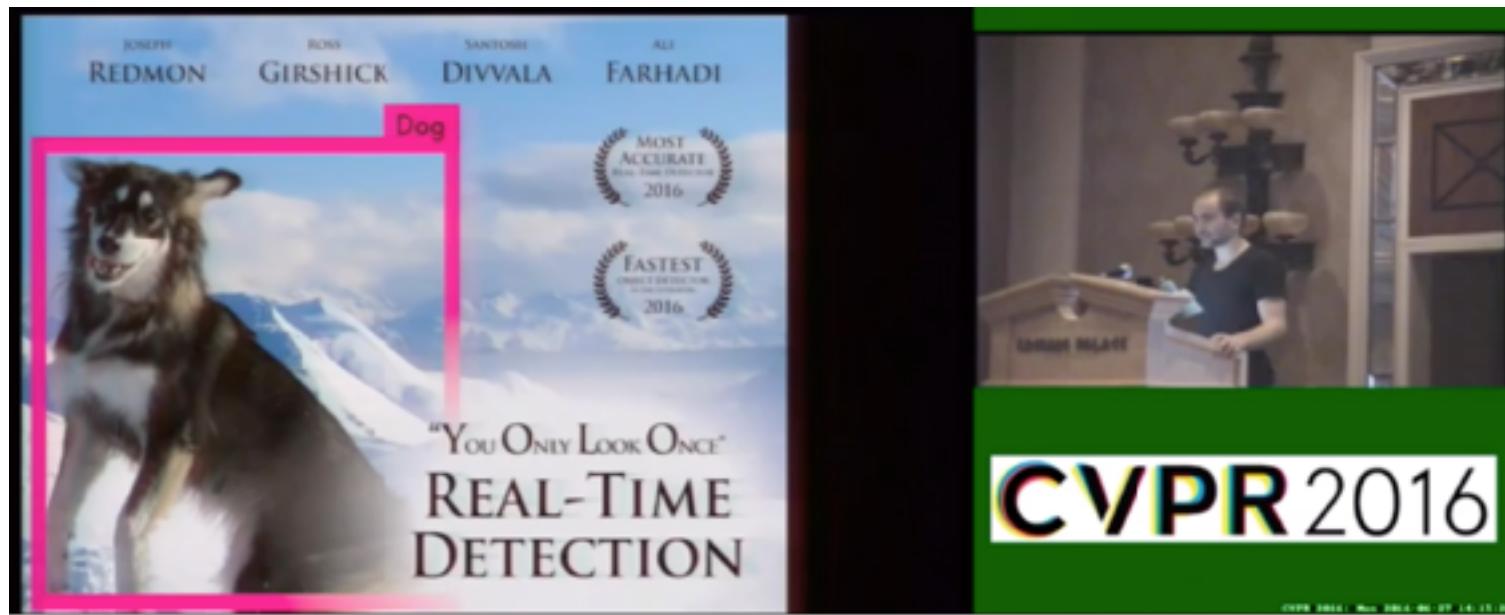


- In parallel to RCNN, some of the same researchers started working on a related, but different approach



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, 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

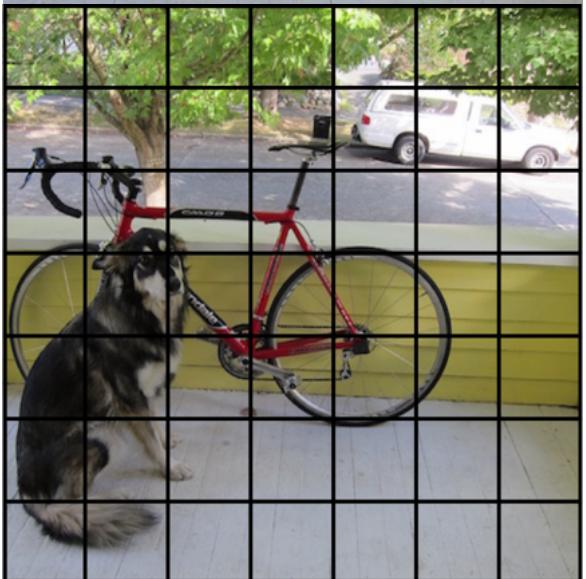
Joseph Redmon^{*†}, Ali Farhadi^{*†}
University of Washington^{*}, Allen Institute for AI[†]
<http://pjreddie.com/yolo9000/>



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

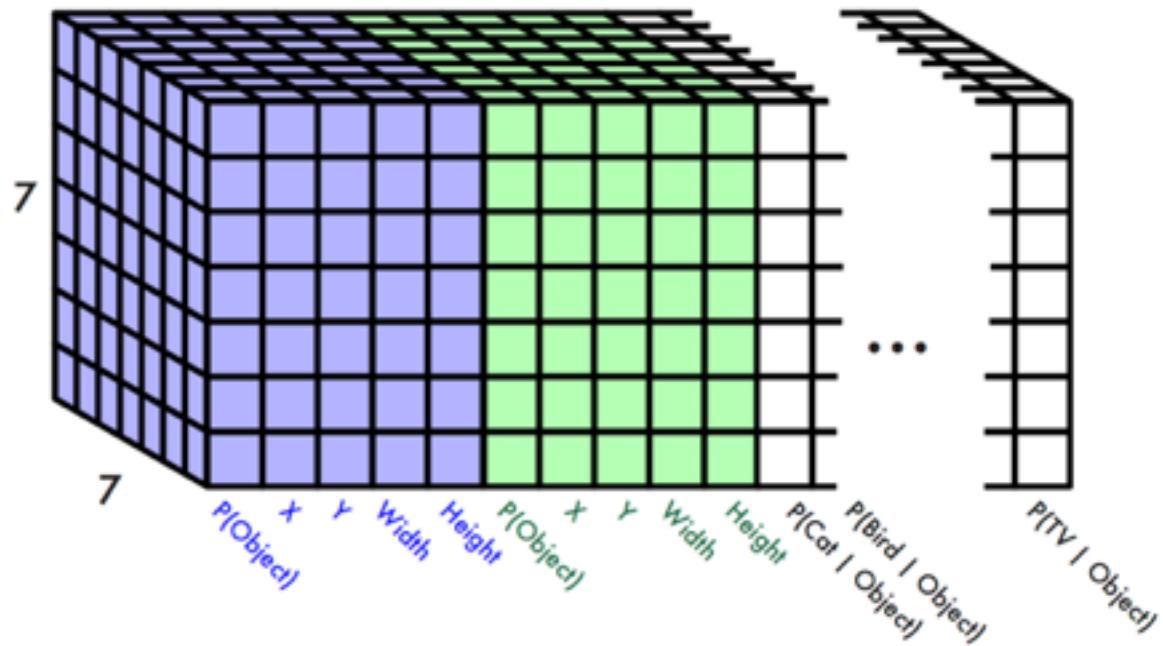


The YOLO Output Tensor



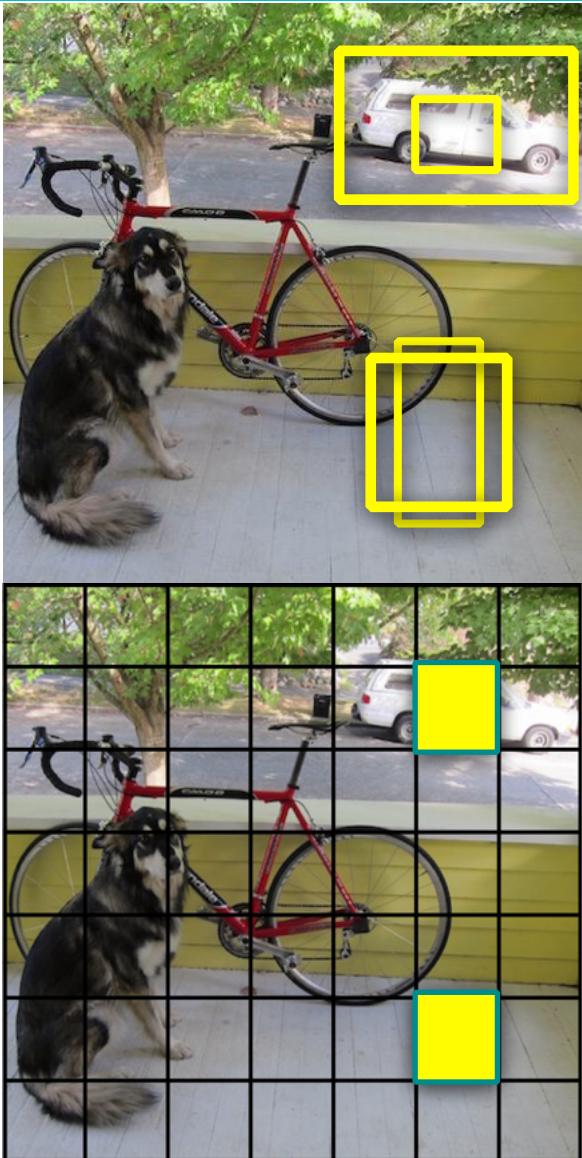
The Output Tensor

First Bounding Box Second Bounding Box Class Probabilities



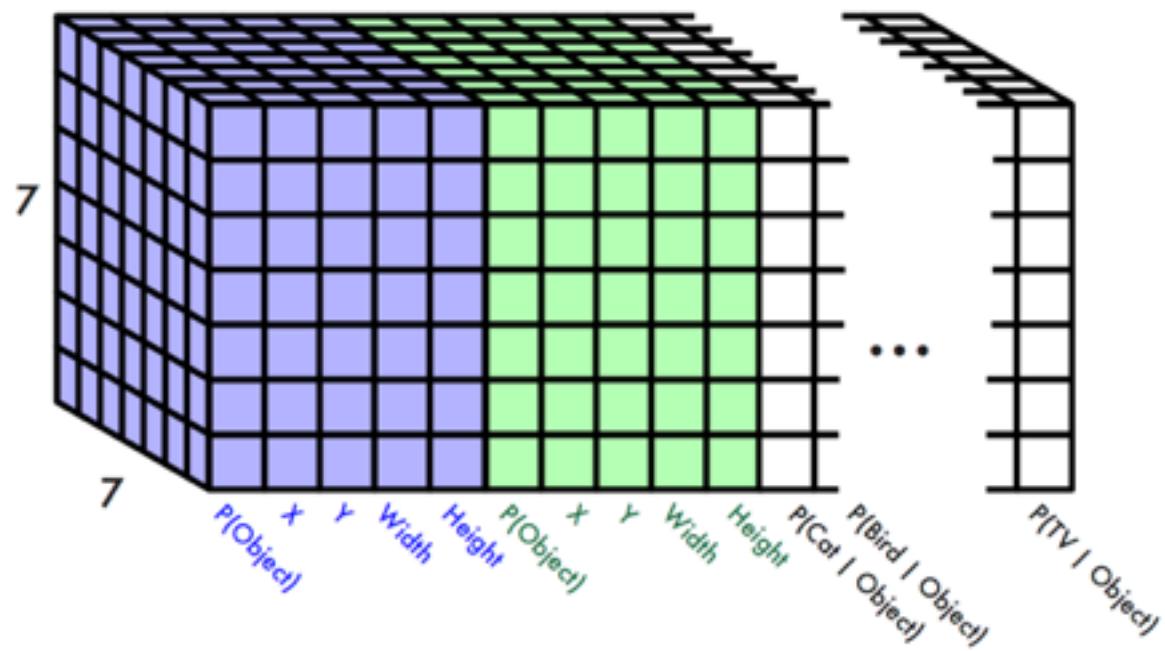
Redmon and Farhadi, YOLO9000: Better, Faster, Stronger, 2016,
December 25 — Merry Christmas?

The YOLO Output Tensor



The Output Tensor

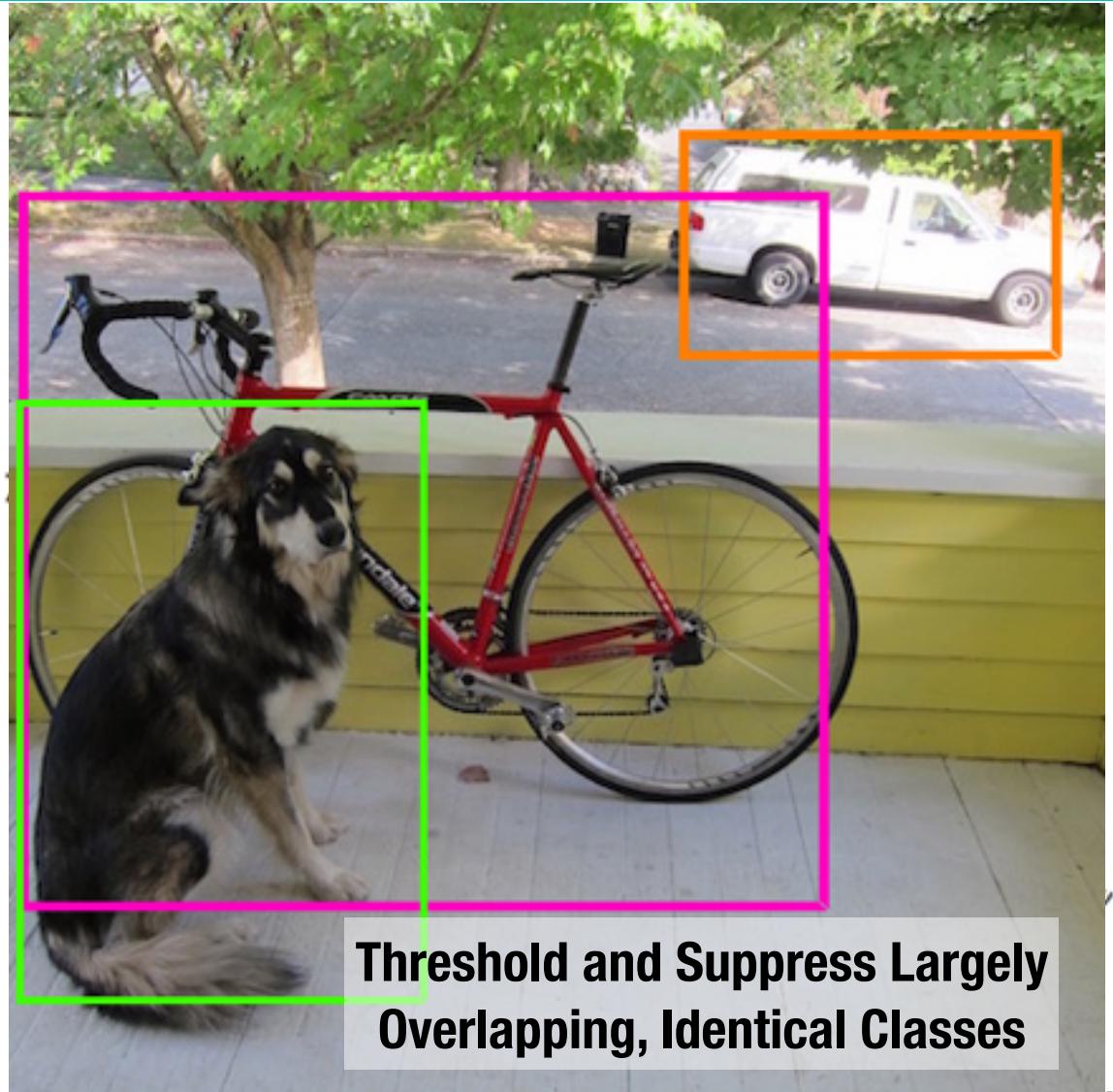
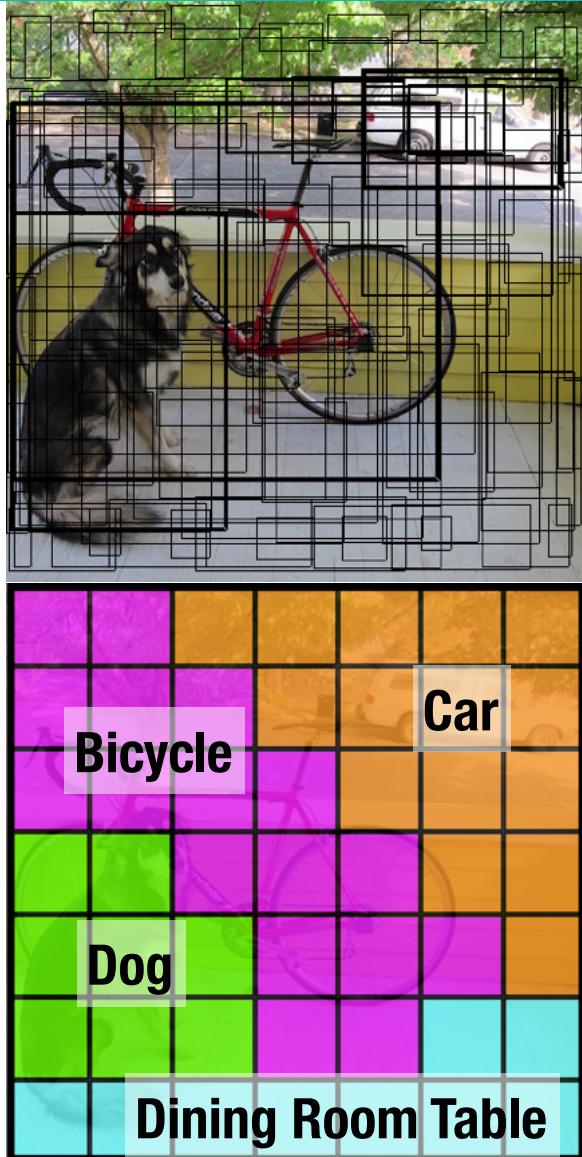
First Bounding Box Second Bounding Box Class Probabilities



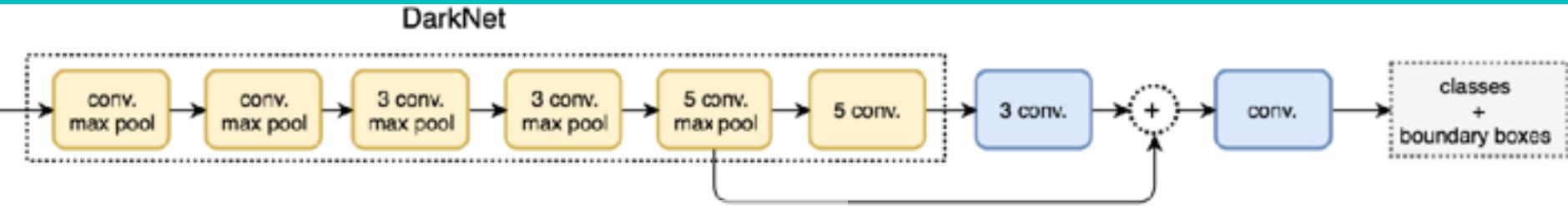
Redmon and Farhadi, YOLO9000: Better, Faster, Stronger, 2016,
December 25 — Merry Christmas?



The YOLO Output Tensor



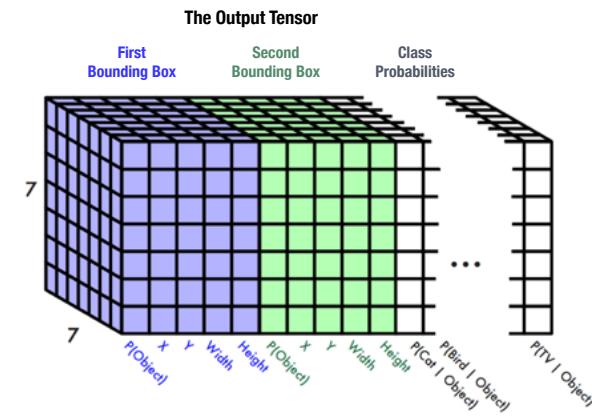
The YOLO Architecture



**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:

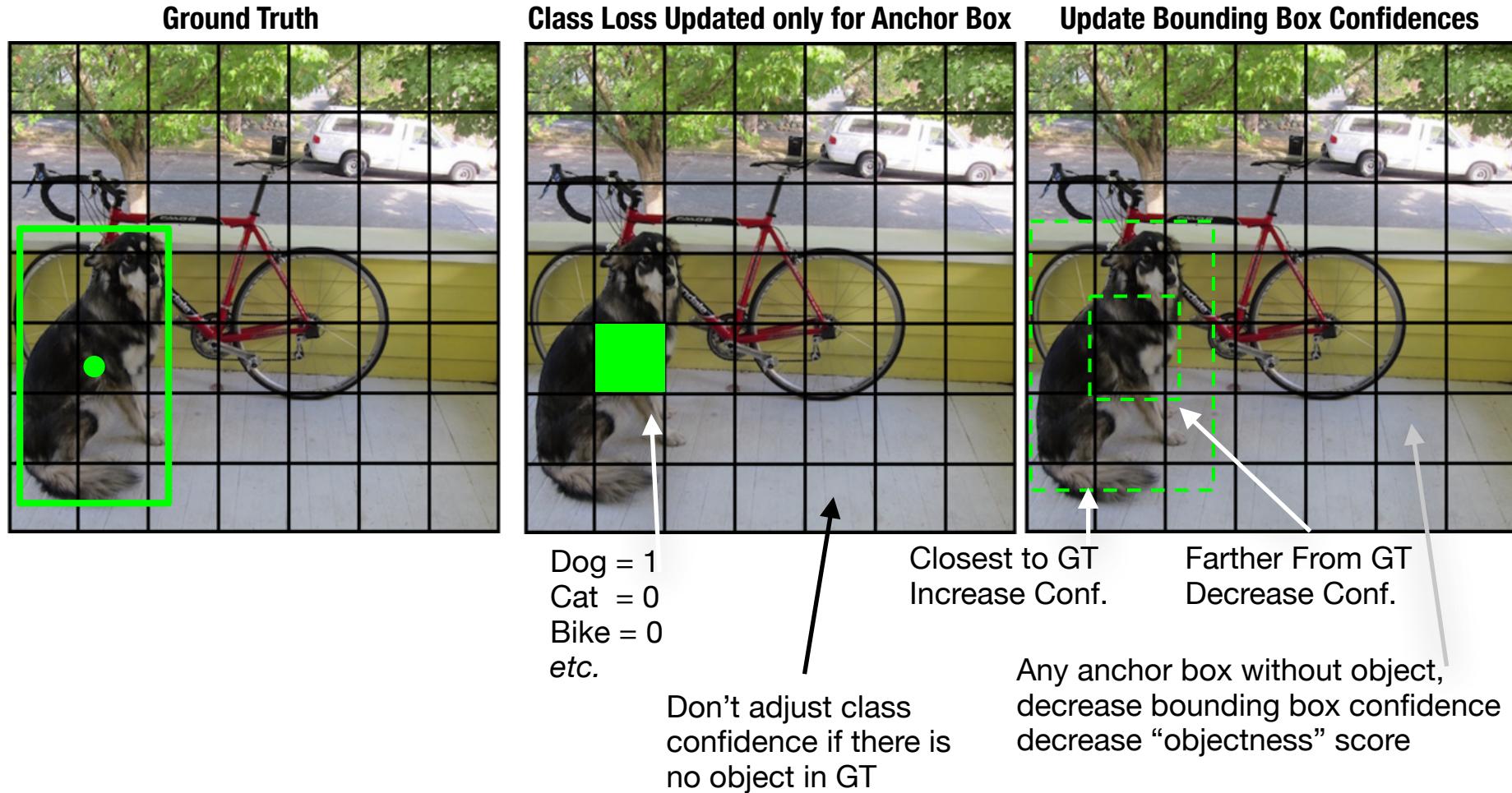
**Last layers:
Trained on images
with bounding boxes**



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

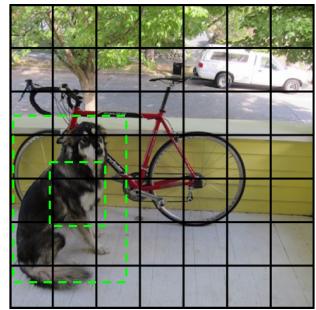


Training the YOLO Architecture

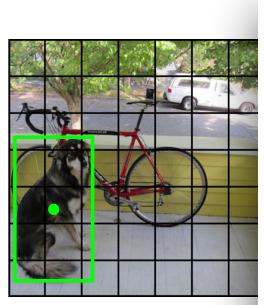


The YOLO Loss Function

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^{th} cell

B boxes per cell, j^{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

Localization Loss

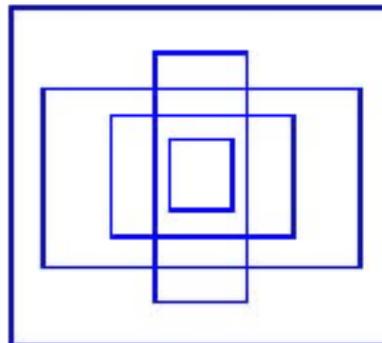
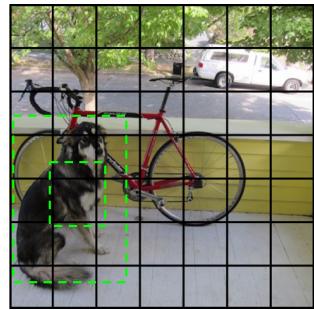
Object Detection Loss

Classification Loss

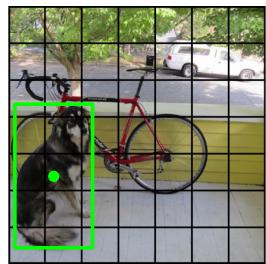


Updated YOLO Localization

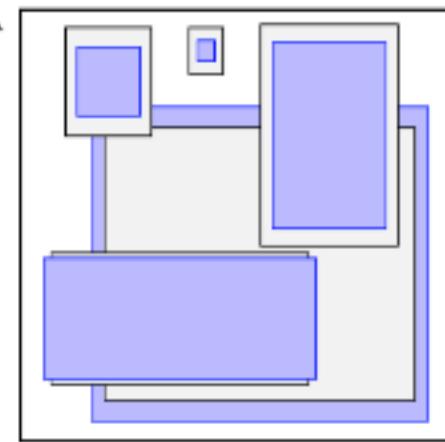
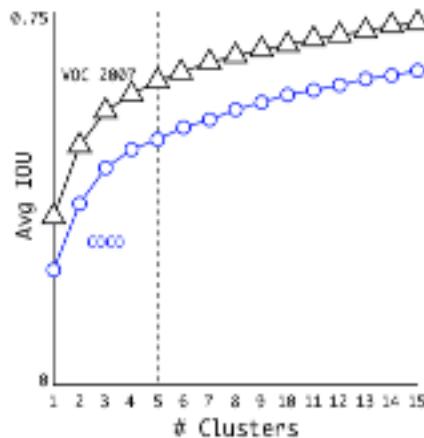
Update Bounding Box



- 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 w and h scaling of predefined shape



**“Good” Shape Priors
Found via Clustering**



Class Loss



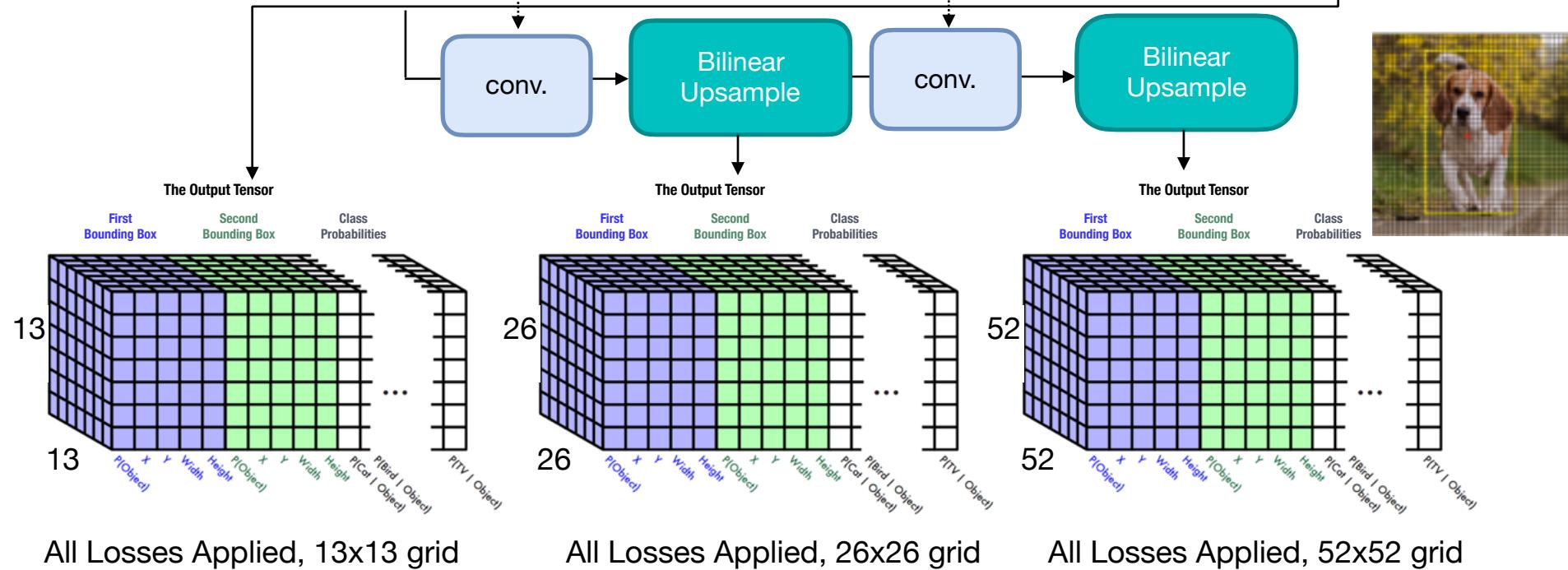
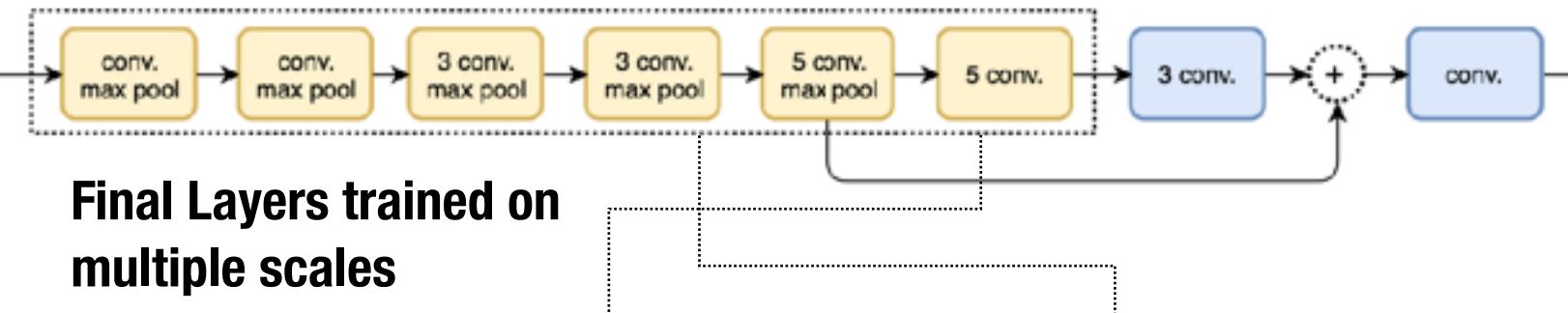
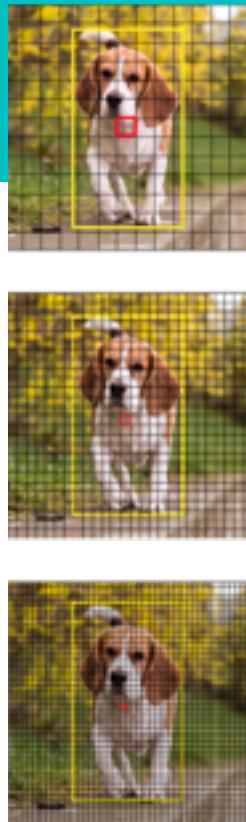
$$\begin{aligned} & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_{ij} \right)^2 \\ & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_{ij} \right)^2 + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}}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

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

Classification Loss



The YOLOv3 Architecture

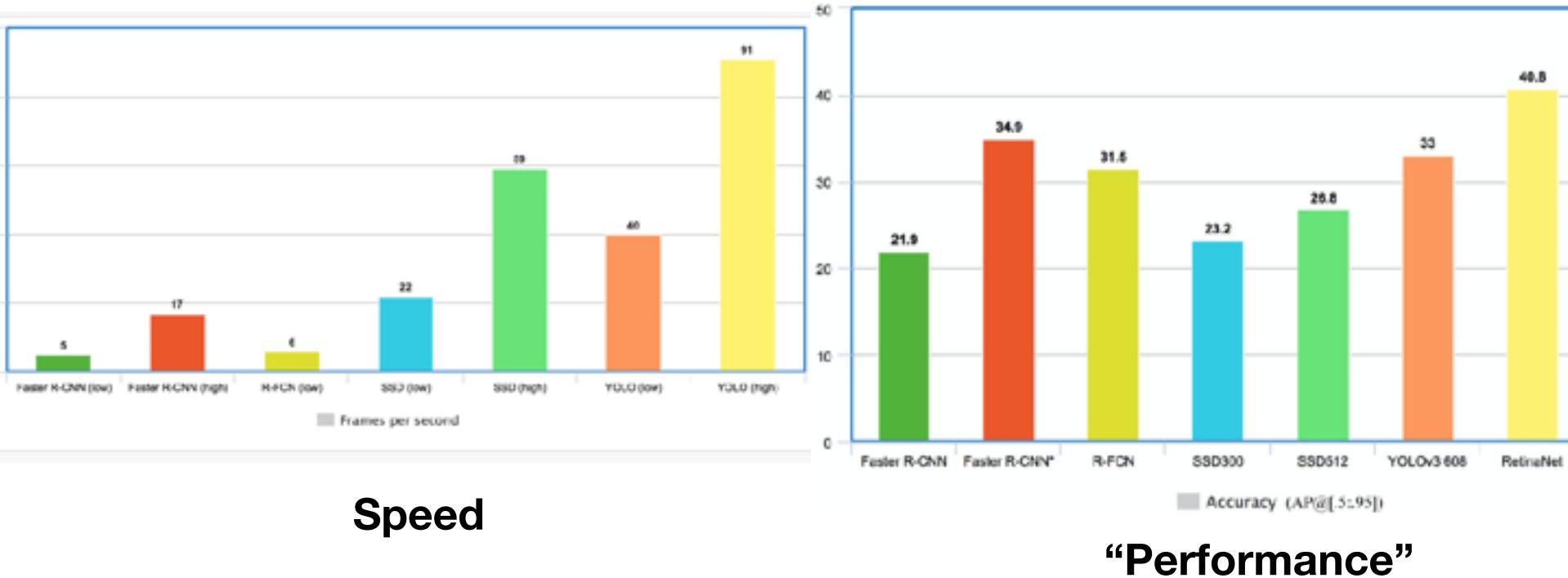


https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088



2020: YOLO, Faster-RCNN

- Remember, this is an ongoing line of research and everybody wants to win the franchise war



https://medium.com/@jonathan_hui/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359

22

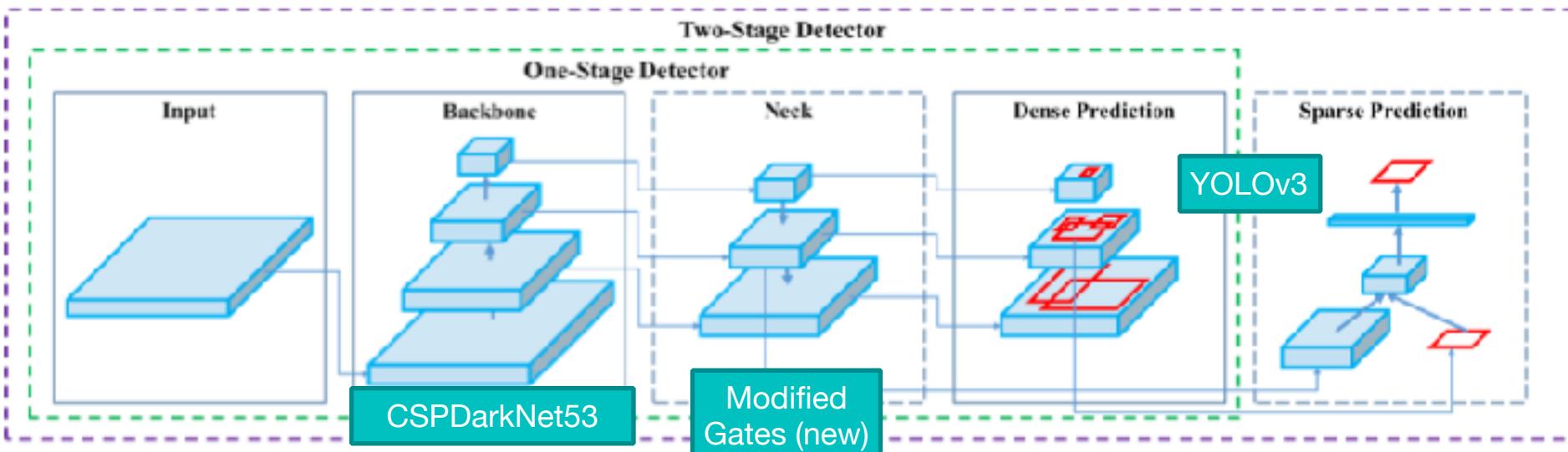


More YOLO Methods

I don't care at all



Is there a YOLOv4?

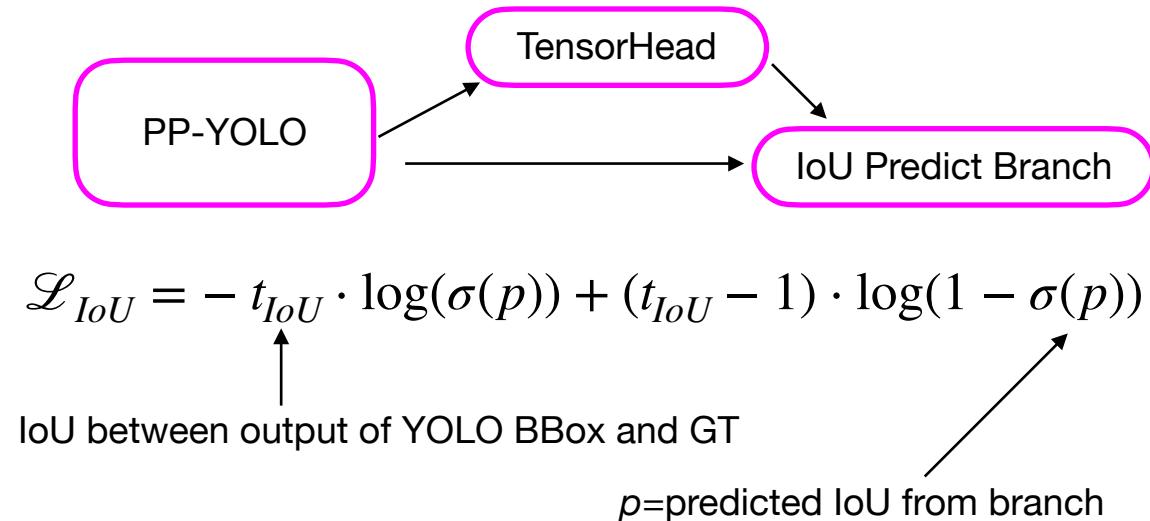
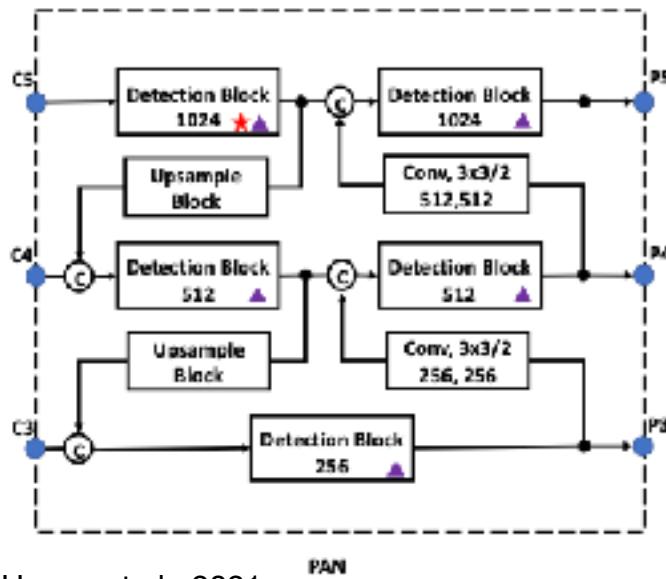


- **Yes** and it has improvements based on some simple elements
- There is also a “YOLOv5” that caused a lot of controversy because it was essentially a port of existing YOLO to PyTorch
 - Torch-YOLO?
- And then a PP-YOLO, and PP-YOLOv2
 - from Baidu using their deep learning framework PaddlePaddle



PP-YOLOv2: Methods

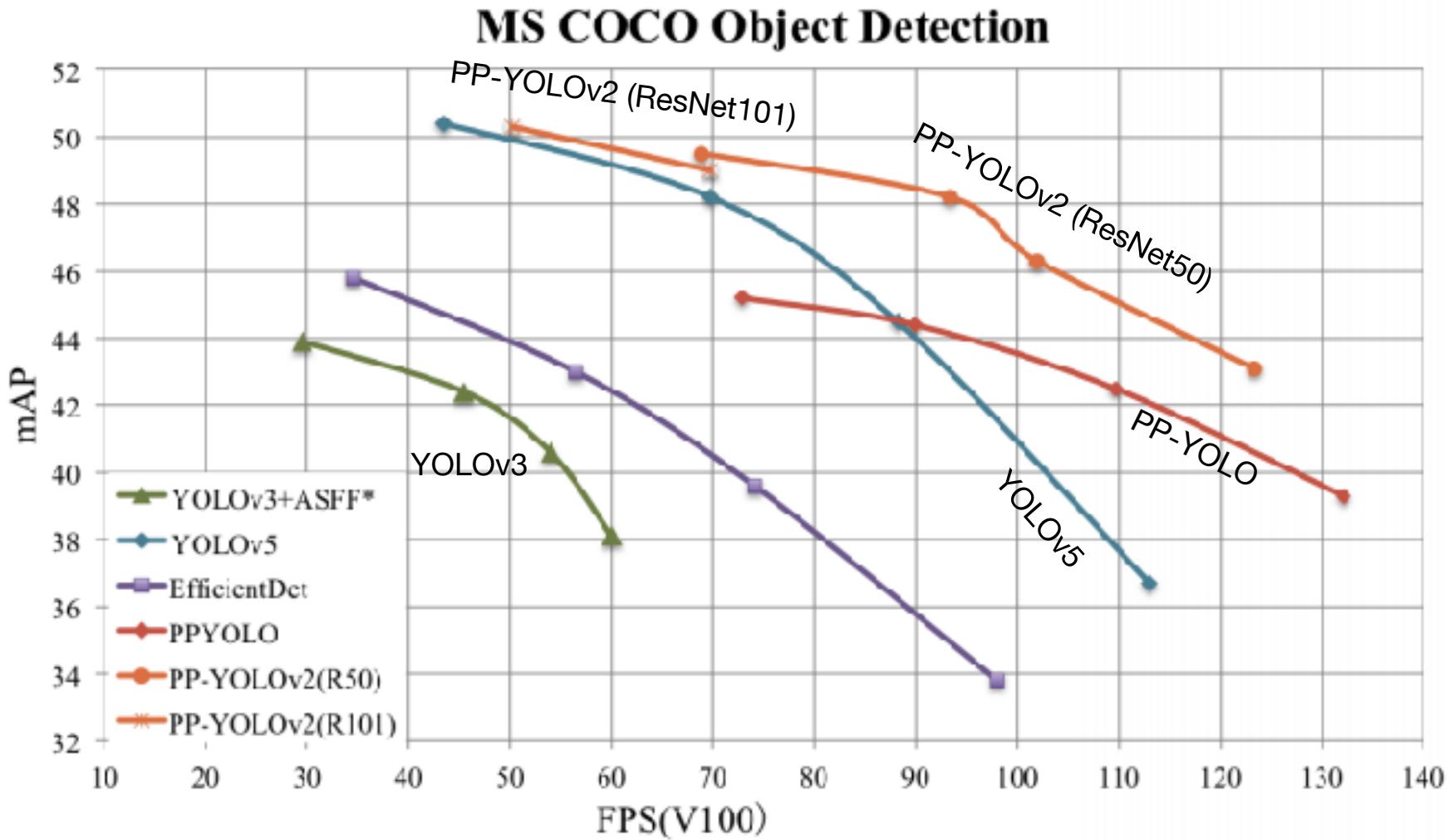
- Path Aggregation Network (PAN) for multi-scale processing
- Varied Input Sized Images and “Mish” activation
- “IoU Aware Branch” that uses the IoU prediction in the back propagation



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



A closing thought from YOLOv3 Report



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

Replies to @kevin_zakka 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

Replies to @skoulaidou

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



Redmon and Farhadi, YOLOv3 Report

"What are we going to do with them?" A lot of Google and Facebook. Technology is in good hands. Protect your personal information saying that's exactly

ding vision research are anything horrible like technology oh wait....

the people using computers good stuff with it, like national park [13], or defend their house [19]. But to questionable use and liability to at least consider and think of ways to mitigate.

(I finally quit Twitter).

val 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.

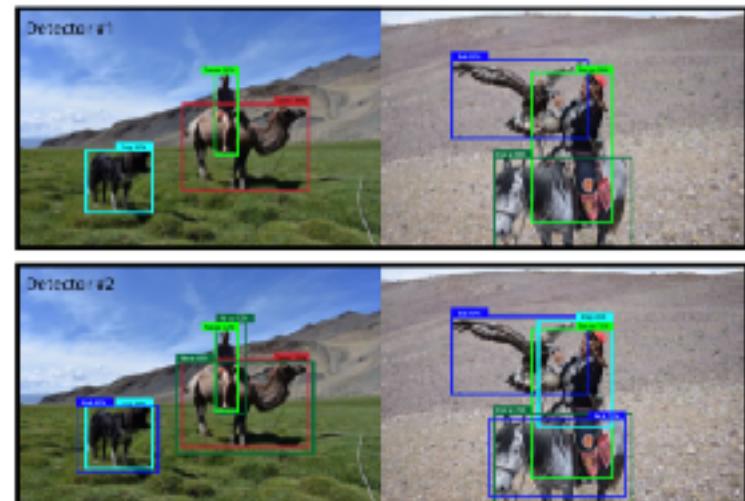
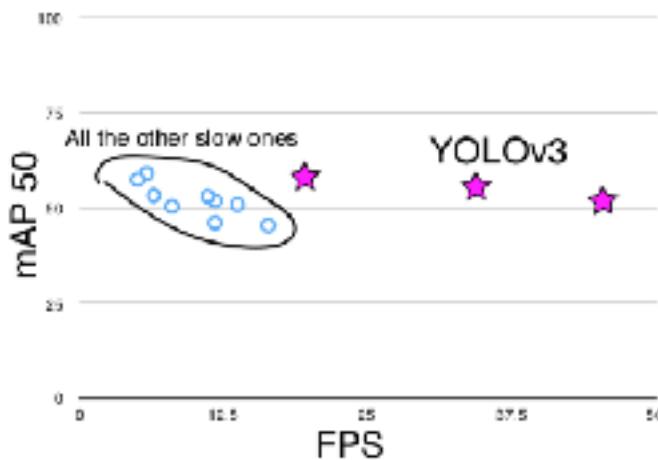


Figure 5. These two hypothetical detectors are perfect according to mAP over these two images. They are both perfect. Totally equal.

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]

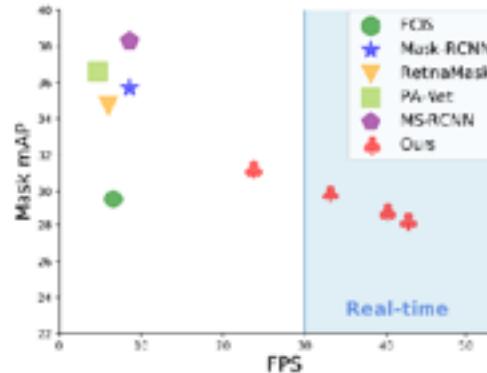


Figure 1: Speed-performance trade-off for various instance segmentation methods on COCO. To our knowledge, ours is the first *real-time* (above 30 FPS) approach with around 30 mask mAP on COCO test-dev.

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

