Deep Learning for Computer Vision

Andrii Liubonko

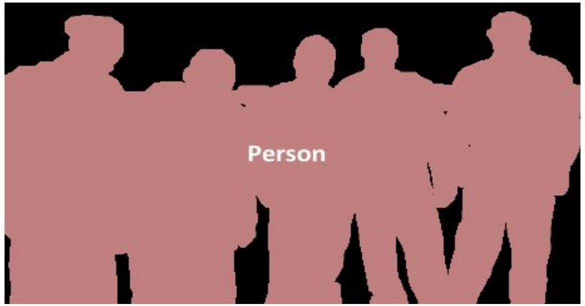
Grammarly

Content

- **Intro to Object Detection**
  - Datasets
  - Metrics
- Two-stage detectors [RCNN Family]
- One-stage detectors
  - YOLO
  - Single Shot Detector [SSD]
- Proposal-free detection
- Summary

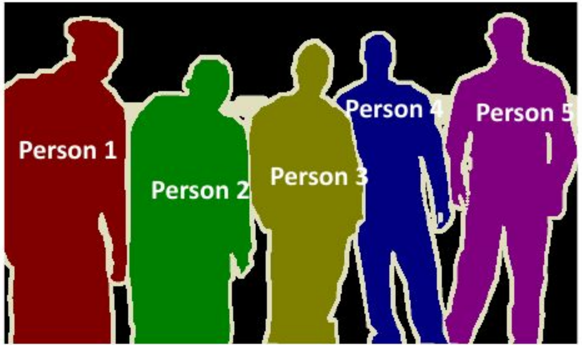
Visual Perception Problems

Classification + Localization



Semantic Segmentation

Object Detection



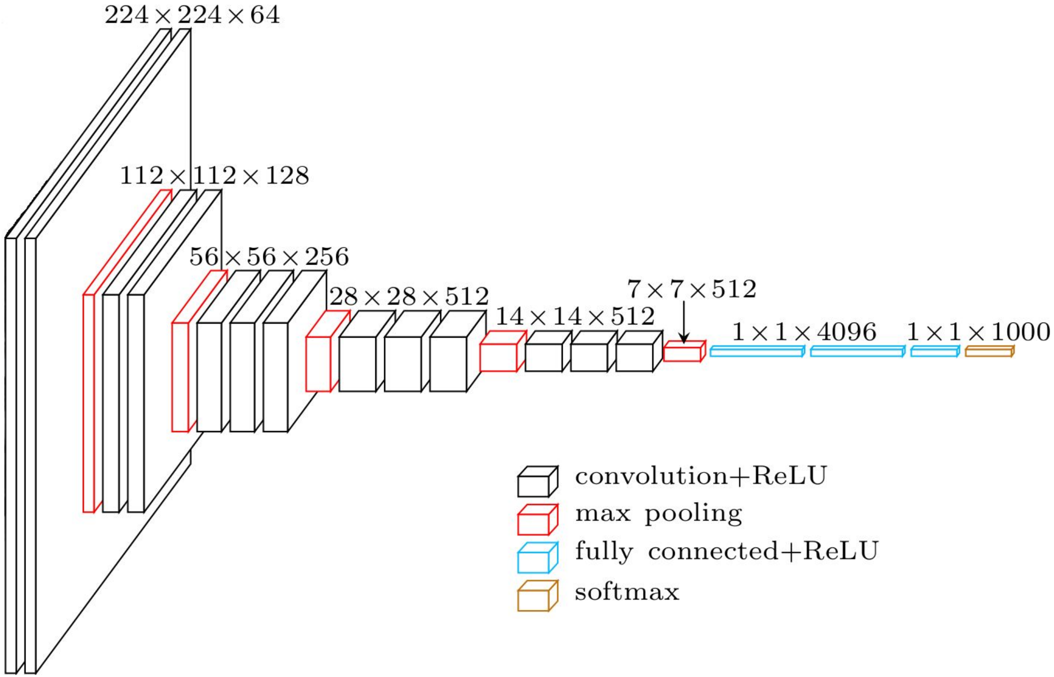
Instance Segmentation

Localization

[Score] dog: 0.98
cow: 0.01
...
[Bounding Box]
(x,y,h,w)

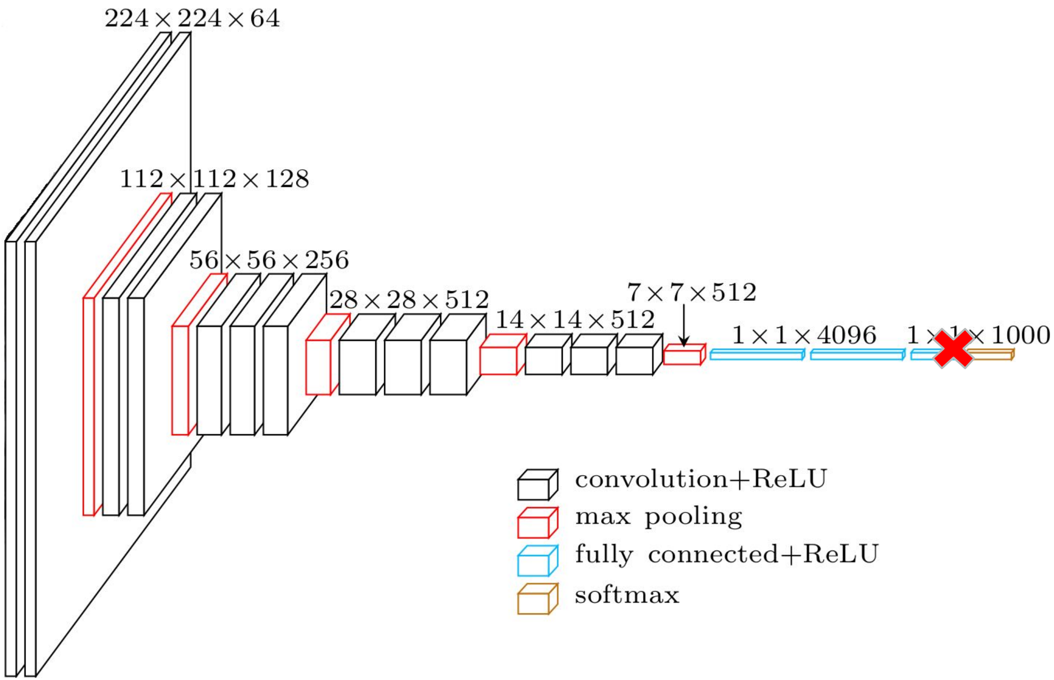
The main assumption is that there is only **one** object in the image

Image Classification



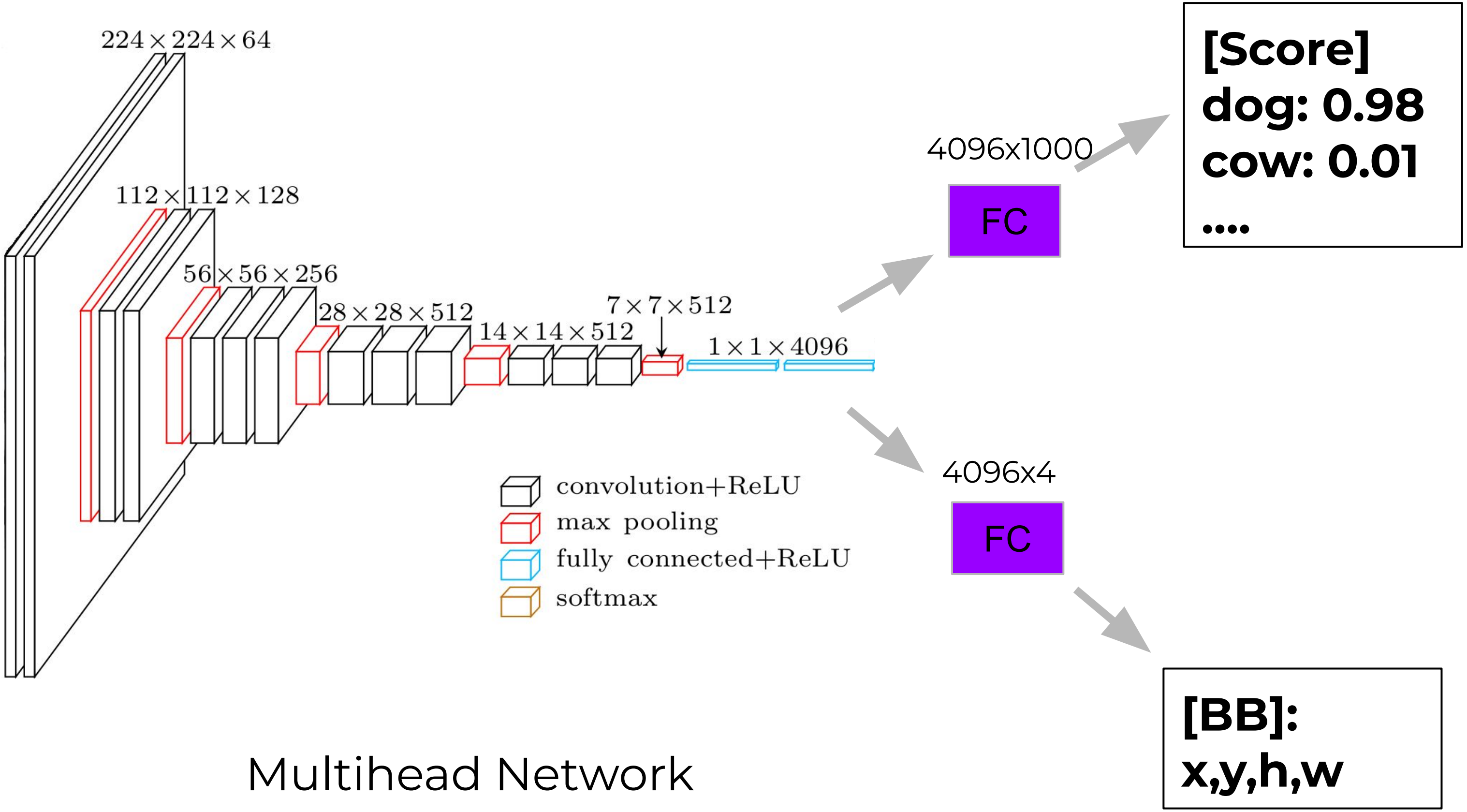
dog: 0.98
cow: 0.01
....

Image Classification -> Classification + Localization



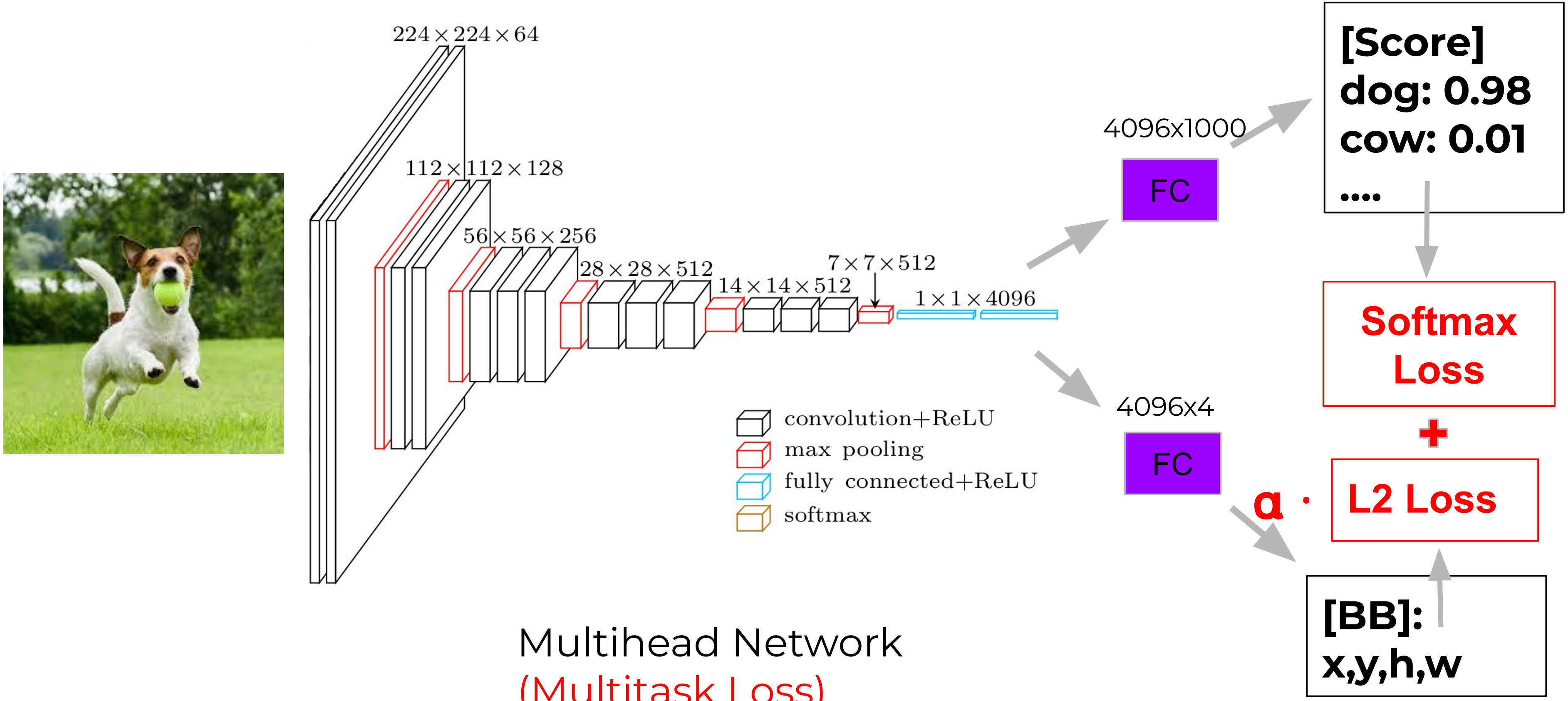
dog: 0.98
cow: 0.01
....

Image Classification -> Classification + Localization



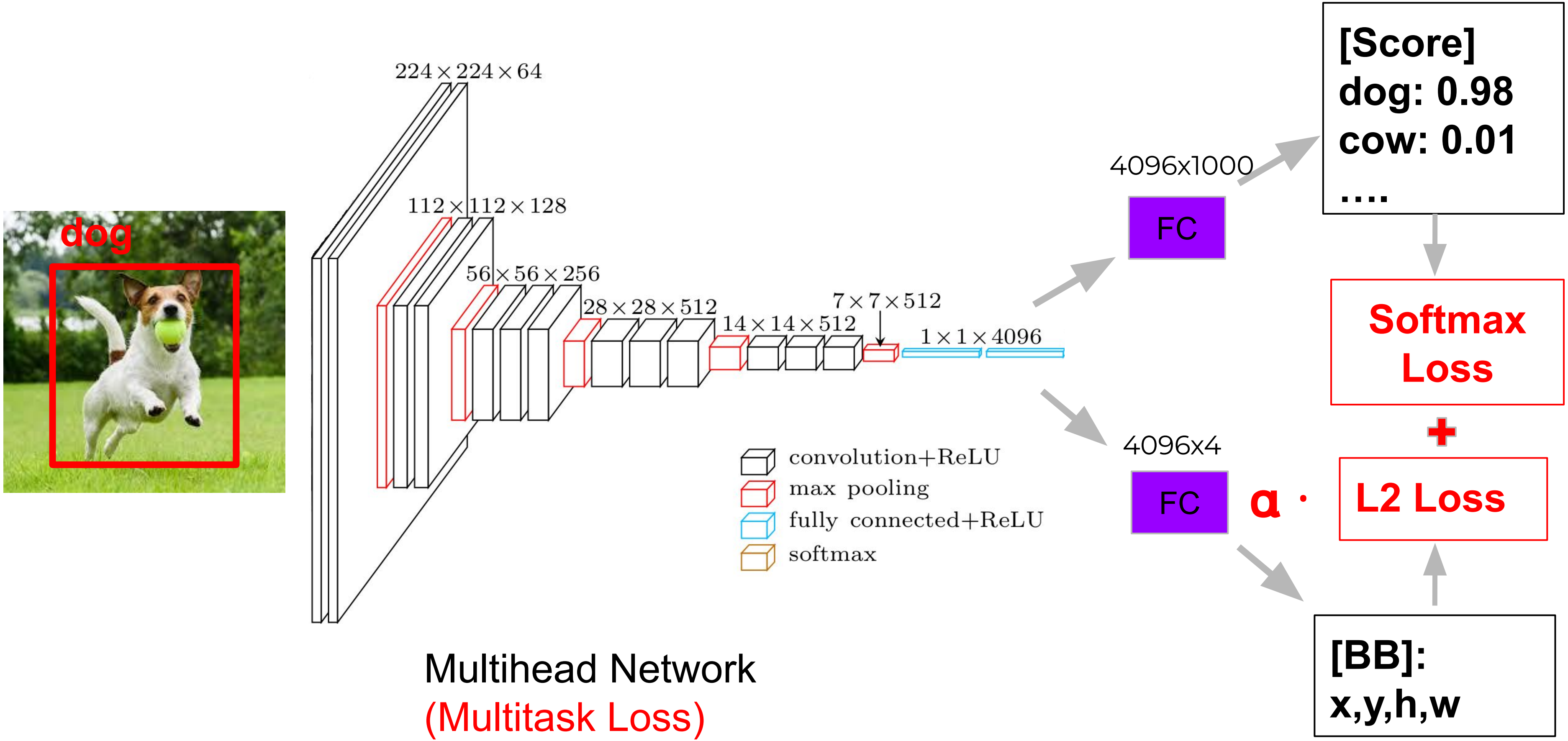
Multihead Network

Image Classification -> Classification + Localization



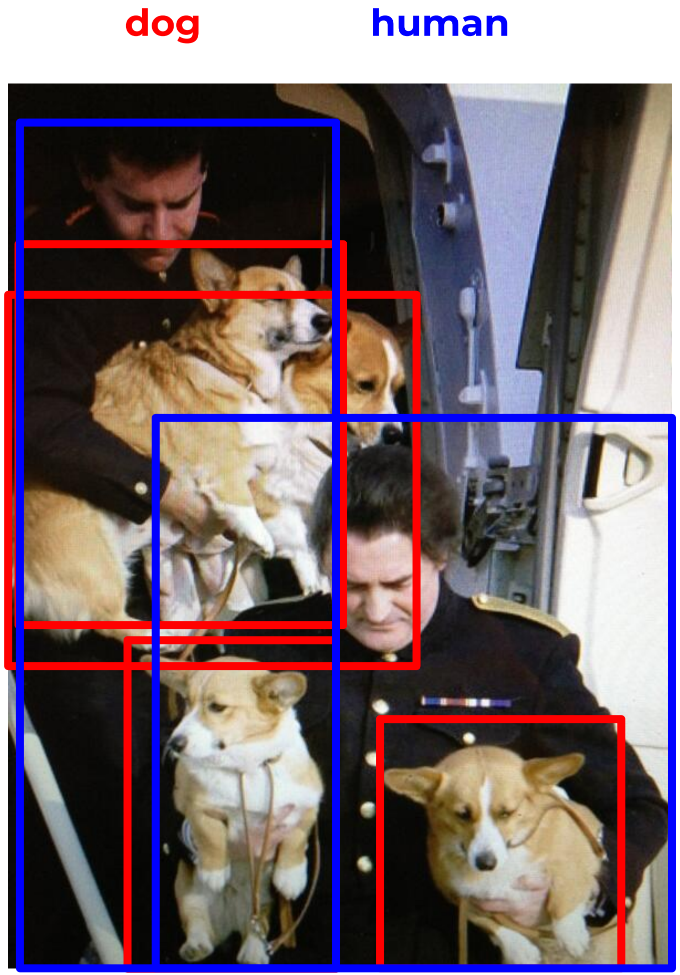
Multihead Network
(Multitask Loss)

Localization [Training]



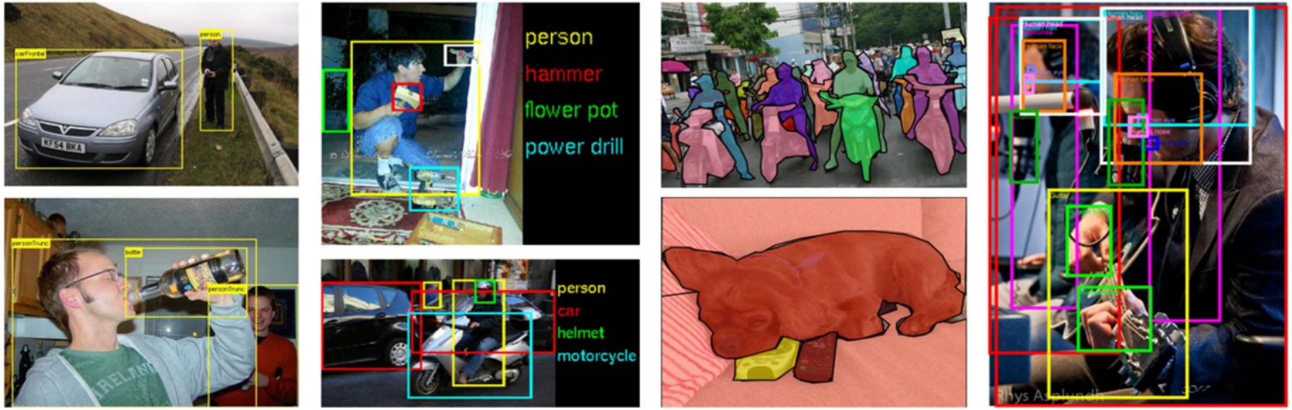
Multihead Network
(Multitask Loss)

Object Detection



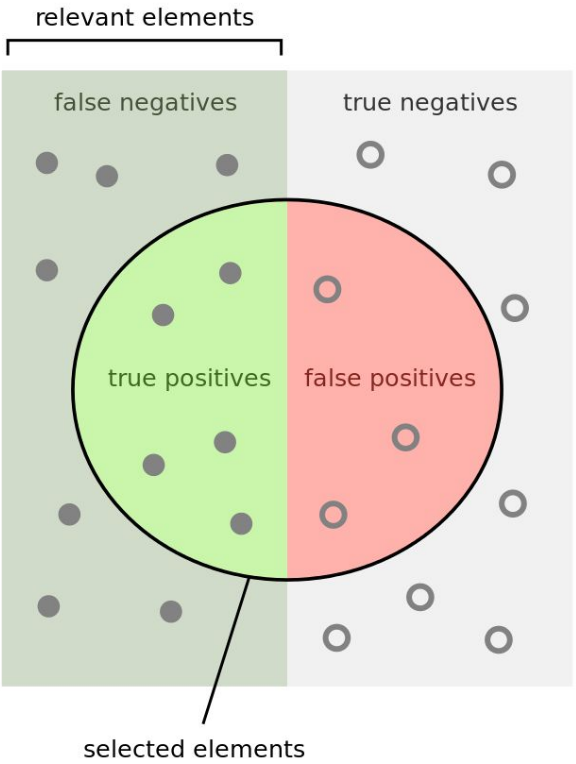
Object Detection [Datasets]

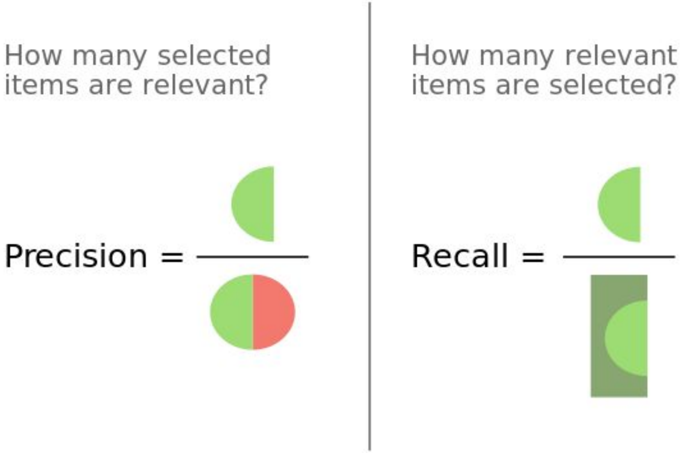
Dataset	train		validation		trainval		test	
	images	objects	images	objects	images	objects	images	objects
VOC-2007	2,501	6,301	2,510	6,307	5,011	12,608	4,952	14,976
VOC-2012	5,717	13,609	5,823	13,841	11,540	27,450	10,991	-
ILSVRC-2014	456,567	478,807	20,121	55,502	476,688	534,309	40,152	-
ILSVRC-2017	456,567	478,807	20,121	55,502	476,688	534,309	65,500	-
MS-COCO-2015	82,783	604,907	40,504	291,875	123,287	896,782	81,434	-
MS-COCO-2018	118,287	860,001	5,000	36,781	123,287	896,782	40,670	-
OID-2018	1,743,042	14,610,229	41,620	204,621	1,784,662	14,814,850	125,436	625,282



arXiv:1904.01569 VOC ILSVRC MS-COCO OID

Object Detection [Metrics]





$$\text{Precision} = \frac{tp}{tp + fp} \qquad \text{Recall} = \frac{tp}{tp + fn}$$

https://en.wikipedia.org/wiki/Precision_and_recall

Object Detection [Metrics]

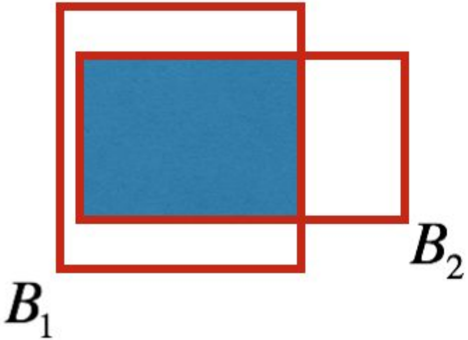
$$IoU = \frac{A \cap B}{A \cup B}$$

A - ground truth (GT)
B - detector result (P)

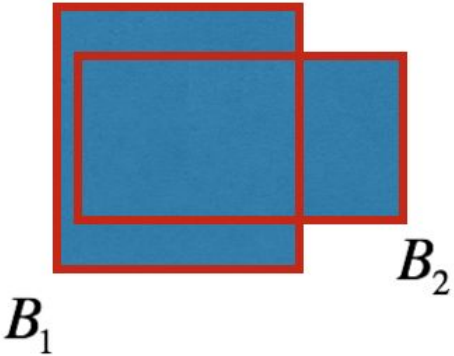
The detection is *true positive* if ***IoU ≥ threshold @0.5***

Object Detection [Metrics]

Intersection

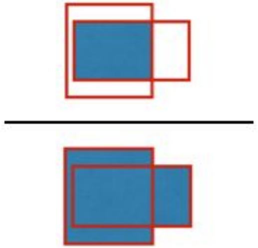


Union



Intersection over Union

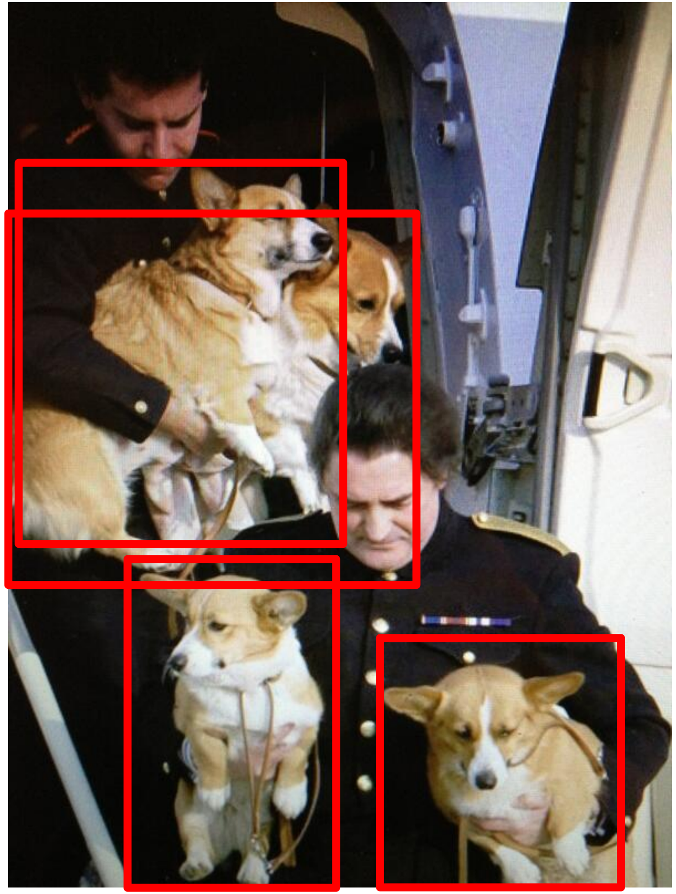
$$IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2} =$$



B_1 - ground truth (GT)
B_2 - detector result (P)

The detection is *true positive* if ***IoU ≥ threshold @0.5***

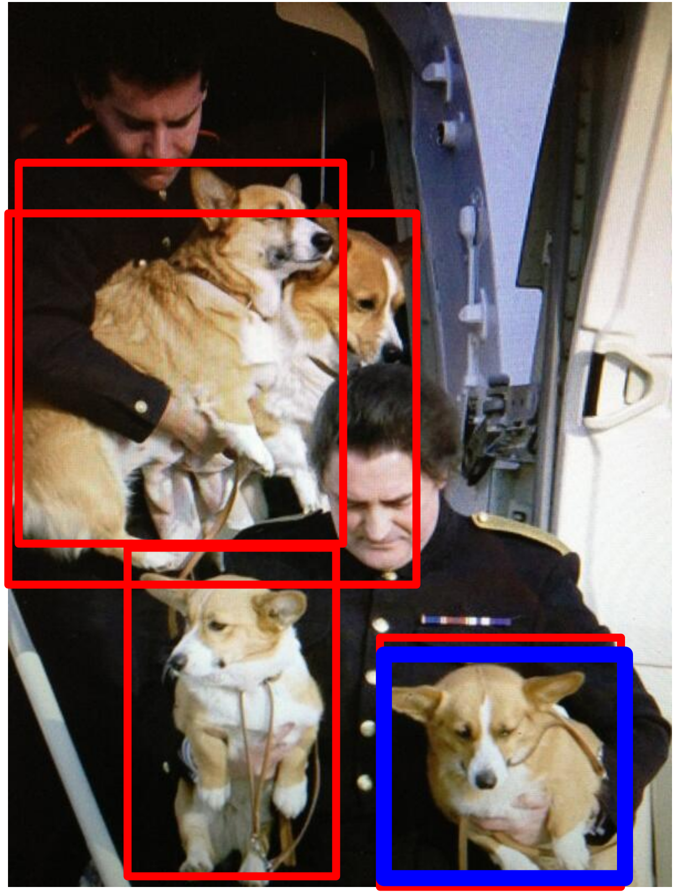
Object Detection [Metrics]



GT

P

Object Detection [Metrics]

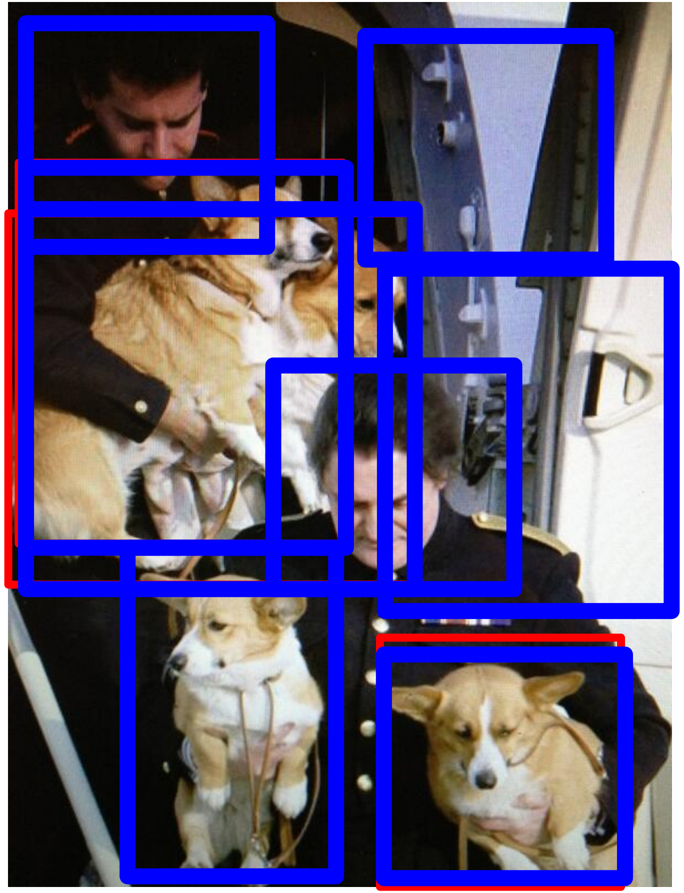


GT

P

Precision ?
Recall ?

Object Detection [Metrics]



GT

P

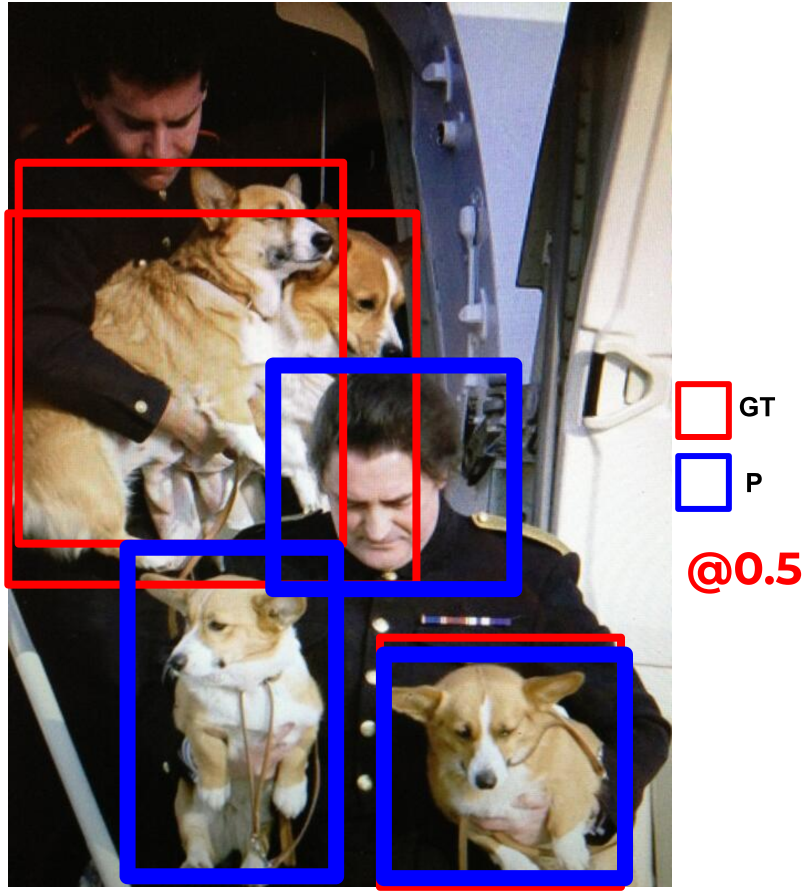
Precision ?

Recall ?

Object Detection

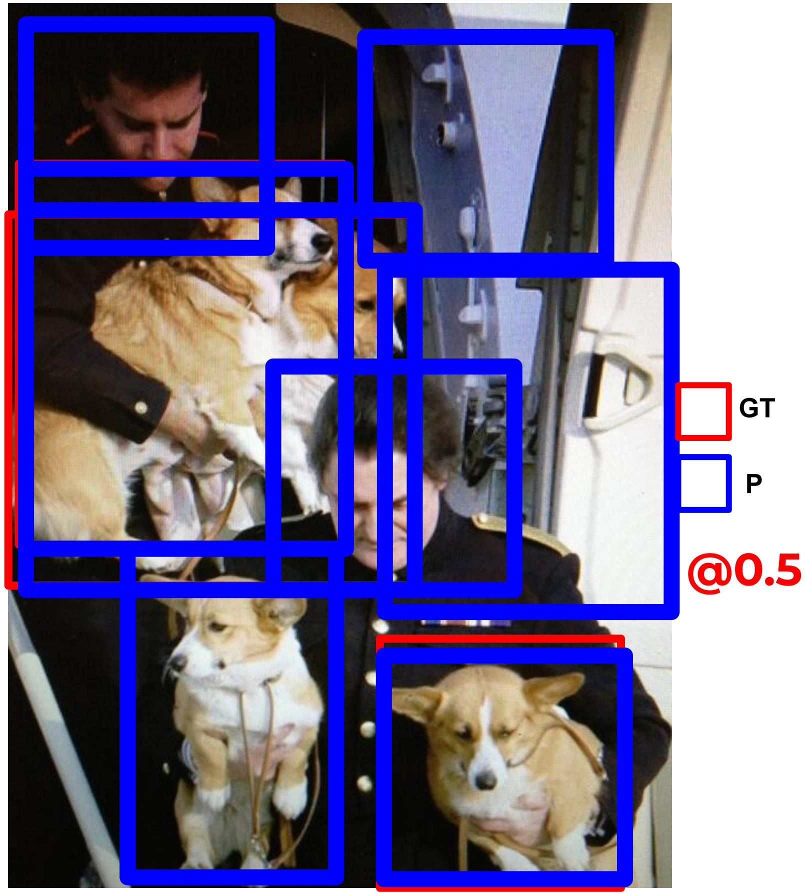
score	correct	Precision	Recall
1.0	True		

Object Detection



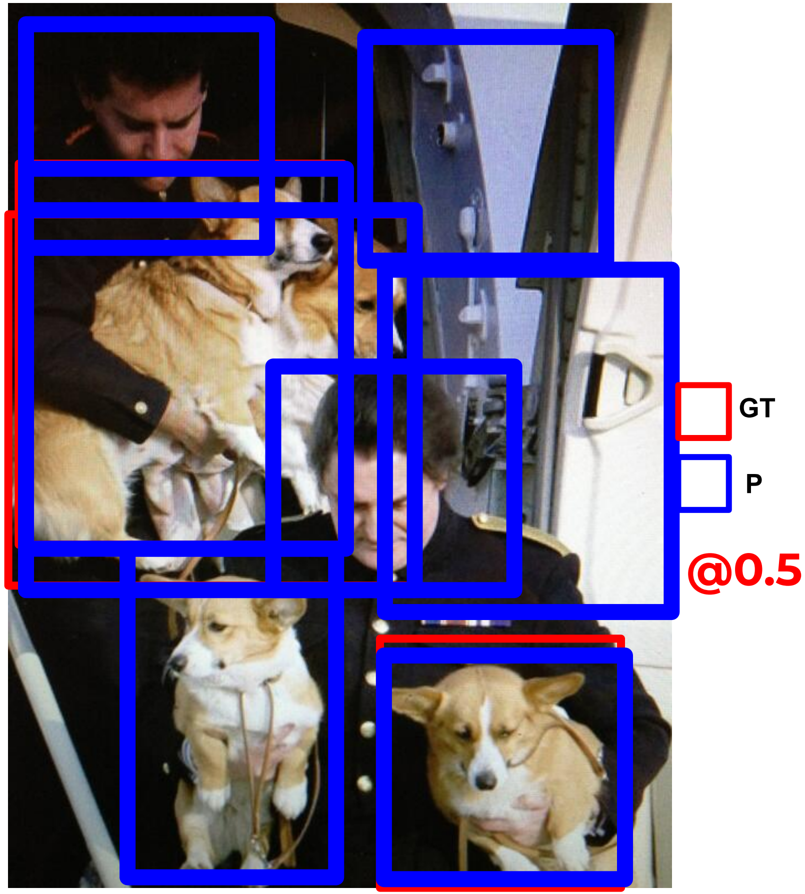
score	correct	Precision	Recall
1.0	True		
0.98	True		
0.97	False		

Object Detection



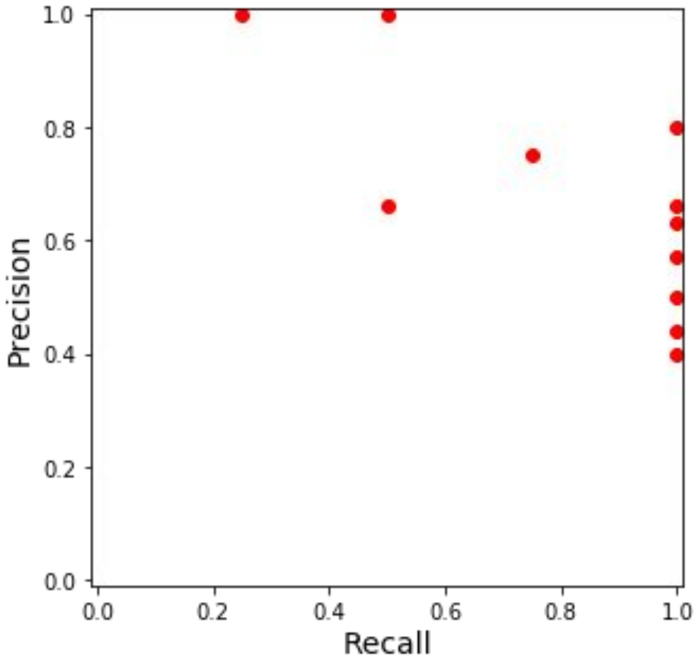
score	correct	Precision	Recall
1.0	True		
0.98	True		
0.97	False		
0.81	True		
0.77	True		
0.67	False		
0.53	True		
0.49	False		
0.33	False		
0.22	False		
0.11	False		

Object Detection



score	correct	Precision	Recall
1.0	True	1.00	0.25
0.98	True	1.00	0.5
0.97	False	0.66	0.5
0.81	True	0.75	0.75
0.77	True	0.80	1.00
0.67	False	0.66	1.00
0.53	True	0.63	1.00
0.49	False	0.57	1.00
0.33	False	0.50	1.00
0.28	False	0.44	1.00
0.14	False	0.4	1.00

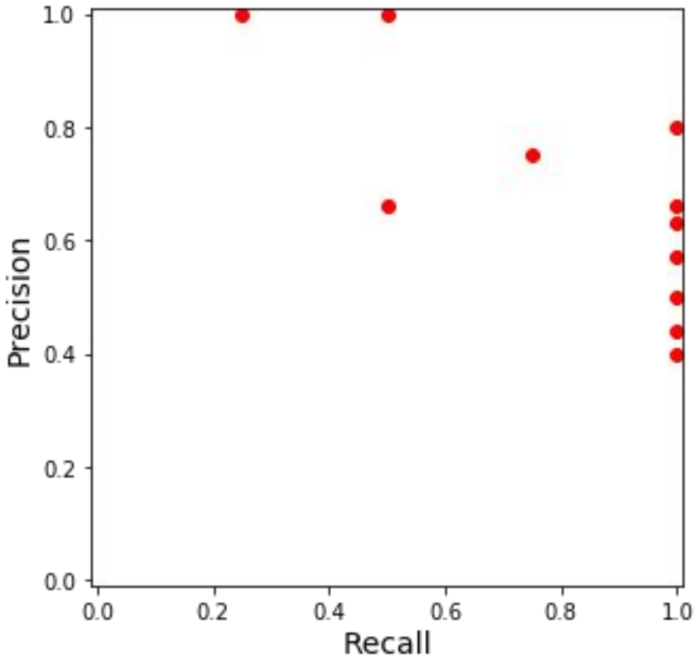
Object Detection

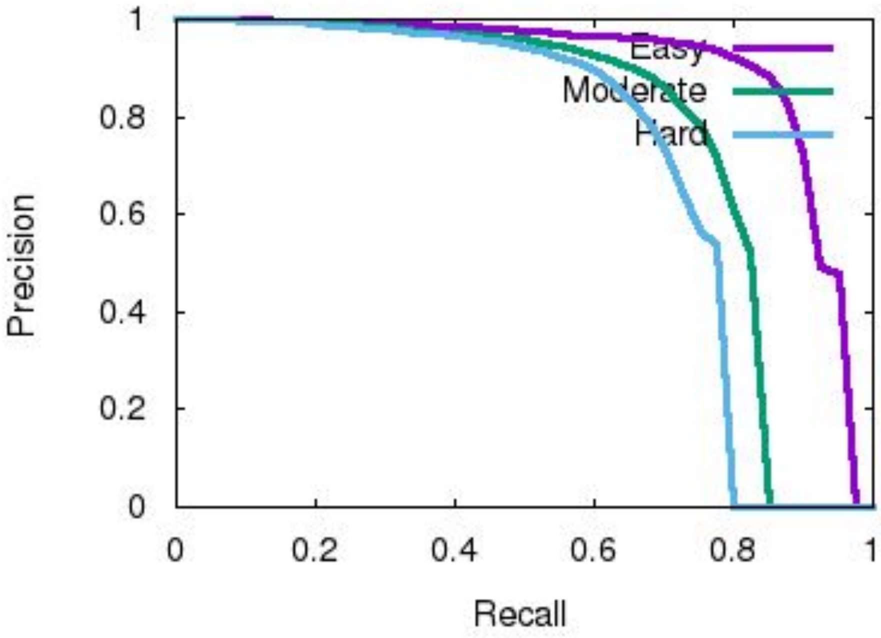


@0.5

score	correct	Precision	Recall
1.0	True	1.00	0.25
0.98	True	1.00	0.5
0.97	False	0.66	0.5
0.81	True	0.75	0.75
0.77	True	0.80	1.00
0.67	False	0.66	1.00
0.53	True	0.63	1.00
0.49	False	0.57	1.00
0.33	False	0.50	1.00
0.28	False	0.44	1.00
0.14	False	0.4	1.00

Object Detection [Metrics]





Object Detection [Metrics]

Average Precision (AP):

AP	% AP at IoU=.50:.05:.95 **(primary challenge metric)**
$AP^{IoU=.50}$	% AP at IoU=.50 (PASCAL VOC metric)
$AP^{IoU=.75}$	% AP at IoU=.75 (strict metric)

AP Across Scales:

AP^{small}	% AP for small objects: area < 32^2
AP^{medium}	% AP for medium objects: 32^2 < area < 96^2
AP^{large}	% AP for large objects: area > 96^2

Average Recall (AR):

$AR^{max=1}$	% AR given 1 detection per image
$AR^{max=10}$	% AR given 10 detections per image
$AR^{max=100}$	% AR given 100 detections per image

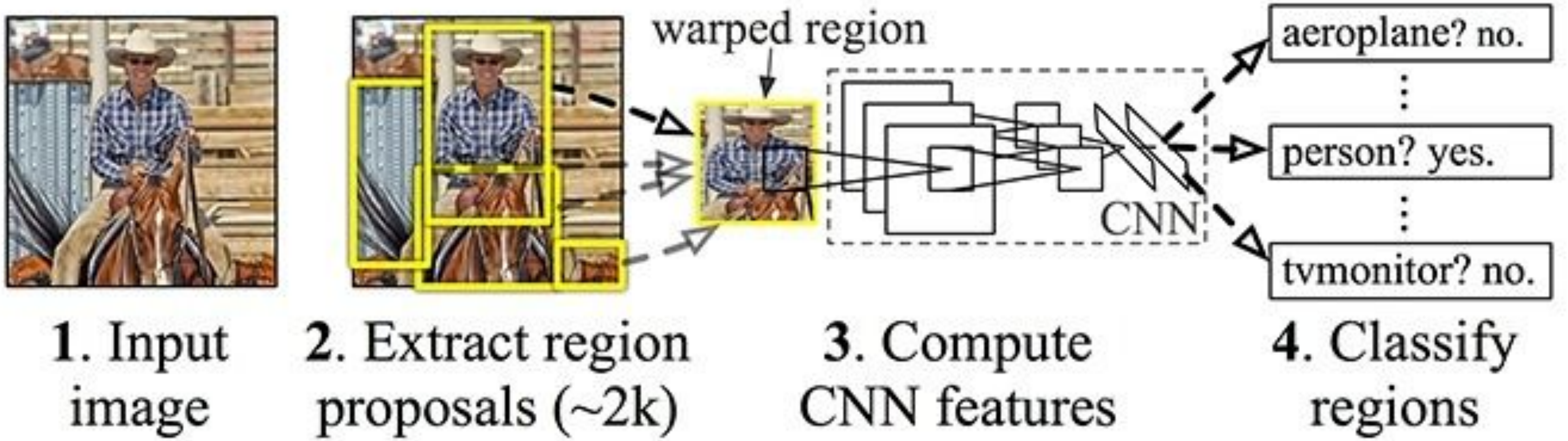
AR Across Scales:

AR^{small}	% AR for small objects: area < 32^2
AR^{medium}	% AR for medium objects: 32^2 < area < 96^2
AR^{large}	% AR for large objects: area > 96^2

Content

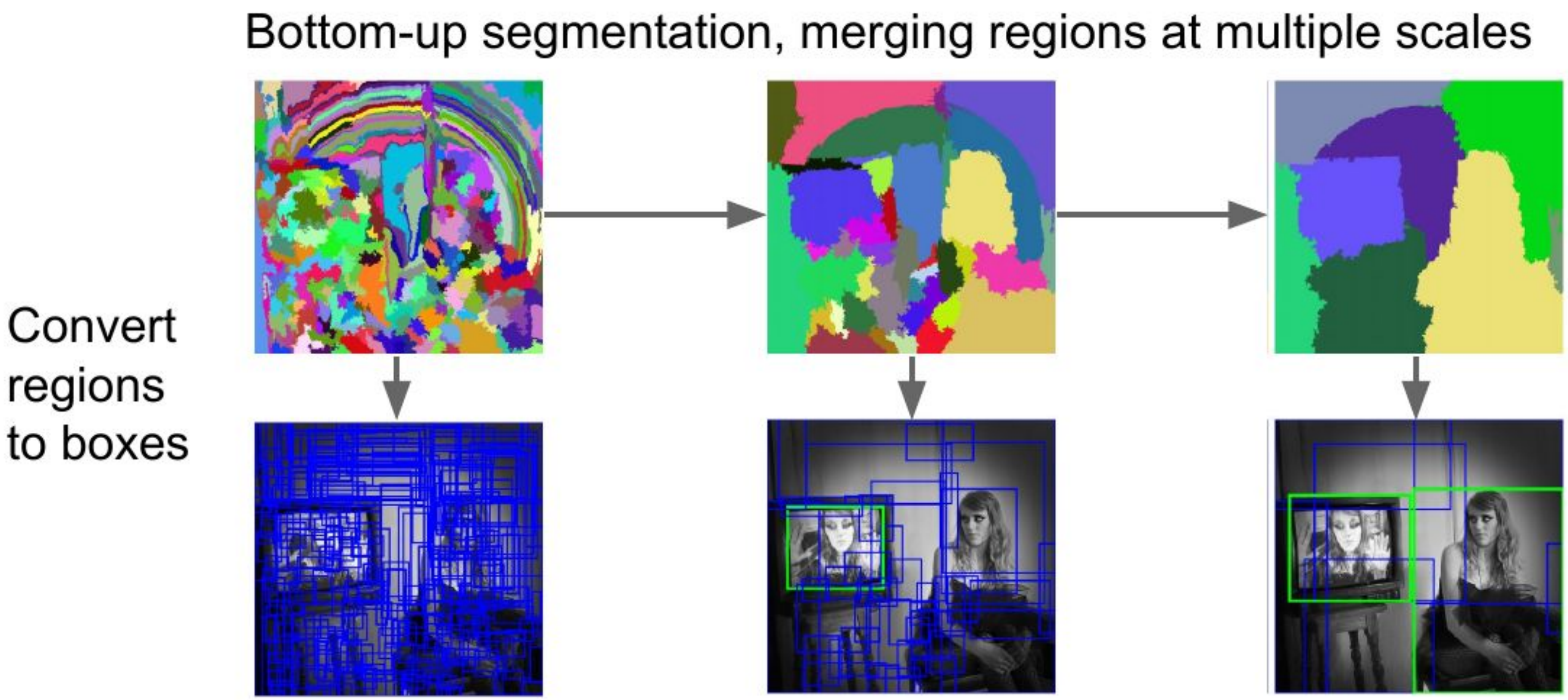
- Intro to Object Detection
  - Datasets
  - Metrics
- **Two-stage detectors [RCNN Family]**
- One-stage detectors
  - YOLO
  - Single Shot Detector [SSD]
- Proposal-free detection
- Summary

R-CNN: Regions with CNN features



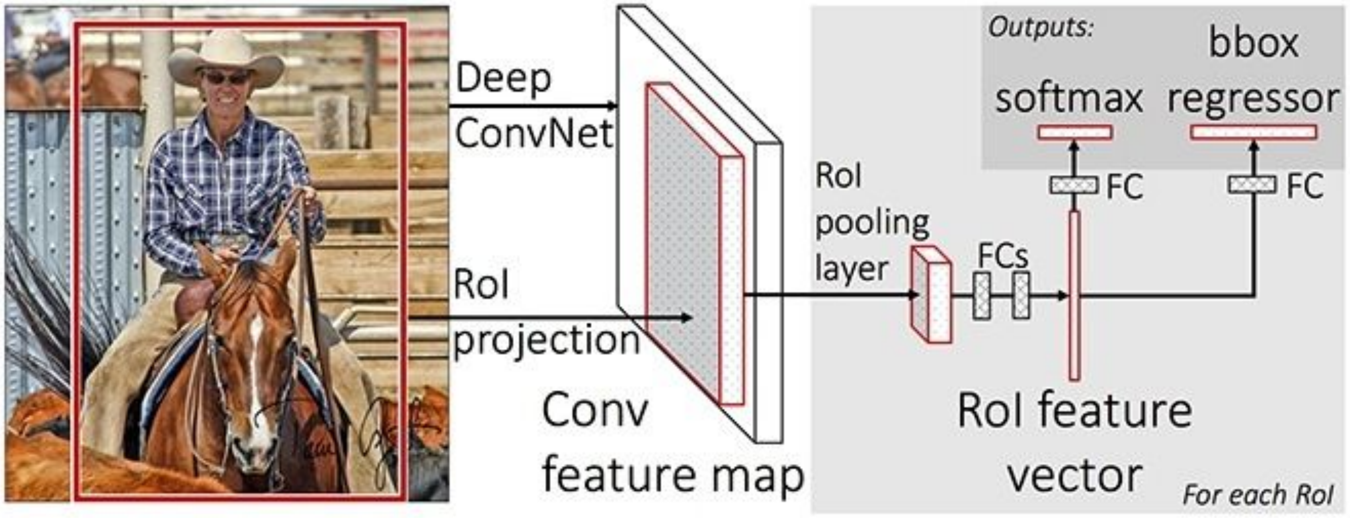
- Extract possible objects using a region proposal method (the most popular one being Selective Search).
- Extract features from each region using a CNN.
- Classify each region with SVMs.

Object Detection [Region Proposal]



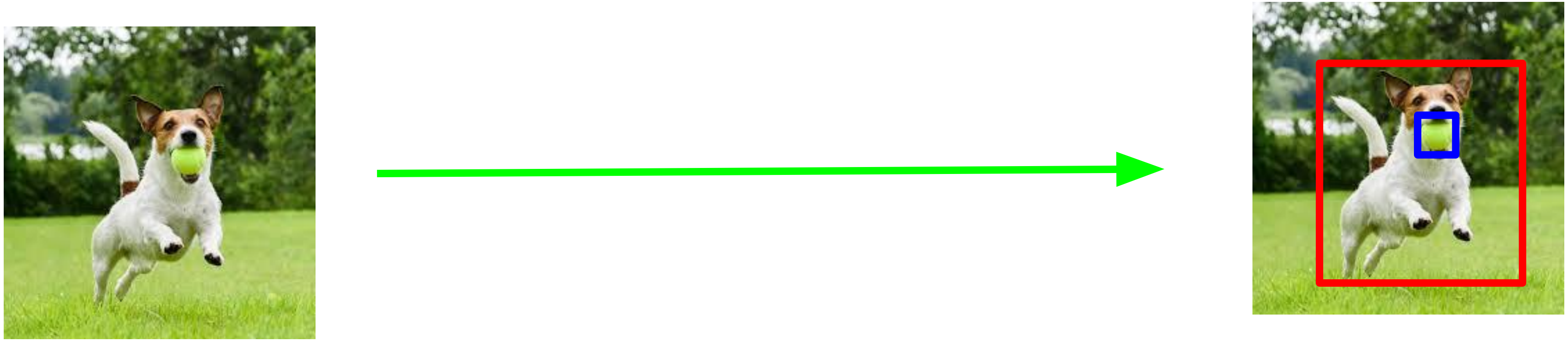
Uijlings et al, "Selective Search for Object Recognition", IJCV 2013

Fast R-CNN



- Use Selective Search to generate object proposals;
- Apply the CNN on the complete image and then used both **Region of Interest (RoI) Pooling** on the feature map;
- Use feed forward network for classification and regression;

Faster R-CNN (Overall architecture)

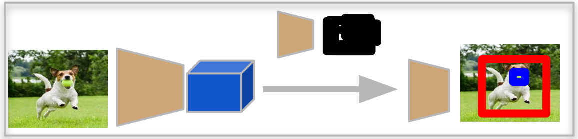


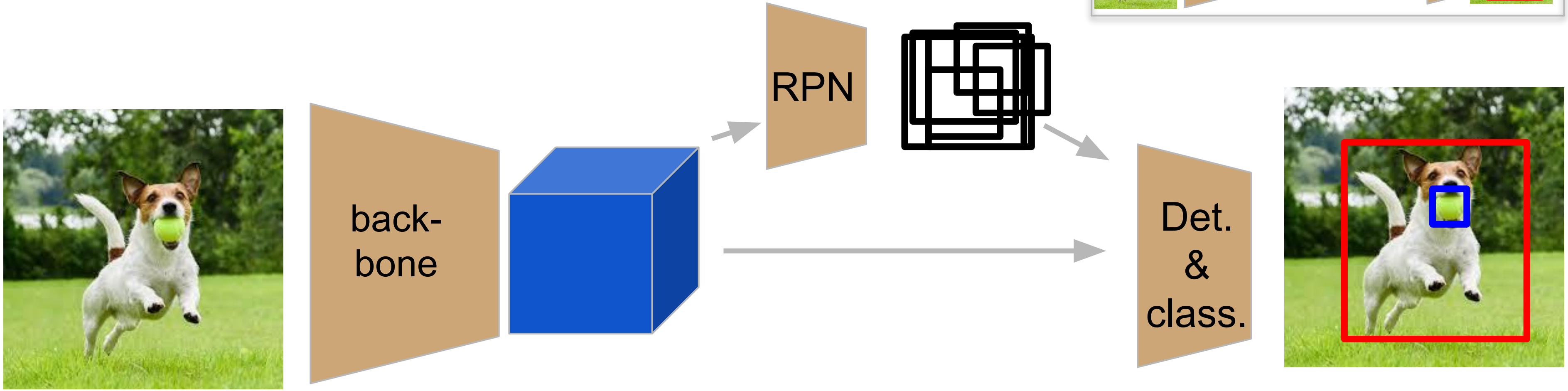
dog, score: 0.98

ball, score: 0.6

1506.01497

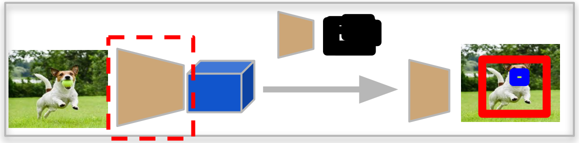
Faster R-CNN [Overall architecture]

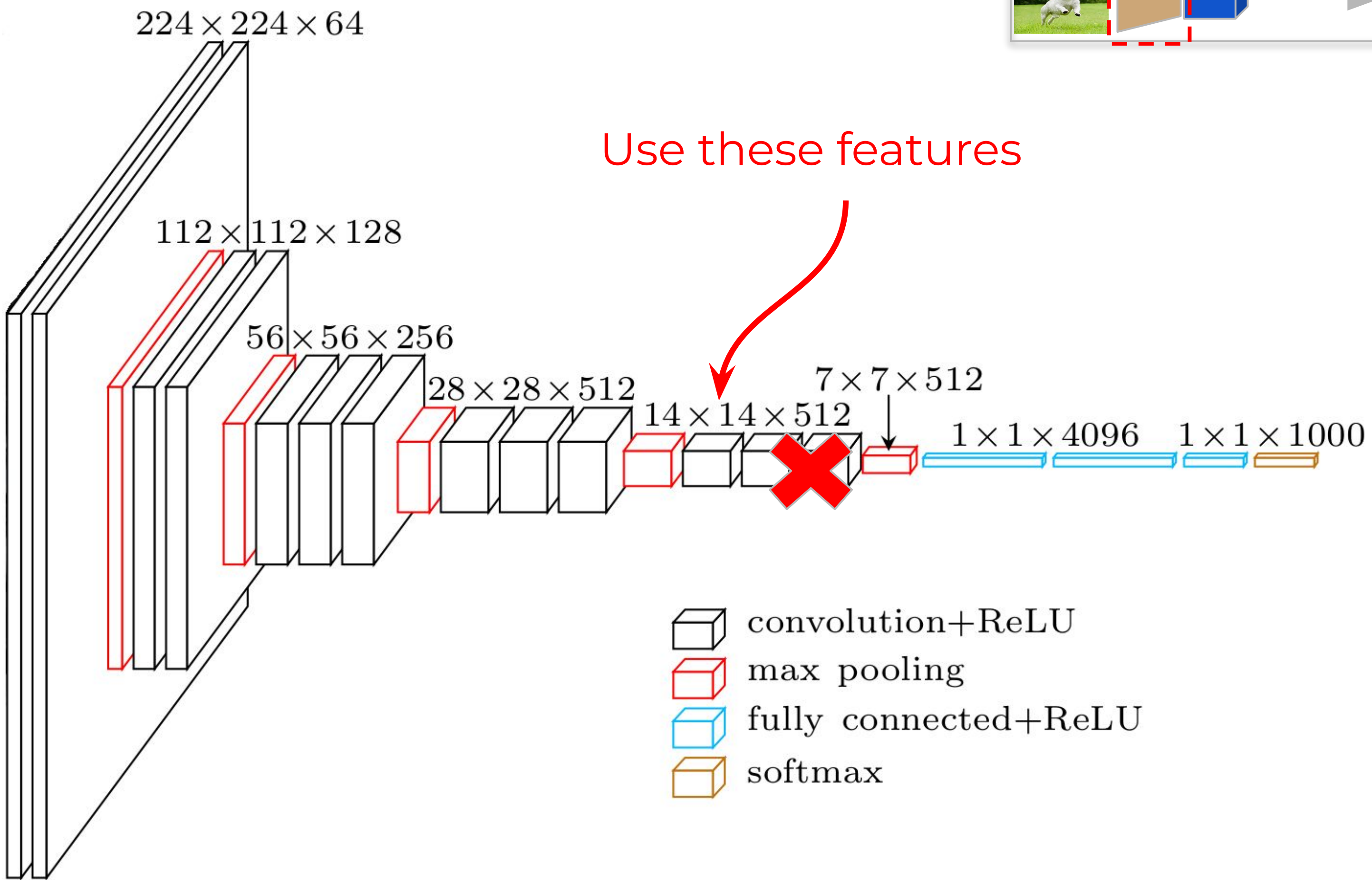




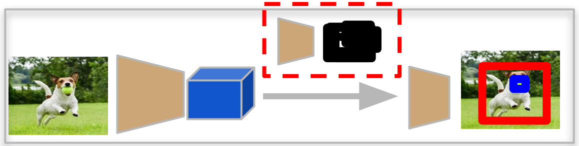
dog, score: 0.98
ball, score: 0.8

Faster R-CNN [Base Network]





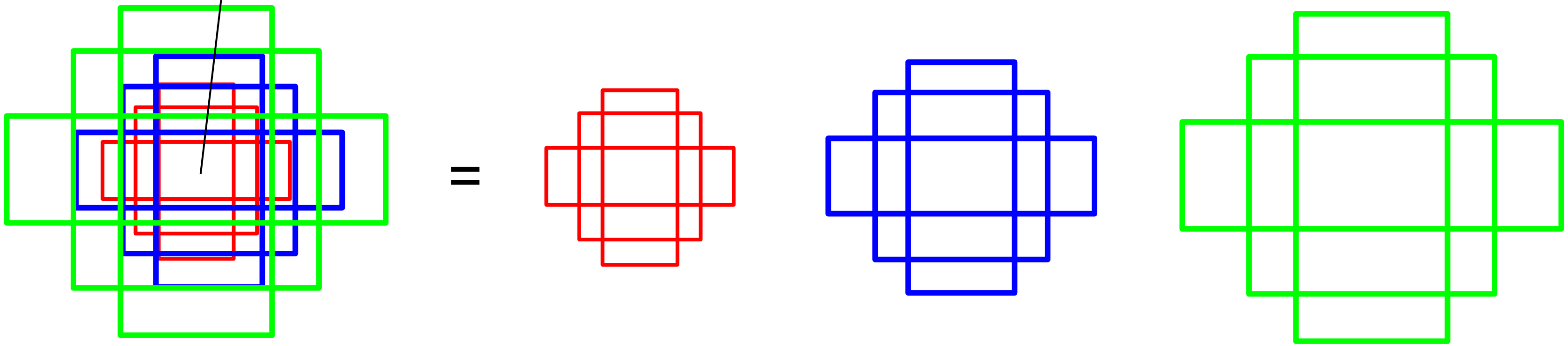
Faster R-CNN [Anchors]



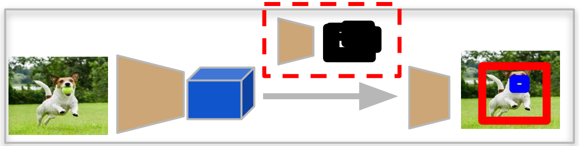
Anchors are fixed bounding boxes that are placed throughout the image with different sizes and ratios that are going to be used for reference when first predicting object locations.

- set of sizes (e.g. 64px, 128px, 256px)
- set of ratios between width and height of boxes (e.g. 0.5, 1, 1.5)

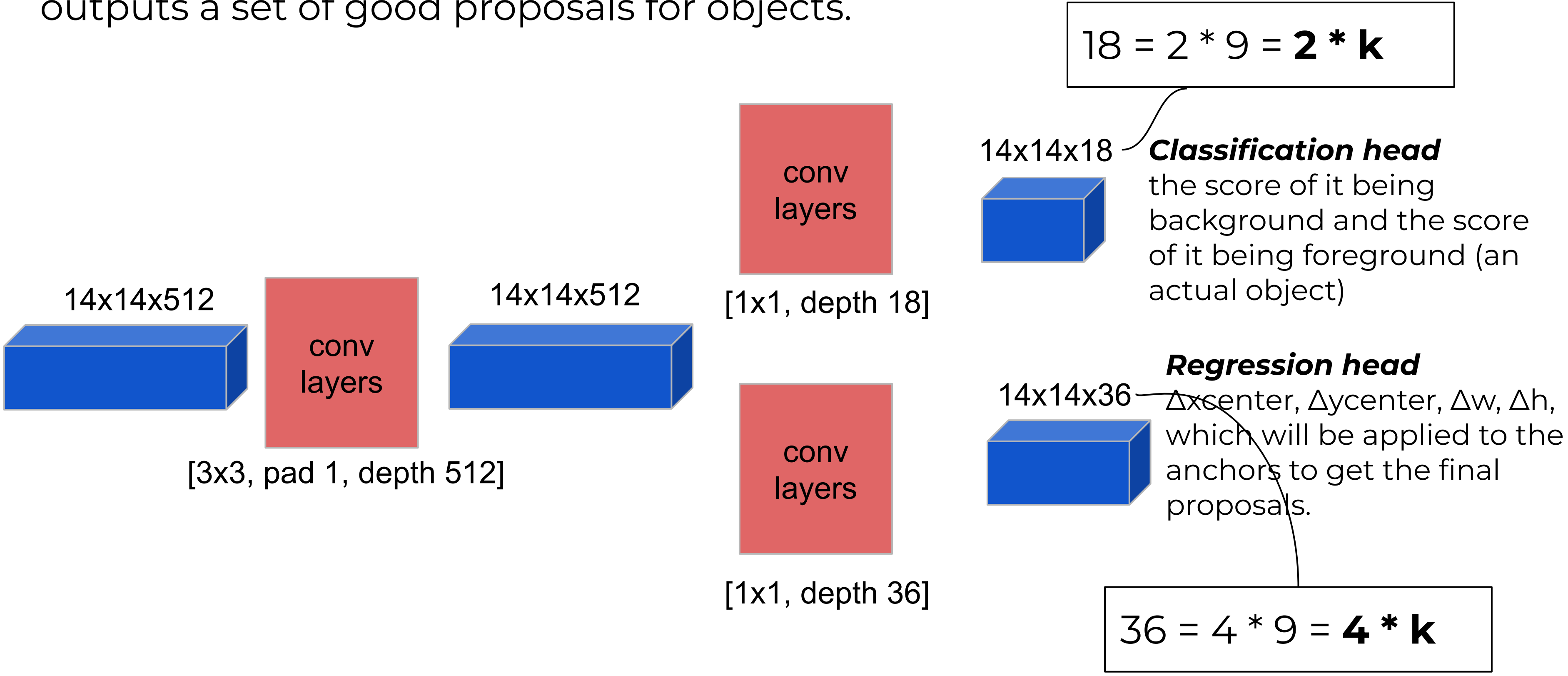
= 9



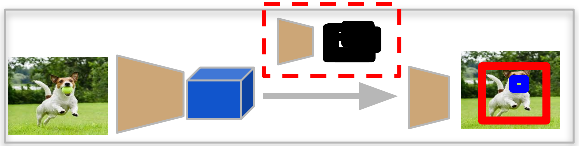
Faster R-CNN [Region Proposal Network]

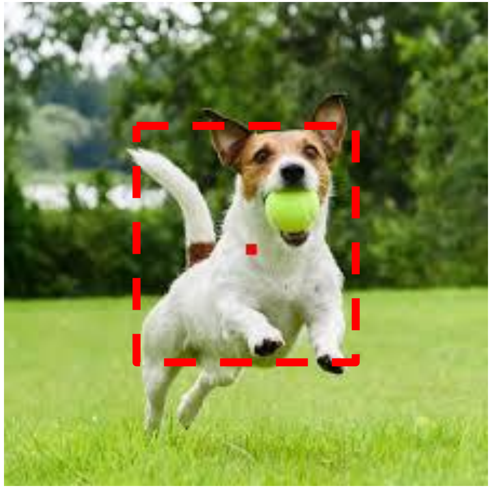


RPN takes all the reference boxes (anchors) and outputs a set of good proposals for objects.



Faster R-CNN [Region Proposal Network]





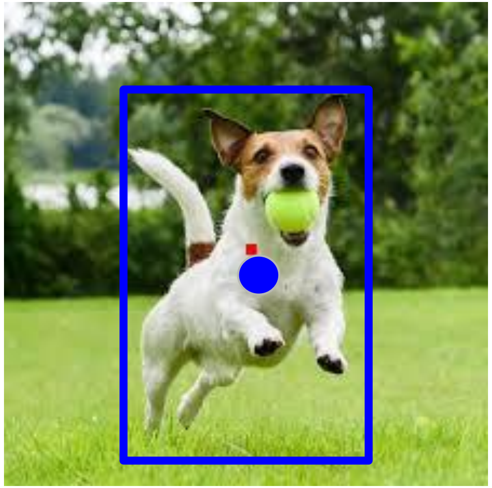
default anchor
position and size:
Xa, Ya, Wa, Ha

<u>RPN output:</u>

score (object) = 0.9
score (background) = 0.1

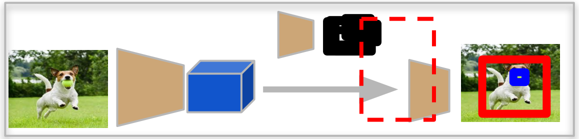
Δxcenter = 3
Δycenter = 10
Δw = 10
Δh = 40

$$X = \Delta xcenter * Wa + Xa$$
$$Y = \Delta ycenter * Wa + Ya$$
$$W = Wa * \exp(\Delta w)$$
$$H = Ha * \exp(\Delta h)$$

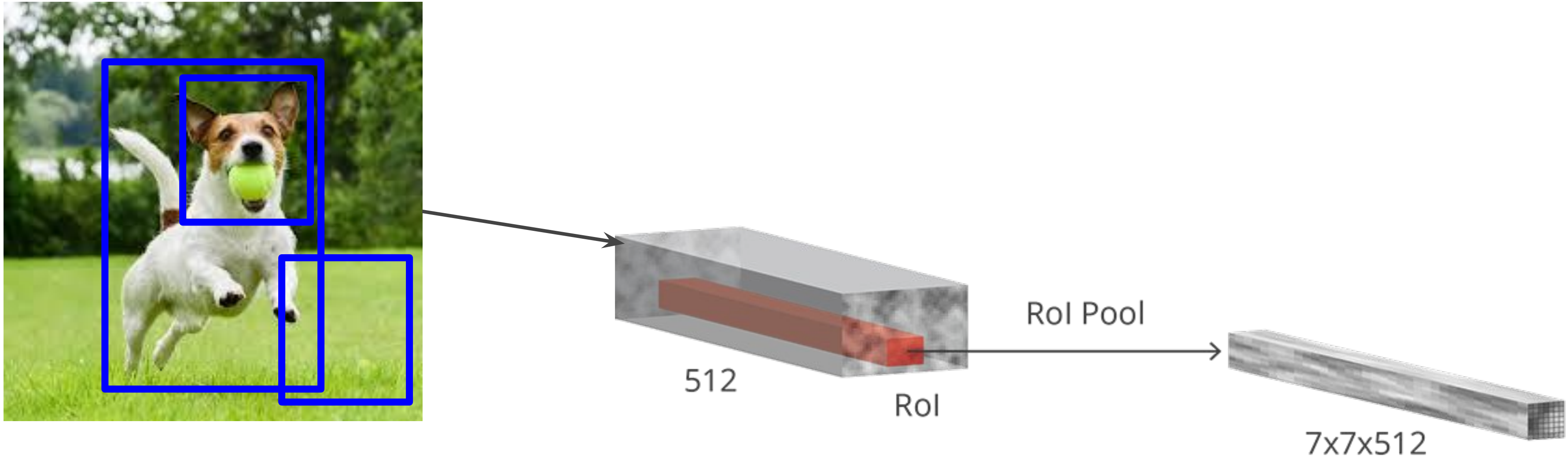


proposal box
position and size:
X, Y, W, H

Faster R-CNN [Region of Interest Pooling]

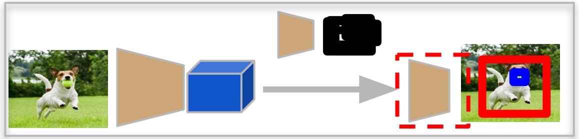


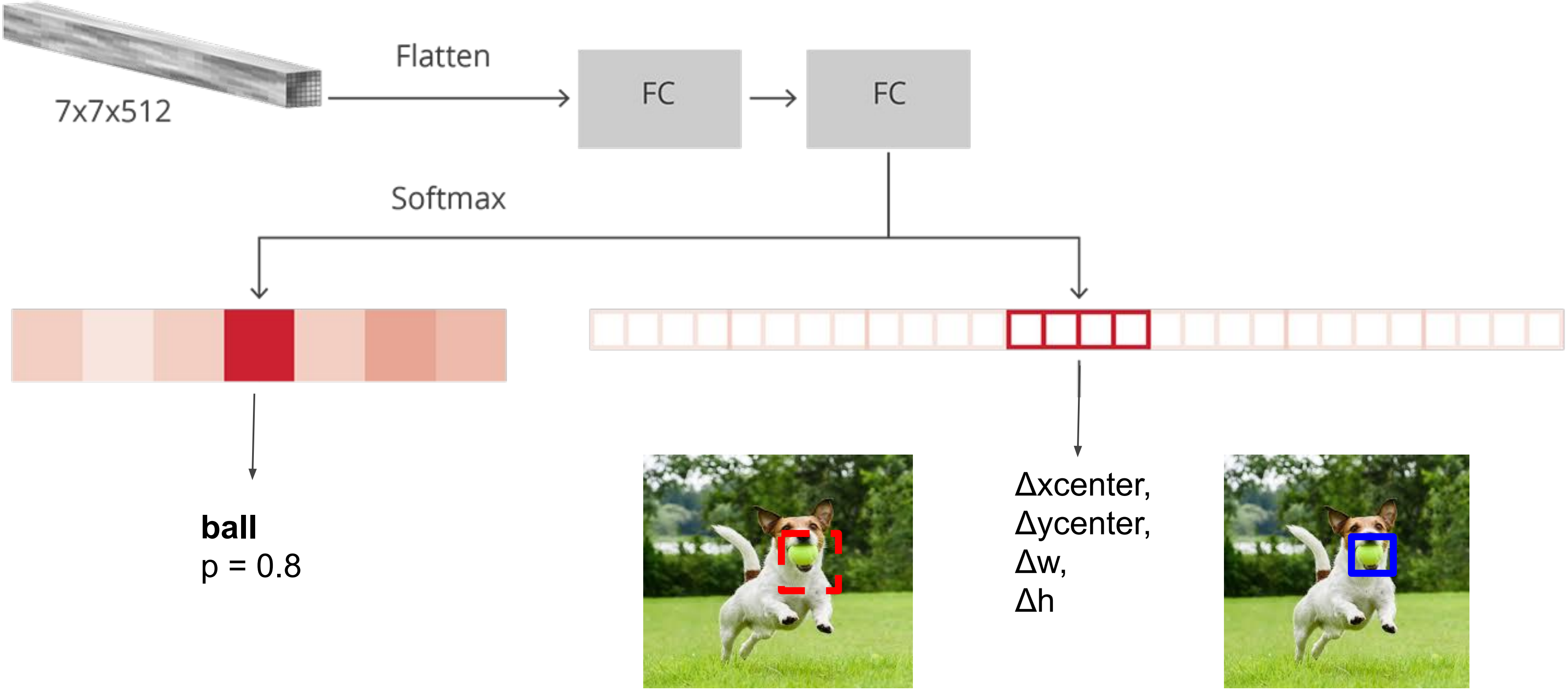
Fixed size feature maps are needed for the det.&class. part (R-CNN) in order to classify them into a fixed number of classes.



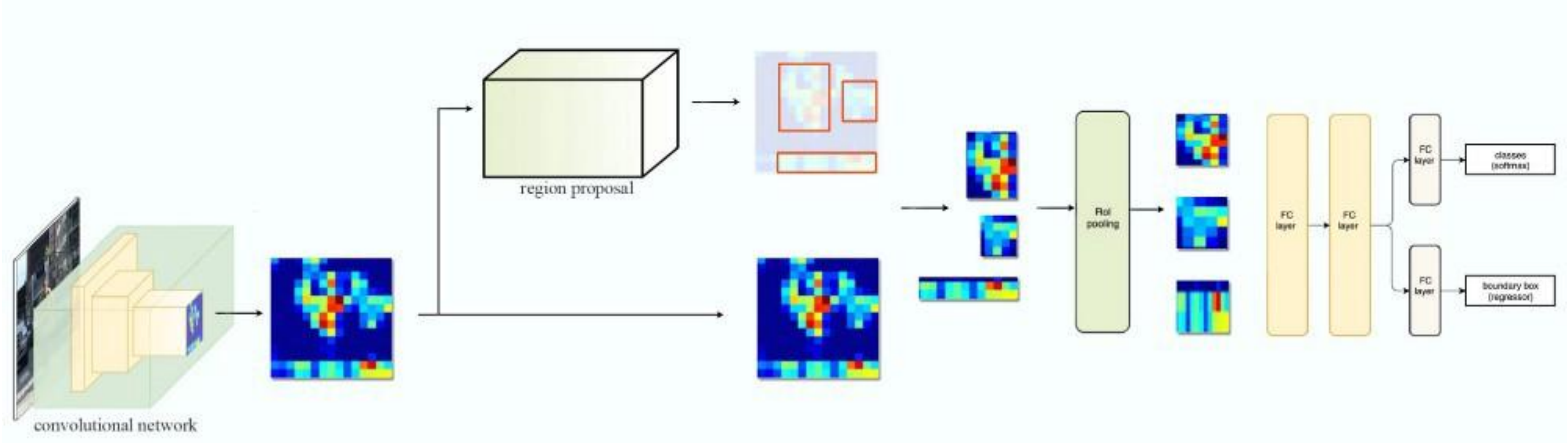
crop the convolutional features using each proposal and then resize to a fixed sized 14×14×D using interpolation (usually bilinear). After cropping, max pooling with a 2x2 kernel is used to get a final 7×7×D for each proposal.

Faster R-CNN [Region-based CNN]



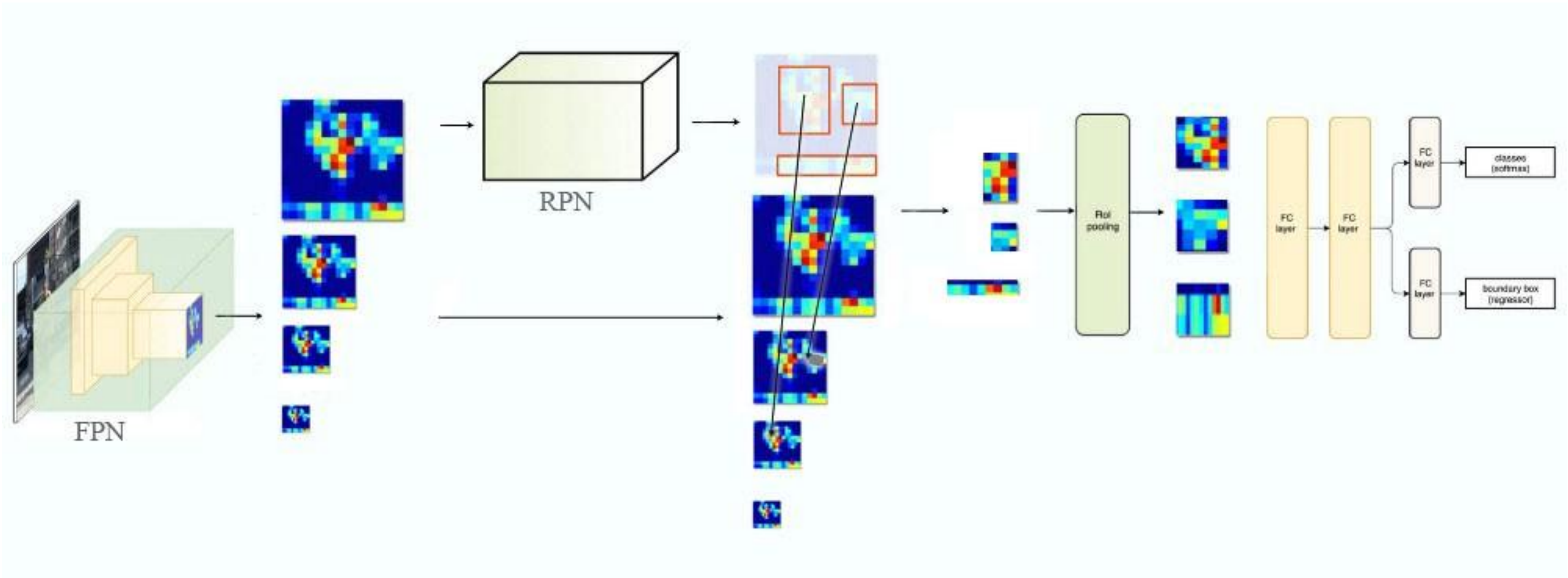


Faster RCNN (normal)



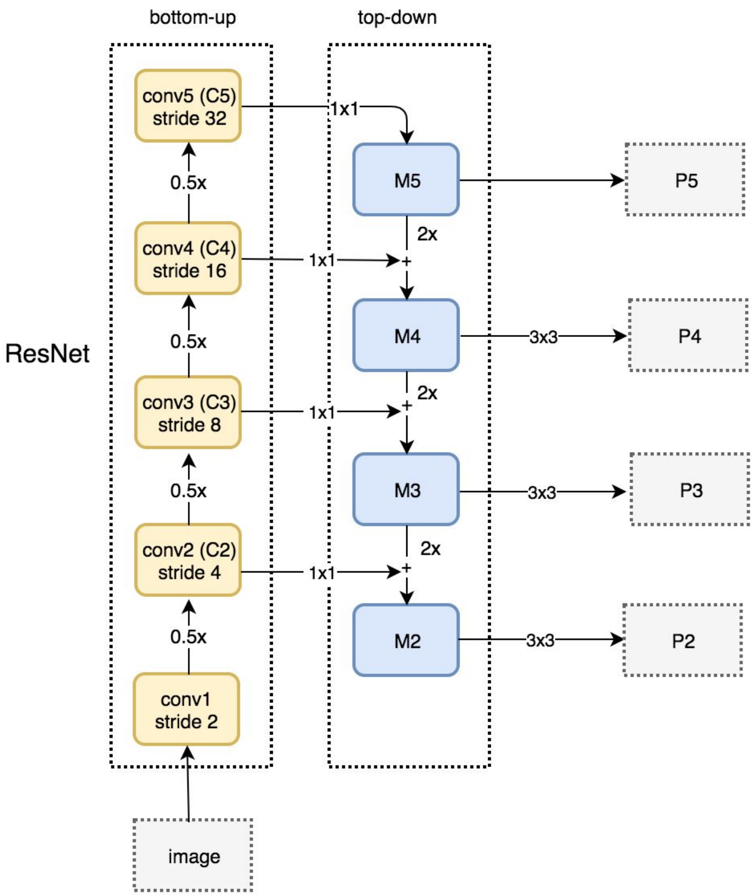
https://medium.com/@jonathan_hui/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c

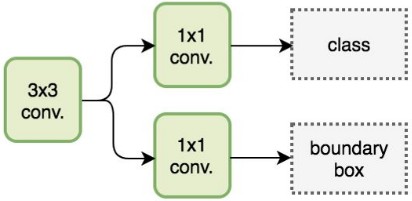
FPN for Faster RCNN



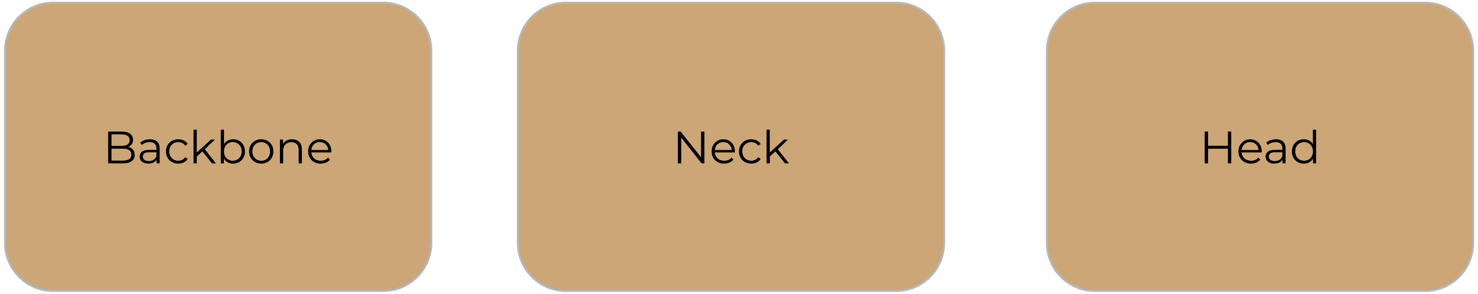
https://medium.com/@jonathan_hui/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c

Feature Pyramid Network (FPN)





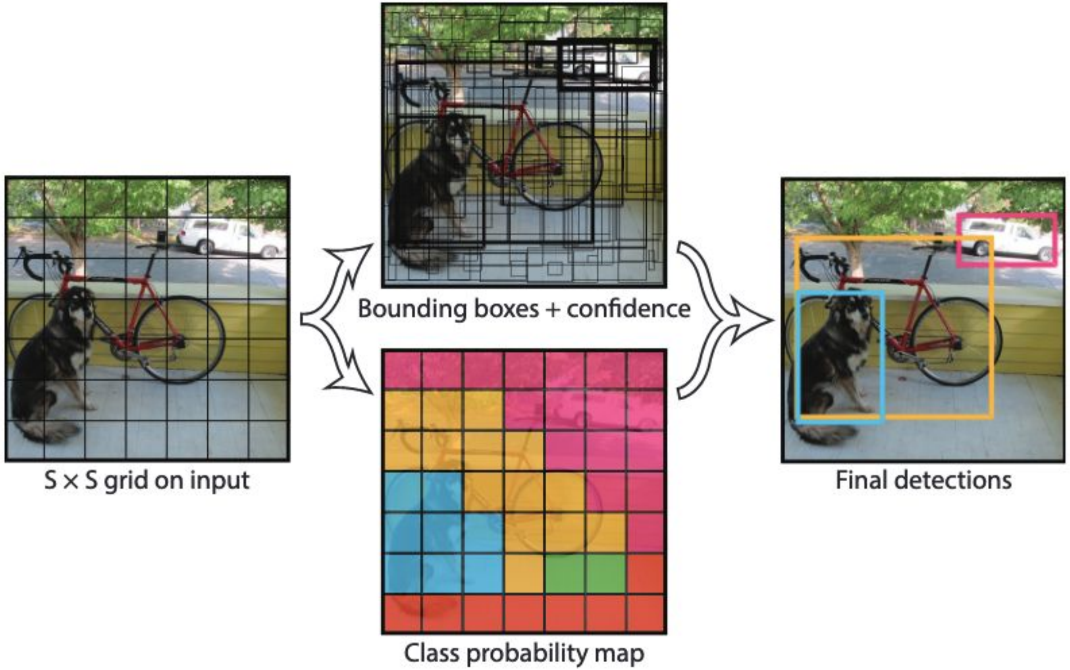
Faster RCNN



Content

- Intro to Object Detection
  - Datasets
  - Metrics
- Two-stage detectors [RCNN Family]
- **One-stage detectors**
  - YOLO
  - Single Shot Detector [SSD]
- Proposal-free detection
- Summary

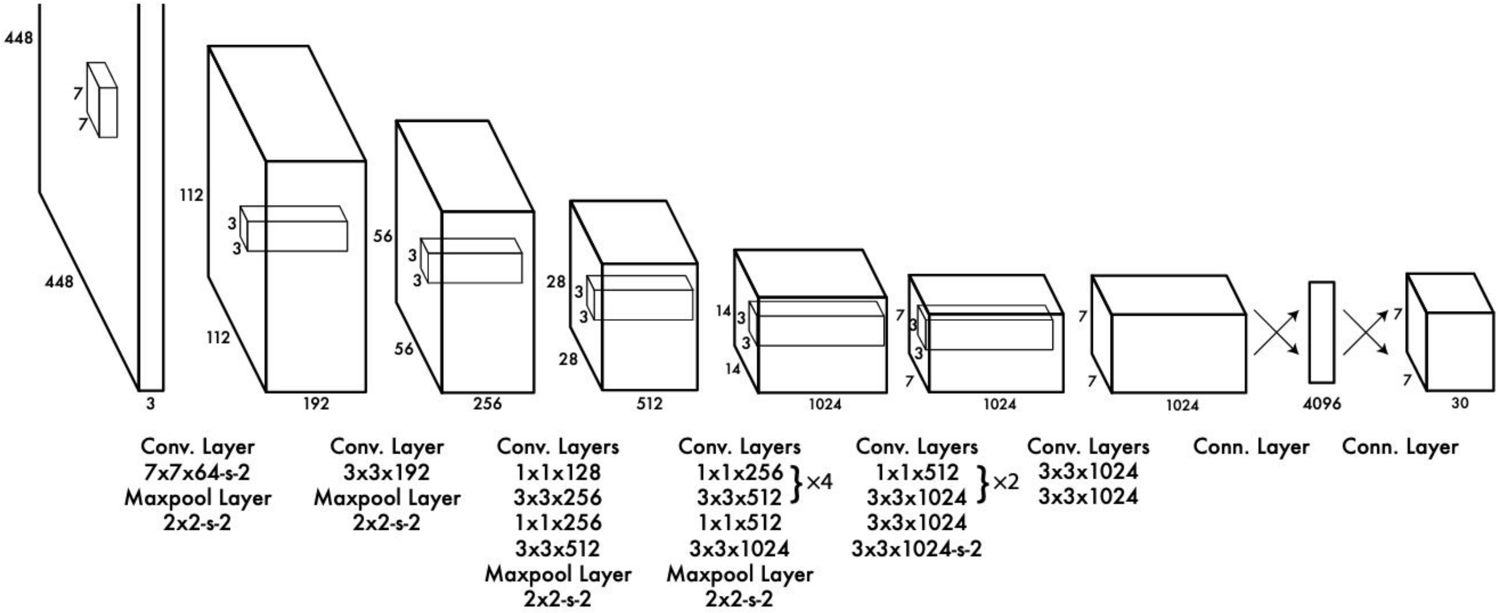
YOLO



YOLO models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor

1506.02640

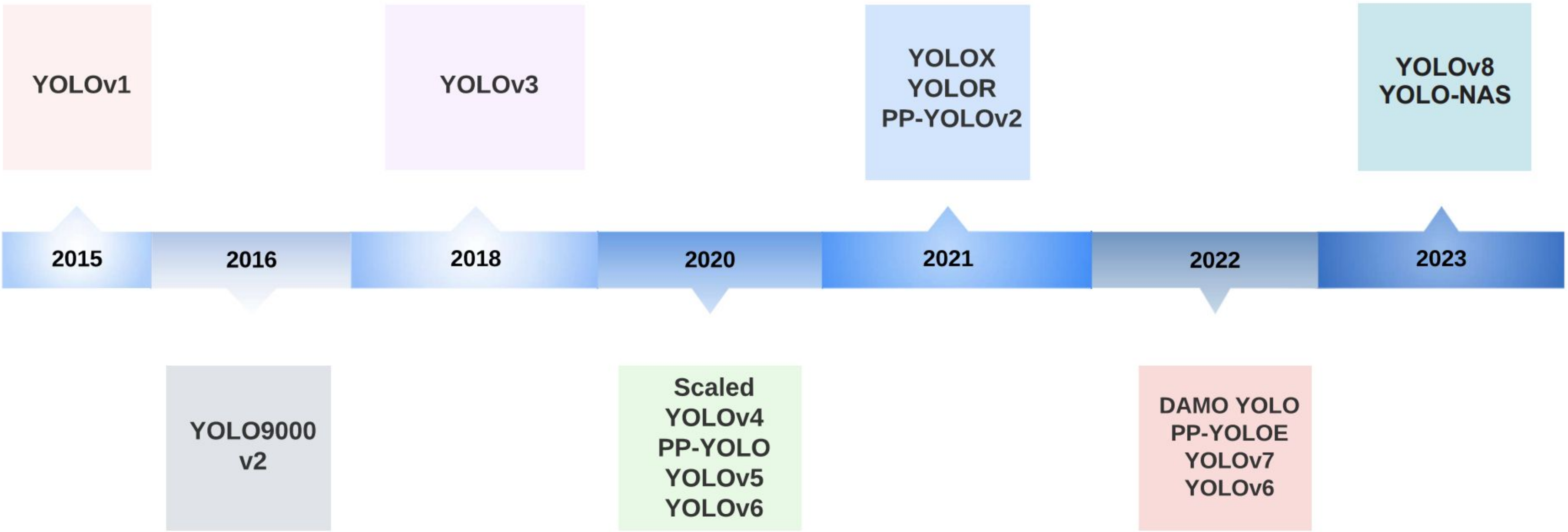
YOLO



YOLO models detection as a regression problem. It divides the image into an S ×S grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an S × S × (B * 5 + C) tensor

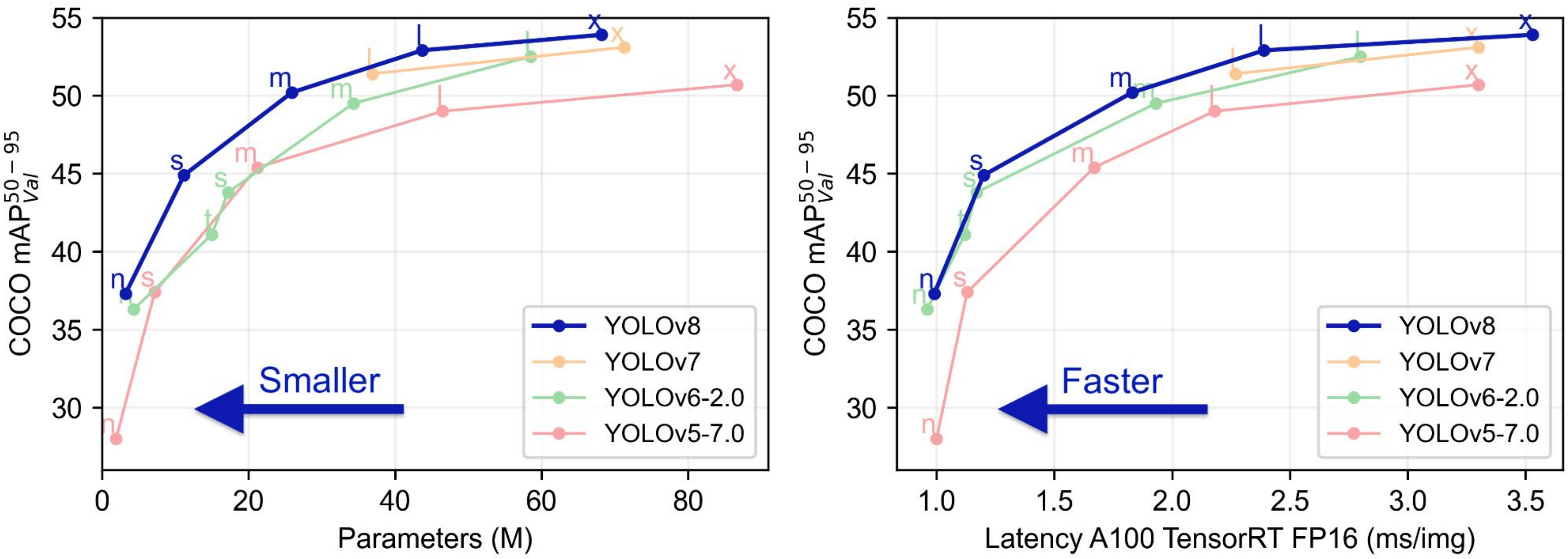
1506.02640

YOLO



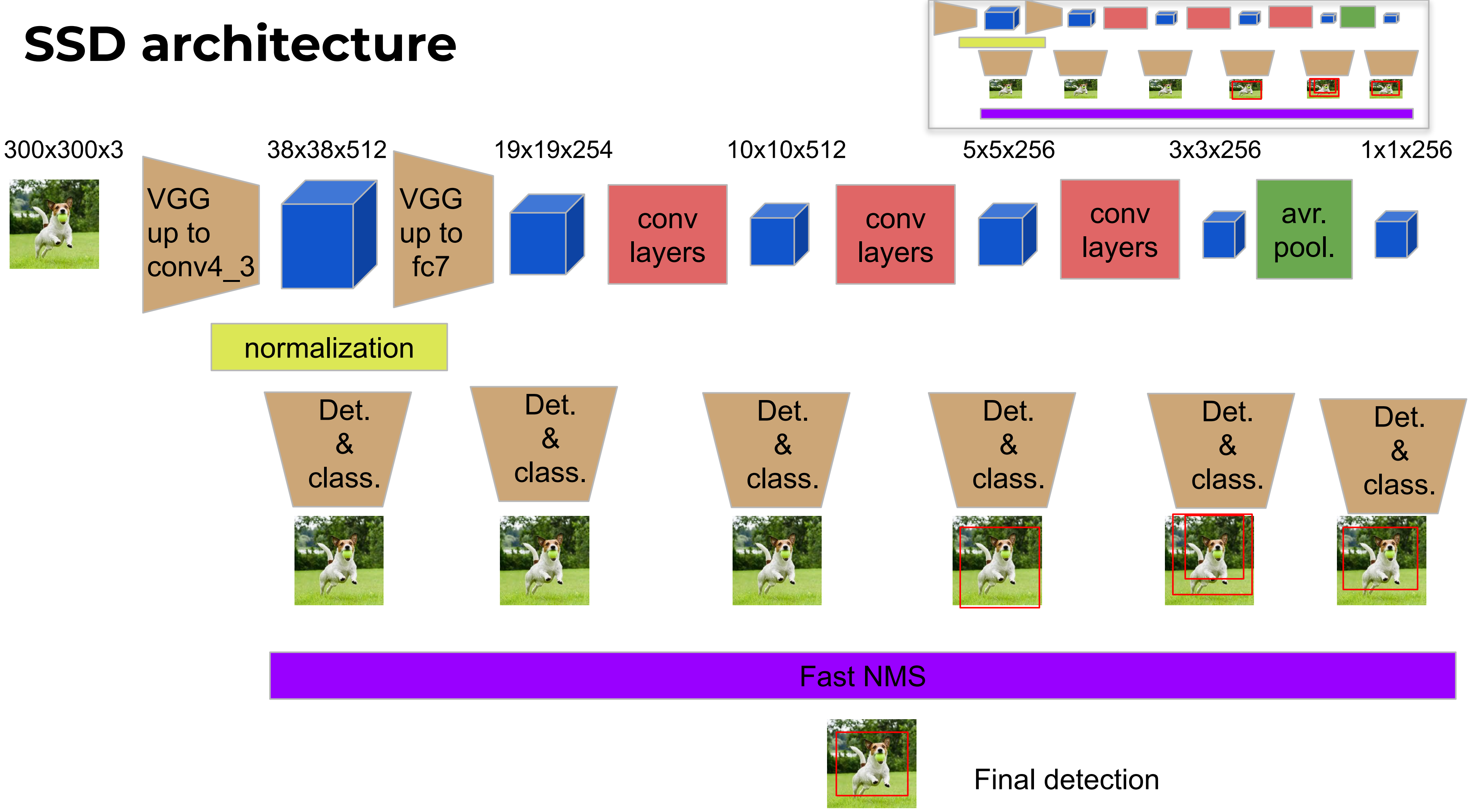
2304.00501

YOLO

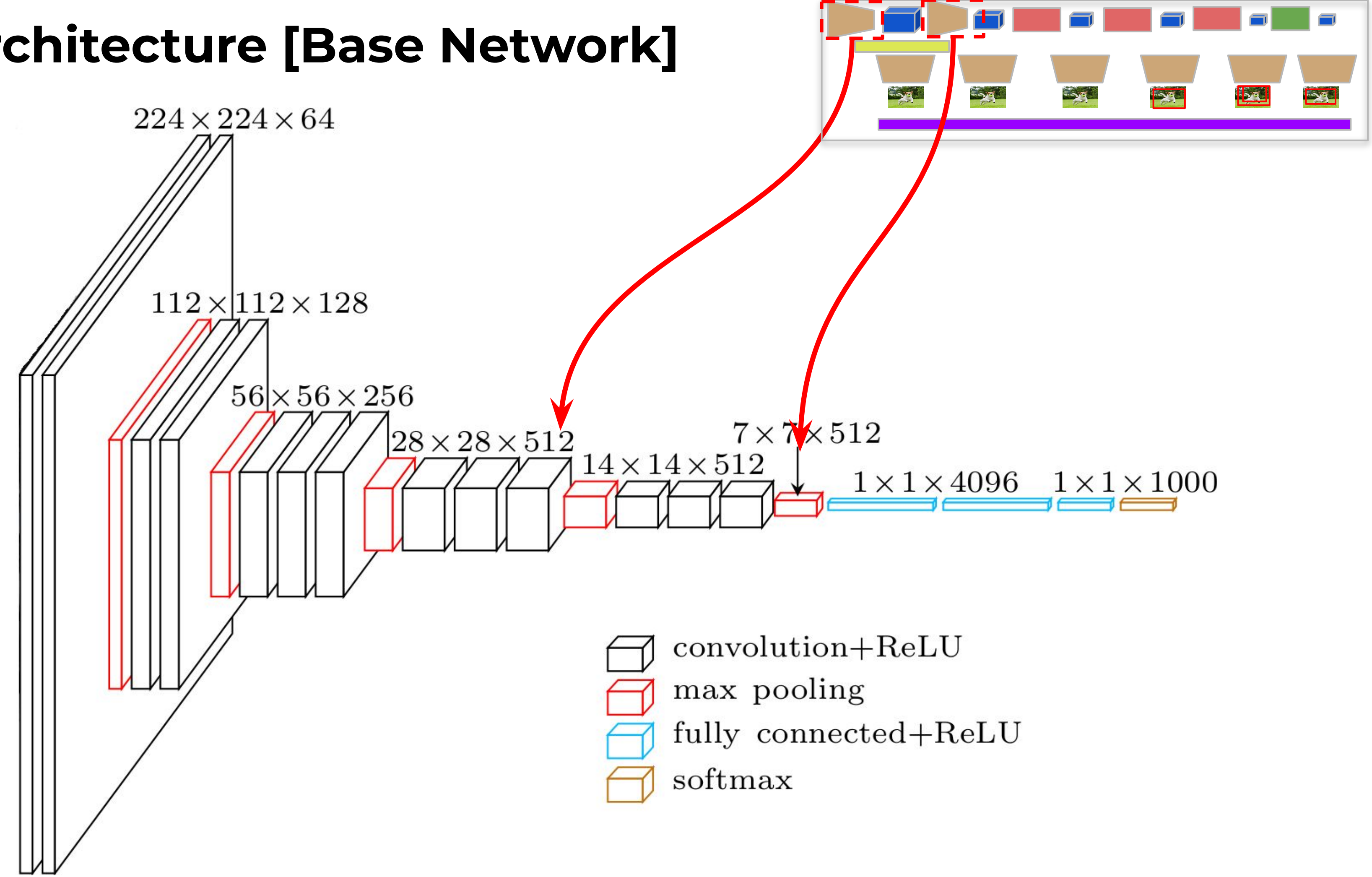


2304.00501

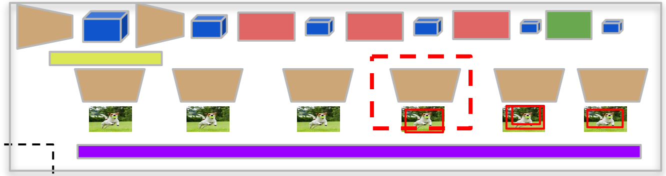
SSD architecture

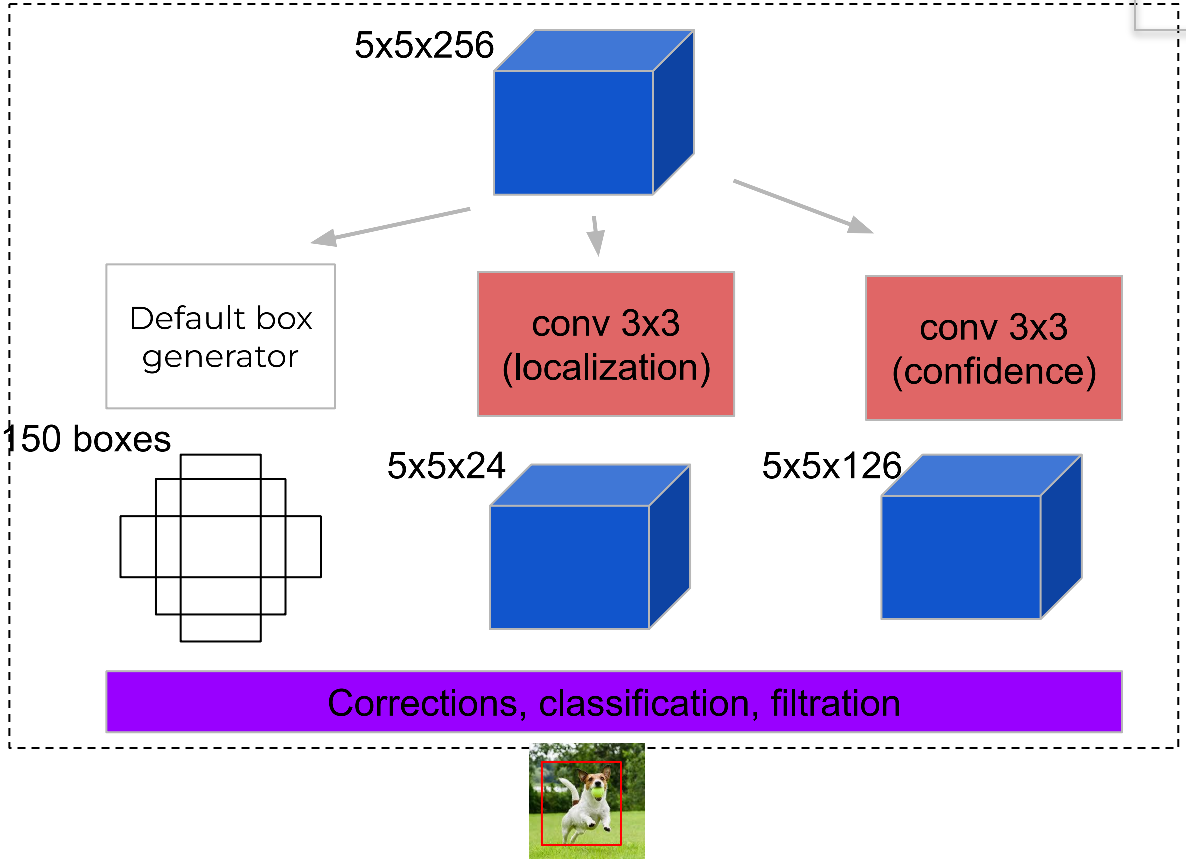


SSD architecture [Base Network]



SSD architecture [Detector & classifier]

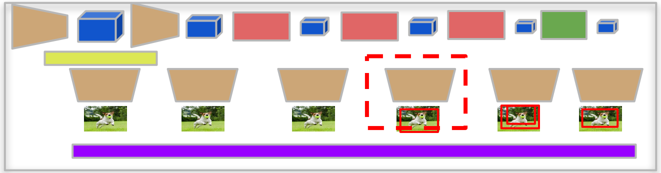


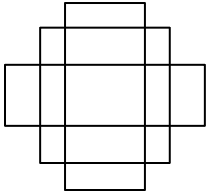


- # default boxes = 6
- 20 + 1 classes
- 4 numbers for each BB

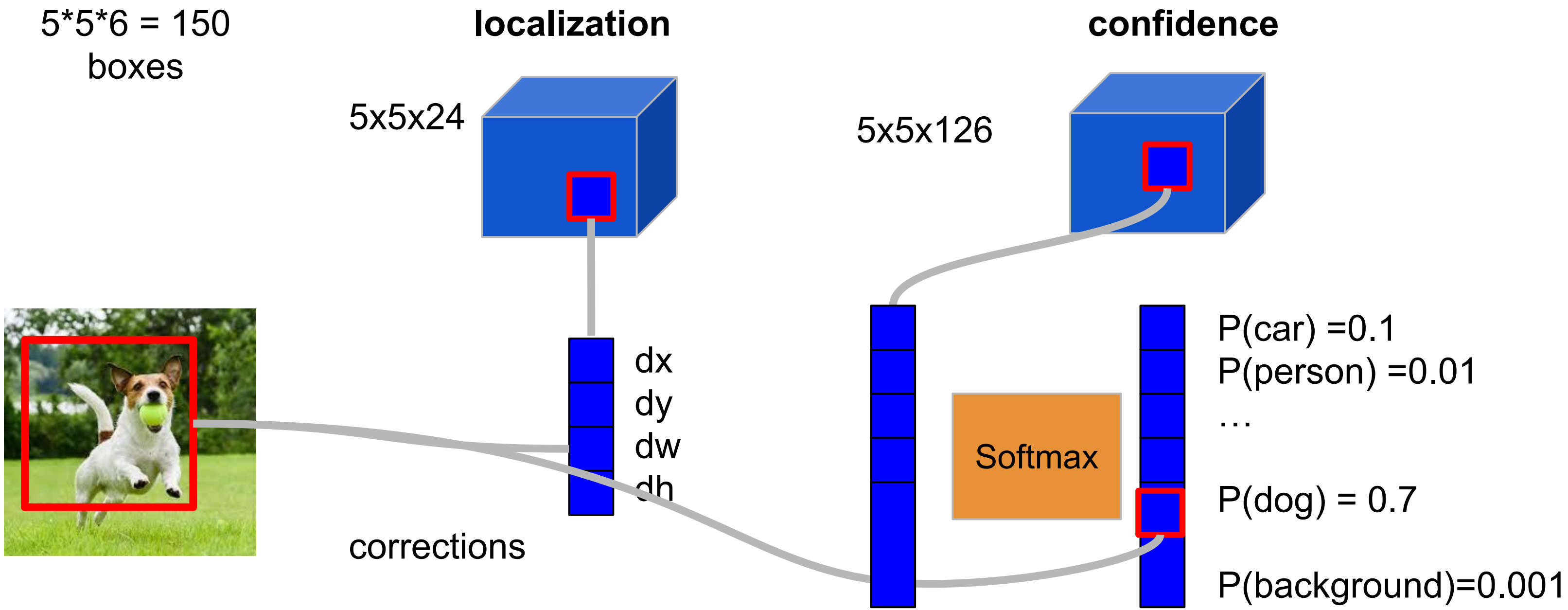
- 126 = 21 x 6
- 24 = 4 x 6
- 150 = 5 x 5 x 6

SSD architecture [Corrections]

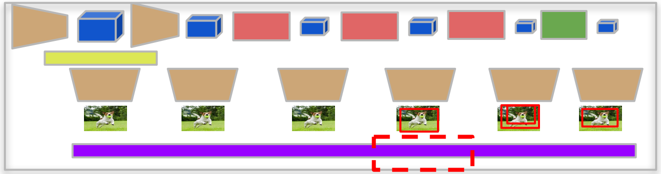


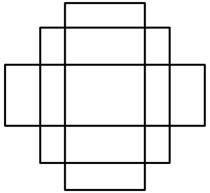


5*5*6 = 150
boxes

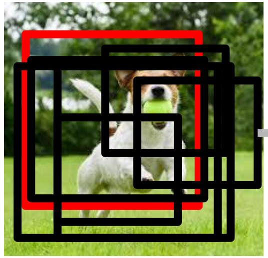


SSD architecture [Corrections]



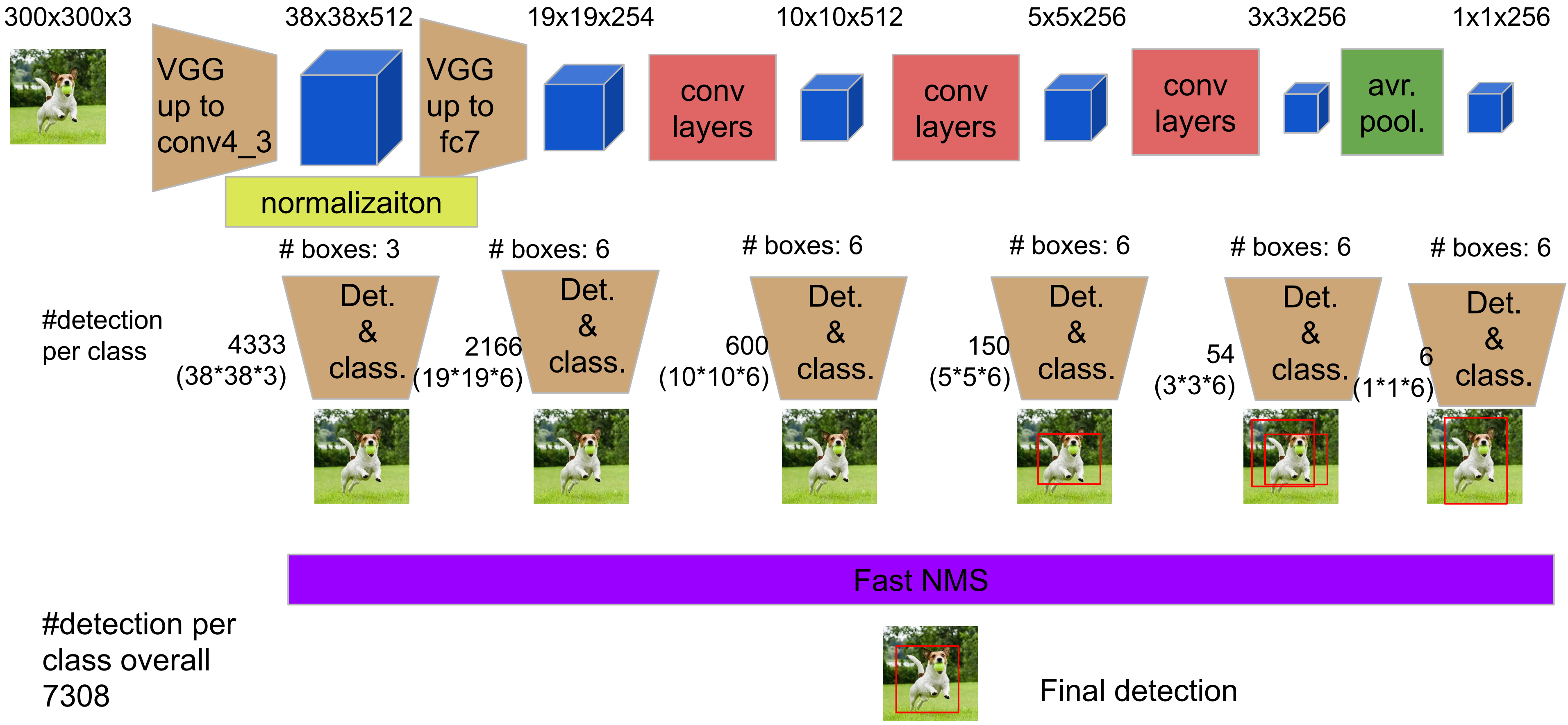


5*5*6 = 150
boxes

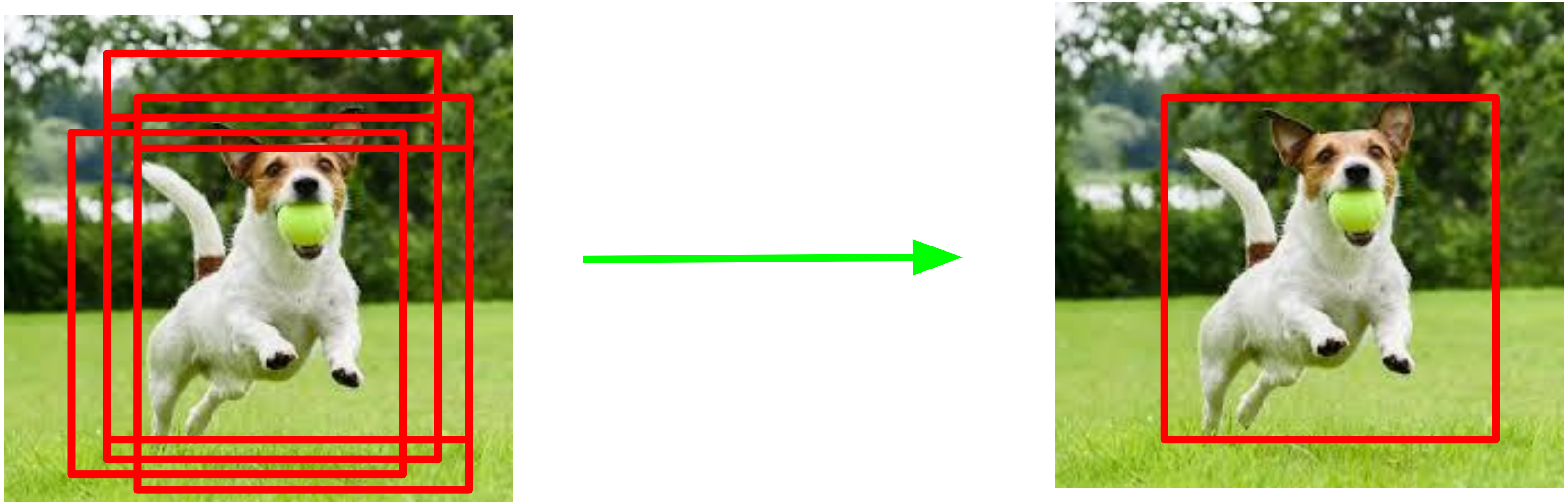


Confidence threshold

SSD architecture



Non-maximum Suppression (NMS)

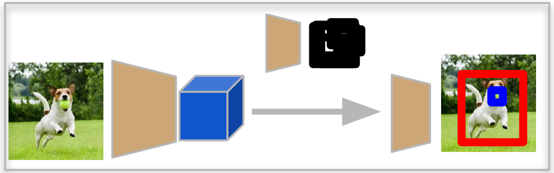


Content

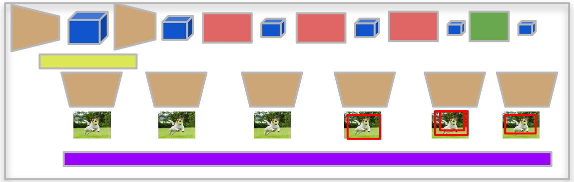
- Intro to Object Detection
  - Datasets
  - Metrics
- Two-stage detectors [RCNN Family]
- One-stage detectors
  - YOLO
  - Single Shot Detector [SSD]
- **Proposal-free detection**
- Summary

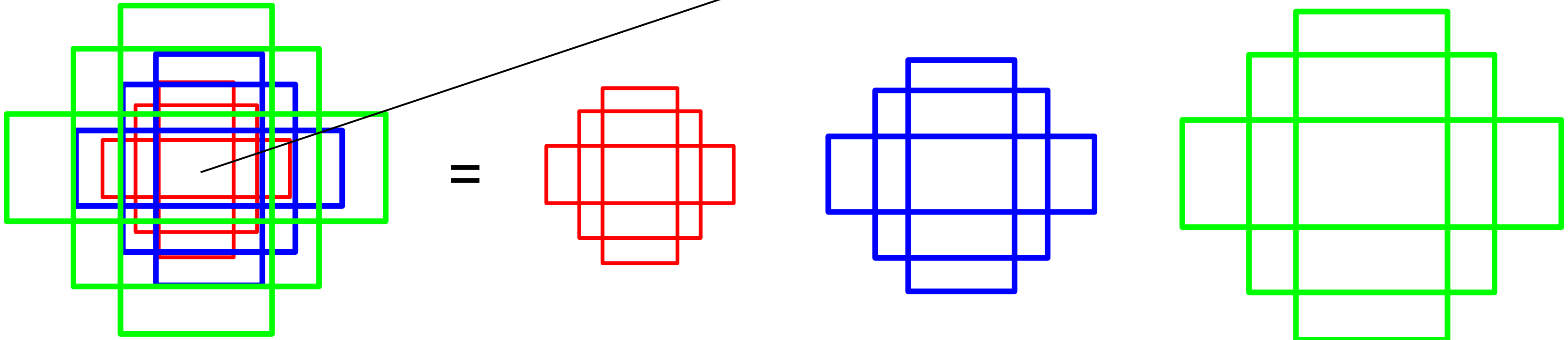
Proposal-based

Faster R-CNN

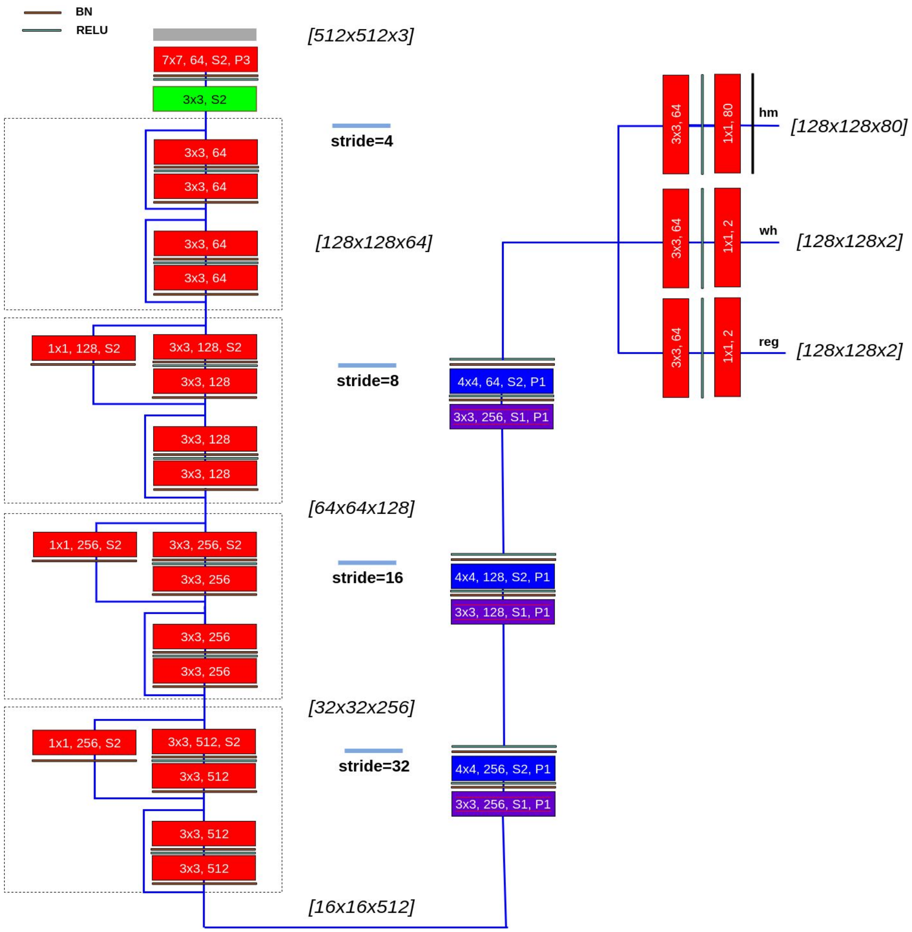


YOLO





CenterNet [architecture]



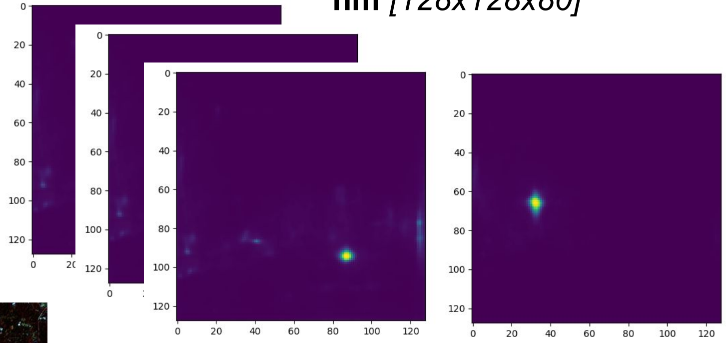
keypoint heatmap [C] local offset [2] object size [2]

$$
\begin{aligned}
(\hat{x}_i + \delta\hat{x}_i - \hat{w}_i/2,\ \ &\hat{y}_i + \delta\hat{y}_i - \hat{h}_i/2,\\
\hat{x}_i + \delta\hat{x}_i + \hat{w}_i/2,\ \ &\hat{y}_i + \delta\hat{y}_i + \hat{h}_i/2)
\end{aligned}
$$

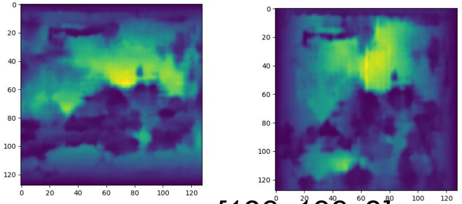
CenterNet (Objects as Points), *arXiv:* [1904.07850]

CenterNet [inference]

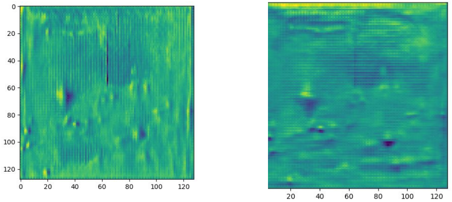
hm *[128x128x80]*

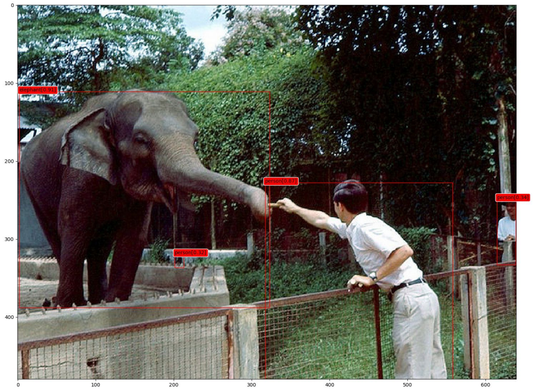


wh *[128x128x2]*



reg *[128x128x2]*





CenterNet [Objects as Points]

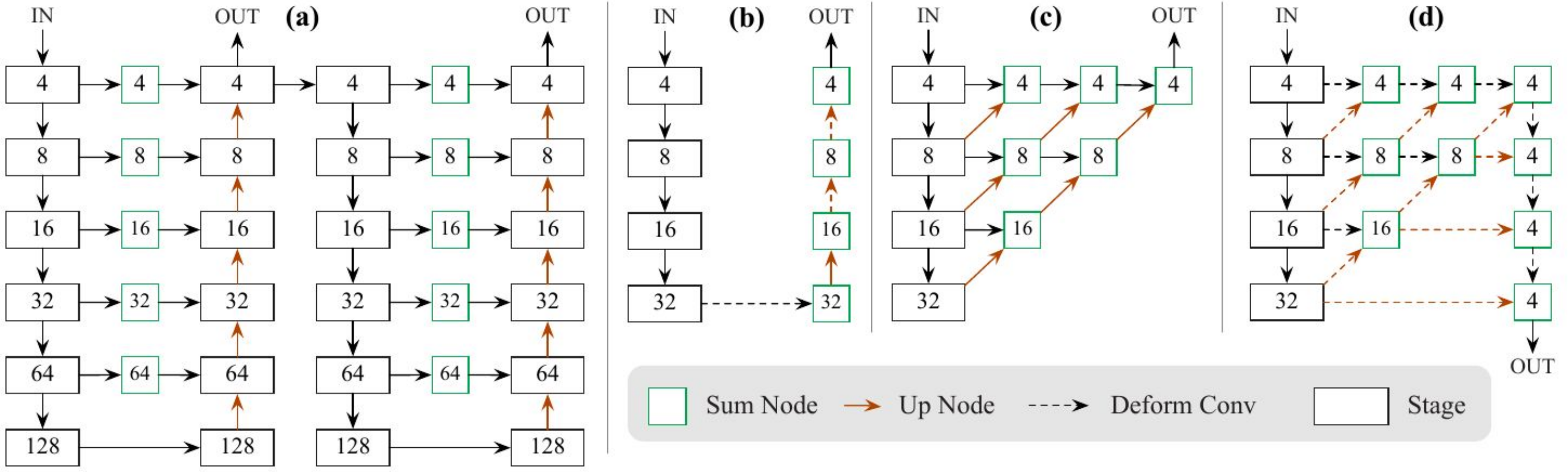
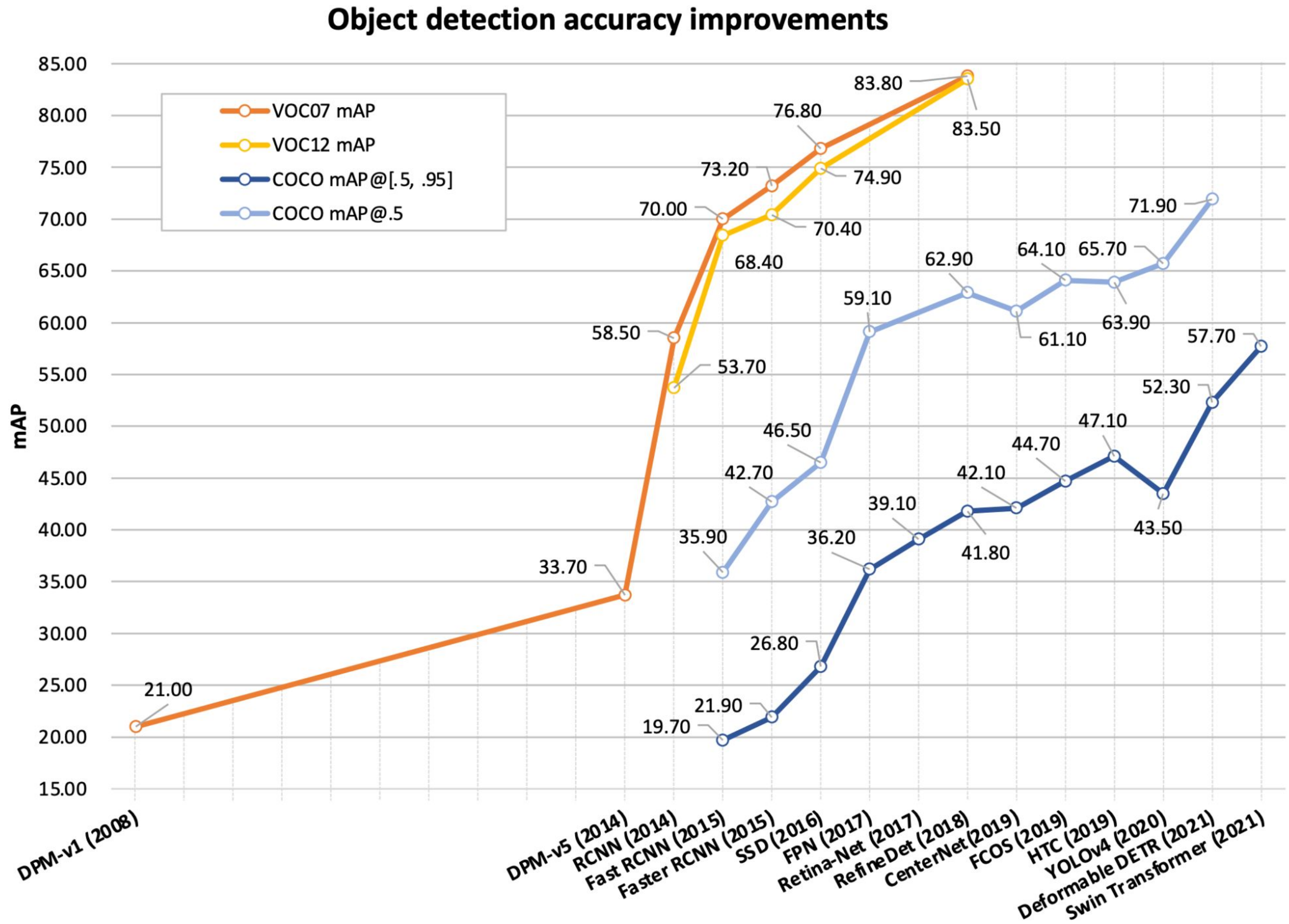


Figure 6: Model diagrams. The numbers in the boxes represent the stride to the image. (a): Hourglass Network [30]. We use it as is in CornerNet [30]. (b): ResNet with transpose convolutions [55]. We add one 3×3 deformable convolutional layer [63] before each up-sampling layer. Specifically, we first use deformable convolution to change the channels and then use transposed convolution to upsample the feature map (such two steps are shown separately in $32 \rightarrow 16$. We show these two steps together as a dashed arrow for $16 \rightarrow 8$ and $8 \rightarrow 4$). (c): The original DLA-34 [58] for semantic segmentation. (d): Our modified DLA-34. We add more skip connections from the bottom layers and upgrade every convolutional layer in upsampling stages to deformable convolutional layer.

CenterNet (Objects as Points), *arXiv:* 1904.07850

Content

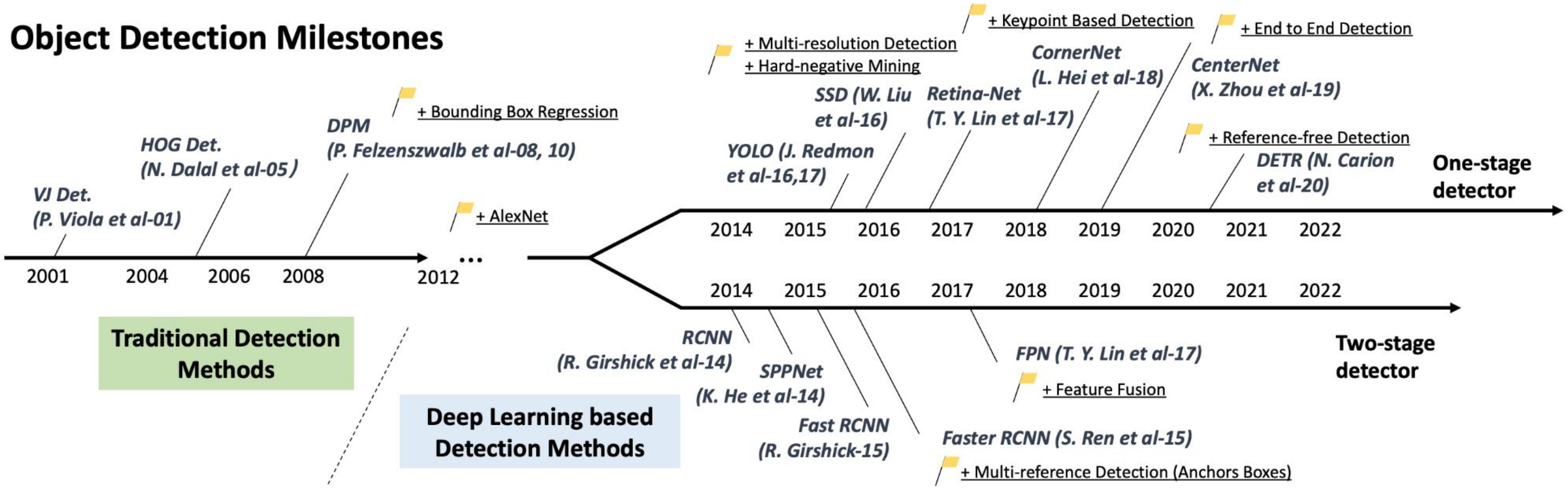
- Intro to Object Detection
  - Datasets
  - Metrics
- Two-stage detectors [RCNN Family]
- One-stage detectors
  - YOLO
  - Single Shot Detector [SSD]
- Proposal-free detection
- **Summary**



Object detection accuracy improvements
mAP
VOC07 mAP
VOC12 mAP
COCO mAP@[.5, .95]
COCO mAP@.5
85.00
80.00
75.00
70.00
65.00
60.00
55.00
50.00
45.00
40.00
35.00
30.00
25.00
20.00
15.00
21.00
33.70
58.50
70.00
73.20
76.80
83.80
53.70
68.40
70.40
74.90
83.50
35.90
42.70
46.50
59.10
62.90
61.10
64.10
63.90
65.70
71.90
19.70
21.90
26.80
36.20
39.10
41.80
42.10
44.70
47.10
43.50
52.30
57.70
DPM-v1 (2008)
DPM-v5 (2014)
RCNN (2014)
Fast RCNN (2015)
Faster RCNN (2015)
SSD (2016)
FPN (2017)
Retina-Net (2017)
RefineDet (2018)
CenterNet(2019)
FCOS (2019)
HTC (2019)
YOLOv4 (2020)
Deformable DETR (2021)
Swin Transformer (2021)

1905.05055

Object Detection Milestones



1905.05055