COMP SCI 1400
AI Technologies —Object Detection

Dr. Kamal Mammadov

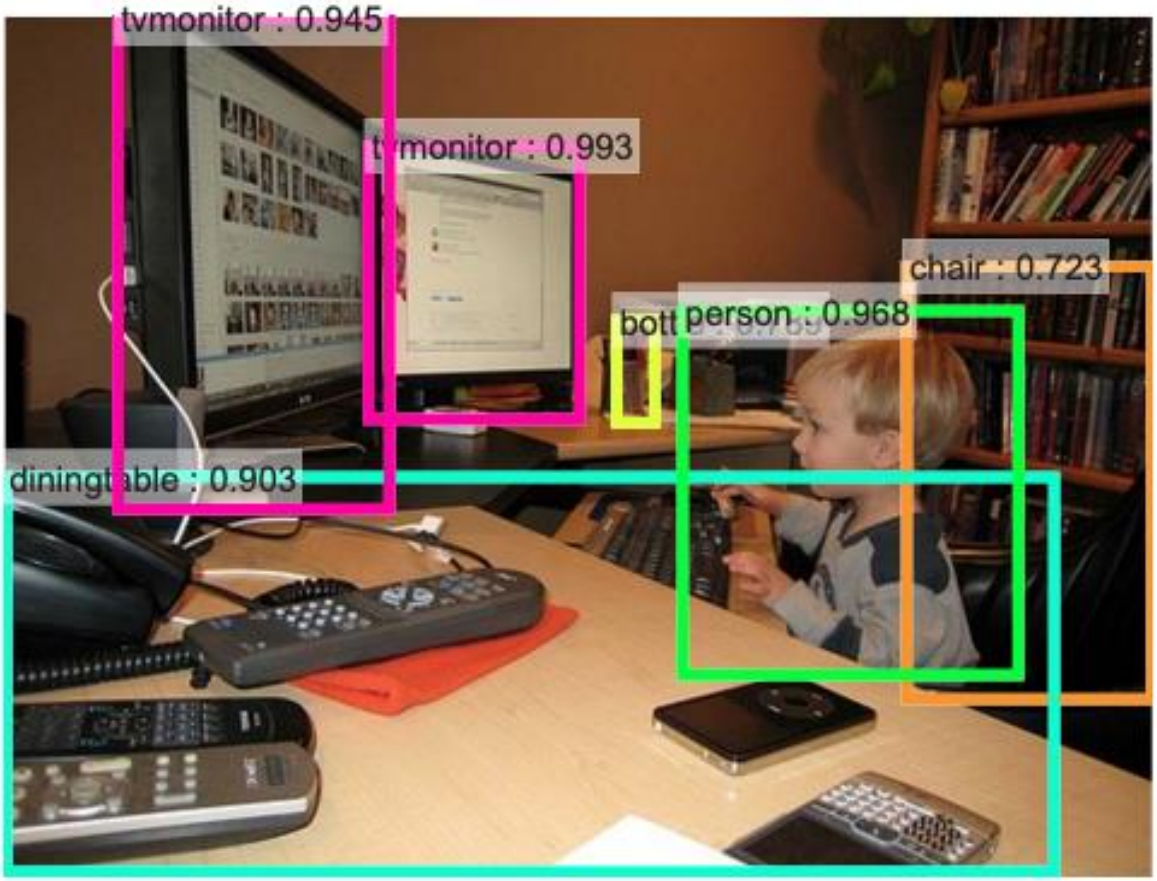
Outline

- What is Object Detection
- Why important
- Challenges
- Brief technique history
- Representative techniques

Object Dectection

- Definition

What objects are where



Roles of Object Detection

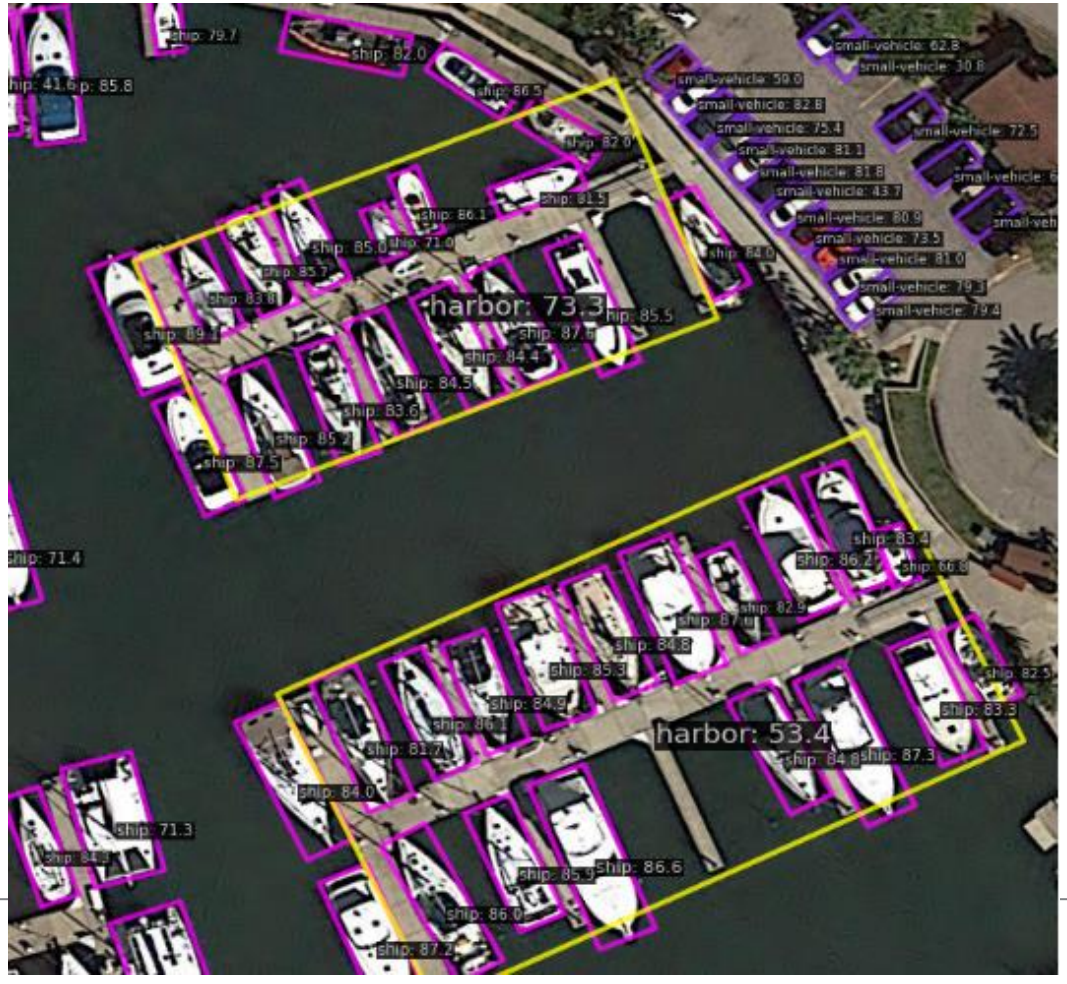
- What can we do with object detection?

Autonomous Driving

Roles of Object Detection

- What can we do with object detection?

Counting



Roles of Object Detection

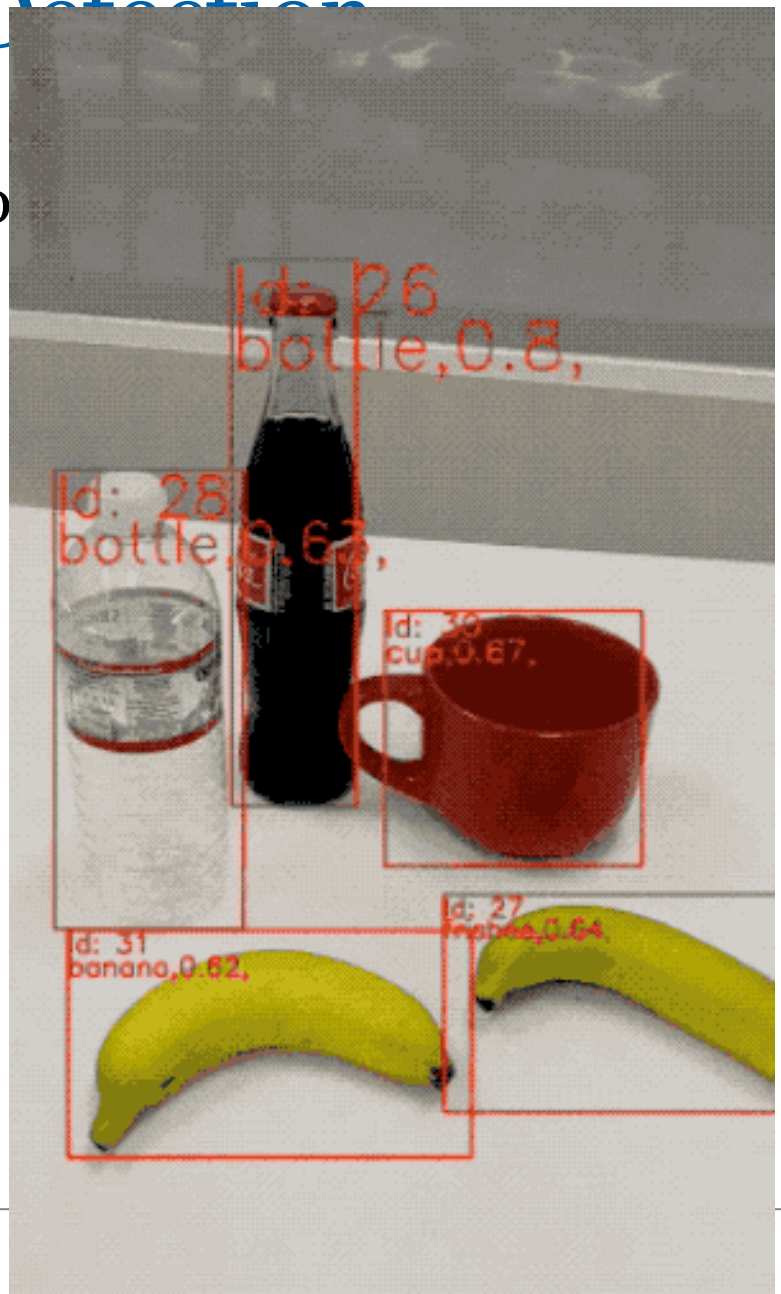
- What can we do with object detection?

Instance segmentation (zoom background setting)

Roles of Object Detection

- What can we do with o

Object tracking



Roles of Object Detection

- What can we do with object detection?

Image captioning

"two young girls are playing with lego toy."

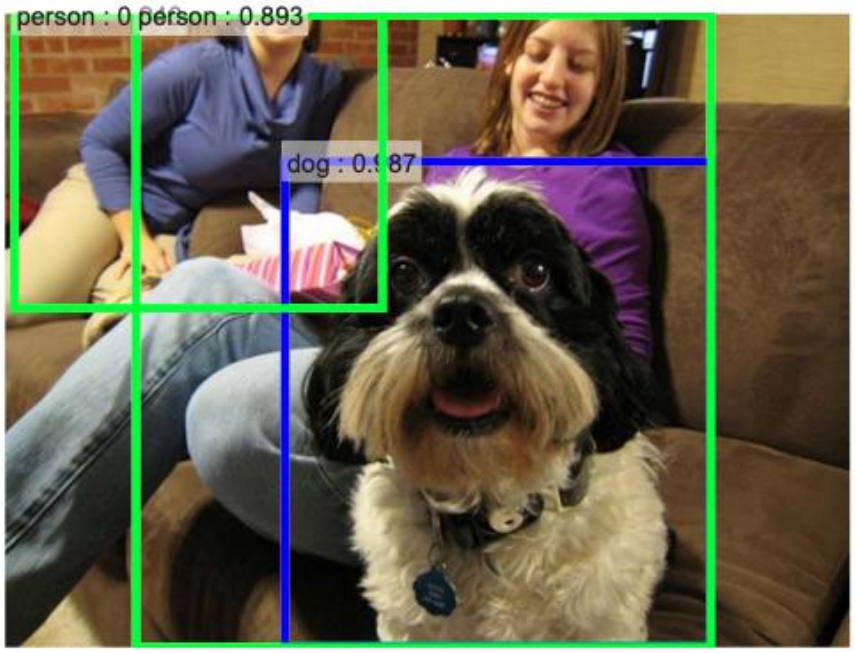
Challenges in Object Detection

- Do we know how many objects are in an image beforehand?

  No

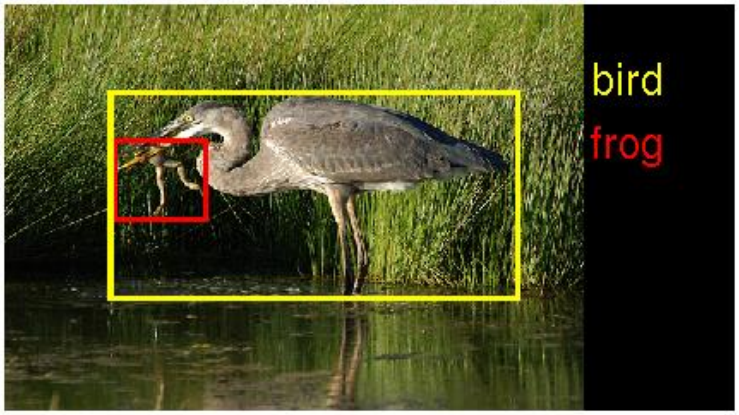
- Do we know the scale of objects?

  No



PASCAL VOC and ILSVRC

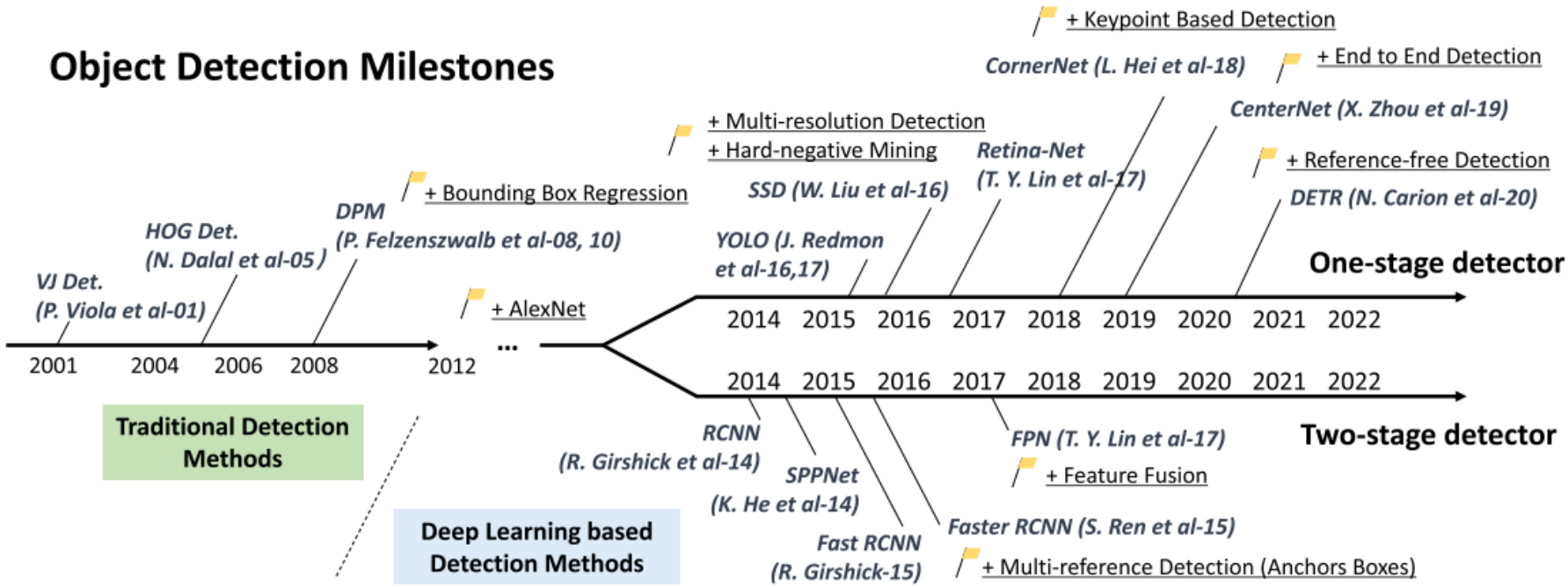
- PASCAL Visual Object Classes 2007-2012
  - 20 classes: person, animal, vehicle, indoor objects
  - VOC 2007: 9,963 images, 24,640 objects
  - VOC 2012: 11,530 images, 27450 objects
- ILSVRC 2012-2017
  - 200 classes
  - Training: 456,567 images, 478,807 objects
  - Validation: 20,121 images, 55,502 objects



COCO Detection

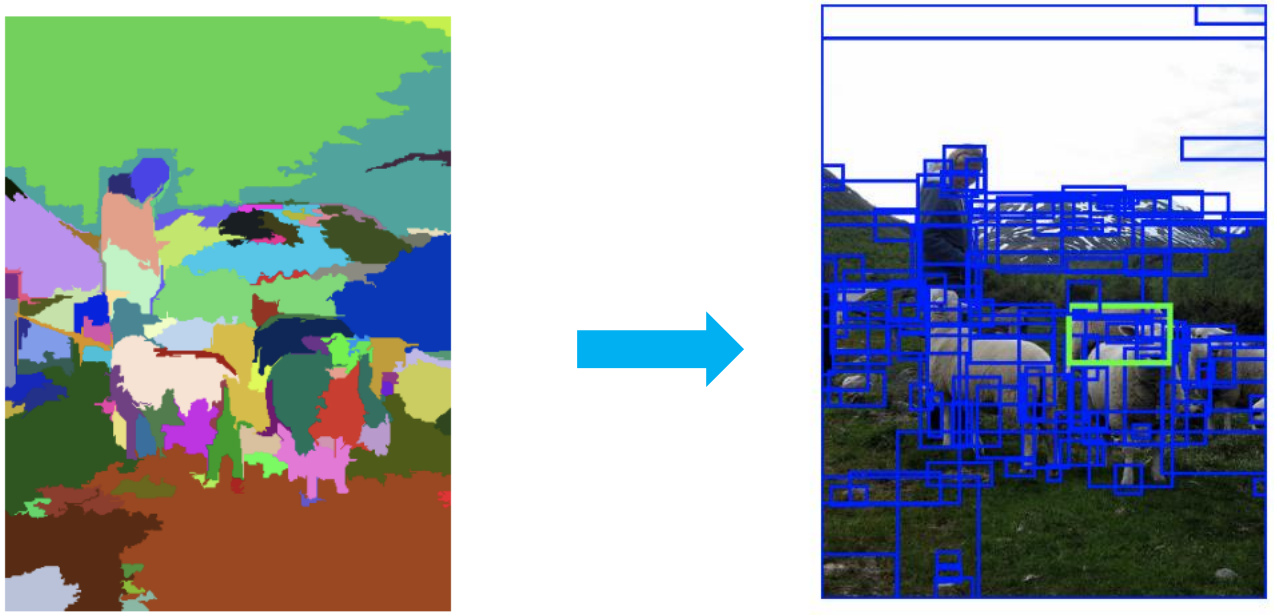
- COCO 2014-2017 dataset
  - 80 classes
  - 118k training images
  - 5k validation images
  - 41k test images

Milestones


Object Detection Milestones
+ Keypoint Based Detection
CornerNet (L. Hei et al-18)
+ End to End Detection
CenterNet (X. Zhou et al-19)
+ Multi-resolution Detection
+ Hard-negative Mining
Retina-Net
(T. Y. Lin et al-17)
+ Reference-free Detection
DETR (N. Carion et al-20)
+ Bounding Box Regression
SSD (W. Liu et al-16)
DPM
(P. Felzenszwalb et al-08, 10)
HOG Det.
(N. Dalal et al-05)
YOLO (J. Redmon
et al-16,17)
One-stage detector
VJ Det.
(P. Viola et al-01)
+ AlexNet
2001
2004
2006
2008
2012
...
2014
2015
2016
2017
2018
2019
2020
2021
2022
Traditional Detection
Methods
RCNN
(R. Girshick et al-14)
FPN (T. Y. Lin et al-17)
Two-stage detector
SPPNet
(K. He et al-14)
+ Feature Fusion
Deep Learning based
Detection Methods
Faster RCNN (S. Ren et al-15)
Fast RCNN
(R. Girshick-15)
+ Multi-reference Detection (Anchors Boxes)

R-CNN

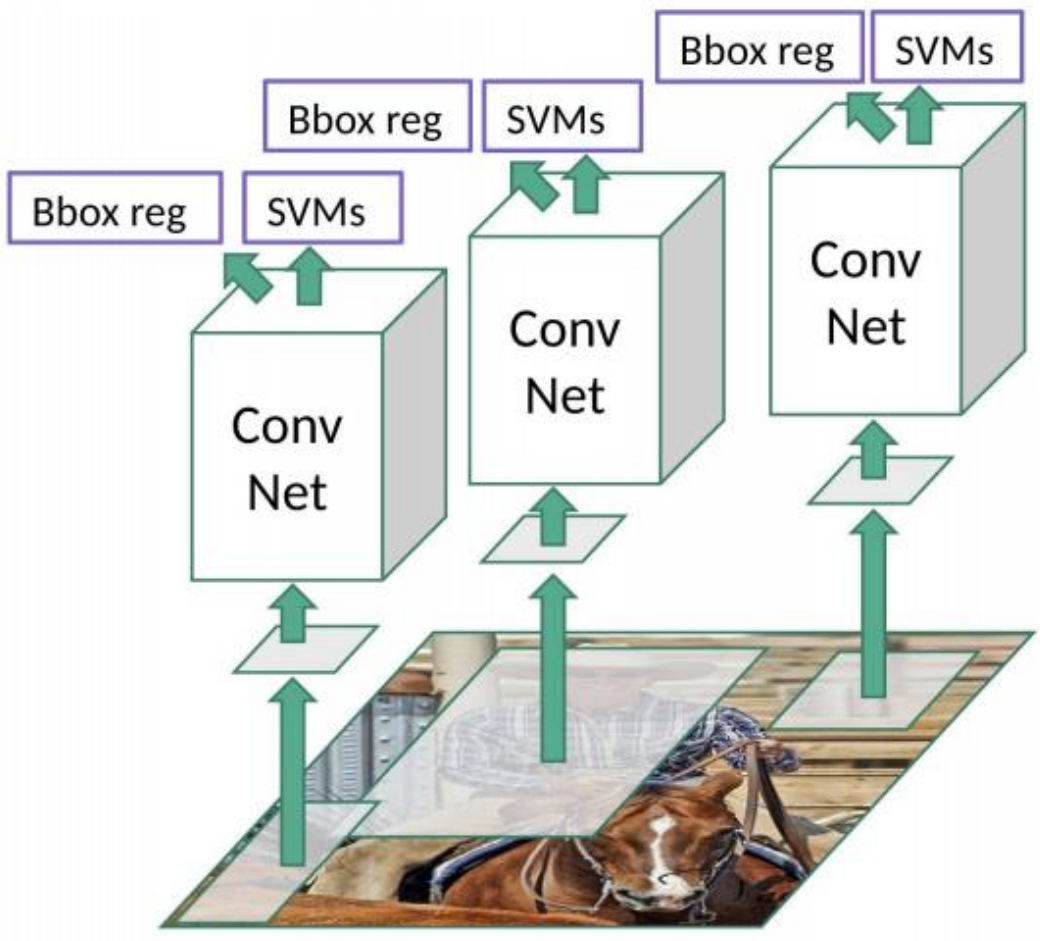
- Selective search
- 2000



object proposals

R-CNN

- Main architecture



R-CNN

- Results
- Metric: mAP

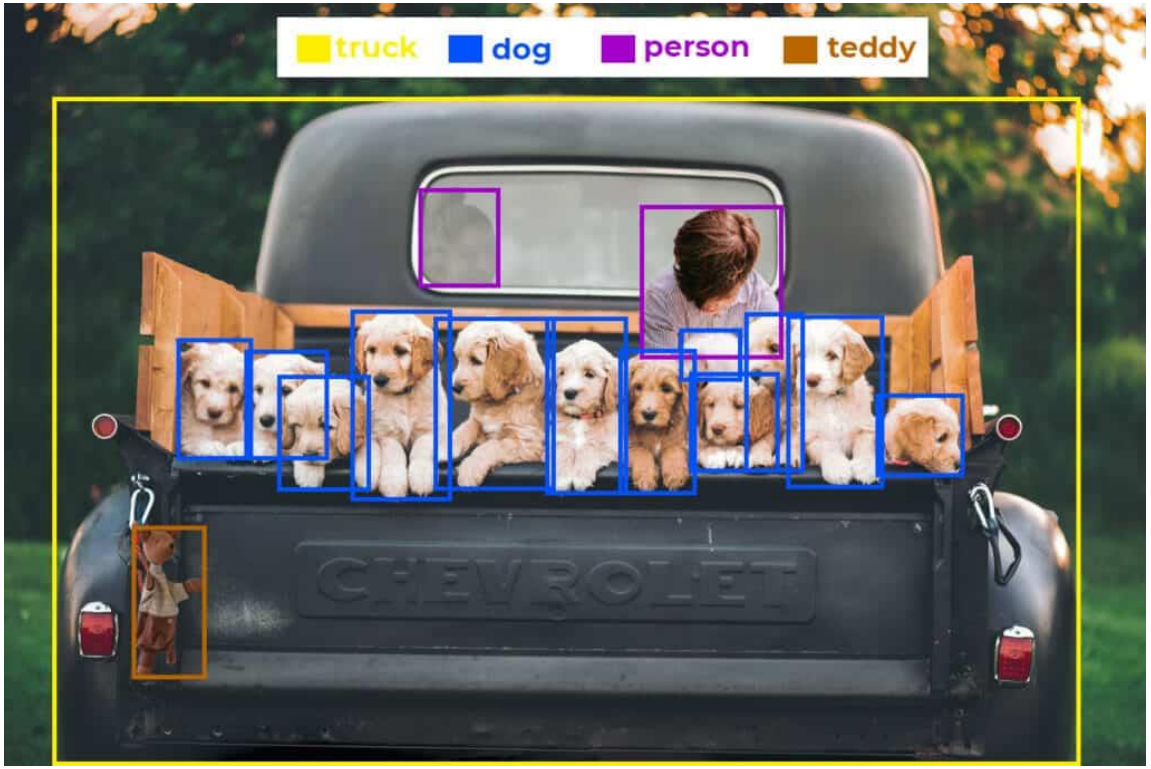
VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN $pool_5$	51.8	60.2	36.4	27.8	23.2	52.8	60.6	49.2	18.3	47.8	44.3	40.8	56.6	58.7	42.4	23.4	46.1	36.7	51.3	55.7	44.2
R-CNN fc_6	59.3	61.8	43.1	34.0	25.1	53.1	60.6	52.8	21.7	47.8	42.7	47.8	52.5	58.5	44.6	25.6	48.3	34.0	53.1	58.0	46.2
R-CNN fc_7	57.6	57.9	38.5	31.8	23.7	51.2	58.9	51.4	20.0	50.5	40.9	46.0	51.6	55.9	43.3	23.3	48.1	35.3	51.0	57.4	44.7
R-CNN FT $pool_5$	58.2	63.3	37.9	27.6	26.1	54.1	66.9	51.4	26.7	55.5	43.4	43.1	57.7	59.0	45.8	28.1	50.8	40.6	53.1	56.4	47.3
R-CNN FT fc_6	63.5	66.0	47.9	37.7	29.9	62.5	70.2	60.2	32.0	57.9	47.0	53.5	60.1	64.2	52.2	31.3	55.0	50.0	57.7	63.0	53.1
R-CNN FT fc_7	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2
R-CNN FT fc_7 BB	**68.1**	**72.8**	**56.8**	**43.0**	**36.8**	**66.3**	**74.2**	**67.6**	**34.4**	**63.5**	**54.5**	**61.2**	**69.1**	**68.6**	**58.7**	**33.4**	**62.9**	**51.1**	**62.5**	**64.8**	**58.5**
DPM v5 [20]	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5	33.7
DPM ST [28]	23.8	58.2	10.5	8.5	27.1	50.4	52.0	7.3	19.2	22.8	18.1	8.0	55.9	44.8	32.4	13.3	15.9	22.8	46.2	44.9	29.1
DPM HSC [31]	32.2	58.3	11.5	16.3	30.6	49.9	54.8	23.5	21.5	27.7	34.0	13.7	58.1	51.6	39.9	12.4	23.5	34.4	47.4	45.2	34.3

How to calculate mAP?

R-CNN

- Metric: mAP

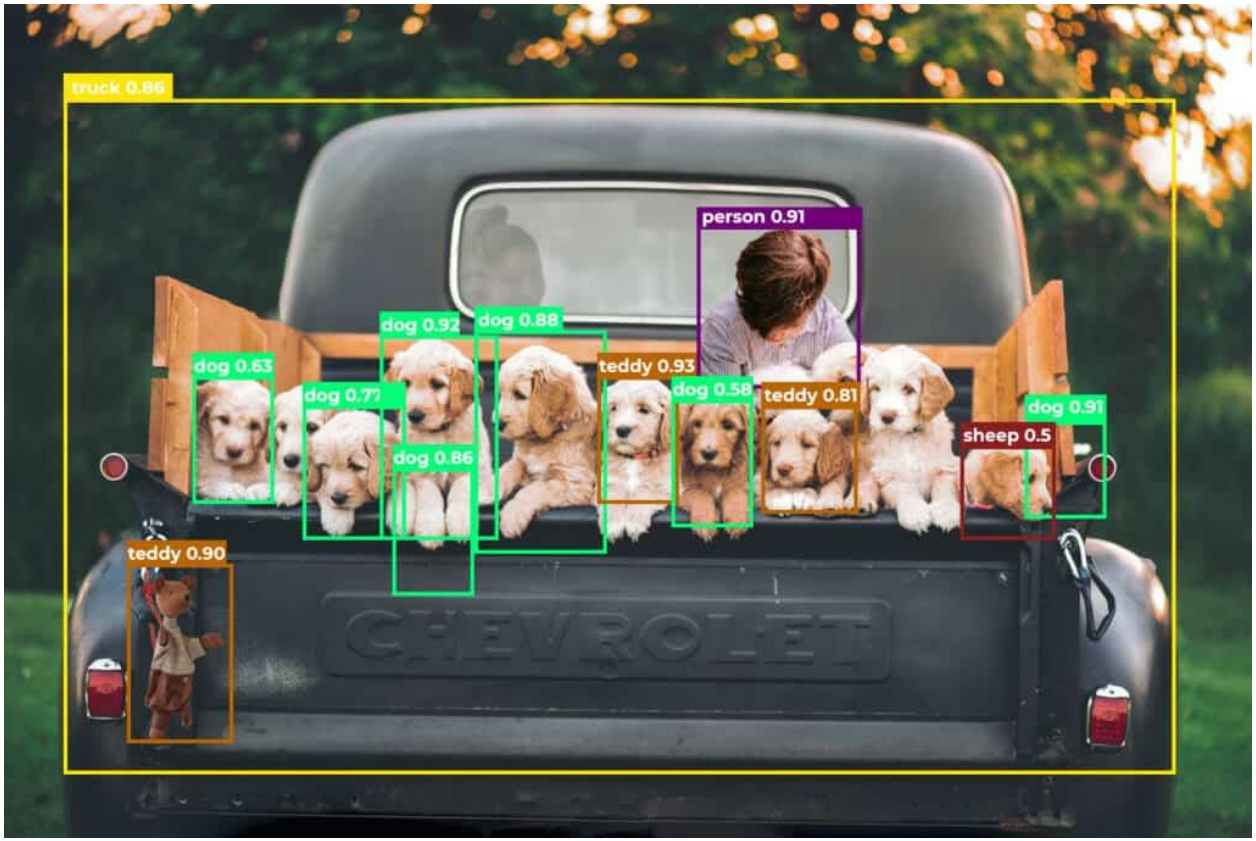
  - 2 person
  - 12 dog
  - 1 teddy
  - 1 truck



Ground truth

R-CNN

- Metric: mAP



Prediction

R-CNN

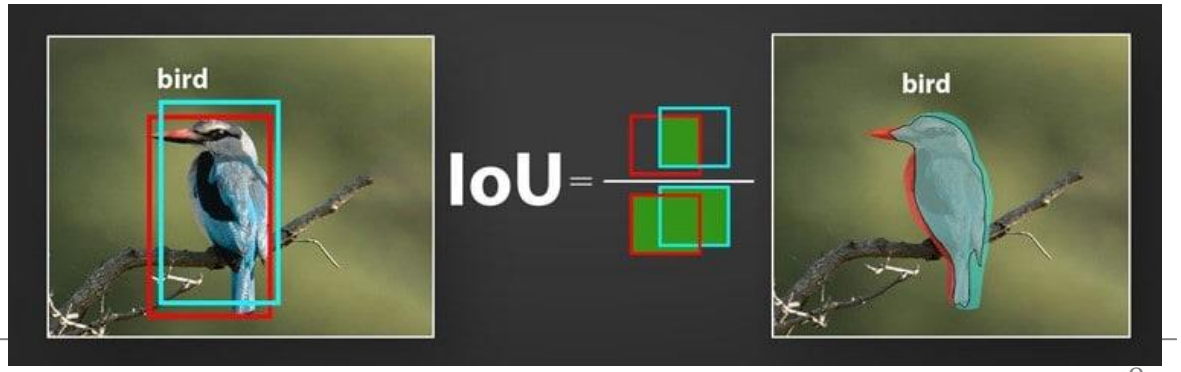
- Metric: mAP
  Average Precision (AP) is calculated class-wise.

Detections							
Conf.	0.63	0.77	0.92	0.86	0.88	0.58	0.91
Matches GT by IoU?	TP	TP	TP	FP	TP	TP	FP

TP: IoU>0.5

Red BB: Ground truth

Cyan BB: Prediction



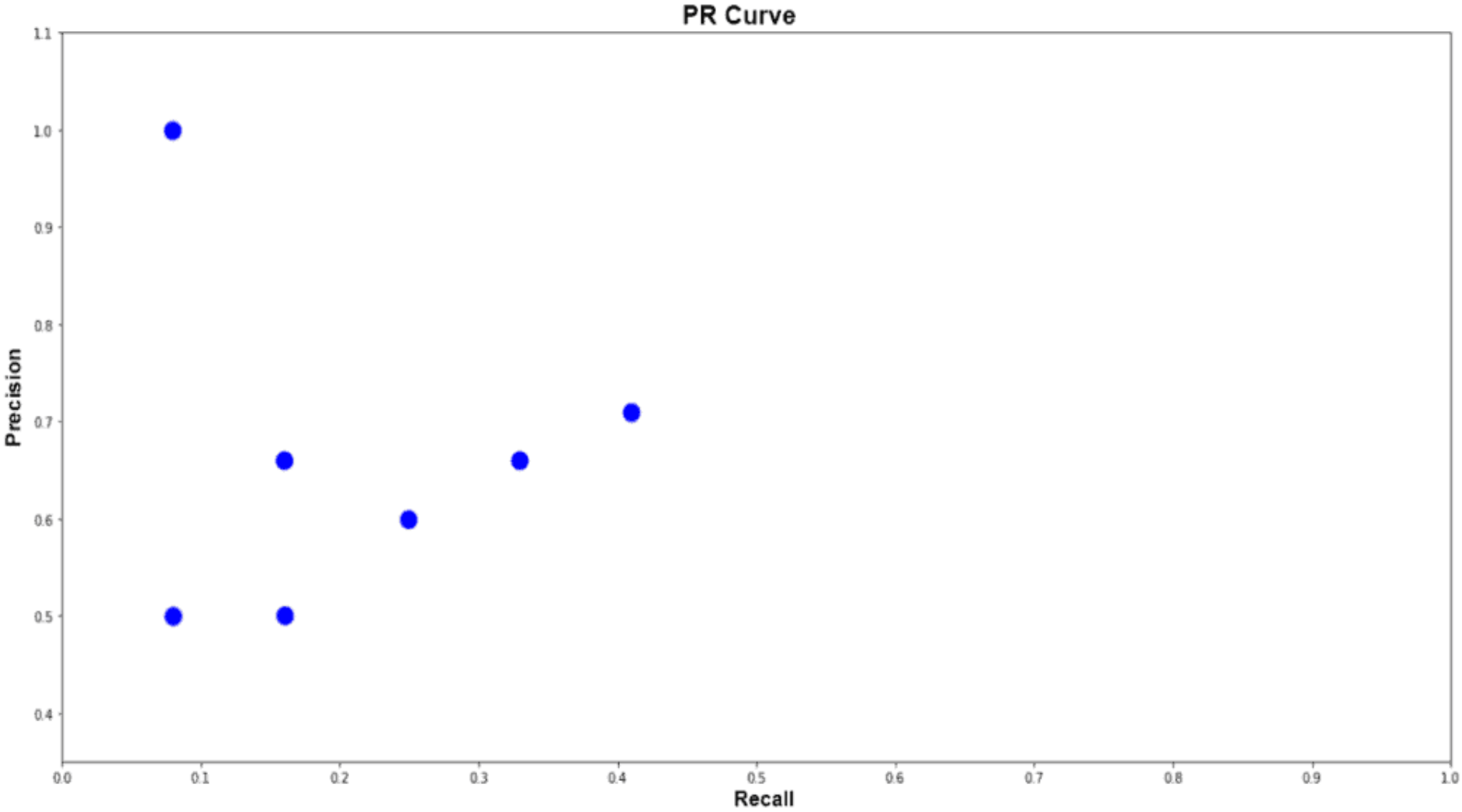
R-CNN

- 
True Positive
Prediction: Hot Dog
Prediction: Hot Dog
False Positive
False Negative
Prediction: Not Hot Dog
Prediction: Not Hot Dog
True Negative

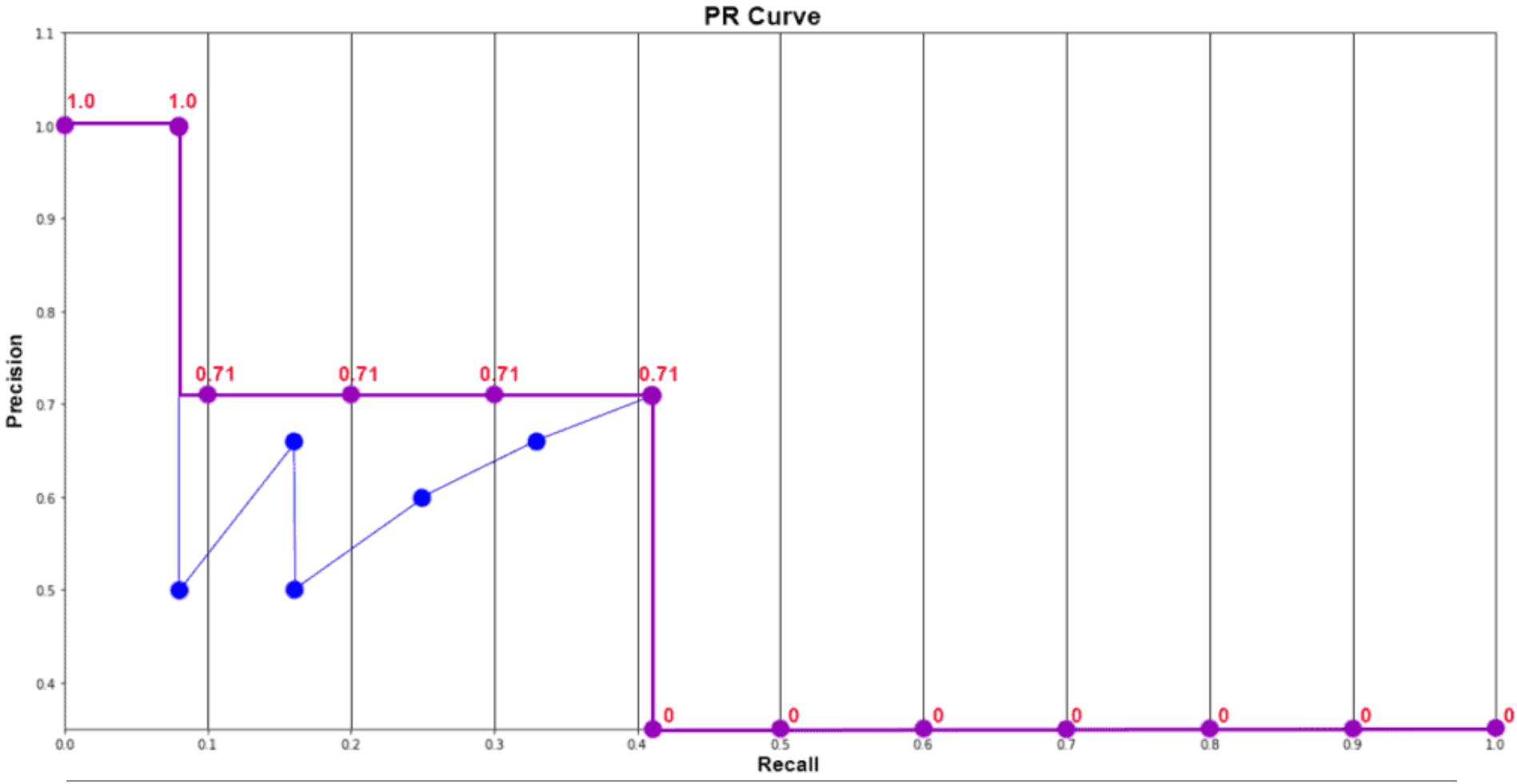
Preds.	Conf.	Matches	Cumulative TP	Cumulative FP	Precision	Recall
	0.92	TP	1	0	1/(1+0) = 1	1/16 = 0.08
	0.91	FP	1	1	1/(1+1) = 0.5	1/16 = 0.08
	0.88	TP	2	1	2/(2+3) = 0.66	2/16 = 0.16
	0.86	FP	2	2	0.5	0.16
	0.77	TP	3	2	0.6	0.25
	0.63	TP	4	2	0.66	0.33
	0.58	TP	5	2	0.71	0.41

R-CNN

- Metric: mAP

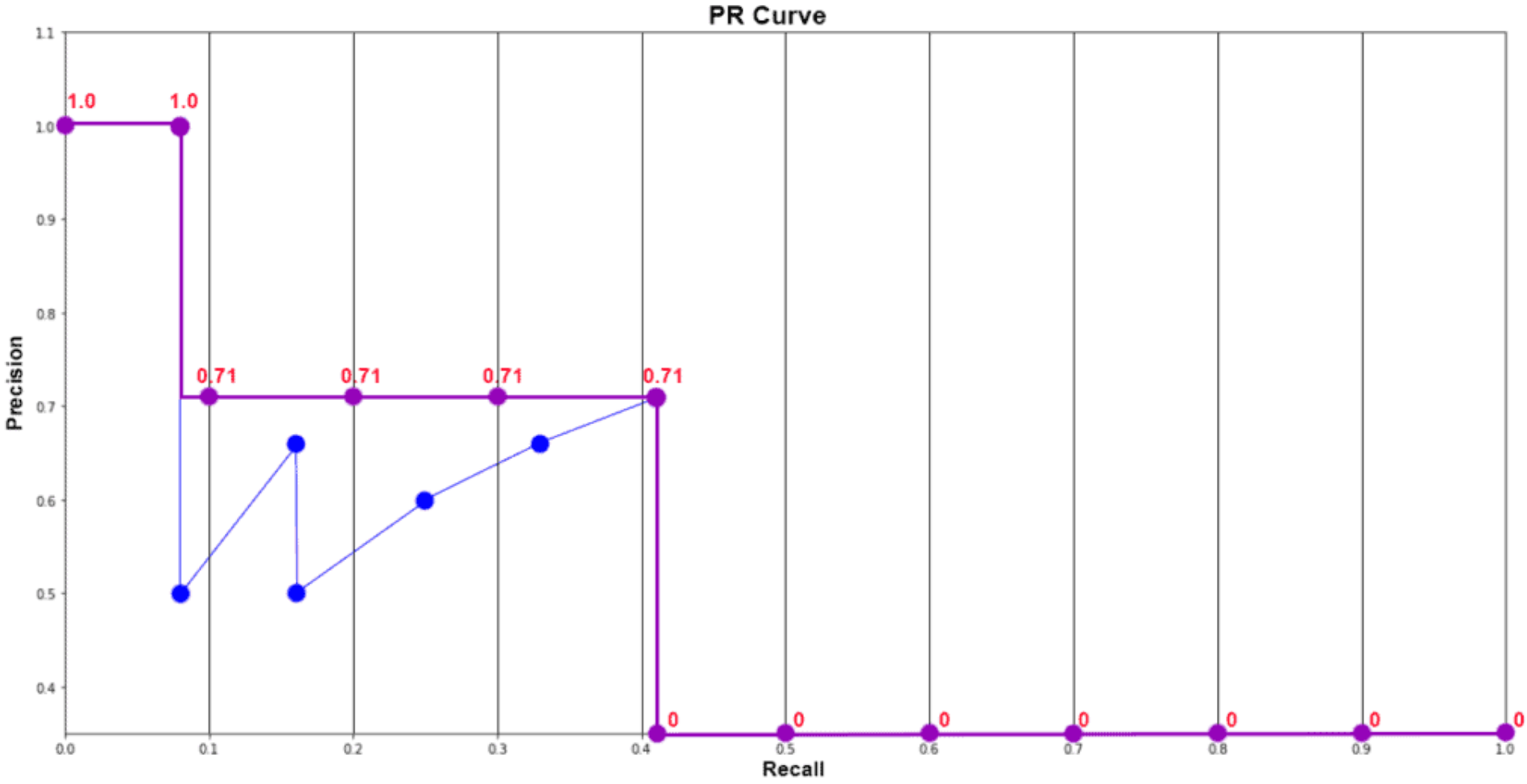


Precision values are interpolated across 11 Recall values, i.e., 0, 0.1, 0.2, 0.3,…,1.0. The interpolated Precision is the **maximum precision value to the right**.



$\mathbf{AP}_{\text{dog}}$ = 1/11 * (Sum of 11 interpolated Precision values)

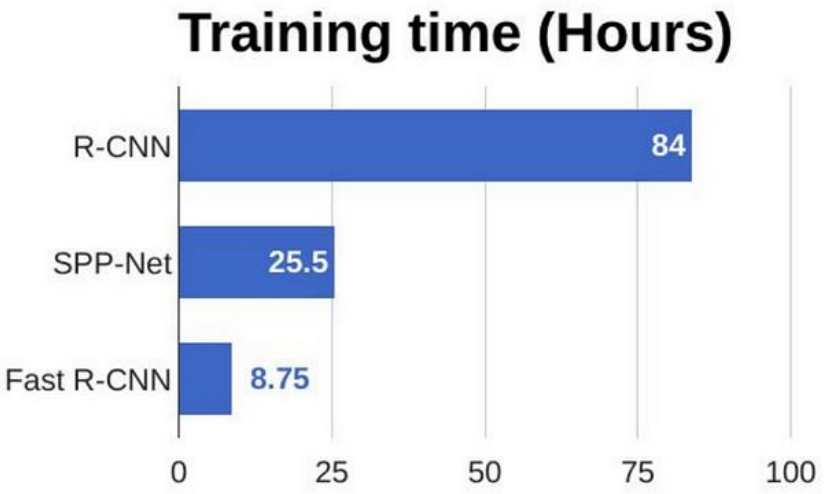
= 1/11 * (1 + 4*0.71 + 6*0)= 0.349= 34.9%

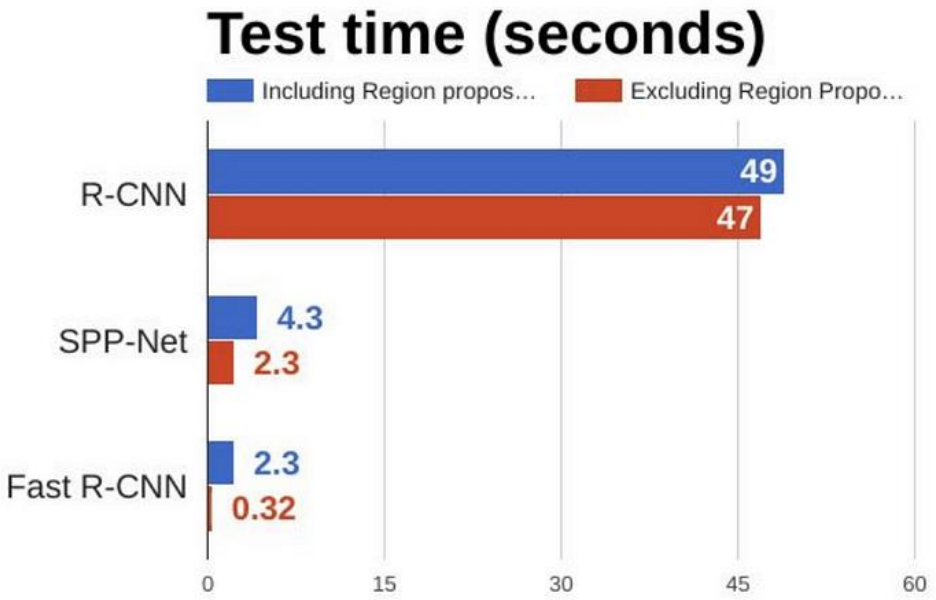


CLASS	dog	person	sheep	truck	teddy
AP	0.349	0.545	0.00	1.00	0.50

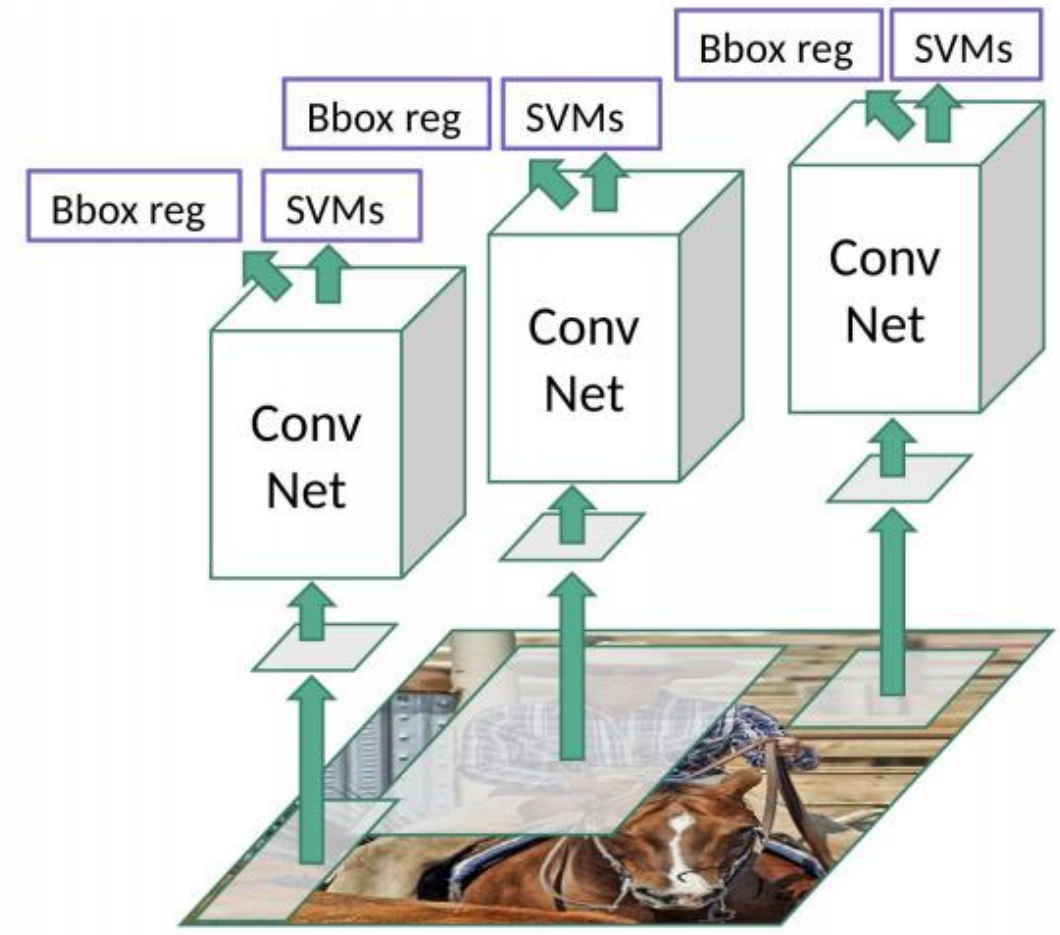
Fast R-CNN

- R-CNN runs slowly. Why?



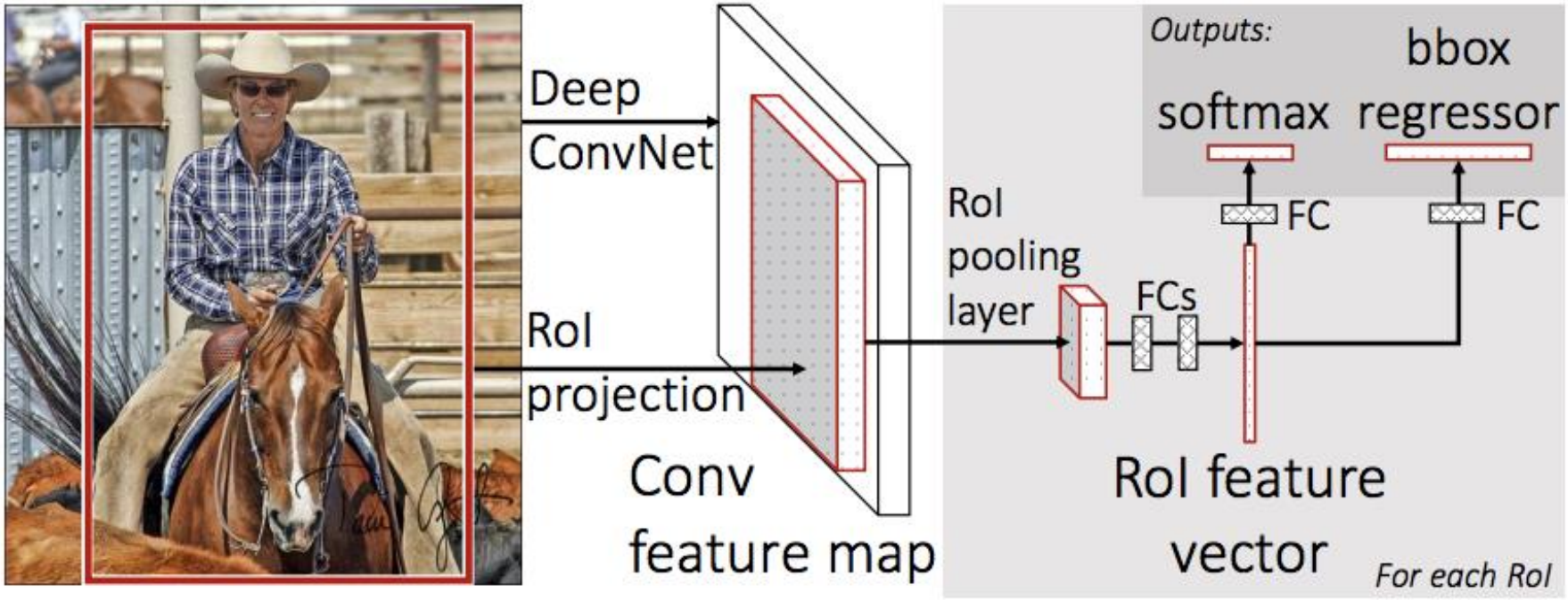


R-CNN -> Fast R-CNN



R-CNN -> Fast R-CNN

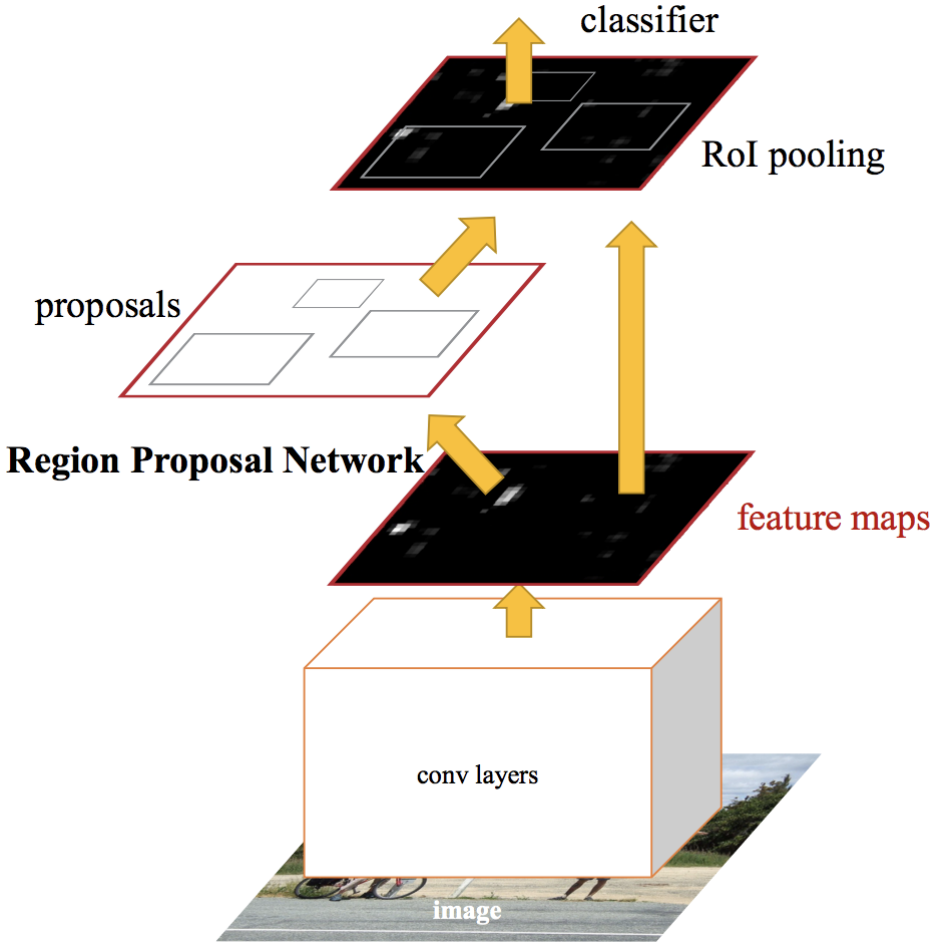
Fast R-CNN



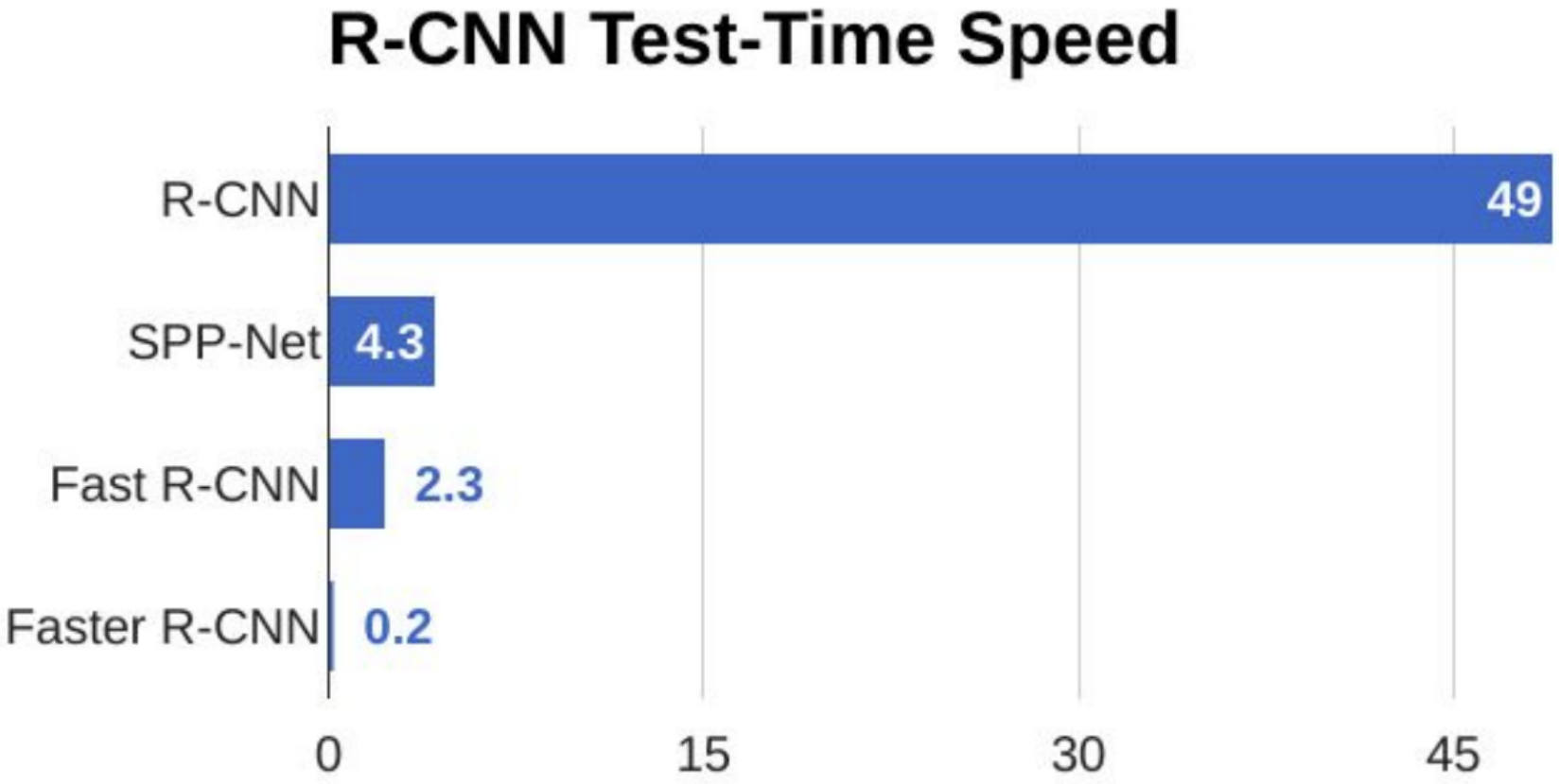
Fast R-CNN -> Faster R-CNN

Fast R-CNN still runs slow due to selective search

Faster R-CNN:
Remove selective search, directly predict proposals from CNN features

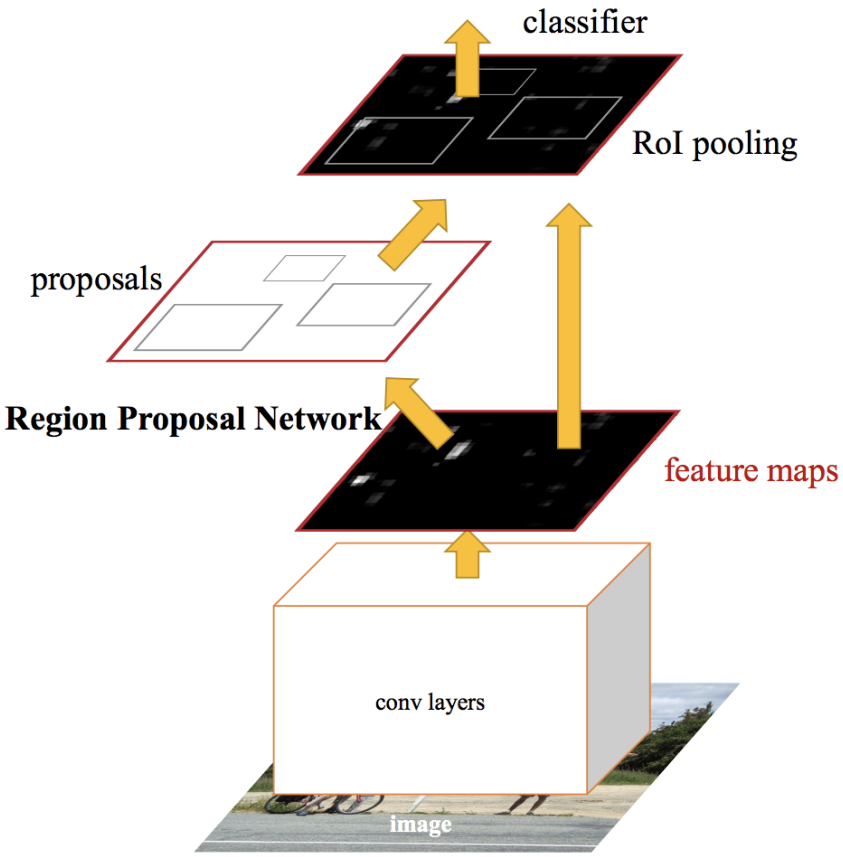


Fast R-CNN -> Faster R-CNN



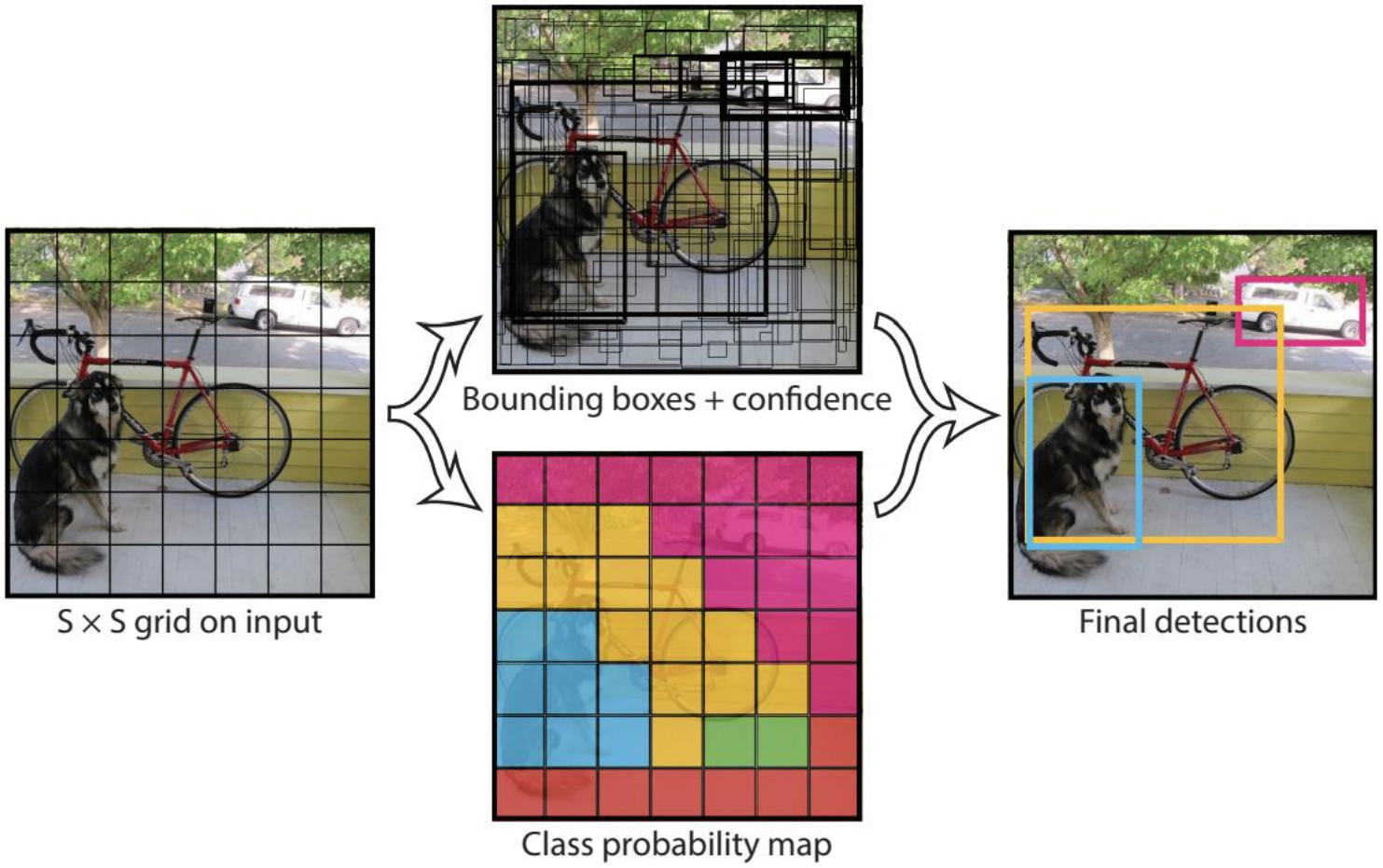
Can be faster?

Can be faster?



Faster RCNN

YOLO: You Only Look Once



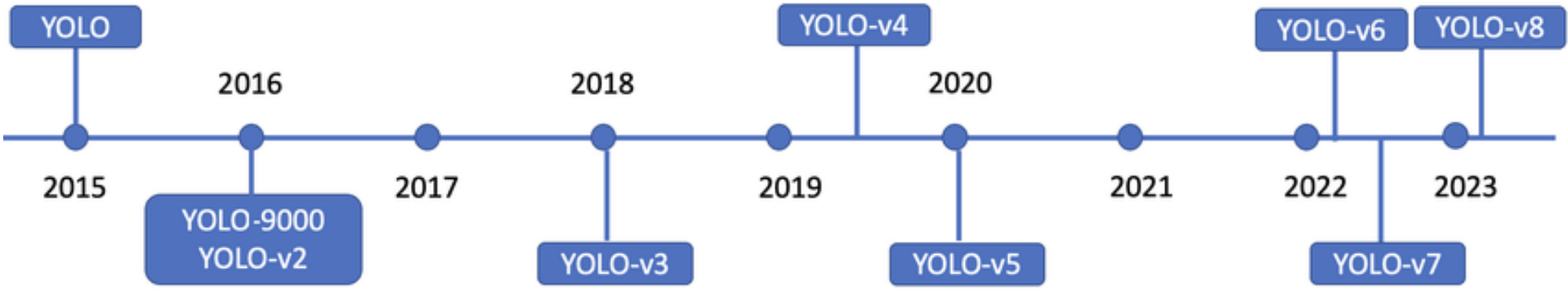
YOLO

YOLO: You Only Look Once

Results

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [31]	2007	16.0	100
30Hz DPM [31]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [38]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[28]	2007+2012	73.2	7
Faster R-CNN ZF [28]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

YOLO: You Only Look Once



- YOLOv2, released in 2016, improved the original model by incorporating batch normalization, anchor boxes, and dimension clusters.
- YOLOv3, launched in 2018, further enhanced the model's performance using a more efficient backbone network, multiple anchors and spatial pyramid pooling.
- YOLOv4 was released in 2020, introducing innovations like Mosaic data augmentation, a new anchor-free detection head, and a new loss function.
- YOLOv5 further improved the model's performance and added new features such as hyperparameter optimization, integrated experiment tracking and automatic export to popular export formats.
- YOLOv6 was open-sourced by Meituan in 2022 and is in use in many of the company's autonomous delivery robots.
- YOLOv7 added additional tasks such as pose estimation on the COCO keypoints dataset.
- YOLOv8 is the latest version of YOLO by Ultralytics. As a cutting-edge, state-of-the-art (SOTA) model, YOLOv8 builds on the success of previous versions, introducing new features and improvements for enhanced performance, flexibility, and efficiency. YOLOv8 supports a full range of vision AI tasks, including detection, segmentation, pose estimation, tracking, and classification. This versatility allows users to leverage YOLOv8's capabilities across diverse applications and domains.

YOLO: You Only Look Once

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

mAP^{val} values are for single-model single-scale on COCO val2017 datase