

Practical Deep learning for Computer Vision

Dr Kien Nguyen

Email: k.nguyenthanh@qut.edu.au

Queensland University of Technology, Australia

Warning

All materials belong to Dr Kien Nguyen and copyright-protected and law-protected. Any sharing has to be inferred to Dr Kien Nguyen.

Deep learning is new SEXY (for a reason!)



Week 3: Object Recognition

Resnet from scratch

Transfer Learning

Google Colab

Week 4: Object Detection

DarkNet & YOLO

Transfer Learning

AWS

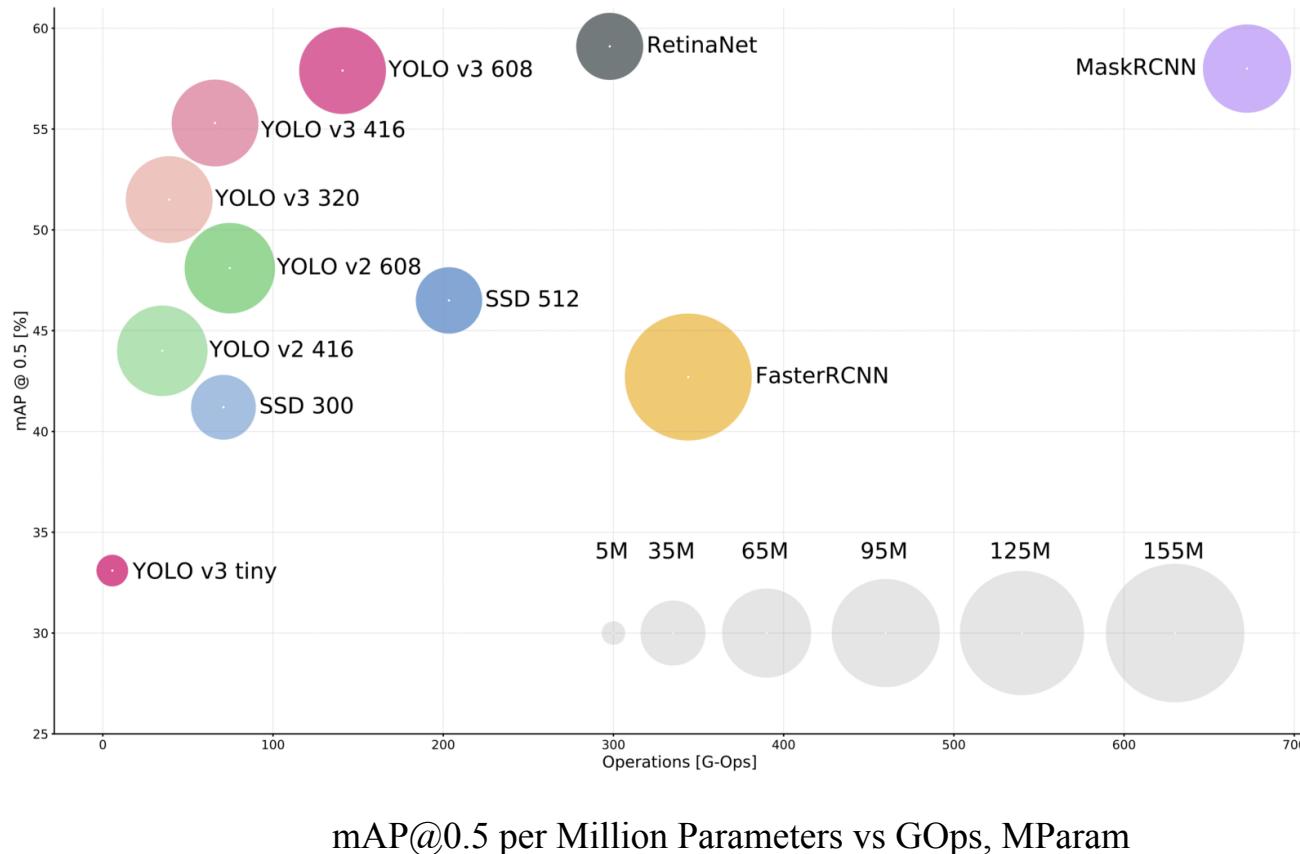
Week 5: GAN

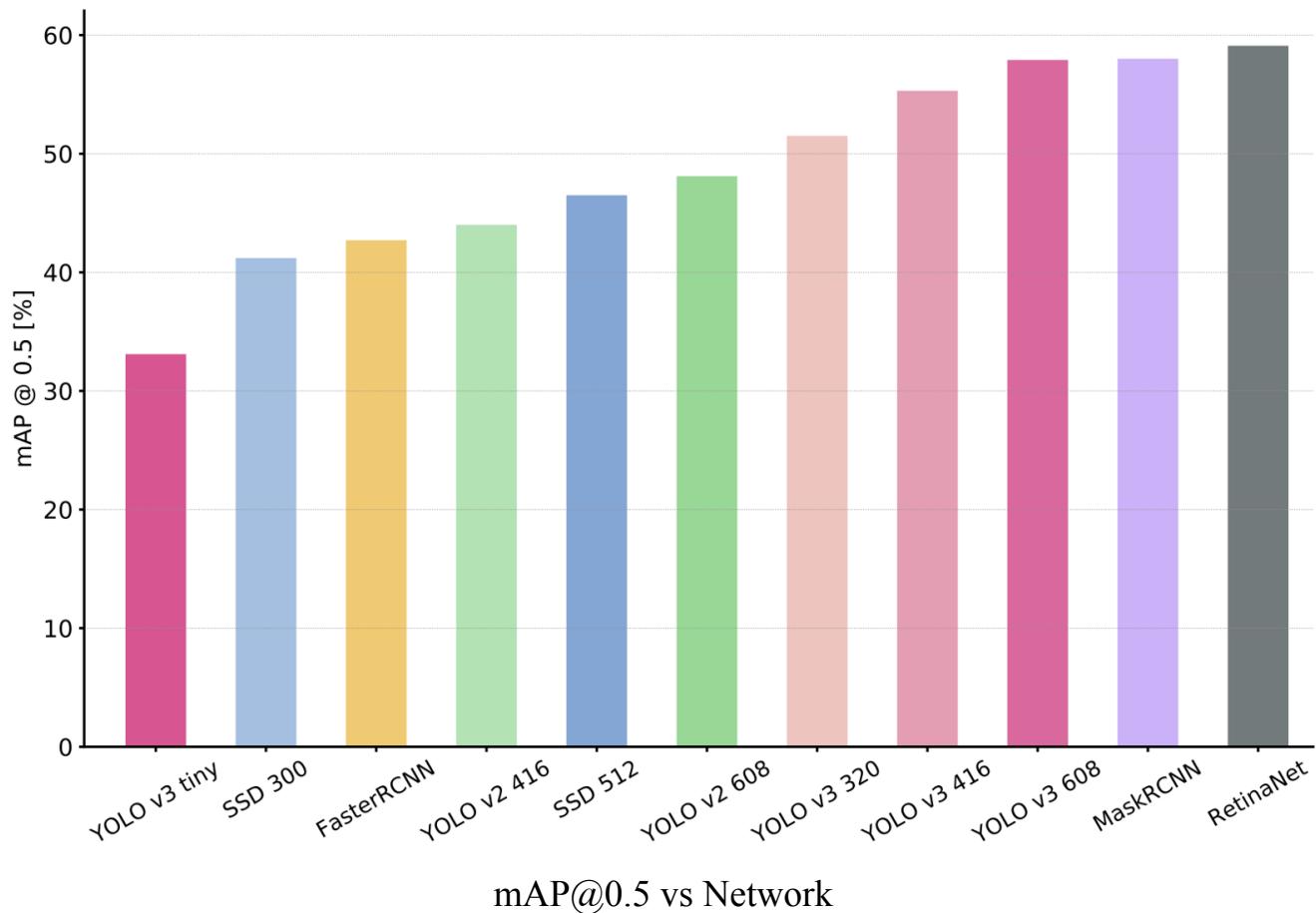
SSD

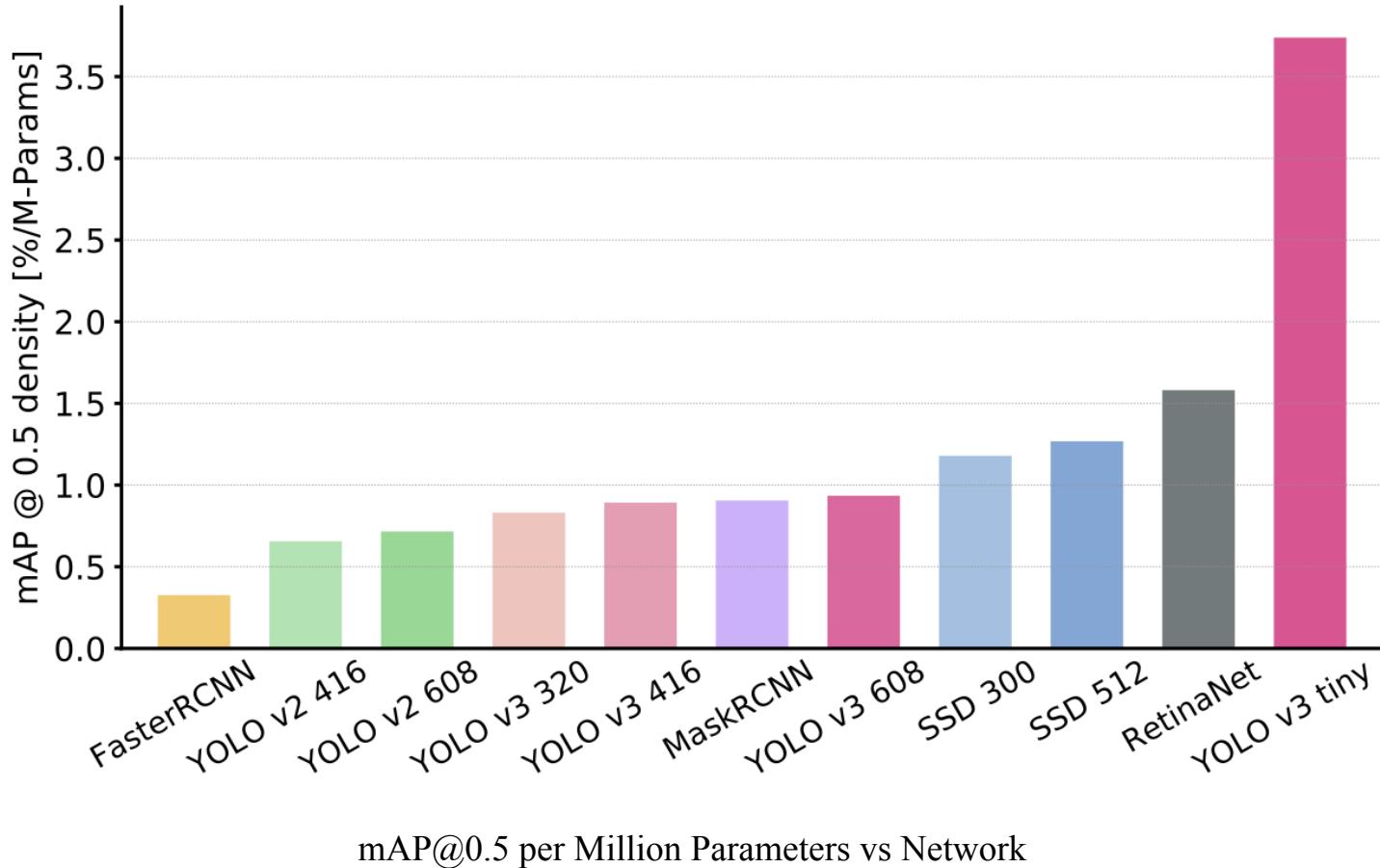
GAN

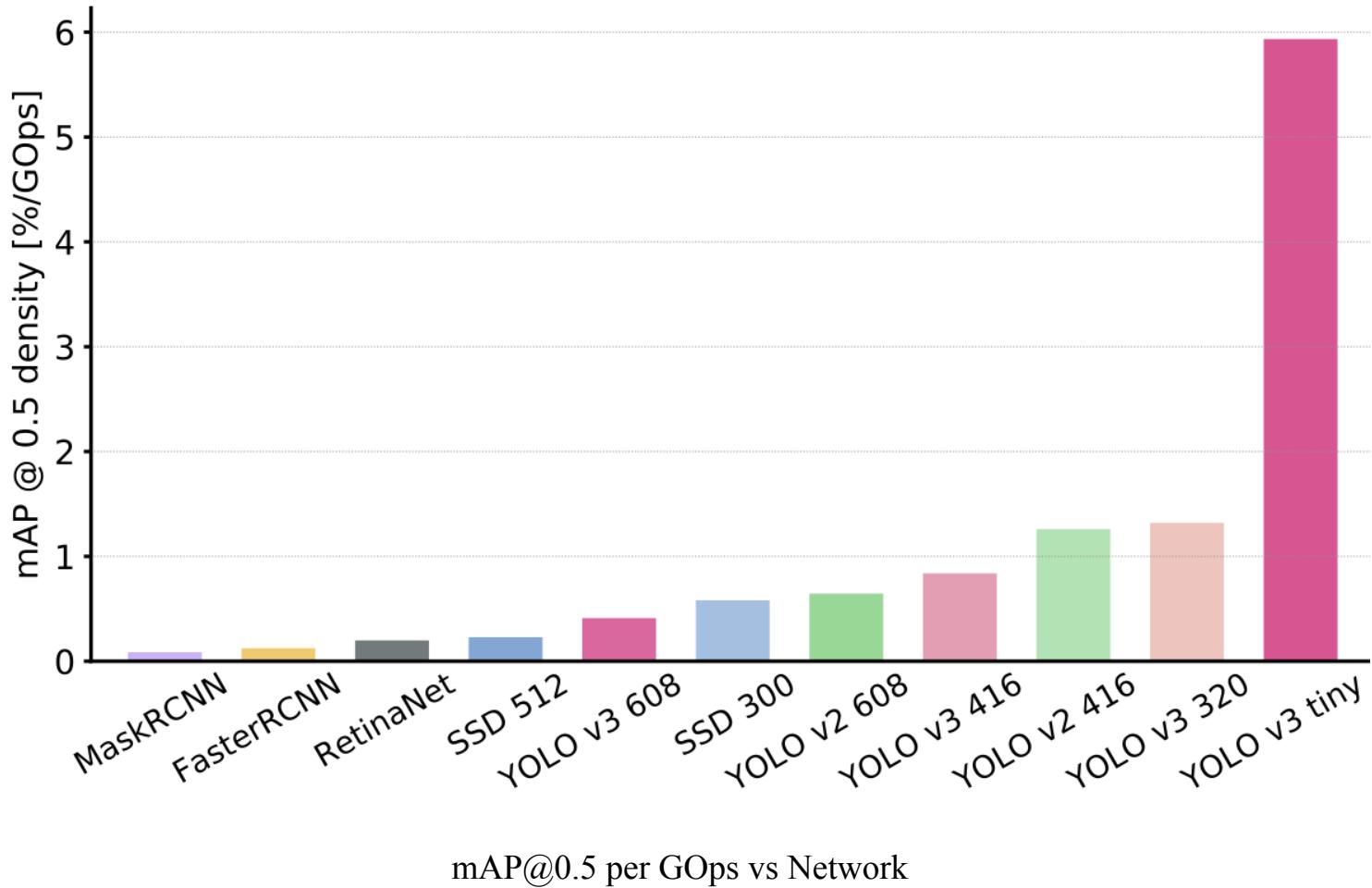
Week 4. Darknet & YOLO

Modern object detector landscape @ MSCOCO 2017 validation set





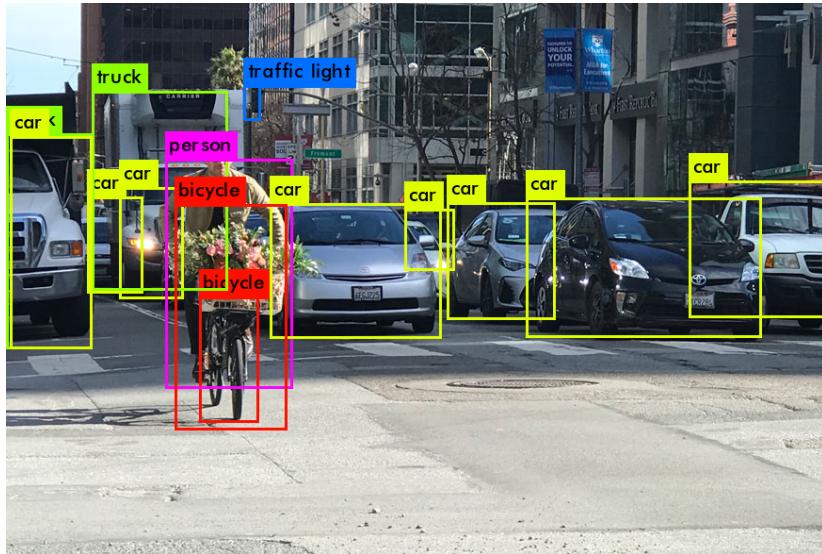




Modern object detectors

- [https://zsc.github.io/megvii-pku-dl-course/slides/Lecture6\(Object%20Detection\).pdf](https://zsc.github.io/megvii-pku-dl-course/slides/Lecture6(Object%20Detection).pdf)

YOLO - How it works

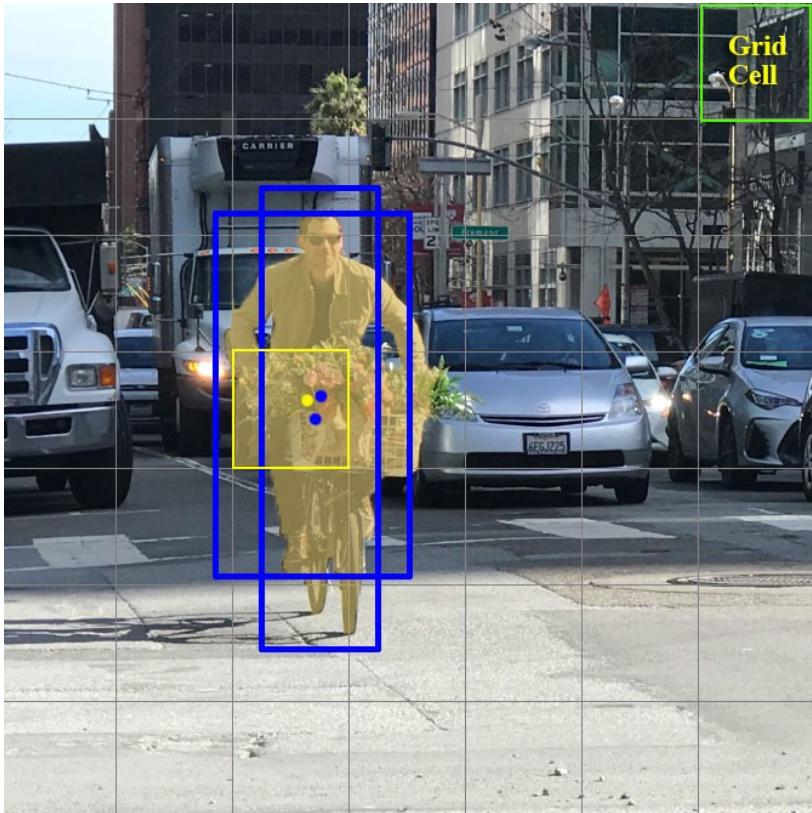


You Only Look Once: Unified, Real-Time Object Detection – CVPR 2016

Class definition

- MSCOCO: 80 classes
 - <https://github.com/pjreddie/darknet/blob/1e729804f61c8627eb257fba8b83f74e04945db7/data/coco.names>
- ImageNet: 9k classes
 - <https://github.com/pjreddie/darknet/blob/1e729804f61c8627eb257fba8b83f74e04945db7/data/9k.names>

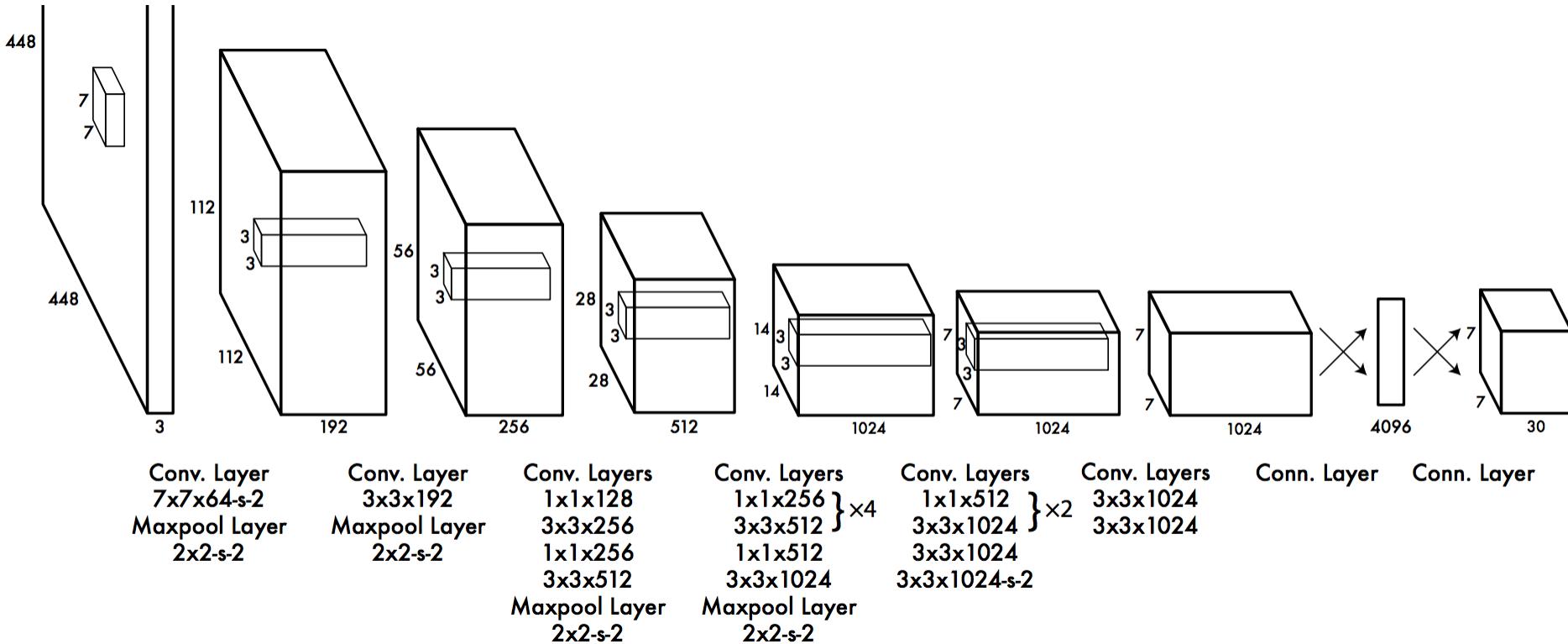
YOLO - How it works



- Divide image into grid: 7×7
- Within each grid cell:
 - Predict **one** object, with potential two different base boxes: $B = 2$
 - Regress from each of the B base boxes to a final box with 5 numbers: ($x, y, h, w, \text{box confidence score}$)
 - Predict scores for each of $C = 20$ classes
- Output: $7 \times 7 \times (5 * B + C)$

$$7 \times 7 \times (5 * 2 + 20) = 7 \times 7 \times 30$$

DarkNet-19



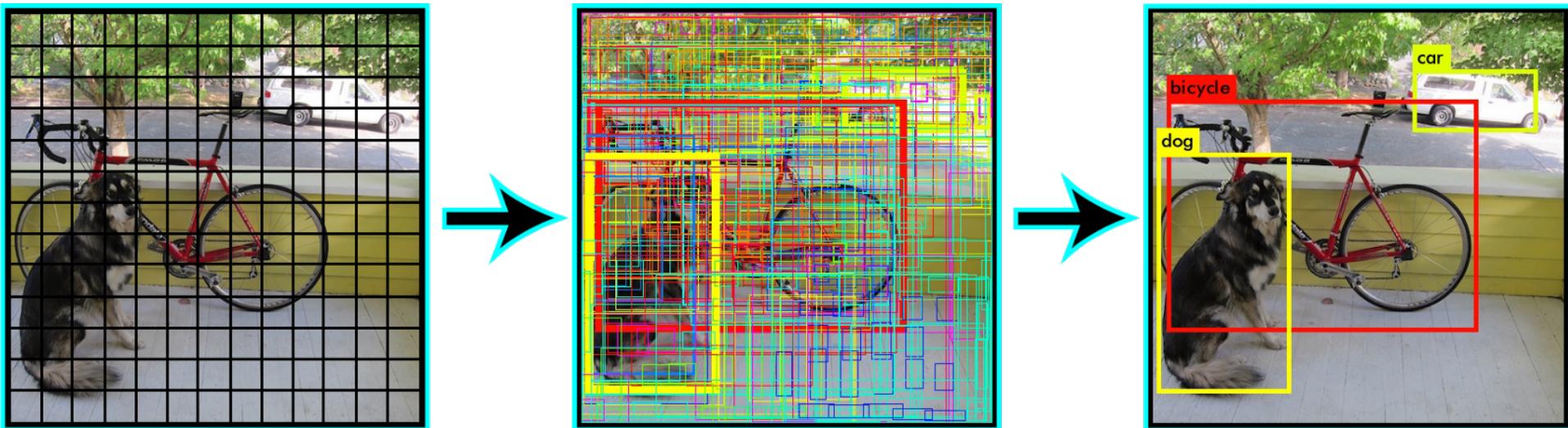
Loss function

- YOLO uses sum-squared error between the predictions and the ground truth to calculate loss. The loss function composes of:
 - the **classification loss**.
 - the **localization loss** (errors between the predicted boundary box and the ground truth).
 - the **confidence loss** (the objectness of the box).

$$\begin{aligned} & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\ & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \end{aligned}$$

YOLO - How it works

- To make a final prediction, we keep those with high box confidence scores (greater than 0.25) as our final predictions (the right picture).



$$\text{class confidence score} = \text{box confidence score} \times \text{conditional class probability}$$

YOLO advantages

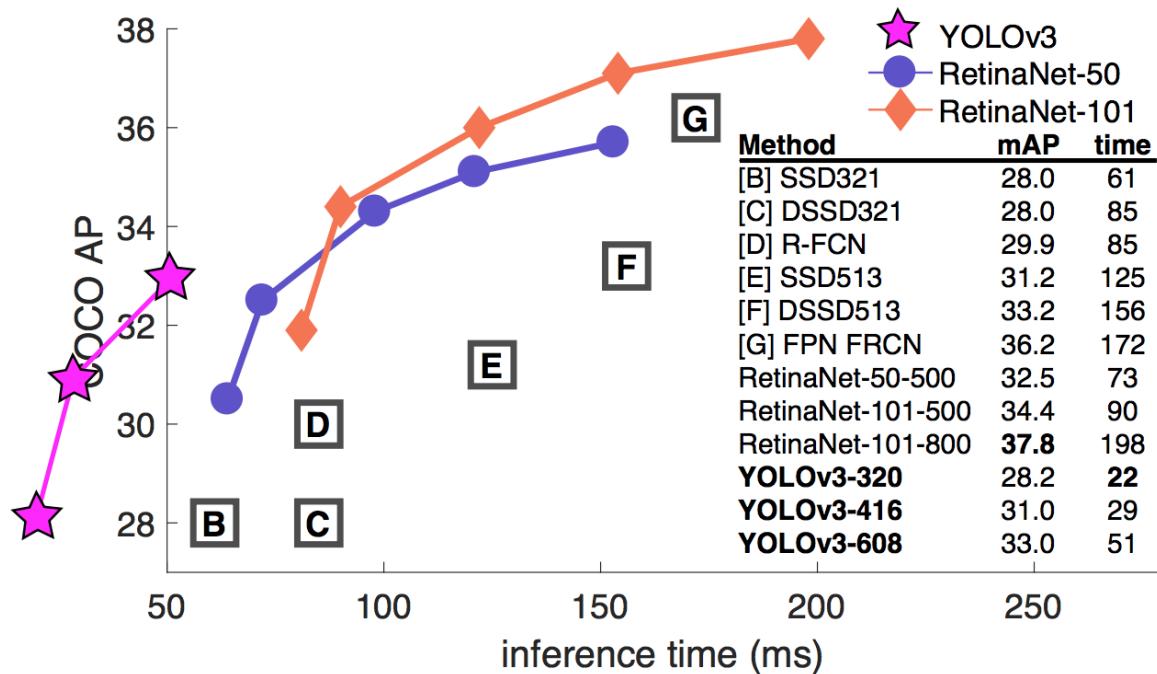
- Fast. Good for real-time processing.
- Predictions (object locations and classes) are made from one single network. Can be trained end-to-end to improve accuracy.
- YOLO is more generalized. It outperforms other methods when generalizing from natural images to other domains like artwork

YOLO9000 (v2)

	YOLO								YOLOv2
batch norm?	✓	✓	✓	✓	✓	✓	✓	✓	✓
hi-res classifier?		✓	✓	✓	✓	✓	✓	✓	✓
convolutional?			✓	✓	✓	✓	✓	✓	✓
anchor boxes?				✓	✓				
new network?					✓	✓	✓	✓	✓
dimension priors?						✓	✓	✓	✓
location prediction?						✓	✓	✓	✓
passthrough?							✓	✓	✓
multi-scale?								✓	✓
hi-res detector?									✓
VOC2007 mAP	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8	78.6

YOLOv3

- New DarkNet-53
- Feature Pyramid Networks (FPN) like Feature Pyramid
- New loss function



DarkNet-19

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2 / 2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2 / 2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2 / 2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2 / 2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2 / 2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

DarkNet-53

Type	Filters	Size	Output
Convolutional	32	3×3	256×256
Convolutional	64	$3 \times 3 / 2$	128×128
1x	32	1×1	
	64	3×3	
	Residual		128×128
2x	128	$3 \times 3 / 2$	64×64
	64	1×1	
	128	3×3	
8x	Residual		64×64
	256	$3 \times 3 / 2$	32×32
	128	1×1	
8x	256	3×3	
	Residual		32×32
	512	$3 \times 3 / 2$	16×16
8x	256	1×1	
	512	3×3	
	Residual		16×16
4x	1024	$3 \times 3 / 2$	8×8
	512	1×1	
	1024	3×3	
	Residual		8×8
Avgpool		Global	
Connected		1000	
Softmax			

YOLO - How it works

- https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088
- Implementations from scratch:
 - Tensorflow: <https://itnext.io/implementing-yolo-v3-in-tensorflow-tf-slim-c3c55ff59dbe>
 - Pytorch: <https://www.kdnuggets.com/2018/05/implement-yolo-v3-object-detector-pytorch-part-1.html>

Demo

- Detection Using A Pre-Trained Model with python
 - Detect objects with a pre-trained YOLOv3 on the COCO dataset (80classes)
- Detection Using A Pre-Trained Model with executable
 - <https://pjreddie.com/darknet/yolo/>

Transfer learning with YOLO – Detect your own objects

- Detect a single object YOLOv2
 - https://medium.com/@manivannan_data/how-to-train-yolov2-to-detect-custom-objects-9010df784f36
- Detect multiple objects YOLOv2
 - https://medium.com/@manivannan_data/how-to-train-multiple-objects-in-yolov2-using-your-own-dataset-2b4fee898f17
- Detect multiple objects with YOLOv3
 - https://medium.com/@manivannan_data/how-to-train-yolov3-to-detect-custom-objects-ccbcafeb13d2

Transfer learning with YOLO – Detect your own objects

- Detect a single object YOLOv2
 - https://medium.com/@manivannan_data/how-to-train-yolov2-to-detect-custom-objects-9010df784f36
- Detect multiple objects YOLOv2
 - https://medium.com/@manivannan_data/how-to-train-multiple-objects-in-yolov2-using-your-own-dataset-2b4fee898f17
- Detect multiple objects with YOLOv3
 - https://medium.com/@manivannan_data/how-to-train-yolov3-to-detect-custom-objects-ccbcafeb13d2

YOLOv3 on Google Colab

- Detect objects with a pre-trained YOLOv3 on the COCO dataset (80classes)
- OpenCV
- All my tutorial codes are here:
 - YOLOv3: https://colab.research.google.com/drive/1mpOqyC9bSM6Gup5M9lBkB16I1yo_V7Db
 - YOLOv3-tiny: <https://colab.research.google.com/drive/1NBGu-26xviYDi4Gi2p5mClid5AtCNe9d>