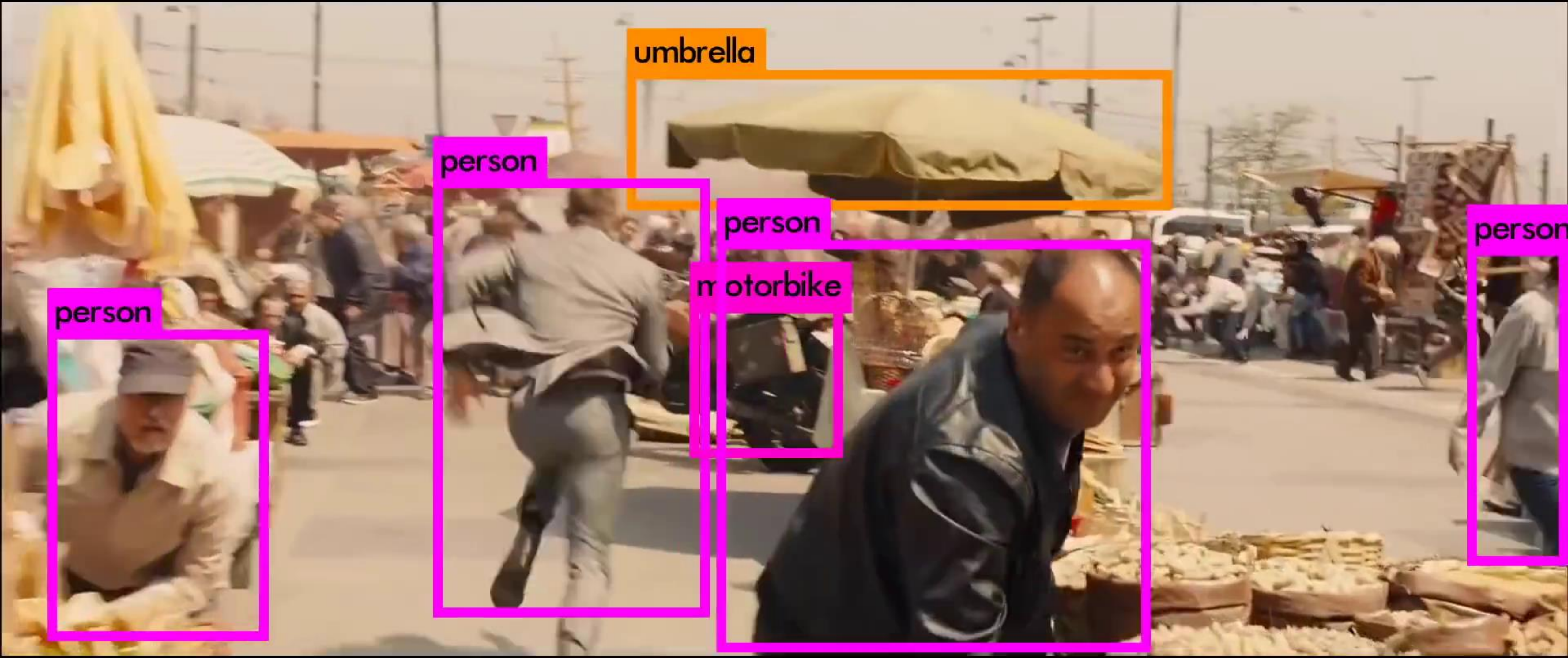
Object Detection

Prof. Kuan-Ting Lai

2020/5/5


kite: 97%
kite: 82%
kite: 90%
kite: 85%
kite: 54%
kite: 92%
person: 76%
person: 54%
pers
person: 97%
person: 86%
person: 99%
person: 99%

YOLO v2



https://www.youtube.com/watch?v=VOC3huqHrss&t=40s

Detection vs Classification

- Classification
  - Ex: ImageNet Large-scale Visual Recognition Challenge (Classify 1000 categories)

- Detection = Binary Classification

Recent Developments of Object Detection

- Deformable Part Model (2010)
- Fast R-CNN (2015)
- Faster R-CNN (2015)
- You Only Look Once: Unified, real-time object detection (2016)
- SSD: Single-Shot Multi-box Detector (2016)
- Mask R-CNN (2017) (Segmentation)
- YOLO9000: Better, Faster, Stronger (2017)
- YOLOv3: An Incremental Improvement (2018)

Ross Girshick

FOLLOW

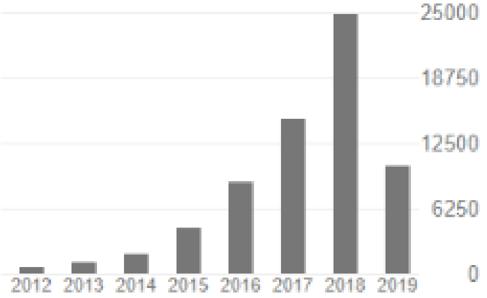
Research Scientist, Facebook AI Research (FAIR)
Verified email at eecs.berkeley.edu - Homepage

computer vision machine learning

TITLE	CITED BY	YEAR
Caffe: Convolutional architecture for fast feature embedding Y Jia, E Shelhamer, J Donahue, S Karayev, J Long, R Girshick, ... Proceedings of the 22nd ACM international conference on Multimedia, 675-678	10729	2014
Faster R-CNN: Towards real-time object detection with region proposal networks S Ren, K He, R Girshick, J Sun Advances in neural information processing systems, 91-99	9521	2015
Rich feature hierarchies for accurate object detection and semantic segmentation R Girshick, J Donahue, T Darrell, J Malik Proceedings of the IEEE conference on computer vision and pattern ...	9074	2014
Object detection with discriminatively trained part-based models PF Felzenszwalb, RB Girshick, D McAllester, D Ramanan Pattern Analysis and Machine Intelligence, IEEE Transactions on 32 (9), 1627 ...	7992	2010
Fast R-CNN R Girshick Proceedings of the IEEE International Conference on Computer Vision, 1440-1448	5432	2015
Microsoft coco: Common objects in context TY Lin, M Maire, S Belongie, J Hays, P Perona, D Ramanan, P Dollár, ... European conference on computer vision, 740-755	4868	2014
You only look once: Unified, real-time object detection J Redmon, S Divvala, R Girshick, A Farhadi Proceedings of the IEEE conference on computer vision and pattern ...	4197	2016
Mask R-CNN K He, G Gkioxari, P Dollár, R Girshick arXiv preprint arXiv:1703.06870	2183	2017
Feature pyramid networks for object detection TY Lin, P Dollár, R Girshick, K He, B Hariharan, S Belongie Proceedings of the IEEE Conference on Computer Vision and Pattern ...	1166	2017
Aggregated residual transformations for deep neural networks S Xie, R Girshick, P Dollár, Z Tu, K He Proceedings of the IEEE conference on computer vision and pattern ...	942	2017

Cited by

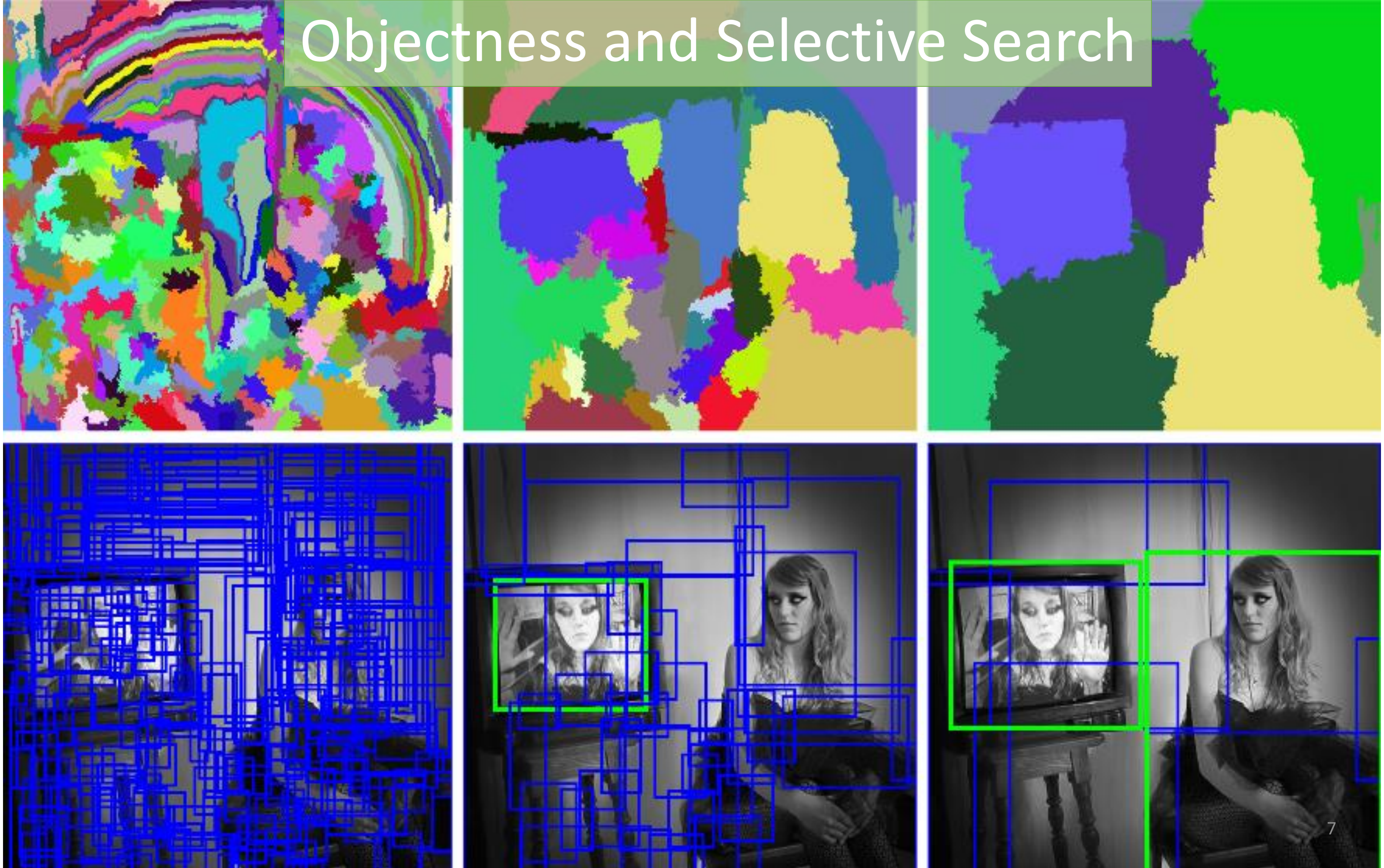
	All	Since 2014
Citations	68366	65802
h-index	53	53
i10-index	72	70



Co-authors VIEW ALL

- Kaiming He — Research Scientist, Facebook AI ...
- Trevor Darrell — Professor of Computer Science, ...
- Jitendra Malik — Professor of EECS, UC Berkeley
- Piotr Dollár — Facebook AI Research
- Jeff Donahue — Research Scientist, DeepMind
- Pedro Felzenszwalb — Brown University
- bharath hariharan — Cornell University
- C. Lawrence Zitnick — Facebook AI Research
- David McAllester — Professor, Toyota Technological I...

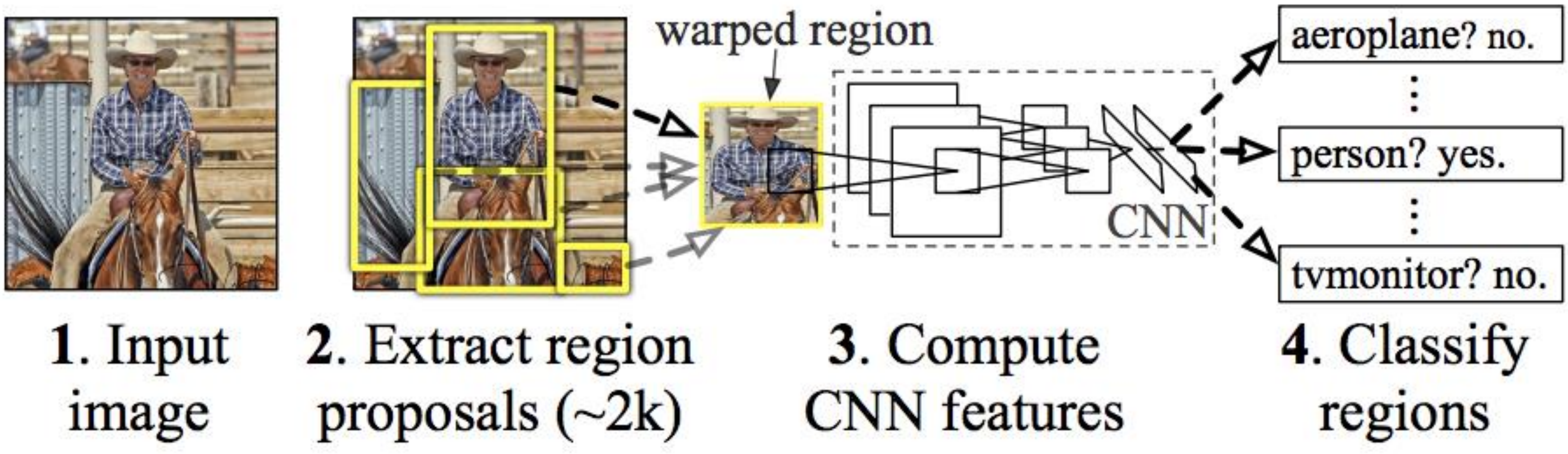
Objectness and Selective Search

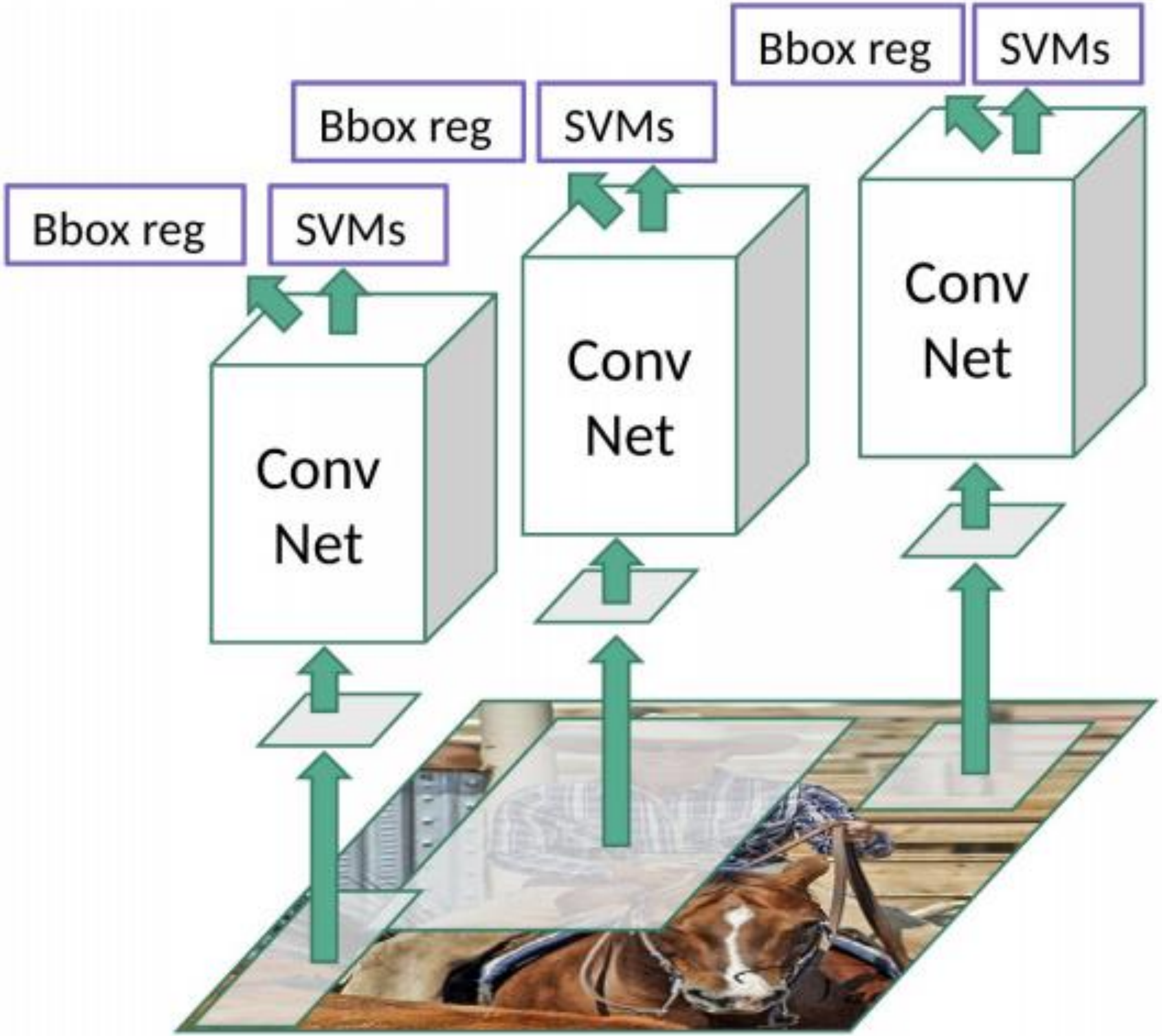


Region Proposal: Multi-scale Objectness Search

- Scan all possible locations and scales for objects

Region Proposal + CNN = R-CNN



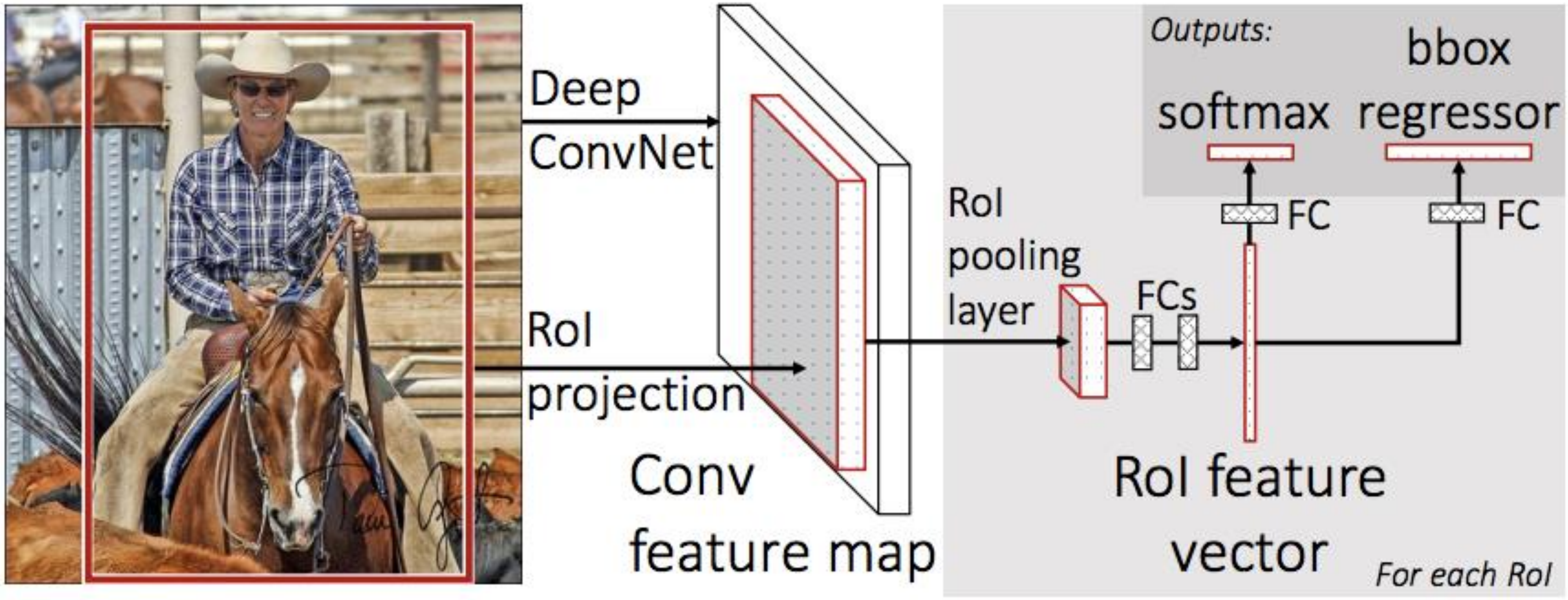

Bbox reg
SVMs
Bbox reg
SVMs
Bbox reg
SVMs
Conv
Net
Conv
Net
Conv
Net

Problems with R-CNN

- 2000 region proposals per image
- It takes around 47 seconds for testing one image
- The selective search algorithm is a fixed algorithm using shallow architecture

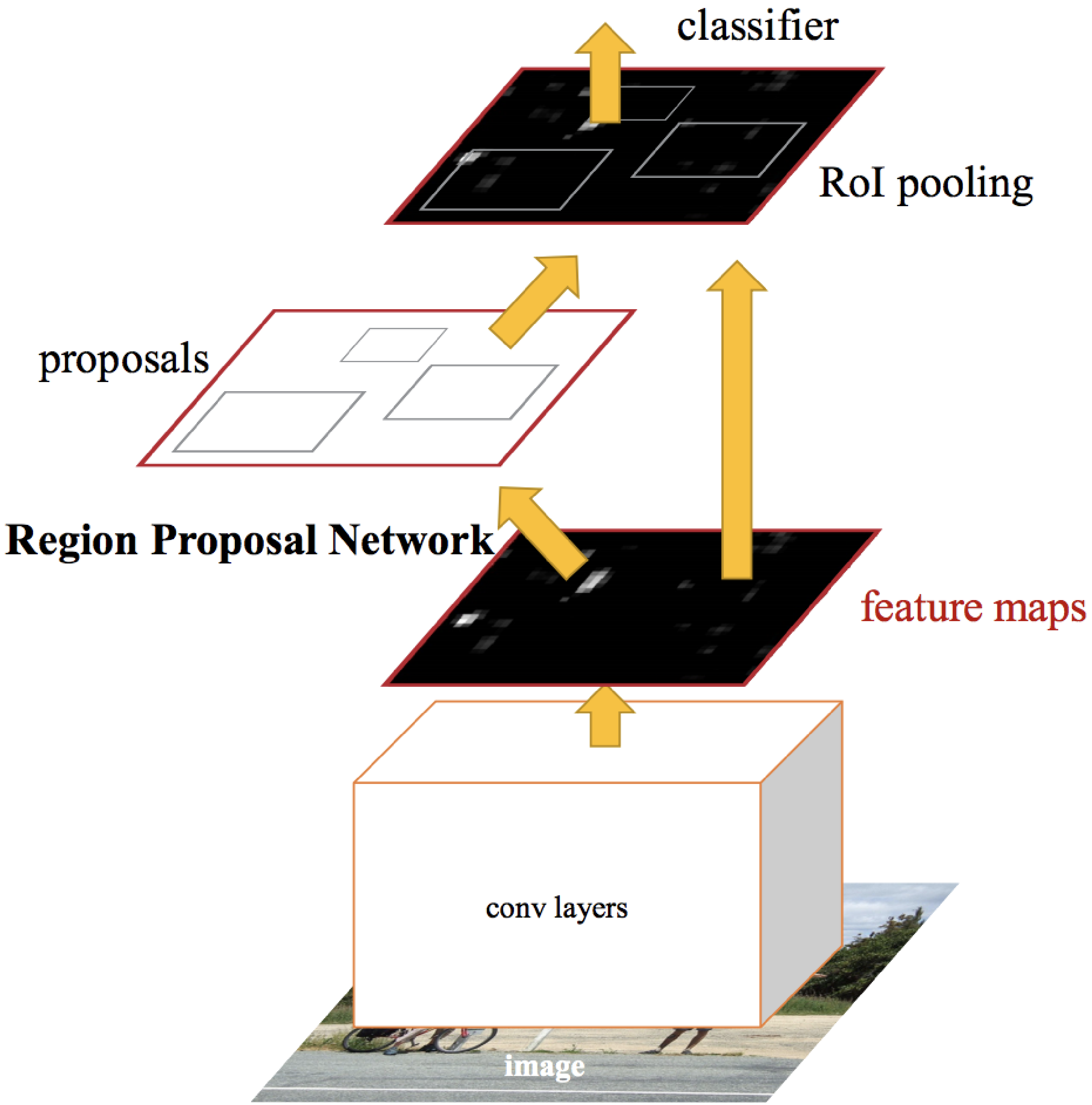
Fast R-CNN

- Instead of running a CNN 2,000 times per image, run just once per image and get all the regions of interest (RoI)

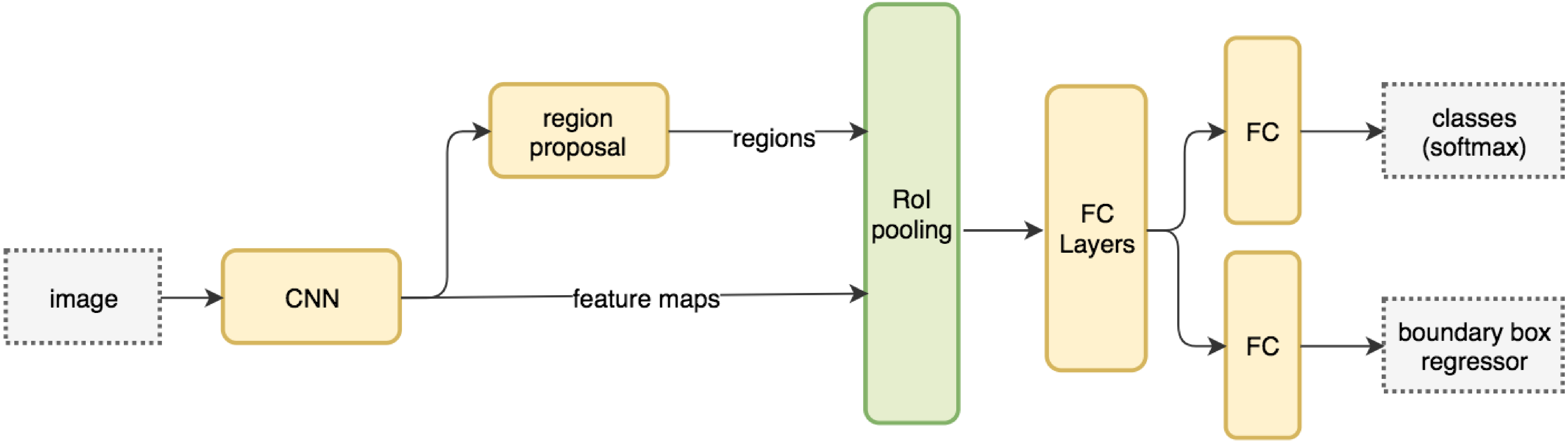


Faster R-CNN

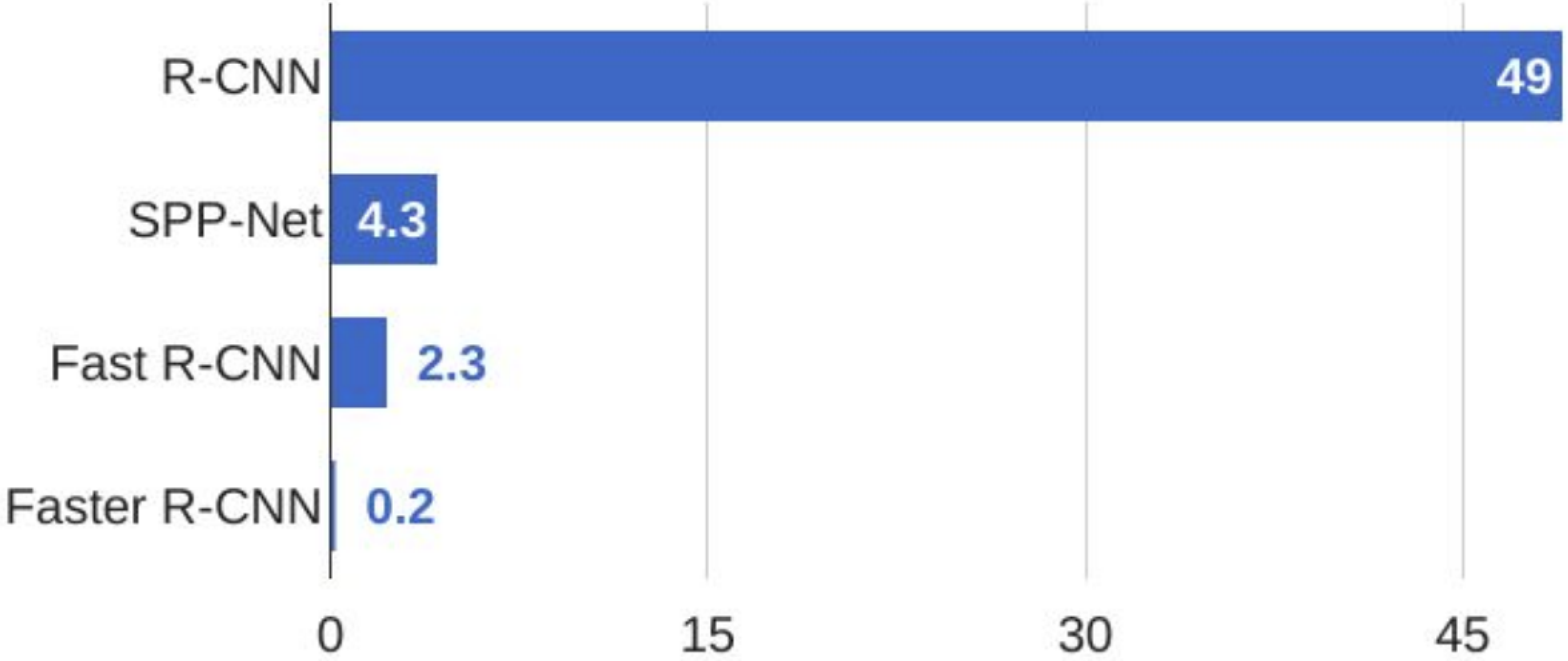
- Replace Selective Search with neural networks



Faster R-CNN Architecture



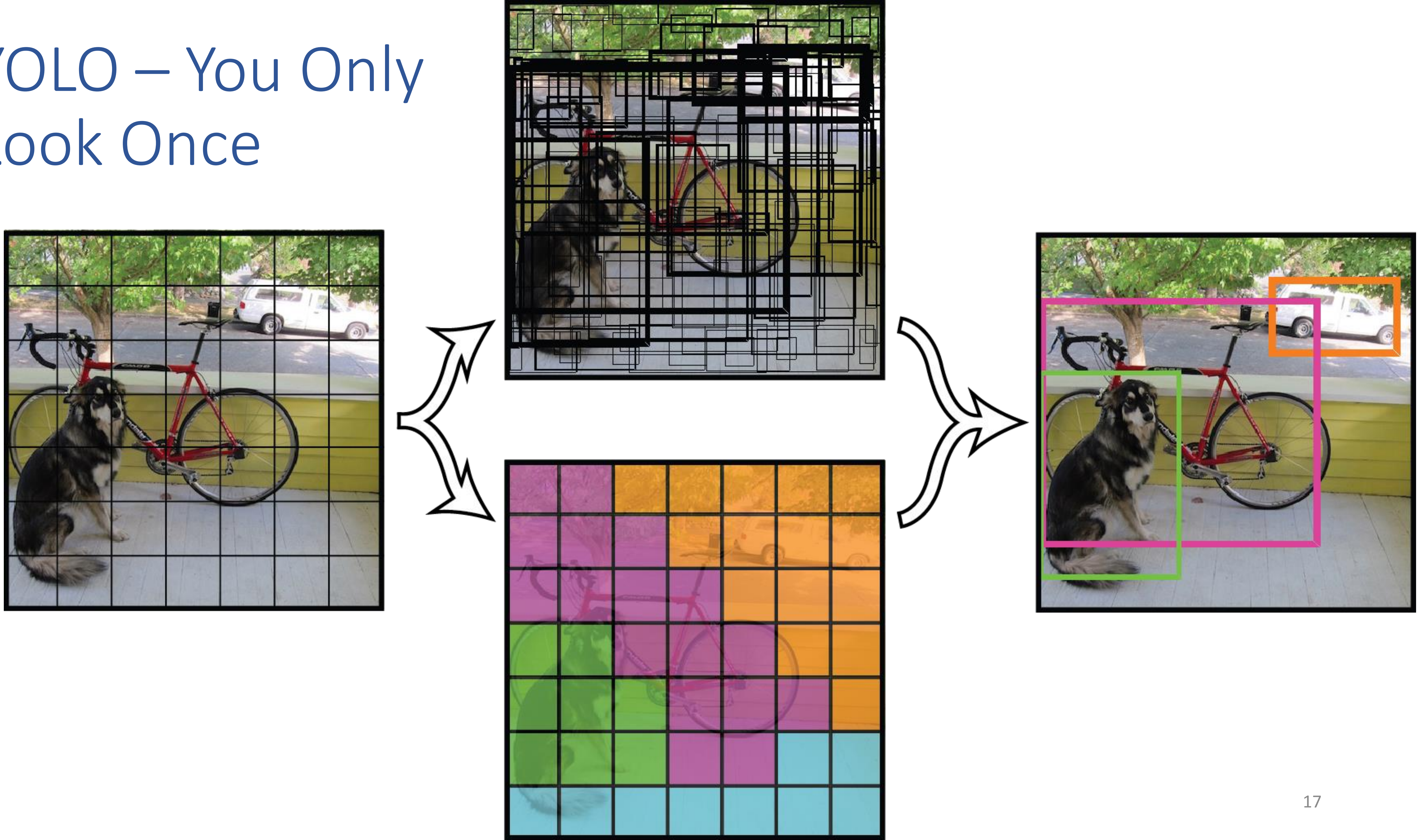
R-CNN Test-Time Speed



Summary

Algorithm	Features	Prediction time	Limitations
RCNN	• Uses selective search to generate regions. • Extracts around 2000 regions from each image.	40-50 secs	High computation time as each region is passed to the CNN separately
Fast RCNN	• Each image is passed only once to the CNN and feature maps are extracted. • Selective search is used on these maps to generate predictions.	2 secs	Selective search is slow and hence computation time is still high.
Faster RCNN	• Replaces the selective search method with region proposal network.	0.2 secs	Object proposal takes time

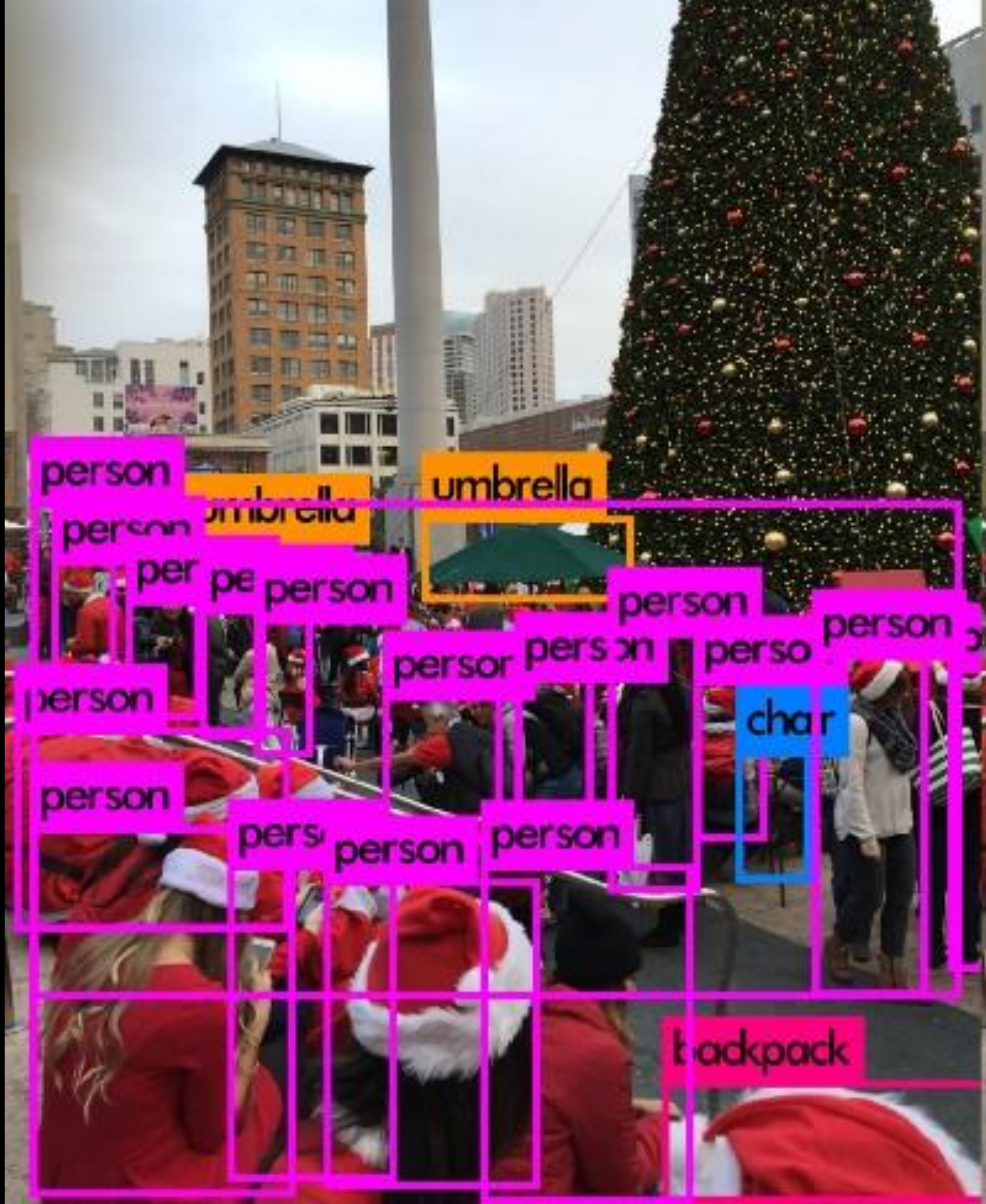
YOLO – You Only Look Once



YOLO v1

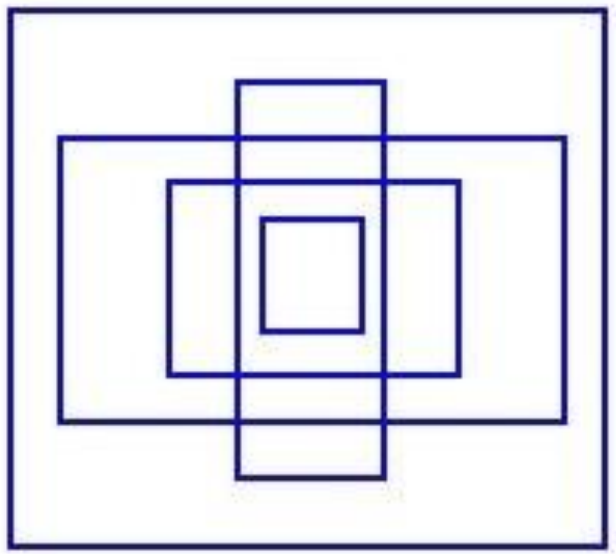
- Divide an image into S x S grid
- Predict bounding box B as (x, y, w, h, confidence)
- Each grid predicts B bounding boxes and C class probabilities
- Final prediction: S x S x (B*5 + C)

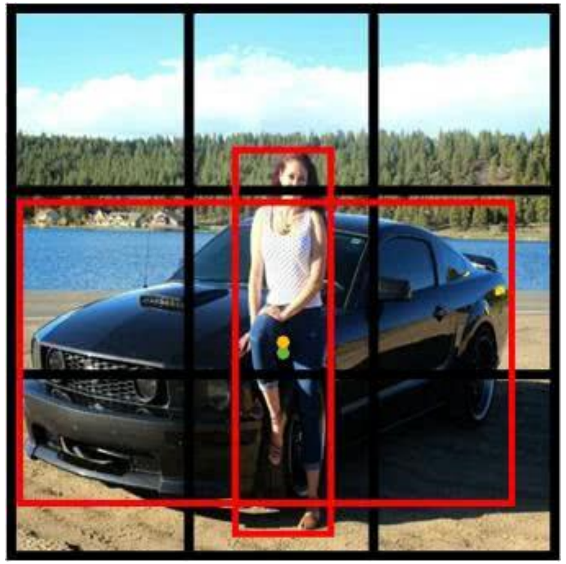
Limitation of YOLO



YOLO v2 – YOLO 9000

- Batch normalization
- High-resolution classifier
- Convolutional with Anchor Boxes

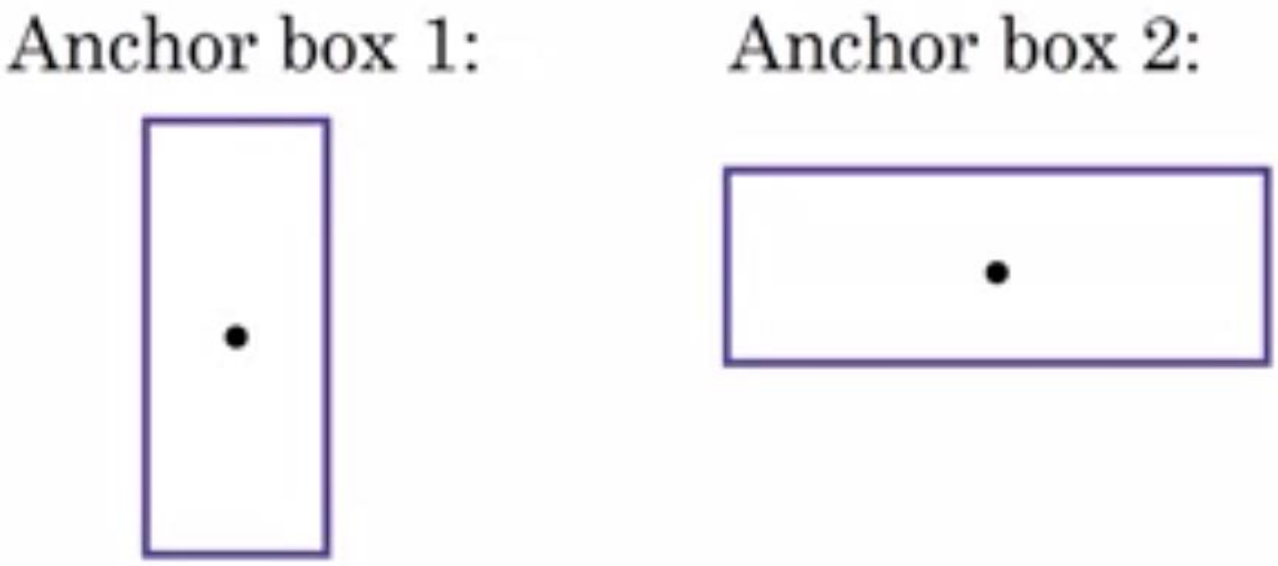




https://heartbeat.fritz.ai/gentle-guide-on-how-yolo-object-localization-works-with-keras-part-2-65fe59ac12d

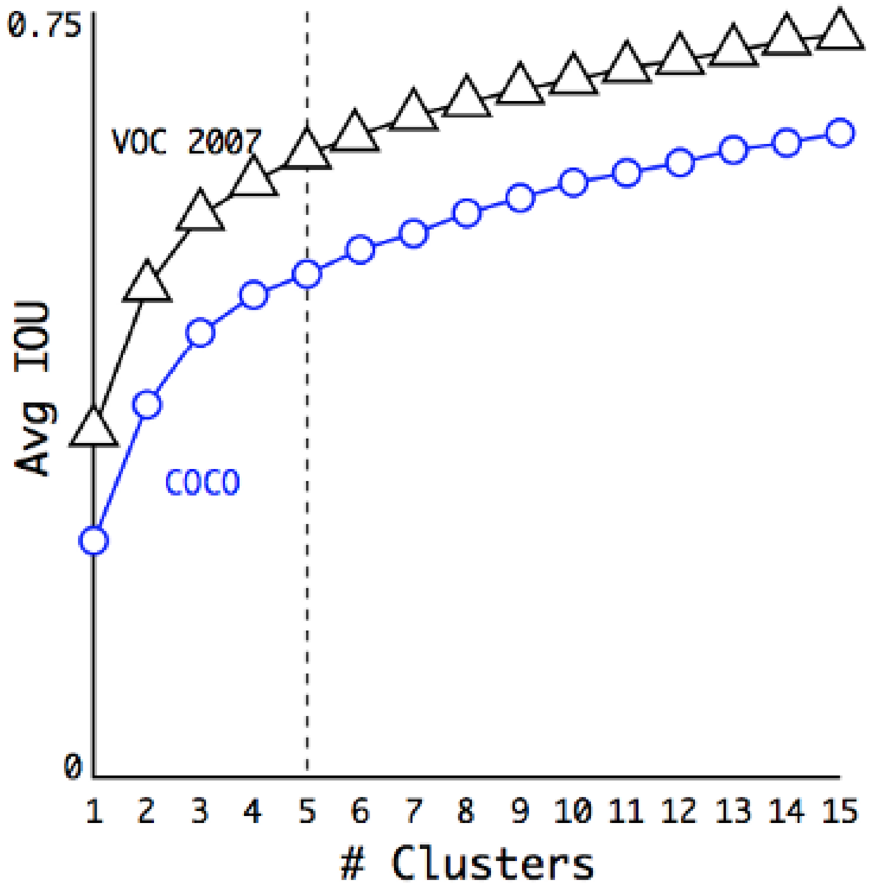
Anchor Boxes

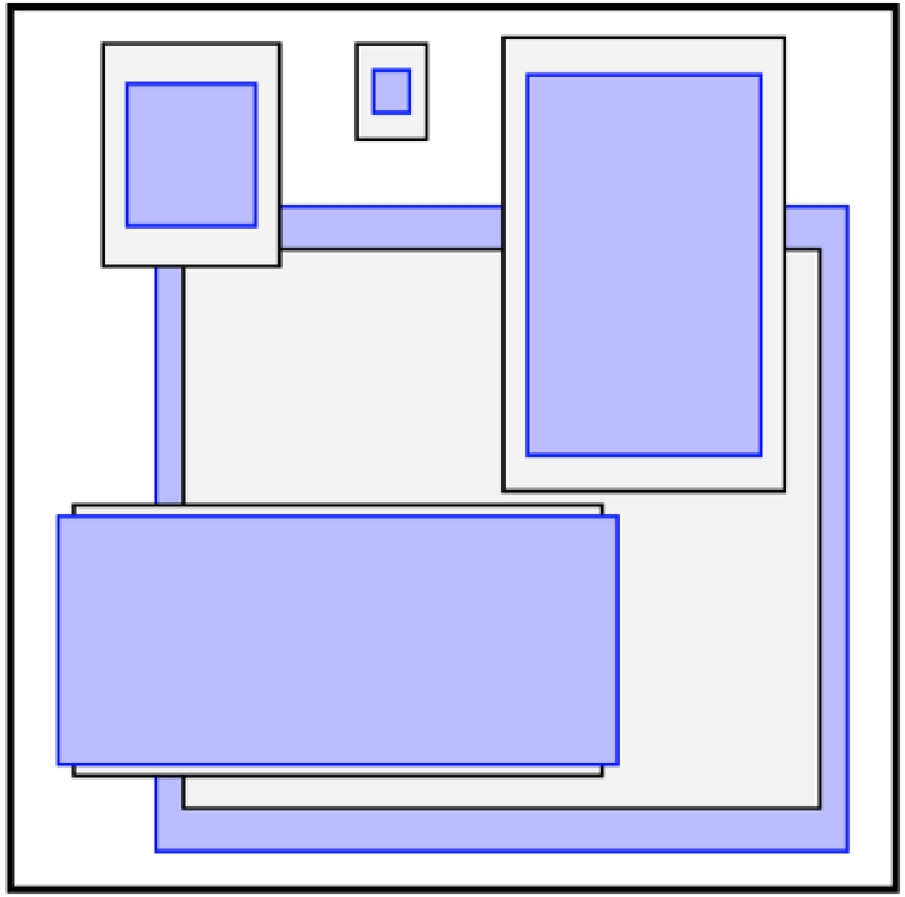
- Detecting objects with different shapes
- Detecting overlapping windows



https://www.coursera.org/lecture/convolutional-neural-networks/anchor-boxes-yNwO0

Using K-means Clustering to Find Anchor Boxes





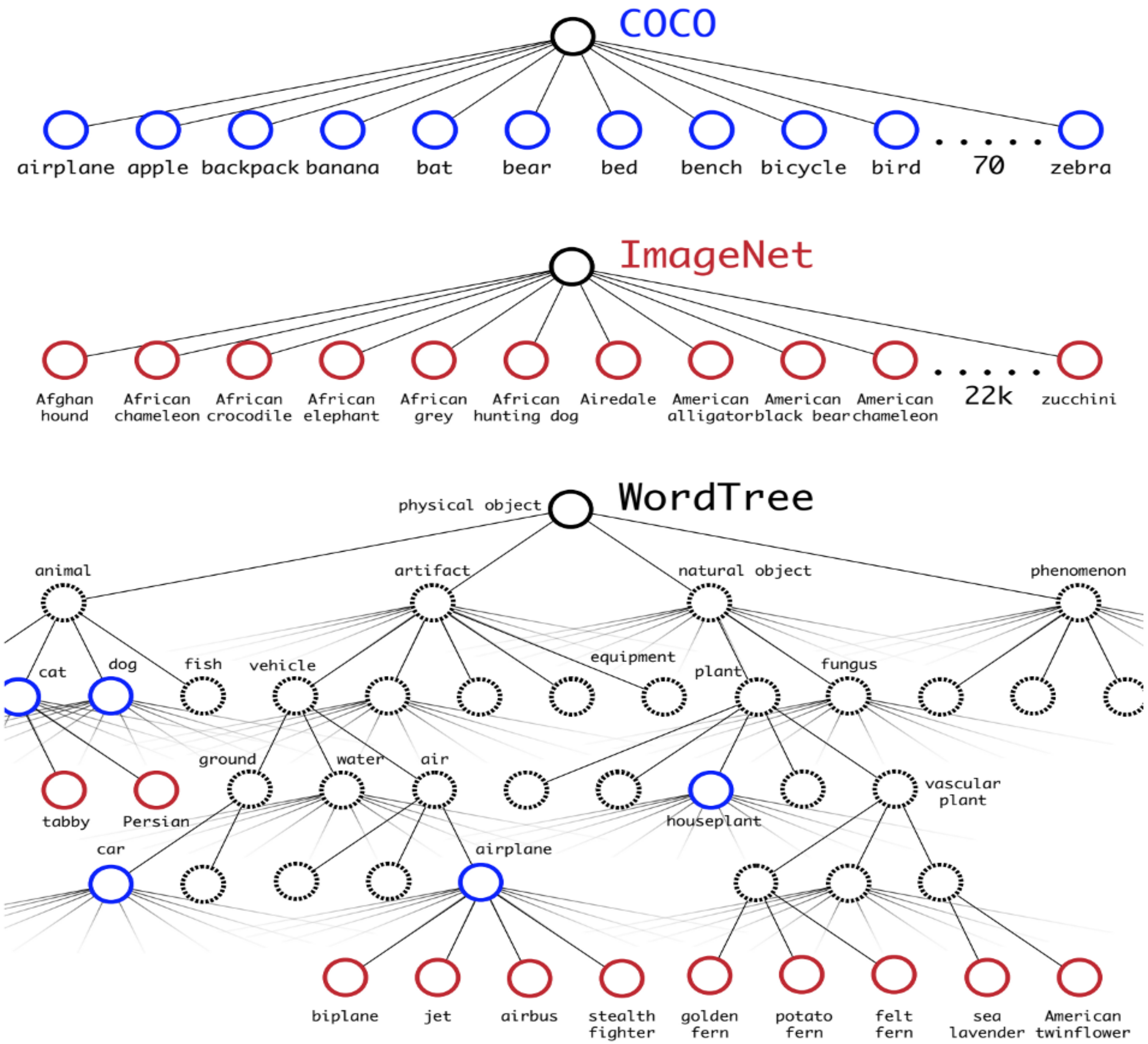
DarkNet

- For ImageNet
  - VGG (30.69 billion FLOPS)
  - GoogLeNet (8.52 billion FLOPS)
  - DarkNet (5.58 billion FLOPS)

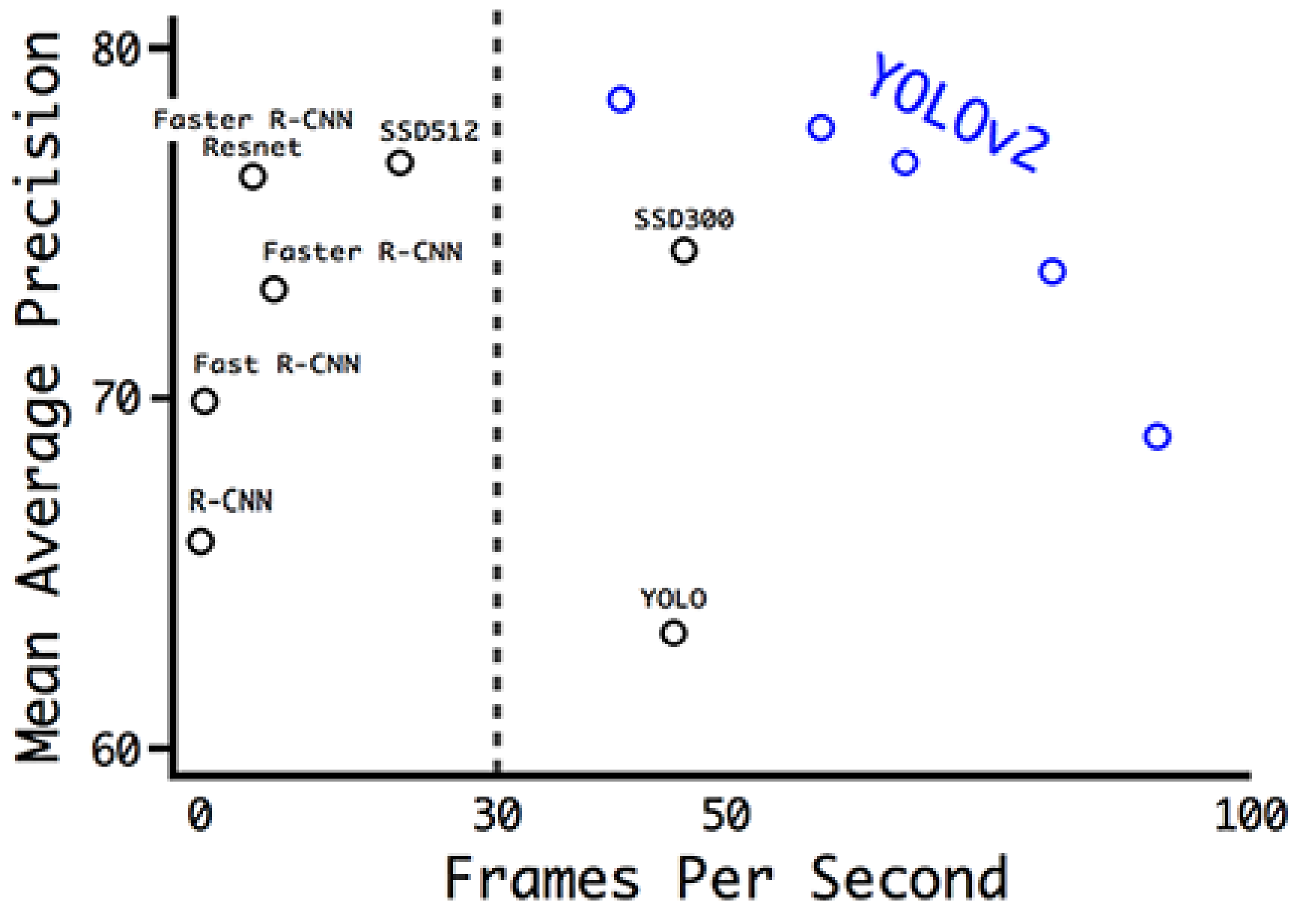
- DarkNet uses mostly 3 × 3 filters to extract features and 1 × 1 filters to reduce output channels

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			

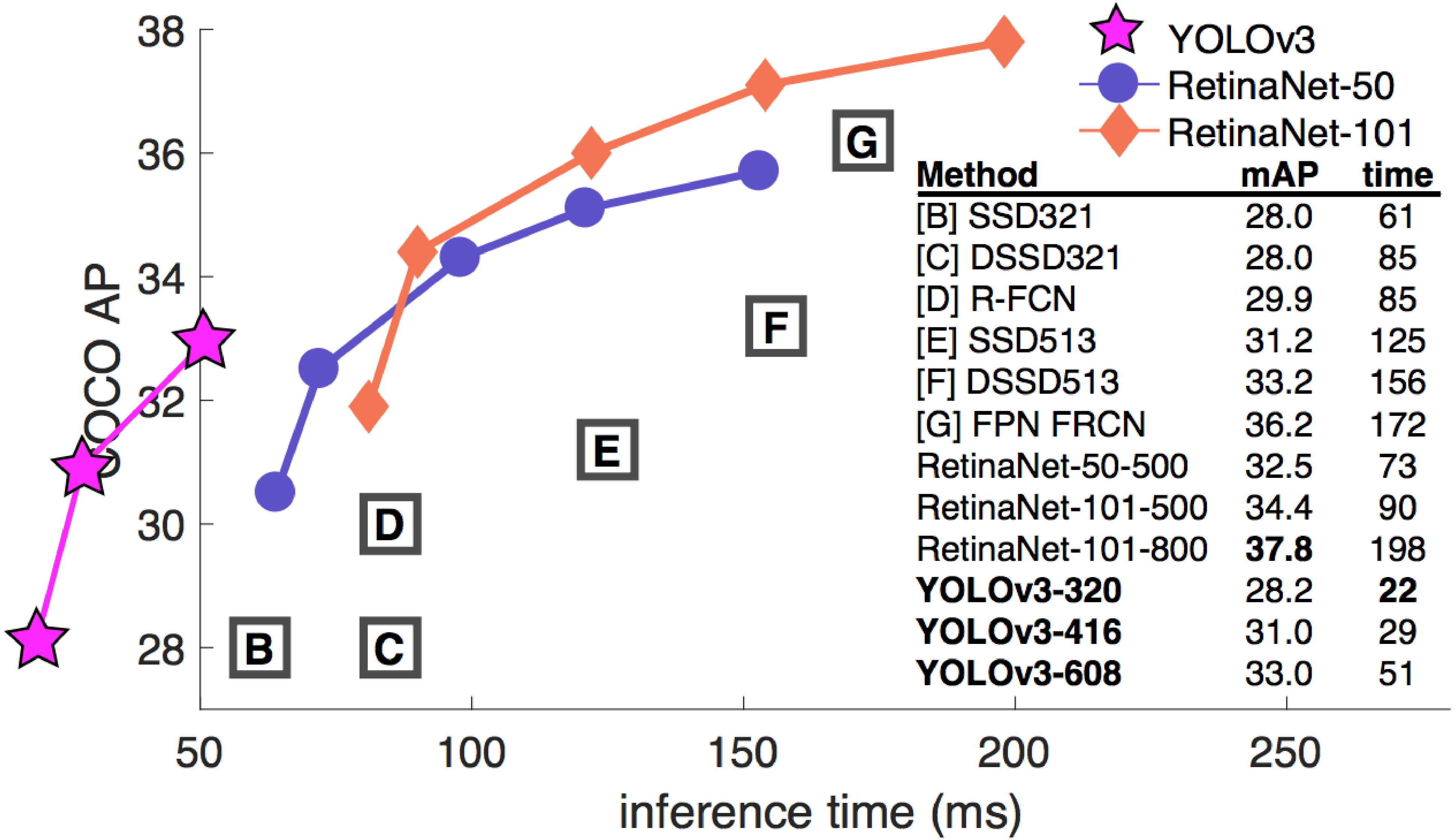
Hierarchical Classification



Performance of YOLOv2 on VOC 2007



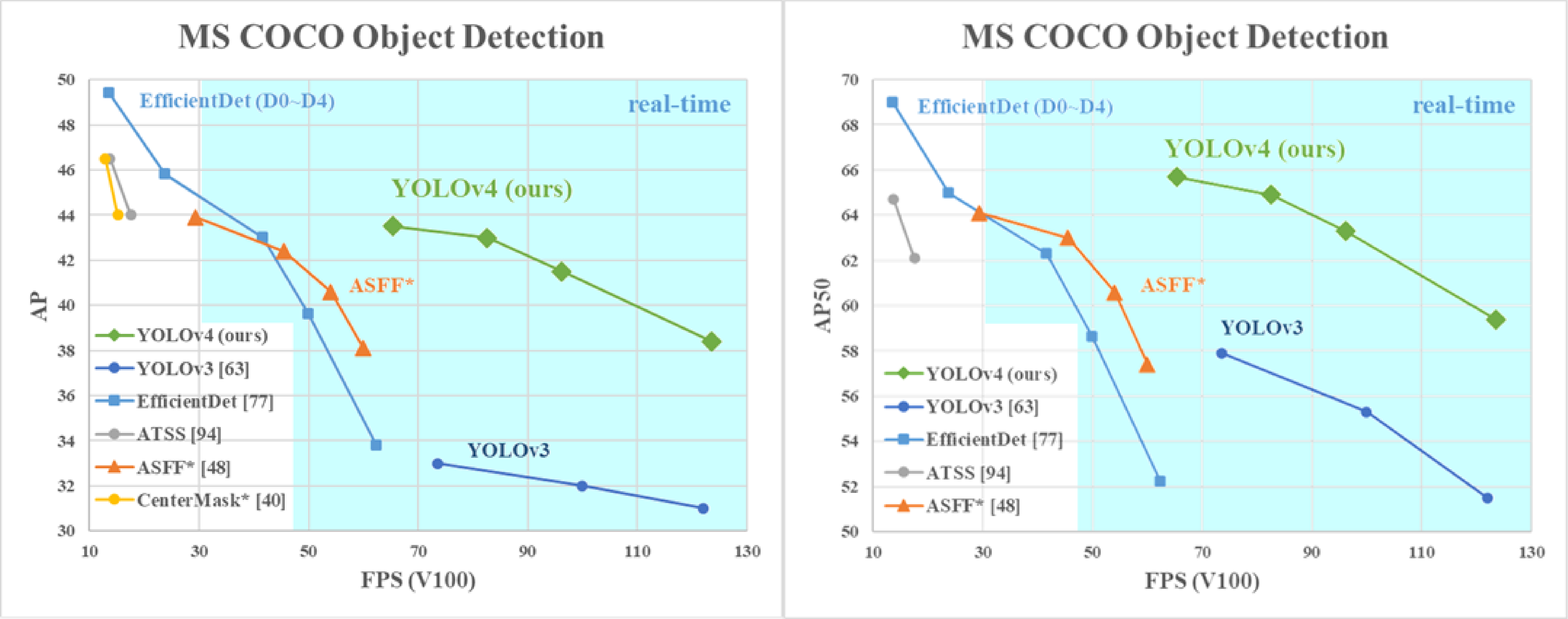
YOLO v3



Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.2	**22**
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51

YOLO v4

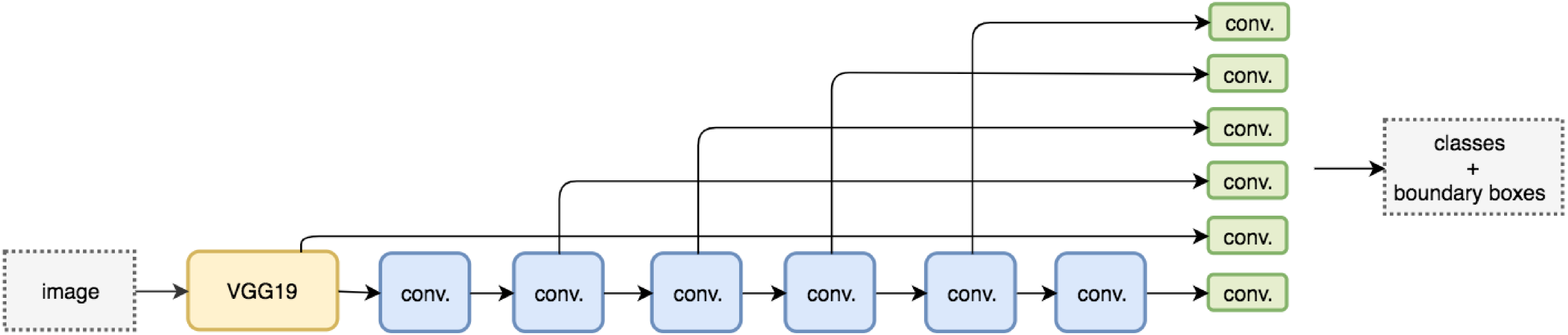
- A. Bochkovskiy, C.-Y. Wang, H.-Y. Mark Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection", 2020
- https://github.com/AlexeyAB/darknet



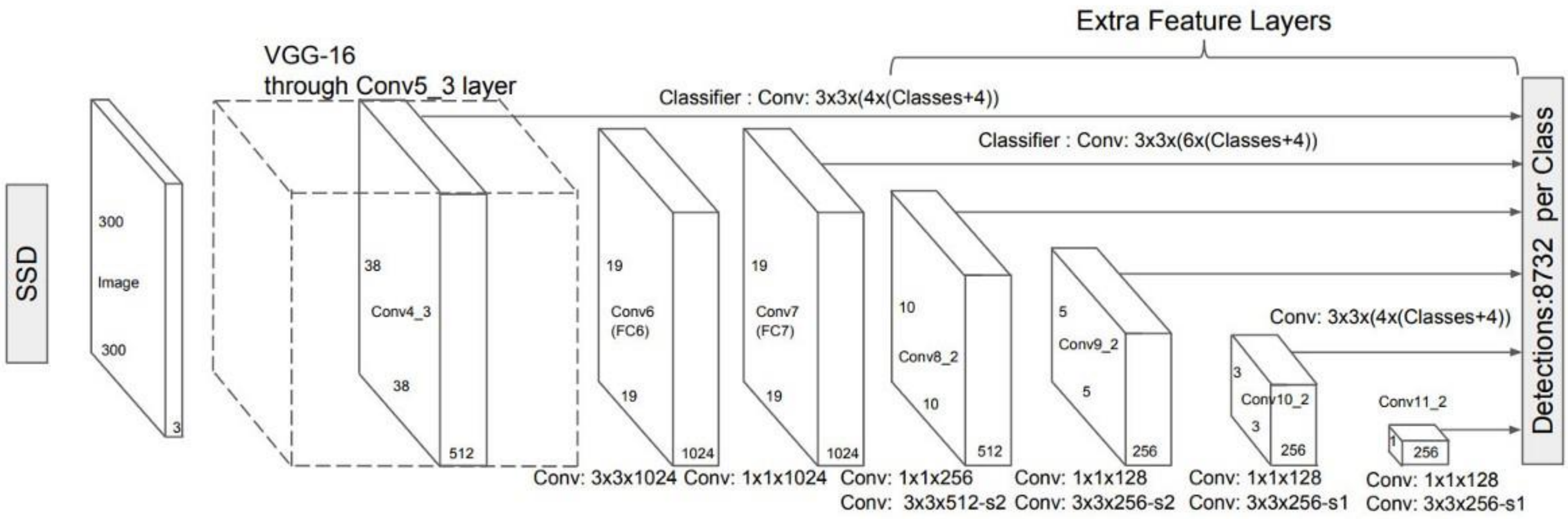
New Techniques Adopted in YOLO v4

- Weighted-Residual-Connections (WRC),
- Cross-Stage-Partial-connections (CSP)
- Cross mini-Batch
- Normalization (CmBN)
- Self-adversarial-training (SAT)
- Mish-activation
- New features:
  - WRC, CSP, CmBN, SAT, Mish activation, Mosaic data augmentation, CmBN, DropBlock regularization, and CIoU loss

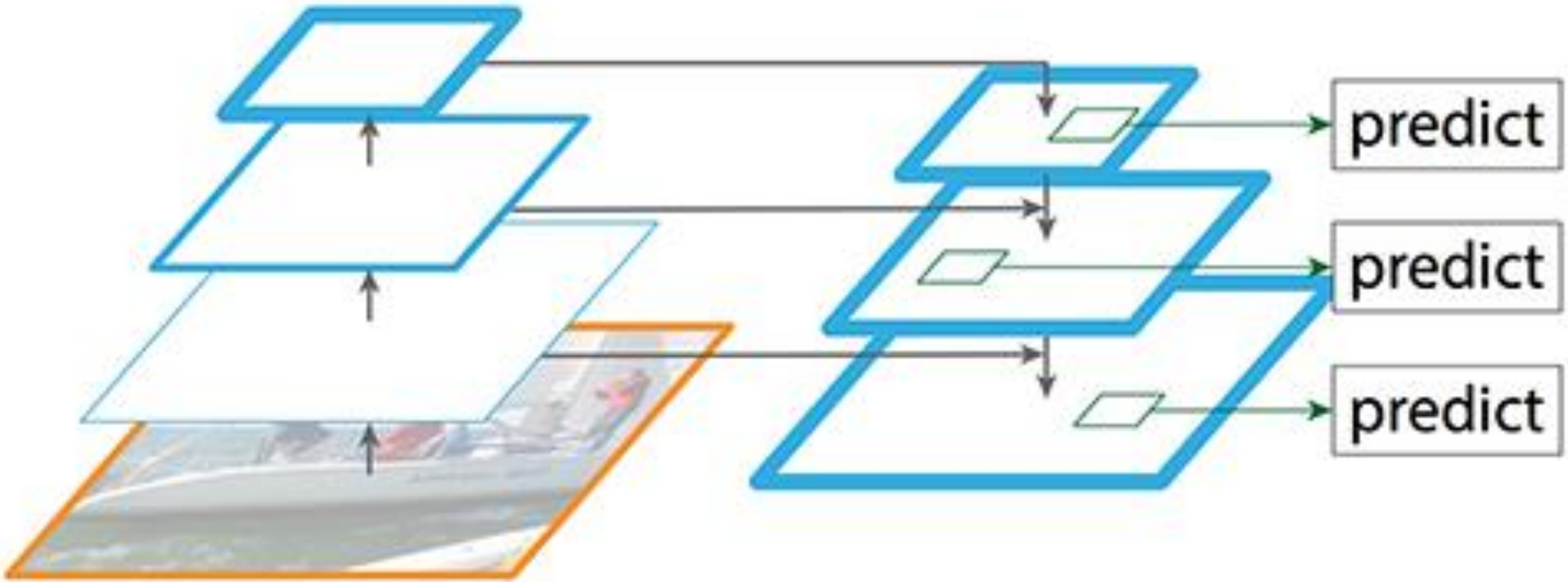
Single-Shot Multi-Box Object Detection (SSD)



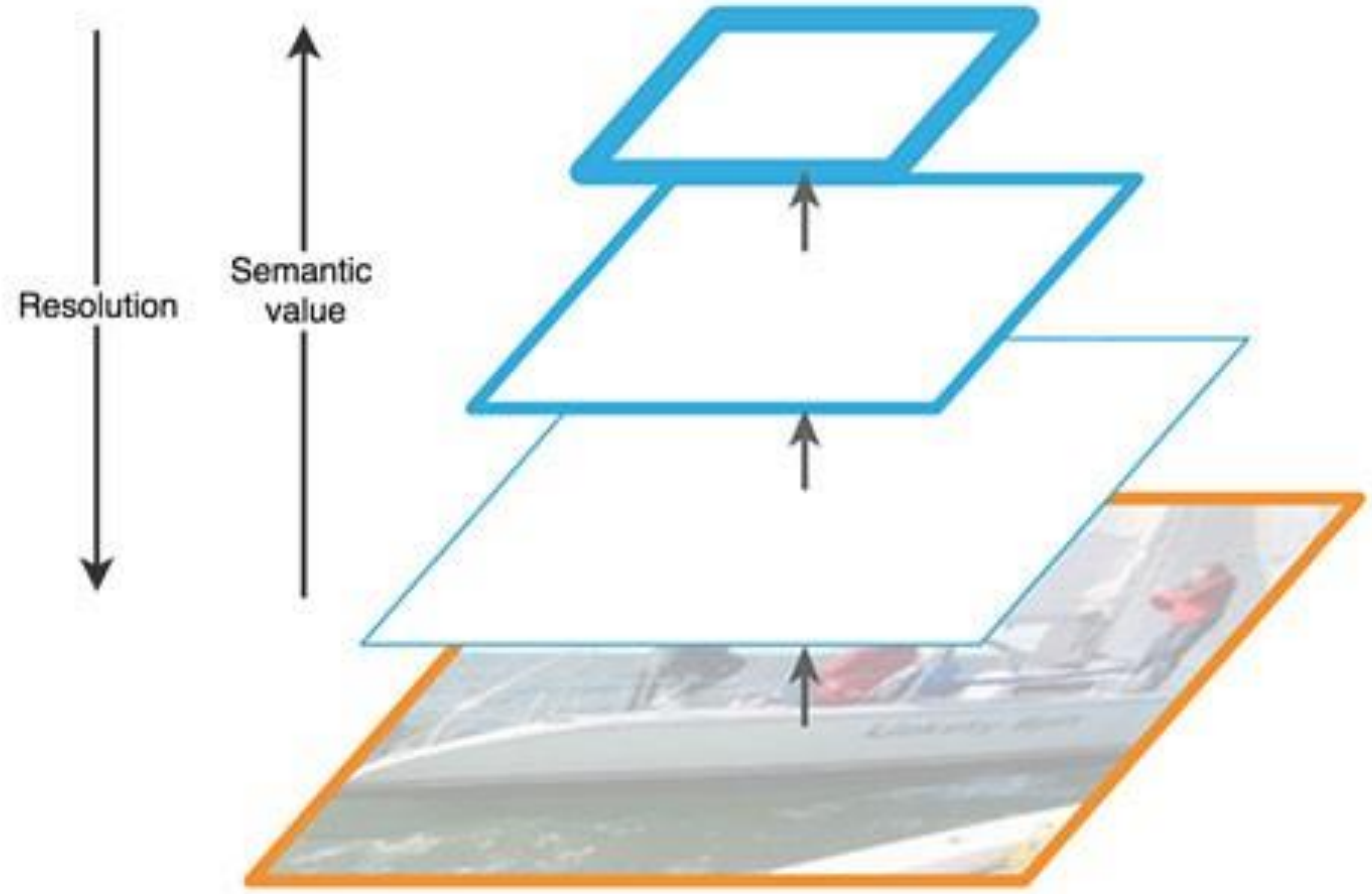
Dimensions of SSD Feature Maps



Feature Pyramid Networks (FPN)

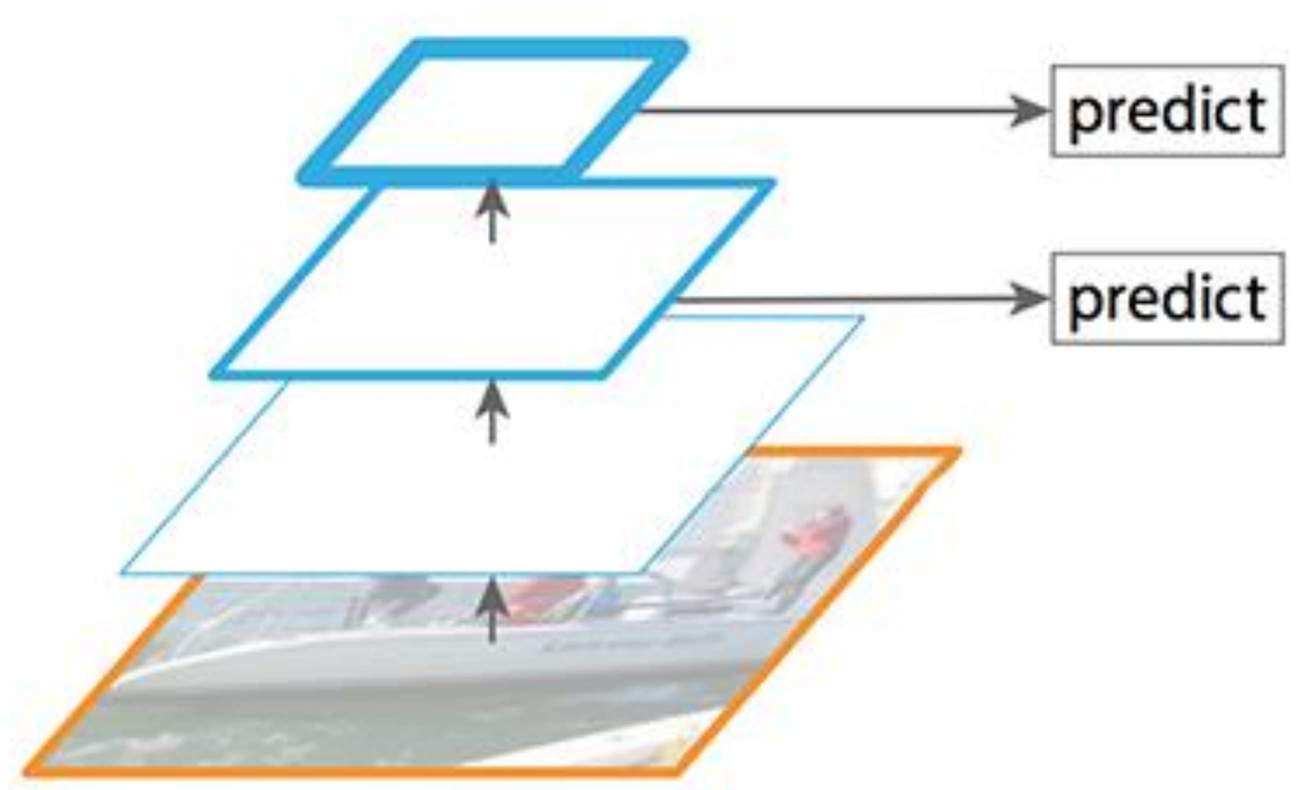


Bottom-up and Top-down

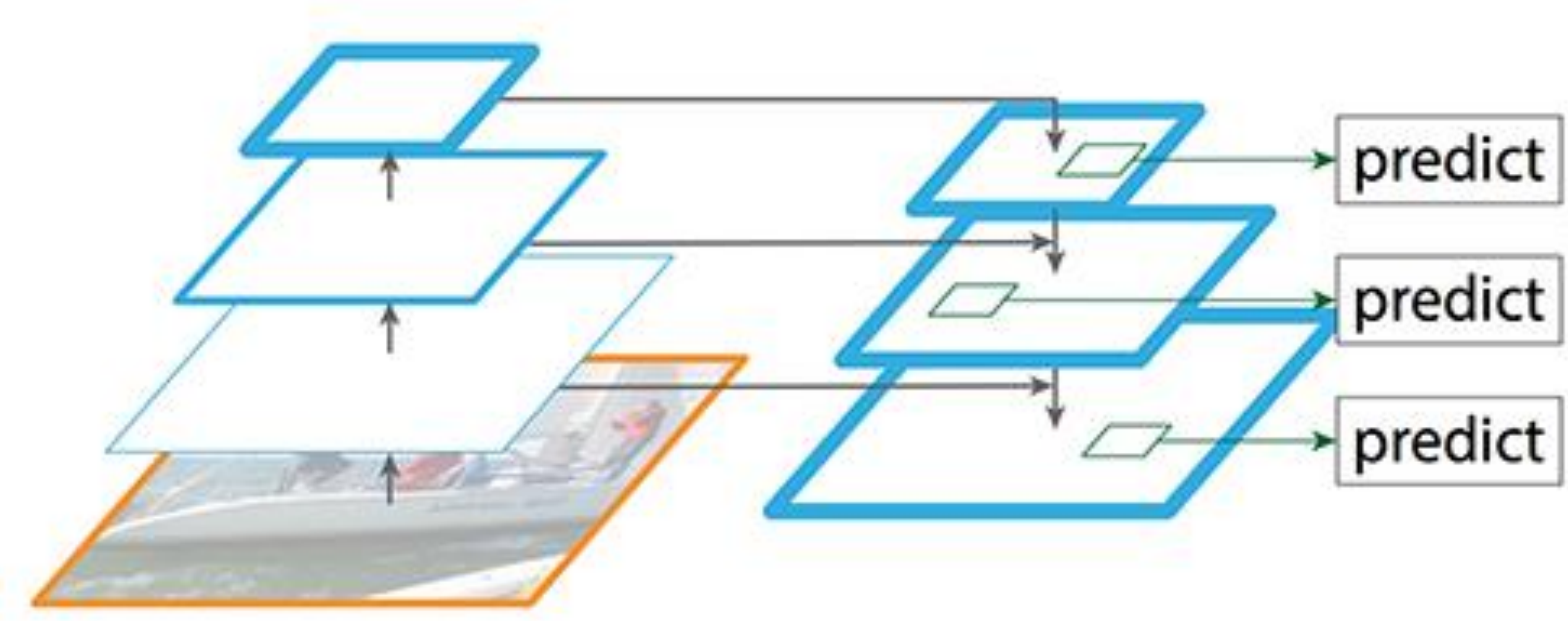


SSD (Bottom-Up)

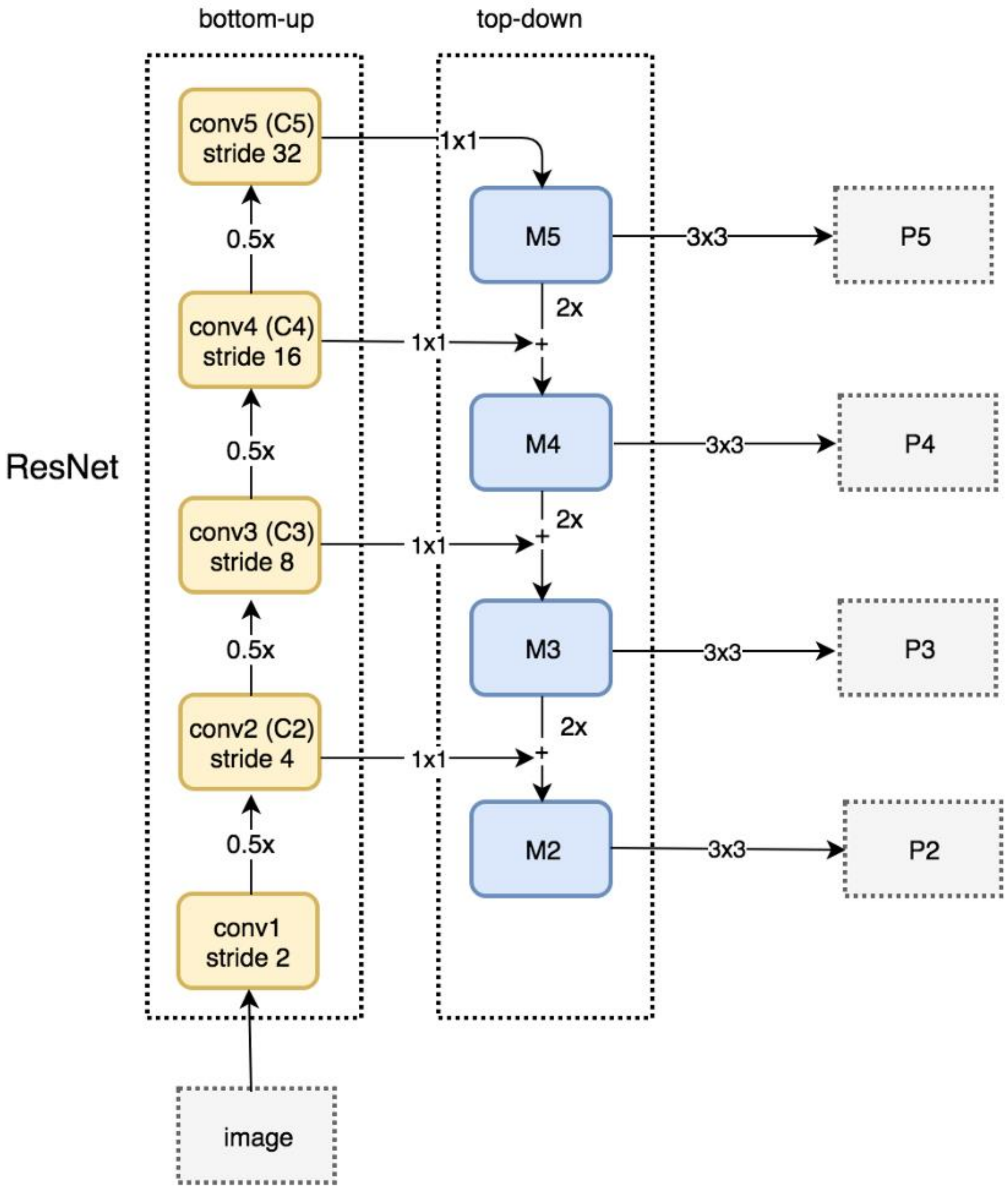
- Using only upper layers as feature maps

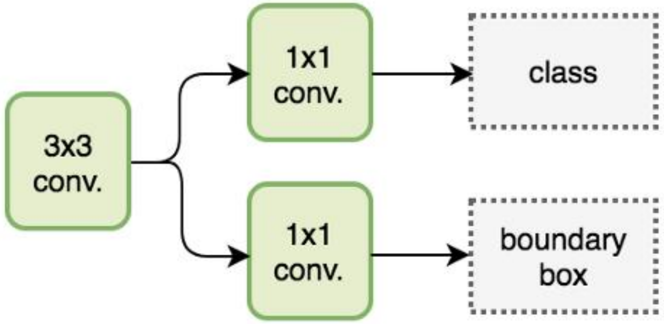


FPN (Top-Down)



FPN Architecture





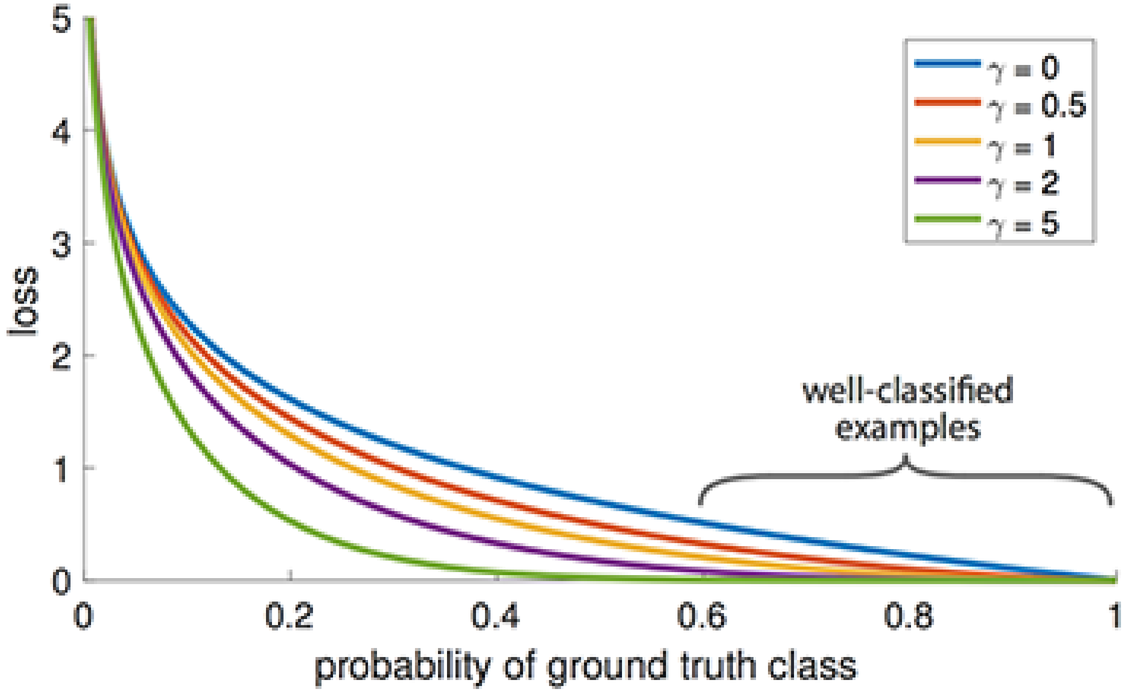
predictor head

Focal Loss

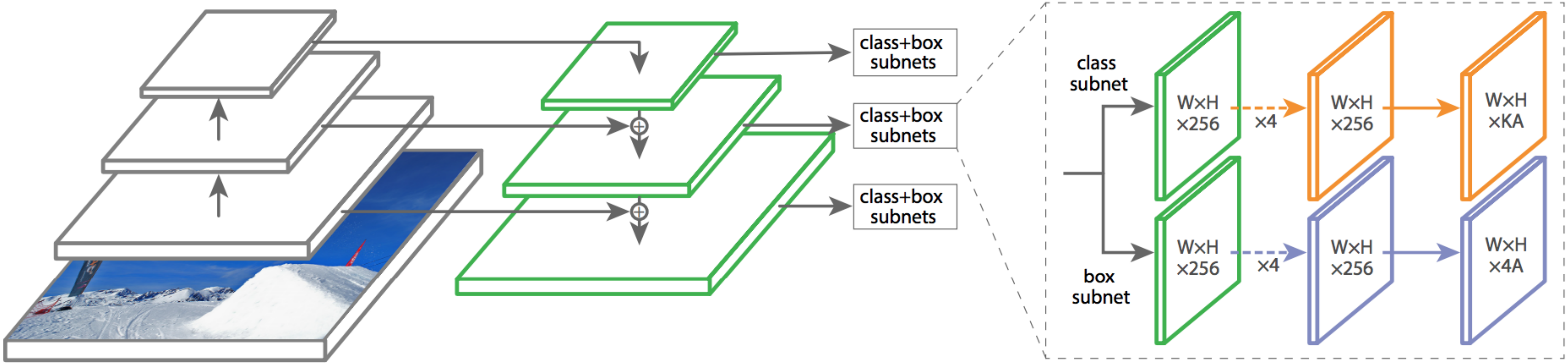
- Solve class imbalance problem by reducing loss for well-trained class

$$CE(p_t) = -\log(p_t)$$
$$FL(p_t) = -(1 - p_t)^{\gamma} \log(p_t)$$

p_t is the predicted class probability for ground truth.
$\gamma > 0$



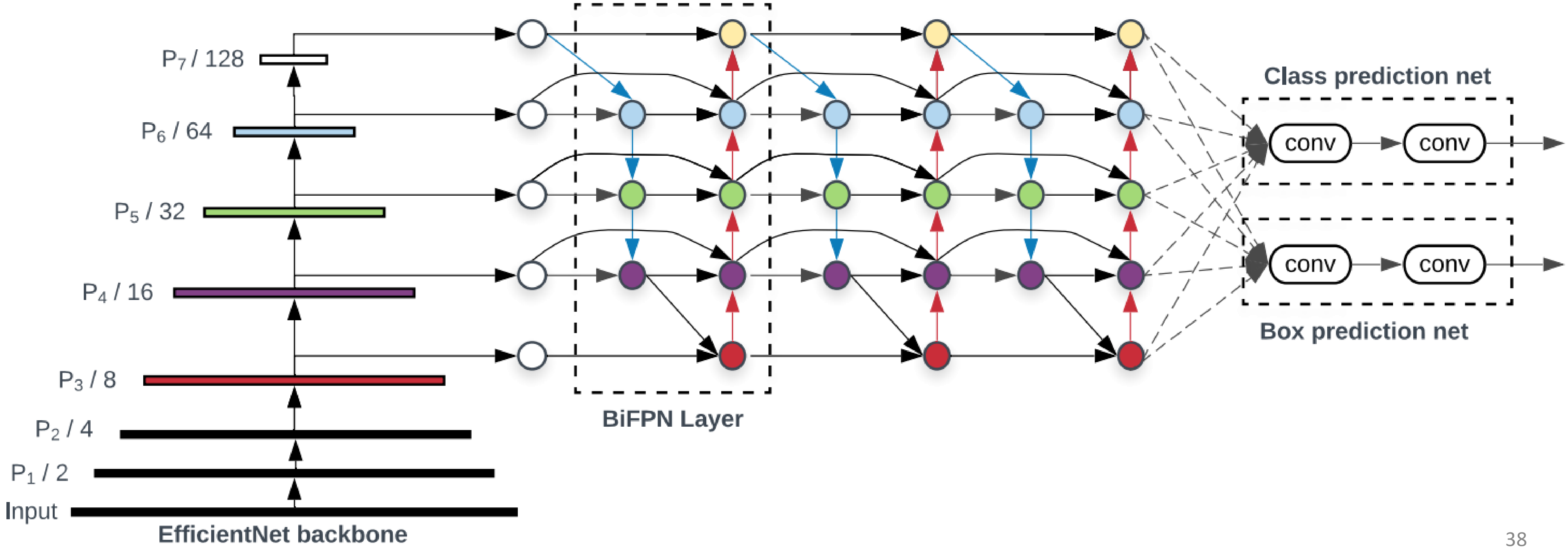
RetinaNet



(a) ResNet (b) feature pyramid net (c) class subnet (top) (d) box subnet (bottom)

EfficientDet

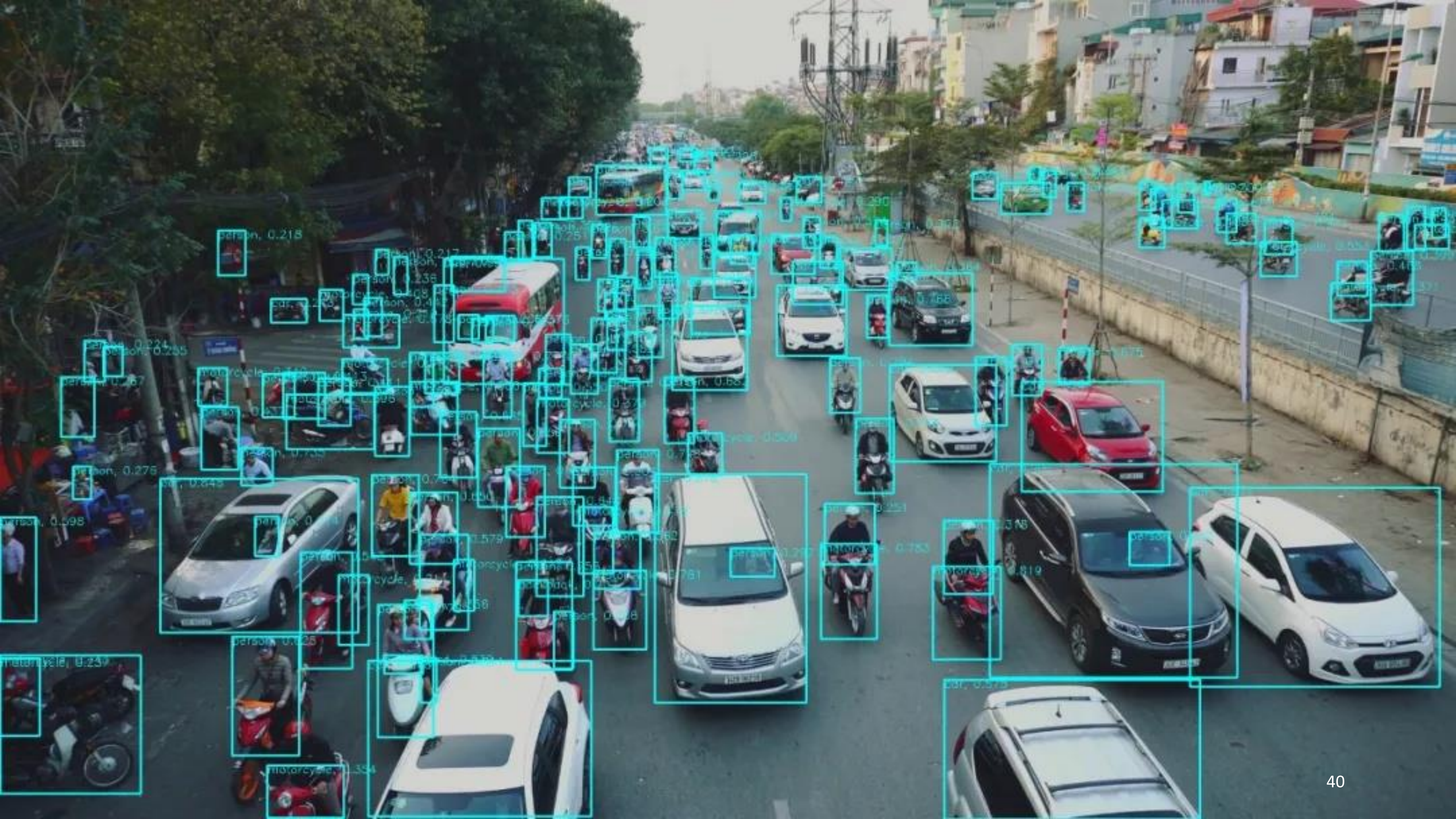
- Based on EfficientNet
  - Mingxing Tan Ruoming Pang Quoc V. Le, “EfficientDet: Scalable and Efficient Object Detection”, Google Research, Brain Team



PyTorch Version of EfficientDet

- 25.86x faster that original TensorFlow version!
- github.com/zylo117

coefficient	Time	FPS	Ratio
Official D0 (tf postprocess)	0.713s	1.40	1X
Official D0 (numpy postprocess)	0.477s	2.09	1.49X
Yet-Another-EfficientDet-D0	***0.028s***	***36.20***	***25.86X***



Segmentation

https://www.analyticsvidhya.com/blog/2019/07/computer-vision-implementing-mask-r-cnn-image-segmentation/

Running Mask R-CNN

https://github.com/matterport/Mask_RCNN.git

Install Prerequisites

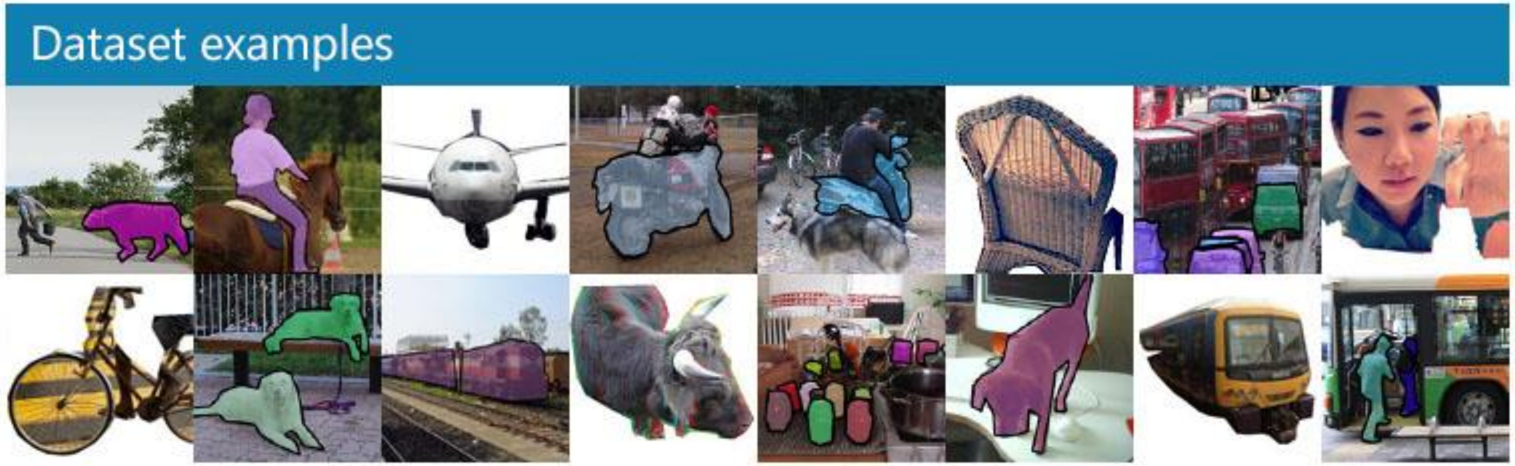
*Create a virtual environment with TensorFlow=1.3 and Keras=2.1

1. git clone https://github.com/matterport/Mask_RCNN.git
2. pip3 install -r requirements.txt
3. python3 setup.py install

Download Pre-trained Weights (MS COCO)

- https://github.com/matterport/Mask_RCNN/releases/download/v2.0/mask_rcnn_coco.h5

COCO
Common Objects in Context



Training Custom Object Detector on Colab

- https://medium.com/analytics-vidhya/custom-object-detection-with-tensorflow-using-google-colab-7cbc484f83d7

Reference

- https://pjreddie.com/
- https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e
- https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/
- https://heartbeat.fritz.ai/gentle-guide-on-how-yolo-object-localization-works-with-keras-part-2-65fe59ac12d
- https://towardsdatascience.com/retinanet-how-focal-loss-fixes-single-shot-detection-cb320e3bb0de
- https://medium.com/@jonathan_hui/what-do-we-learn-from-single-shot-object-detectors-ssd-yolo-fpn-focal-loss-3888677c5f4d