Computer Vision

第十周　目标检测

庞彦

yanpang@gzhu.edu.cn

Spring 2023

01

Object Localization and Detection

目标定位与检测

Object Recognition

目标识别：　　分类问题

Baby

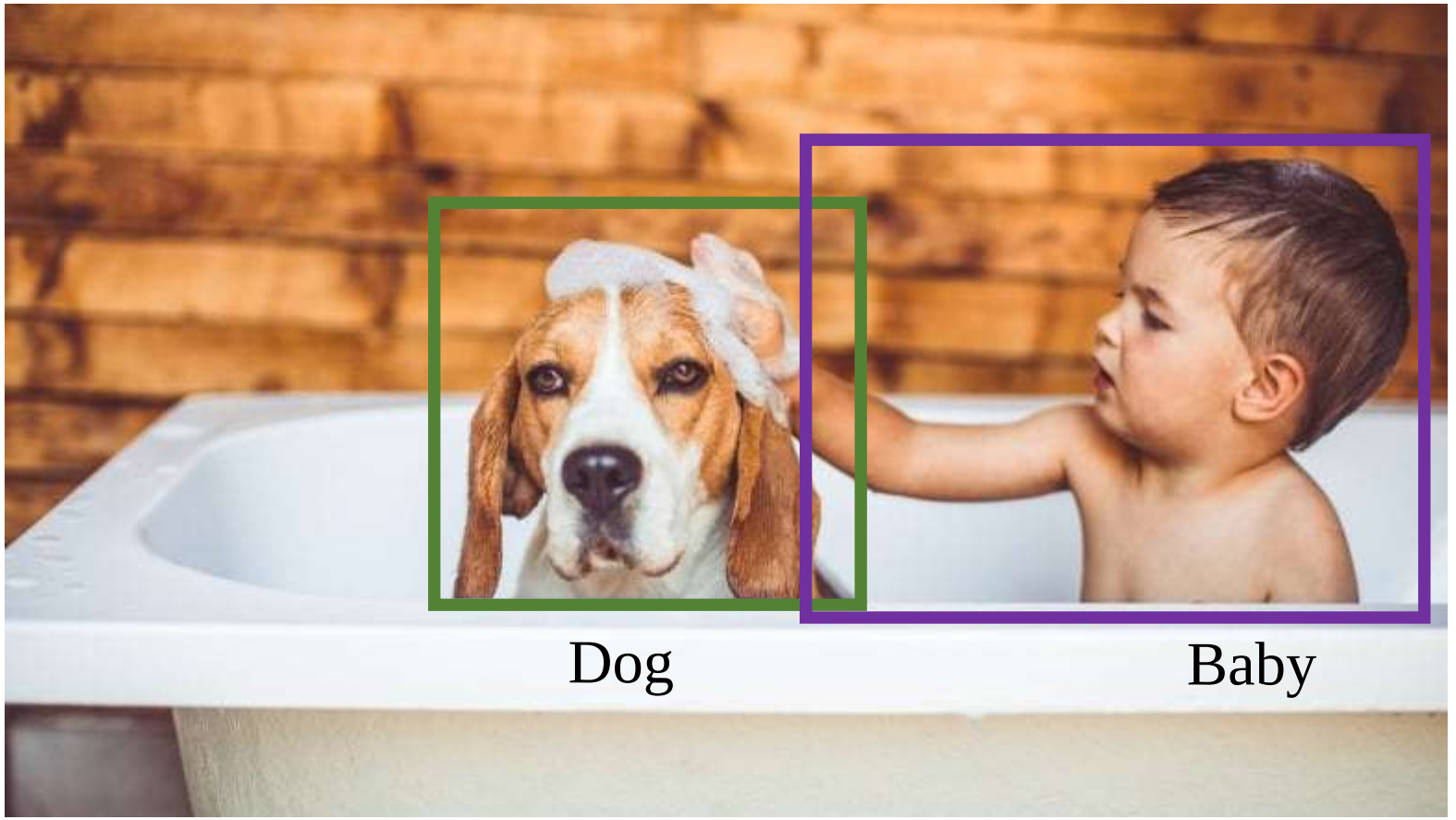
Dog

Lynx

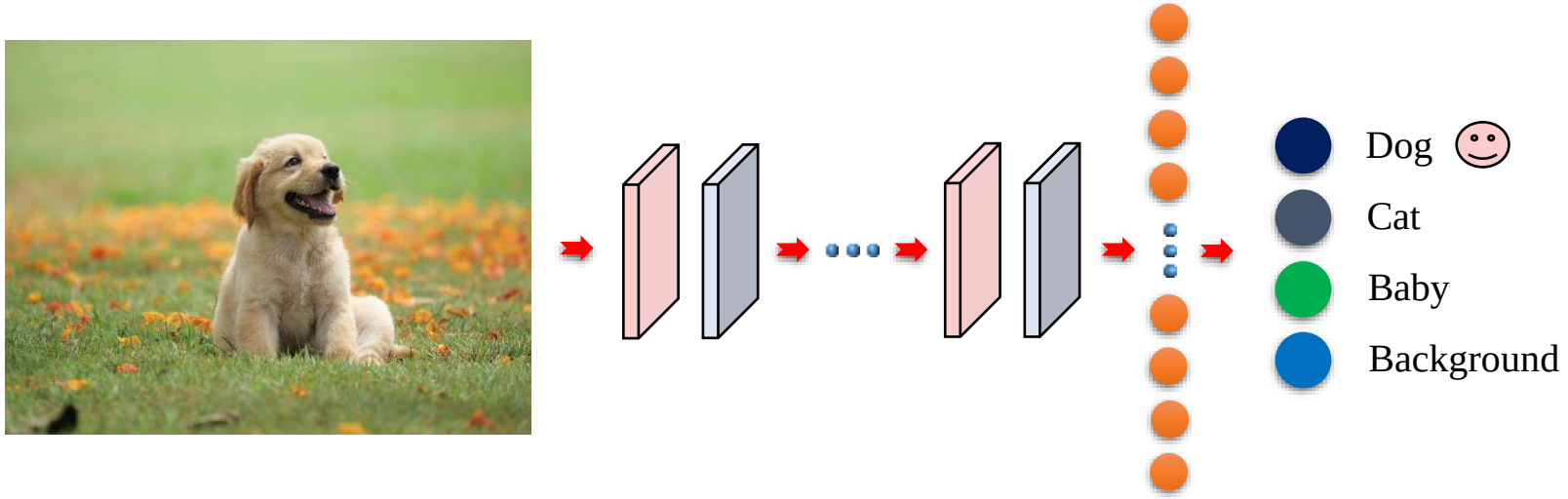
Object Detection

目标识别：　　分类问题

目标检测：　　分类问题
+
定位问题

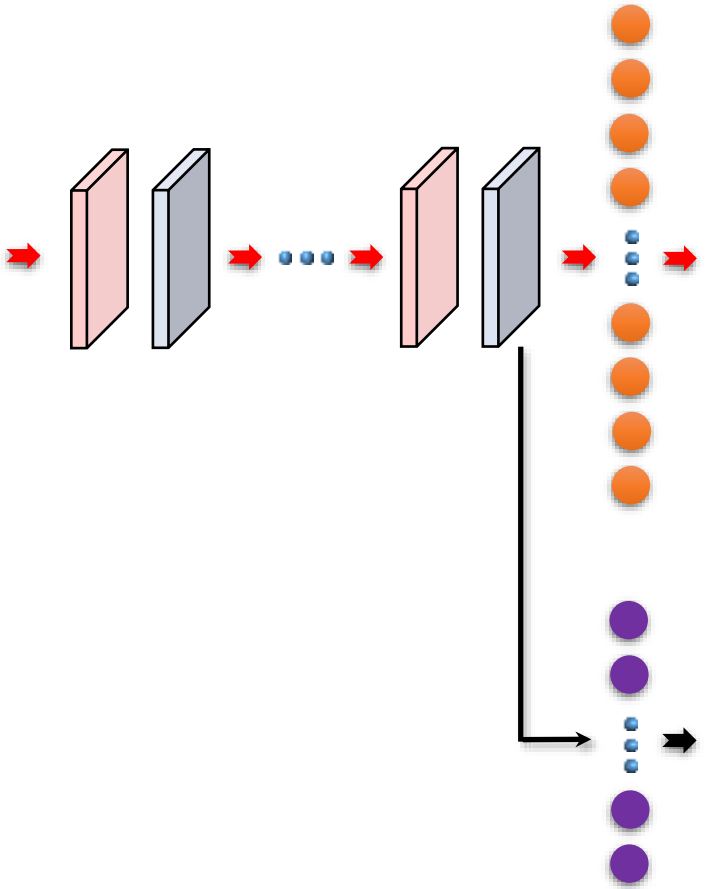


Object Localization



Object Localization

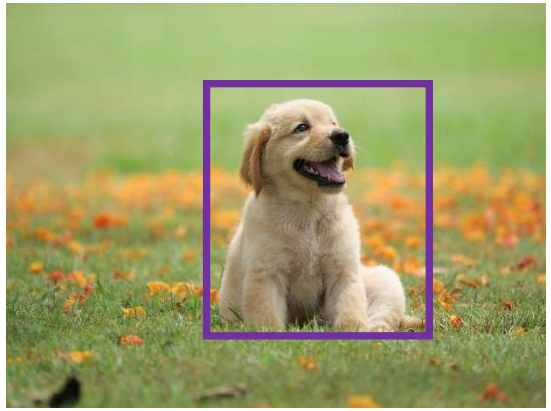
(b_x, b_y, b_w, b_h)
Bounding Box



$$y = \begin{bmatrix} P_c \\ b_x \\ b_y \\ b_w \\ b_h \\ C_1 \\ C_2 \\ C_3 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_w \\ b_h \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad y = \begin{bmatrix} 0 \\ X \\ X \\ X \\ X \\ X \\ X \\ X \end{bmatrix}$$

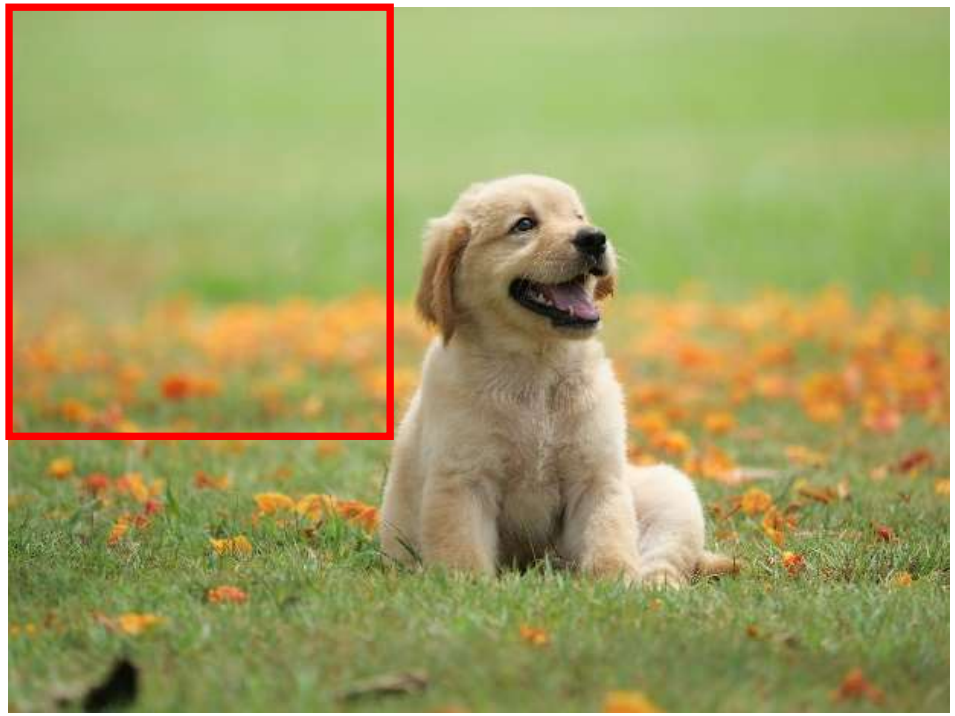
Object Localization

$x =$ 

$$y = \begin{bmatrix} P_c \\ b_x \\ b_y \\ b_w \\ b_h \\ C_1 \\ C_2 \\ C_3 \end{bmatrix}$$

$$min\,\mathcal{L}(\hat{y} - y) = \begin{cases} (\hat{P}_c - P_C)^2 + (\hat{b}_x - b_x)^2 + \cdots + (\hat{C}_3 - C_3)^2, & if\ \ P_C = 1; \\ (\hat{P}_c - P_C)^2, & if\ \ P_C = 0. \end{cases}$$

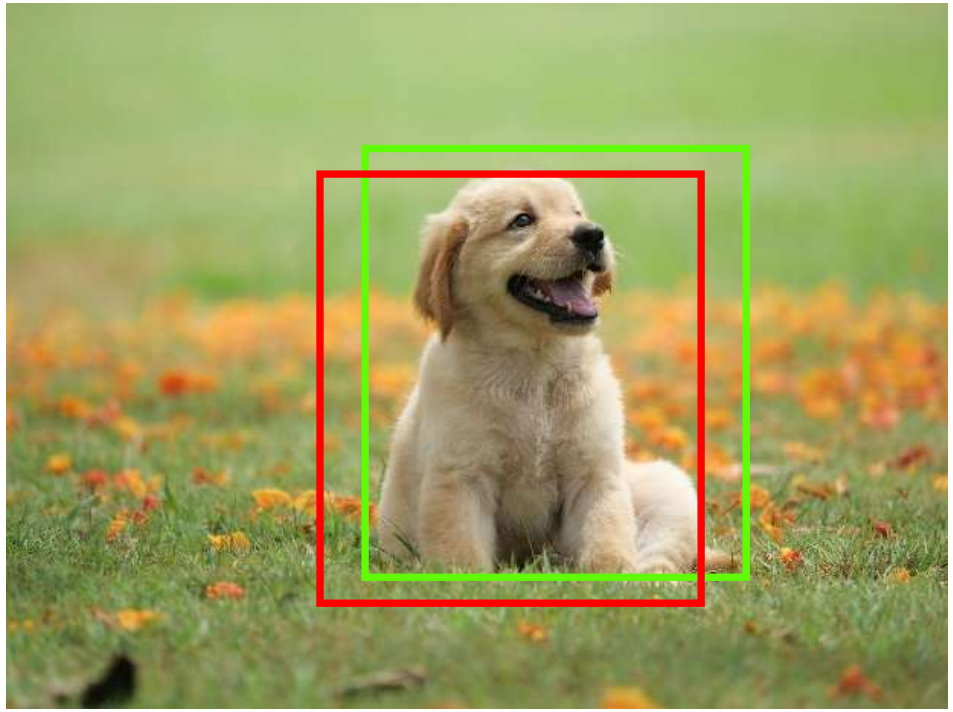
Object Detection



Find the window with the largest IOU

IOU: intersection-over-union

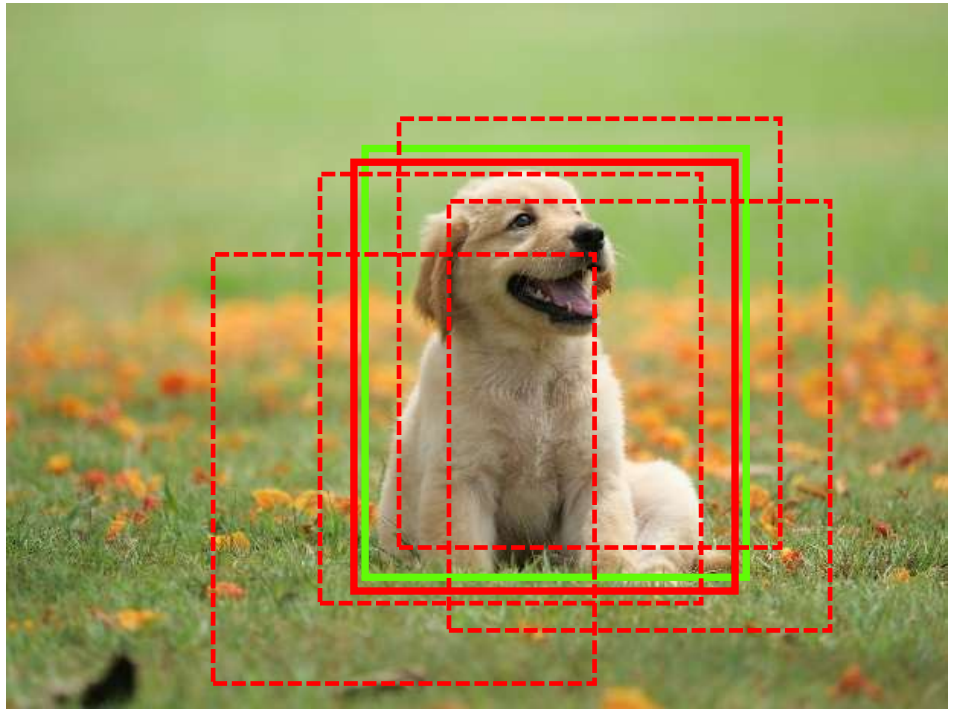
IOU



IOU: intersection-over-union

$$IoU = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$

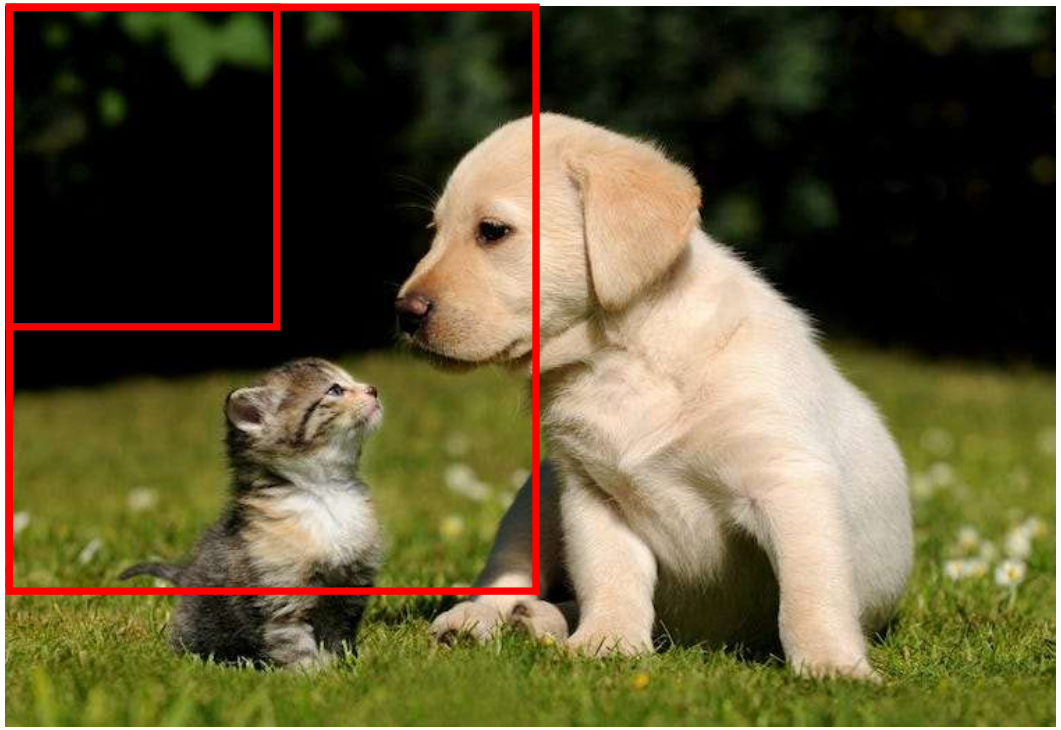
IOU



IOU: intersection-over-union

Find the window with the largest IOU

Object Detection

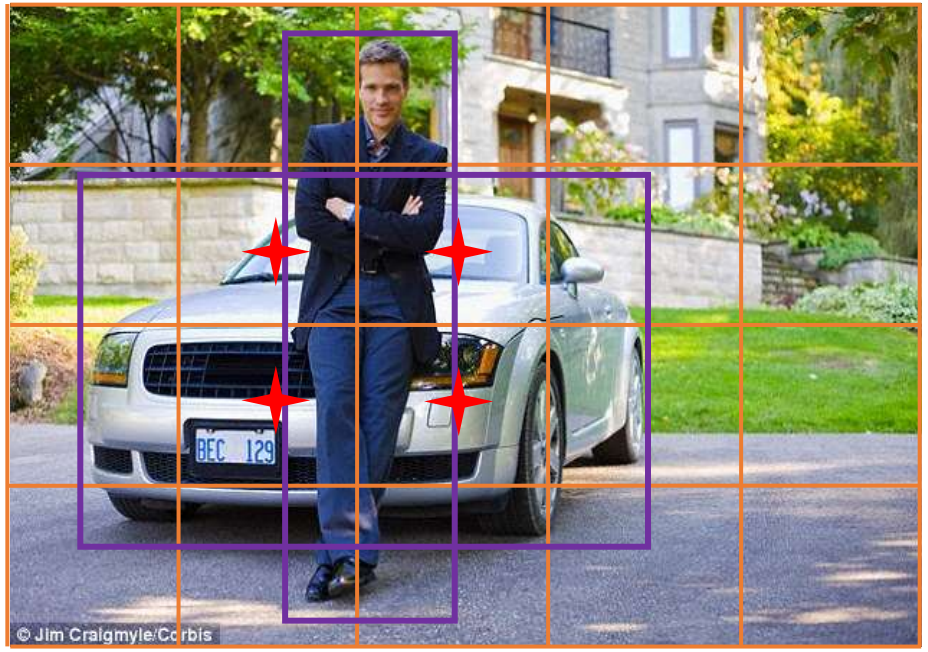


Two Classes CNN

Dog and Kitten Detector

Object Detection

Anchor



4 ×6 Grid Cells

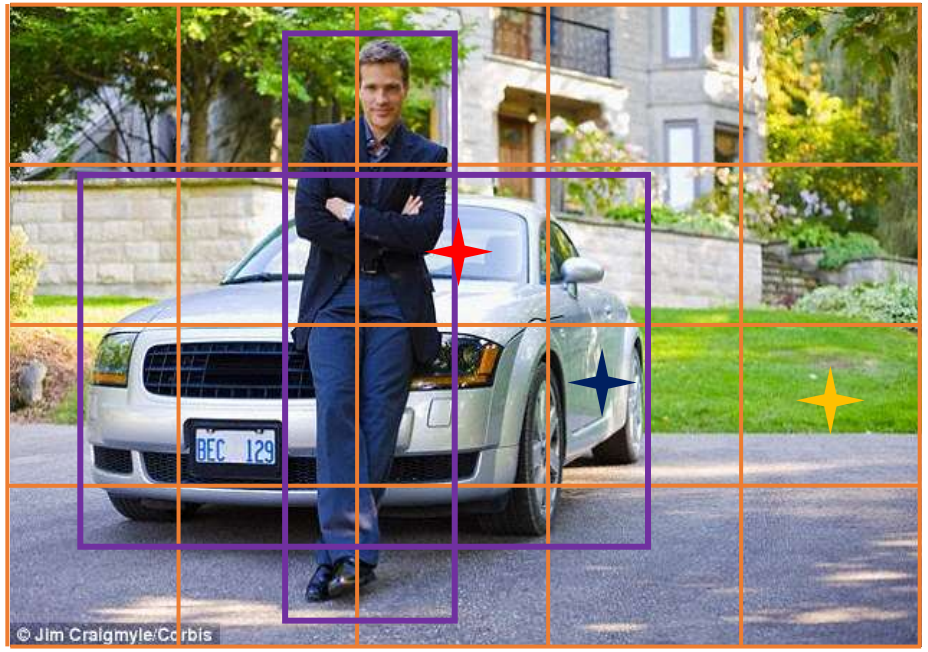
Anchor Box 1

Anchor Box 2

$$y = \begin{bmatrix} P_c \\ b_x \\ b_y \\ b_w \\ b_h \\ C_1 \\ C_2 \\ C_3 \end{bmatrix}$$

$$y = \begin{bmatrix} P_c \\ b_x \\ b_y \\ b_w \\ b_h \\ C_1 \\ C_2 \\ C_3 \\ P_c \\ b_x \\ b_y \\ b_w \\ b_h \\ C_1 \\ C_2 \\ C_3 \end{bmatrix}$$

Anchor



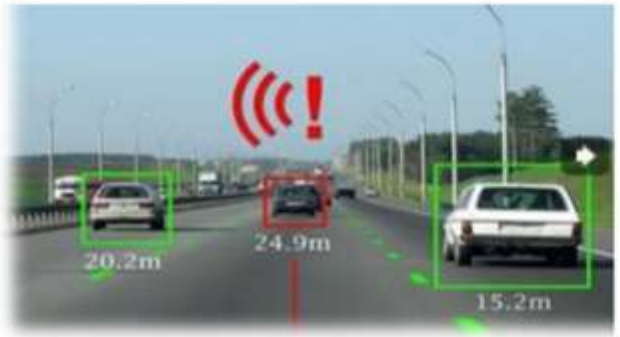
4 ×6 Grid Cells

$$y = \begin{bmatrix} P_c \\ b_x \\ b_y \\ b_w \\ b_h \\ C_1 \\ C_2 \\ C_3 \\ P_c \\ b_x \\ b_y \\ b_w \\ b_h \\ C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_w \\ b_h \\ 1 \\ 0 \\ 0 \\ 1 \\ b_x \\ b_y \\ b_w \\ b_h \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad y = \begin{bmatrix} P_c \\ b_x \\ b_y \\ b_w \\ b_h \\ C_1 \\ C_2 \\ C_3 \\ P_c \\ b_x \\ b_y \\ b_w \\ b_h \\ C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} 0 \\ X \\ X \\ X \\ X \\ X \\ X \\ 1 \\ b_x \\ b_y \\ b_w \\ b_h \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad y = \begin{bmatrix} P_c \\ b_x \\ b_y \\ b_w \\ b_h \\ C_1 \\ C_2 \\ C_3 \\ P_c \\ b_x \\ b_y \\ b_w \\ b_h \\ C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} 0 \\ X \\ X \\ X \\ X \\ X \\ X \\ 0 \\ X \\ X \\ X \\ X \\ X \\ X \\ X \end{bmatrix}$$

Instance

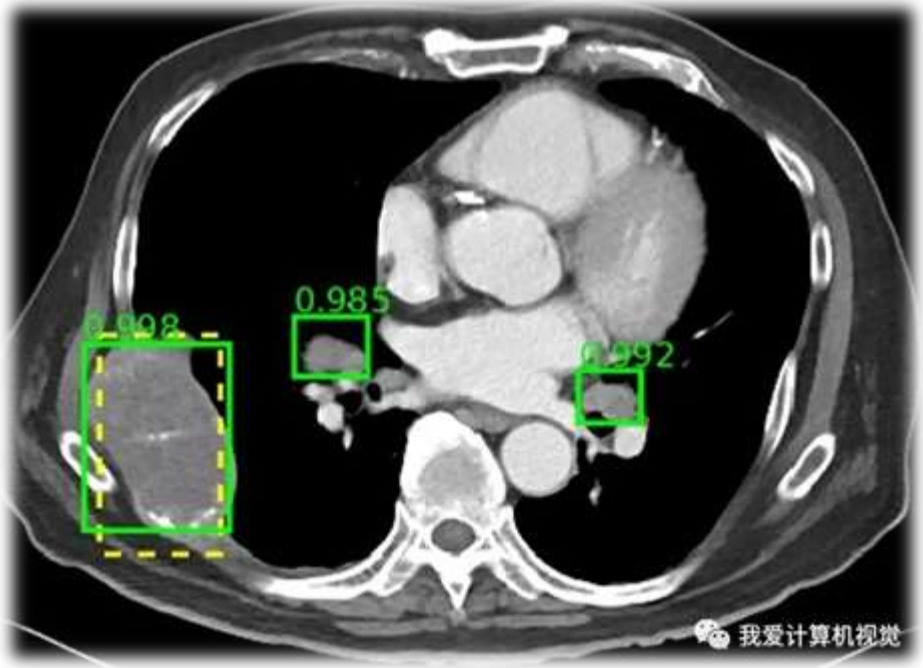
Yolo V4

Applications

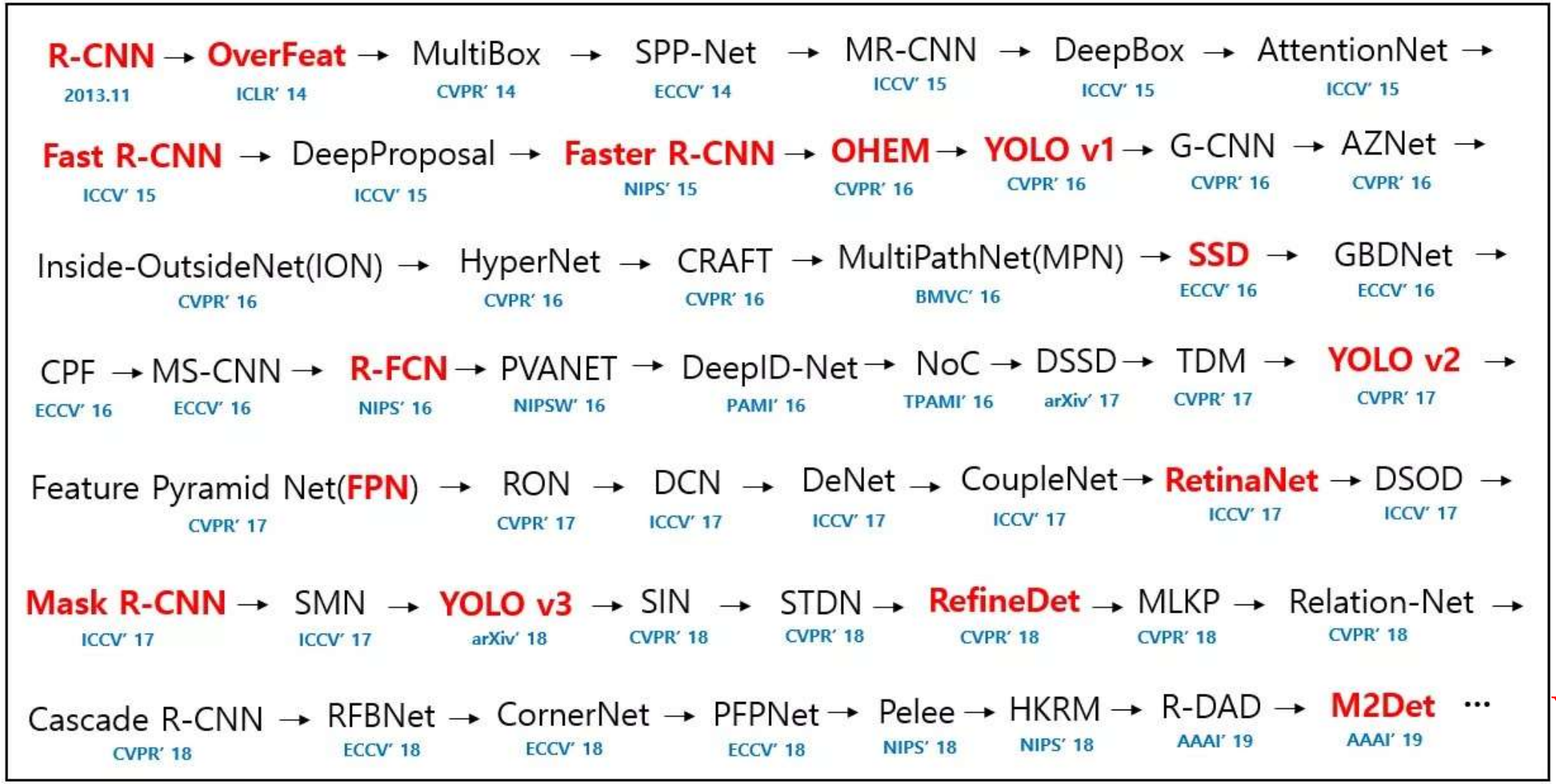


Applications

Applications



Evolution



YOLO v4

2020

Object Detection

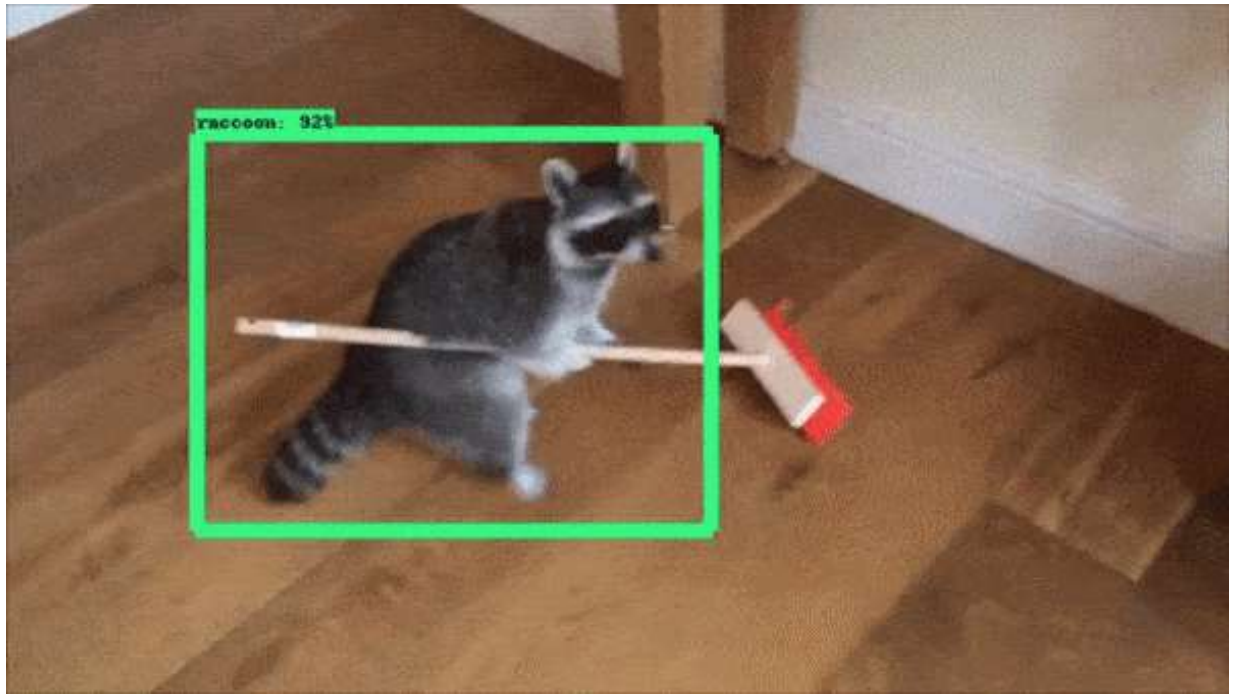
目标检测：　　分类问题　　+　　定位问题

算法流派：

二刀斩（Two Stages）

一刀流（One Stage ）

空手道（Anchor Free）



Object Detection

目标检测：　分类问题　+　定位问题

算法流派：

二刀斩（Two Stages）：

R-CNN, Fast RCNN, Faster RCNN, Mask RCNN...

一刀流（One Stage ）：

SSD, Yolo v4

空手道（Anchor Free）：

CornerNet-Lite, CenterNet

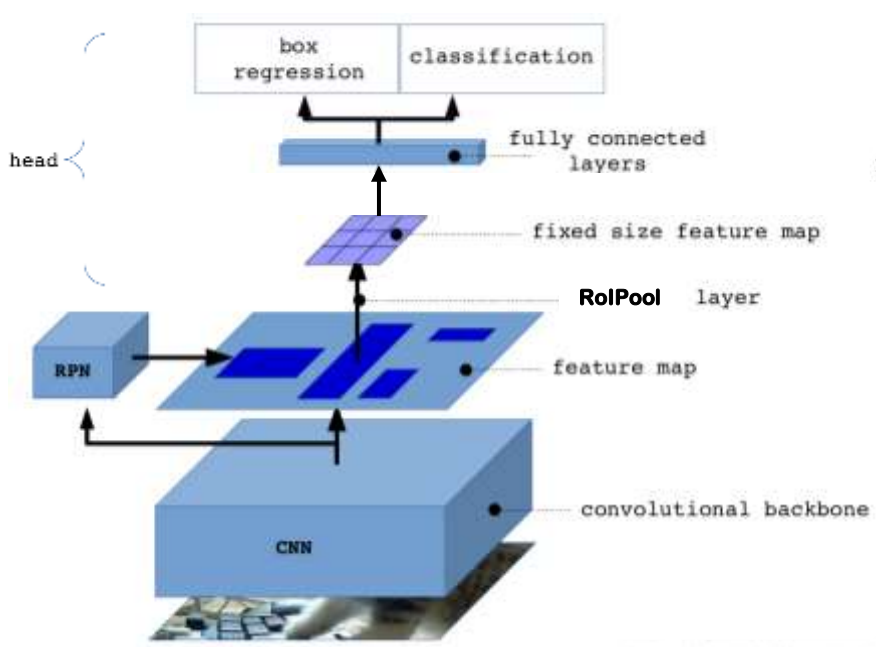
02

Object Detection: Two Stages

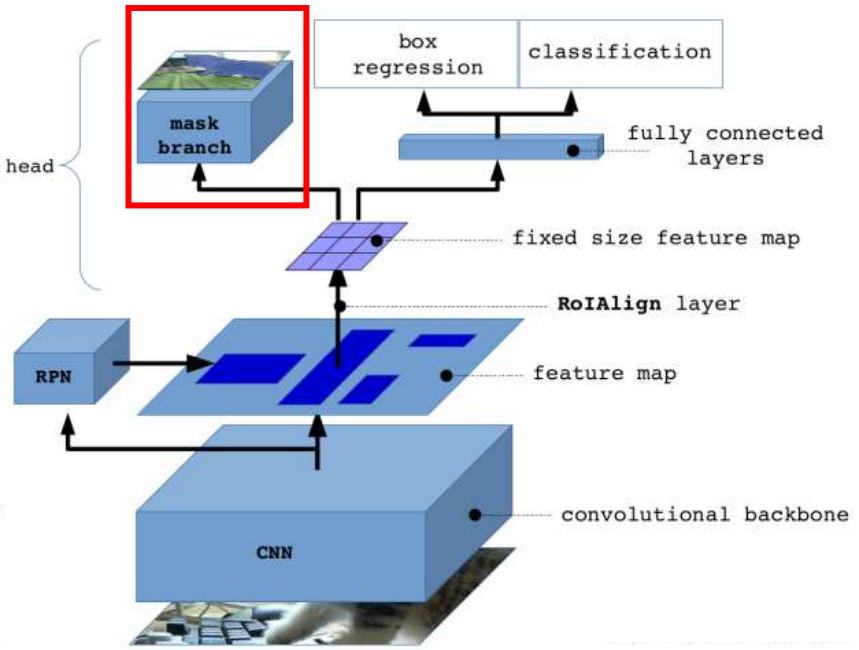
二刀斩： Mask RCNN, Mask Scoring RCNN

Mask RCNN

找人 → 实例分割 → 关键点检测

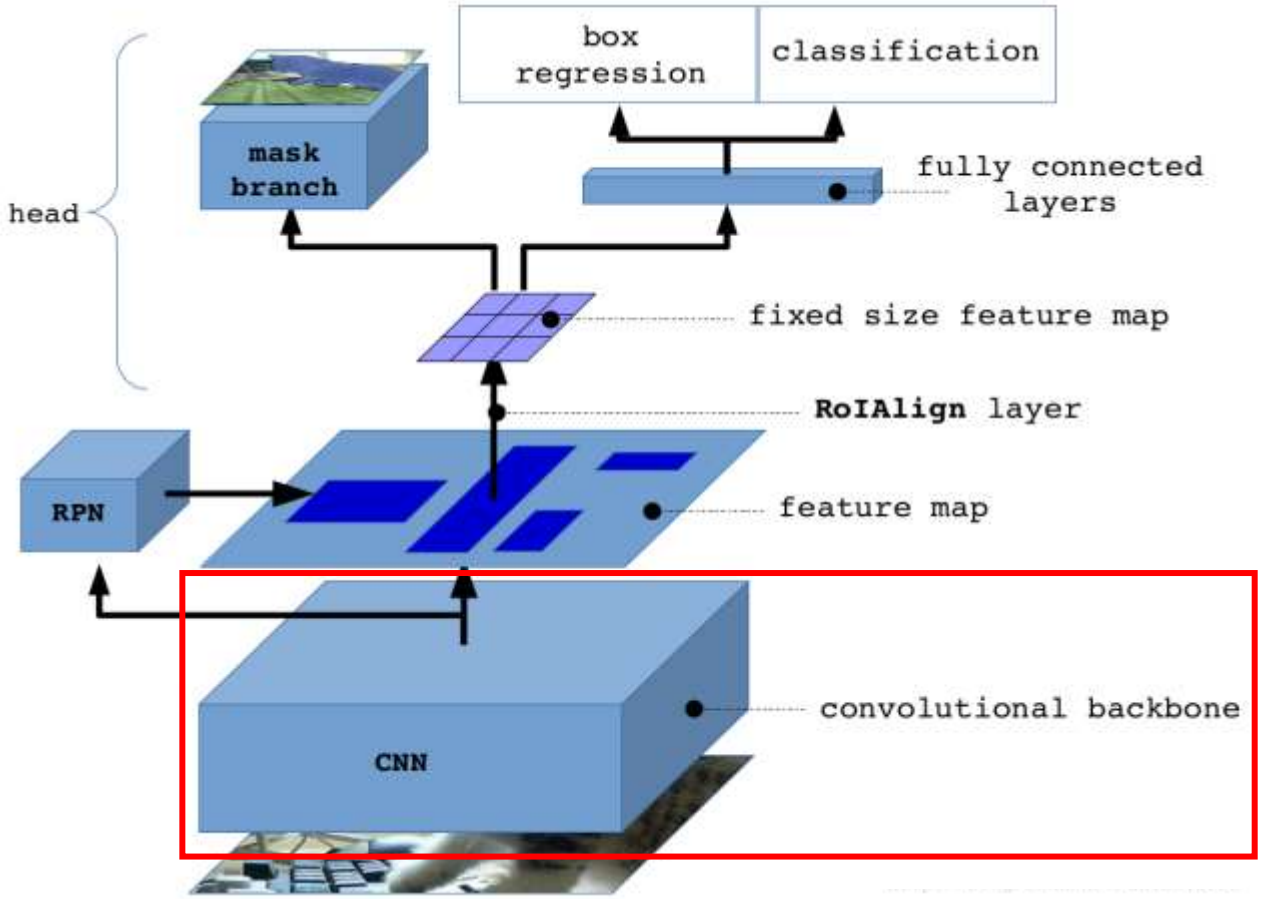


$$Loss = Loss_{regression} + Loss_{classification}$$

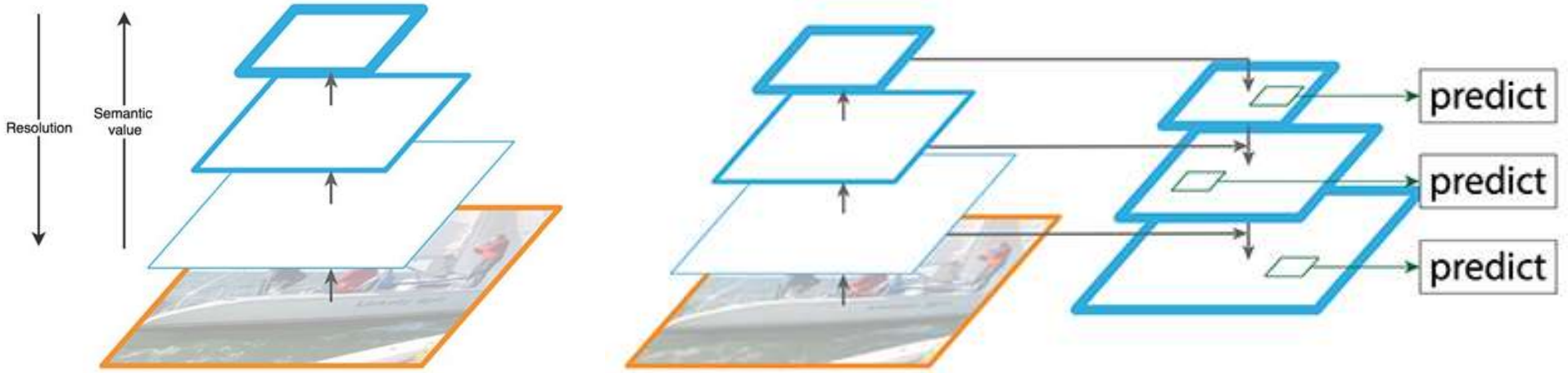


$$Loss = Loss_{regression} + Loss_{classification} + Loss_{mask}$$

Mask RCNN

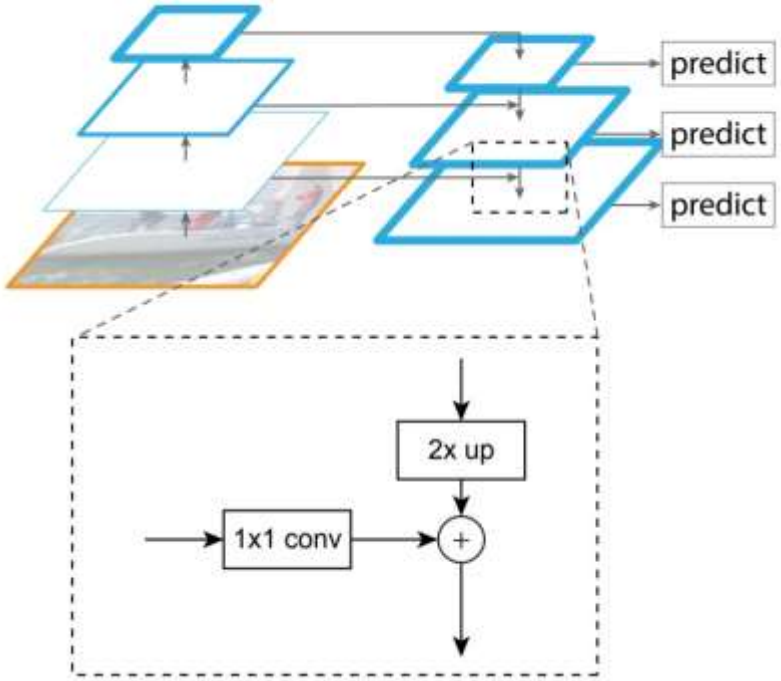

box
regression
classification
mask
branch
head
fully connected
layers
fixed size feature map
RoIAlign layer
RPN
feature map
CNN
convolutional backbone

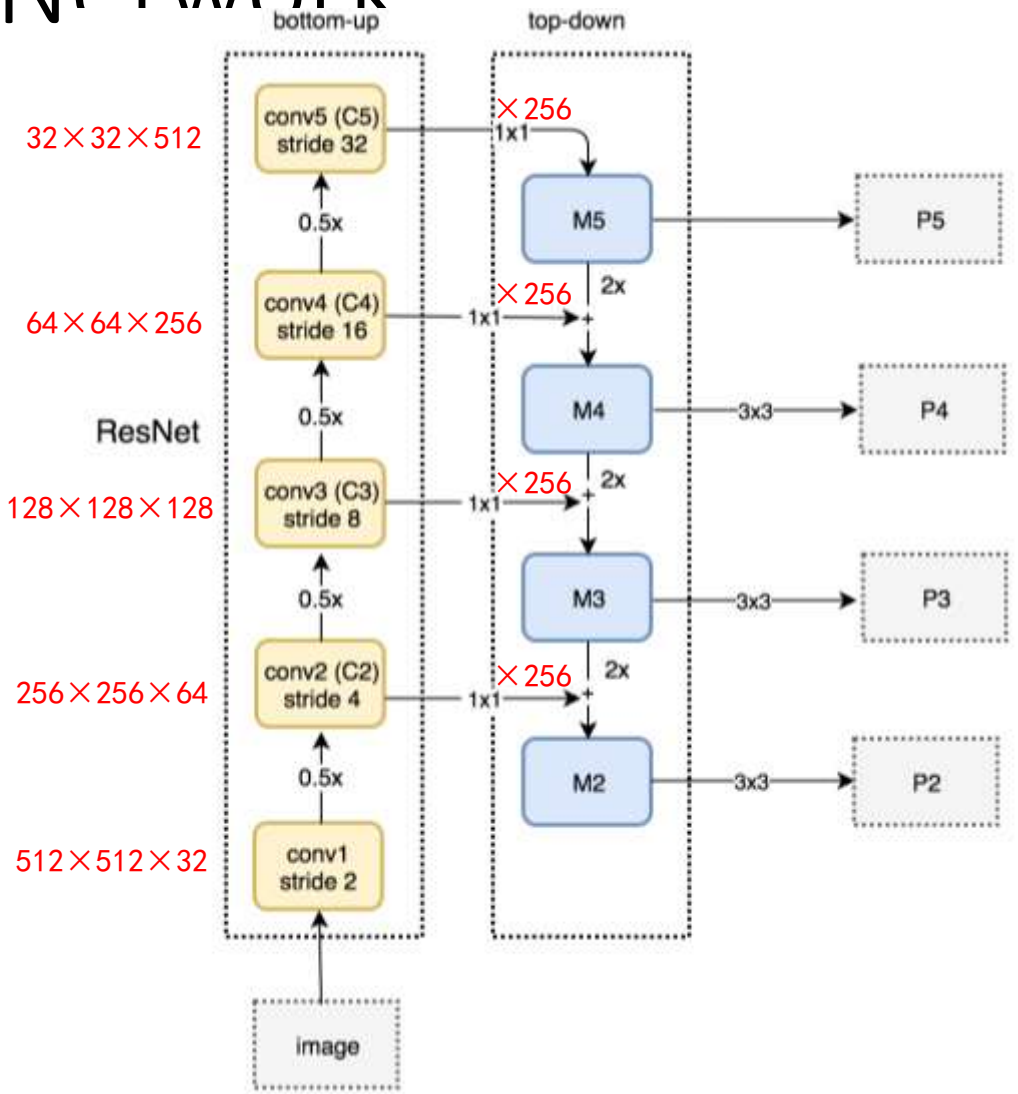
FPN： Feature Pyramid Network



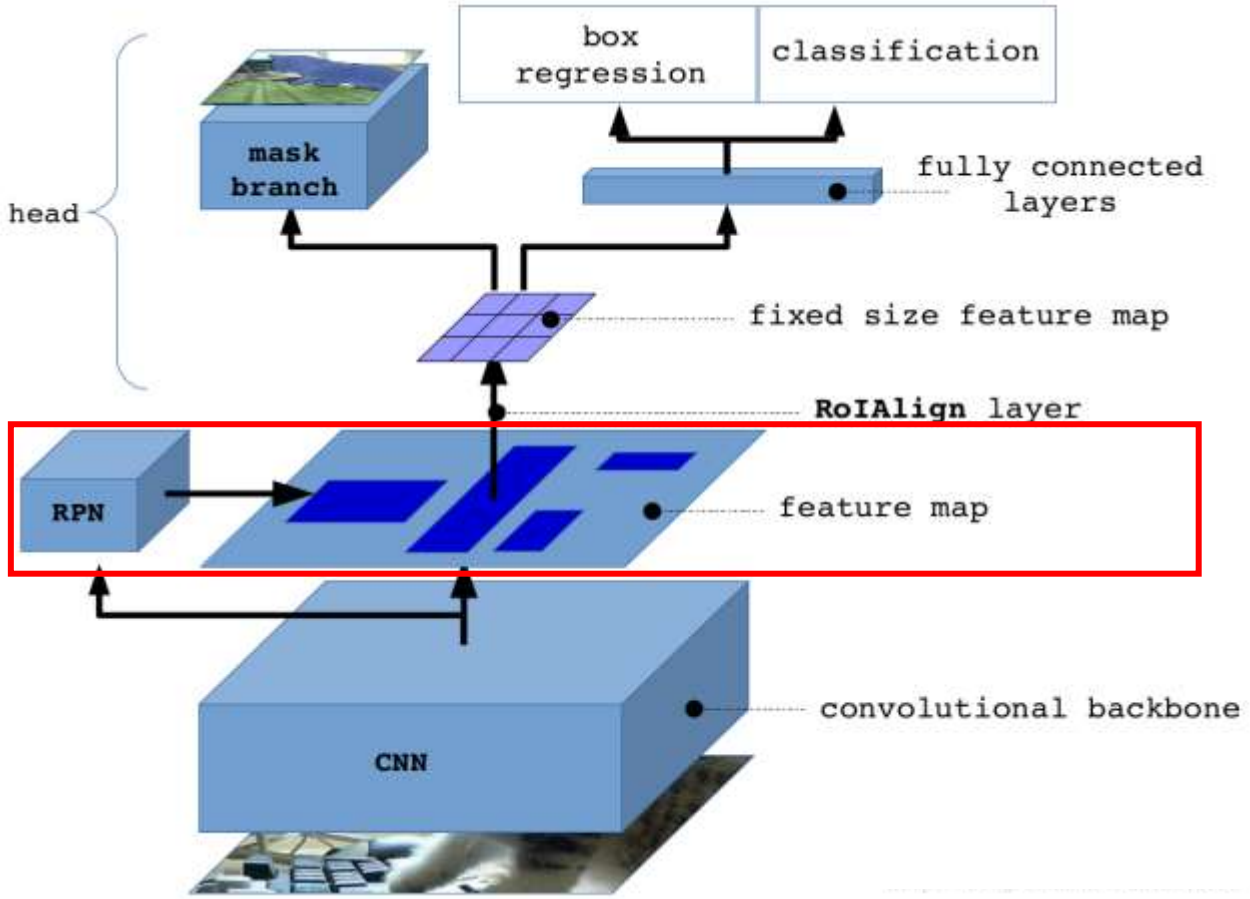
FPN：Feature Pyramid Network

C1-C5的特征图尺寸是不同的；
使用1×1卷积确保depth一样先得到P5，
然后上采样确保特征矩阵大小匹配，
再进行特征矩阵相加，如法炮制得到P4, P3, P2。





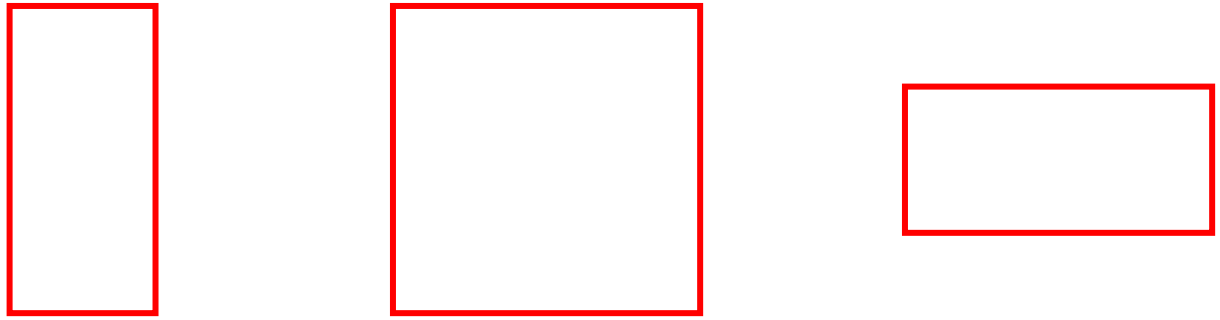
Mask RCNN


box
regression
classification
mask
branch
head
fully connected
layers
fixed size feature map
RoIAlign layer
RPN
feature map
CNN
convolutional backbone

Anchors

Anchors

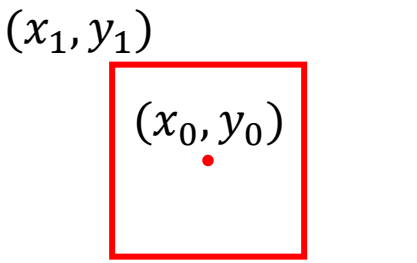
Ratio = Weight/Height = [0.5, 1, 2]



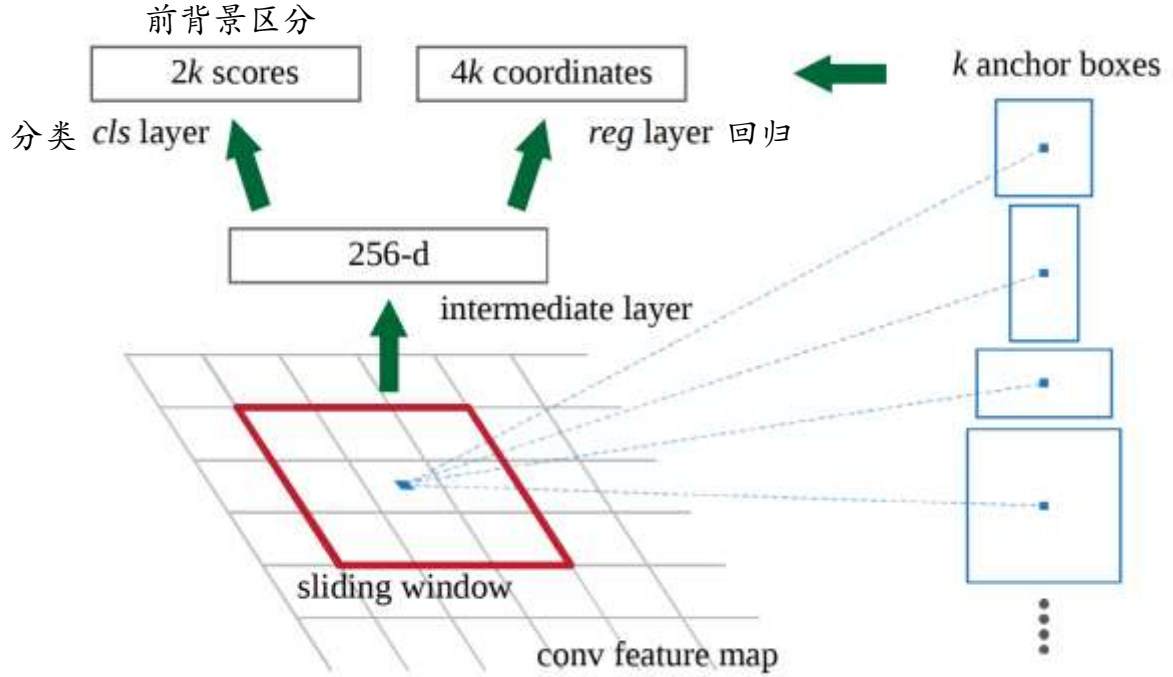
Scales = [32, 64, 256] Pixels

Coordinates $[x_1, y_1, x_2, y_2]$

Coordinates $[x_0, y_0, w, h]$



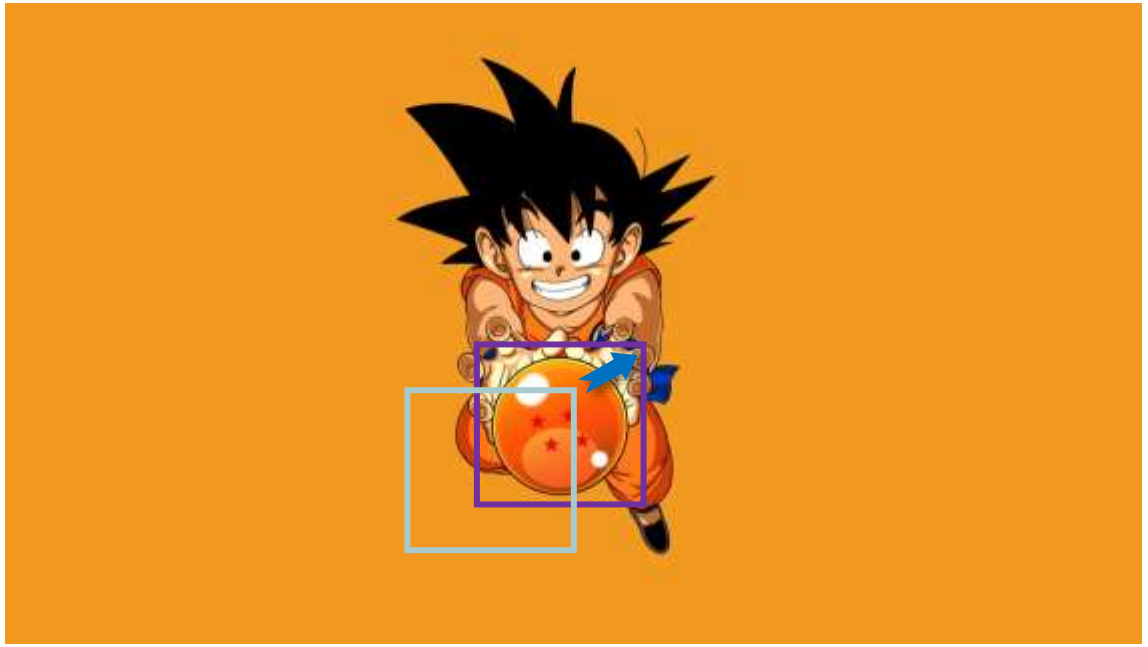
RPN: Region Proposal Network



Proposal Layer

对20W+候选框进行过滤，按照前景对分排序；

取前6000个高分候选框，同时配合其回归值；

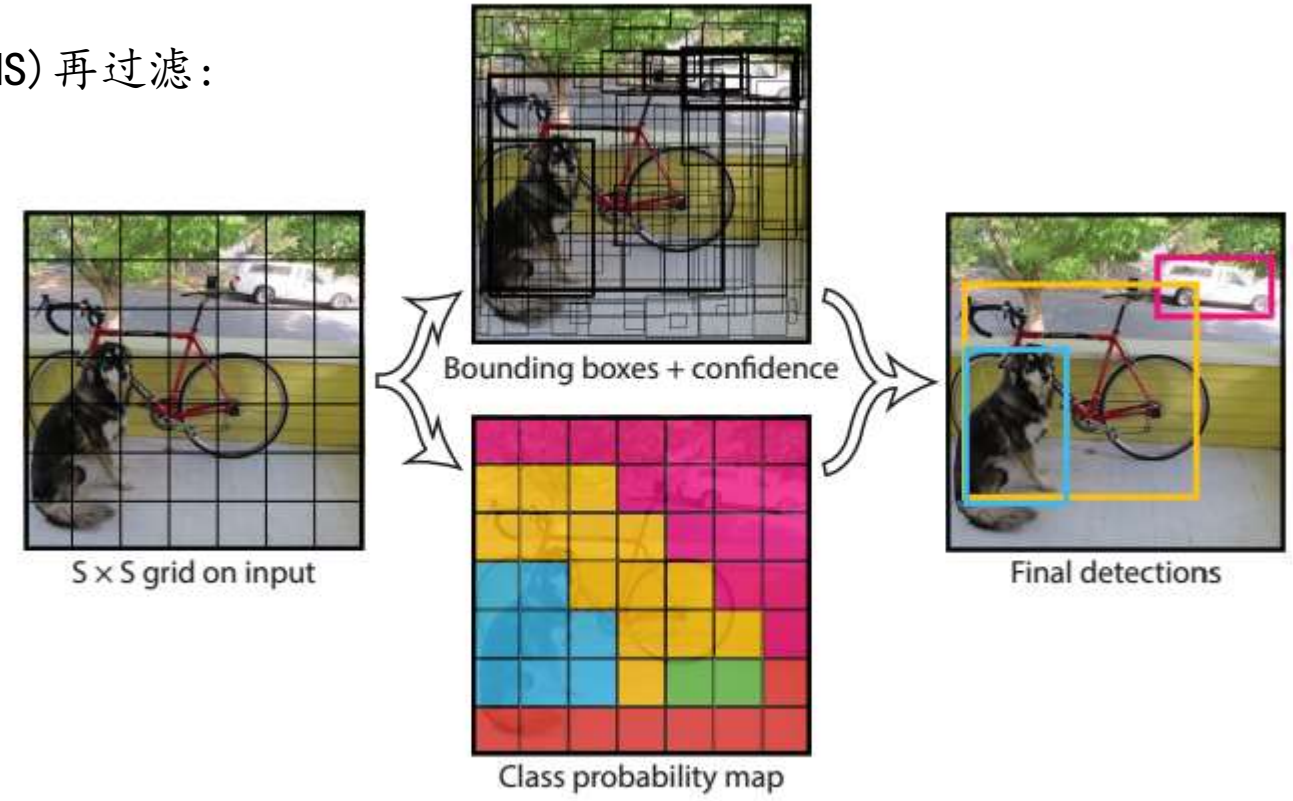


Proposal Layer

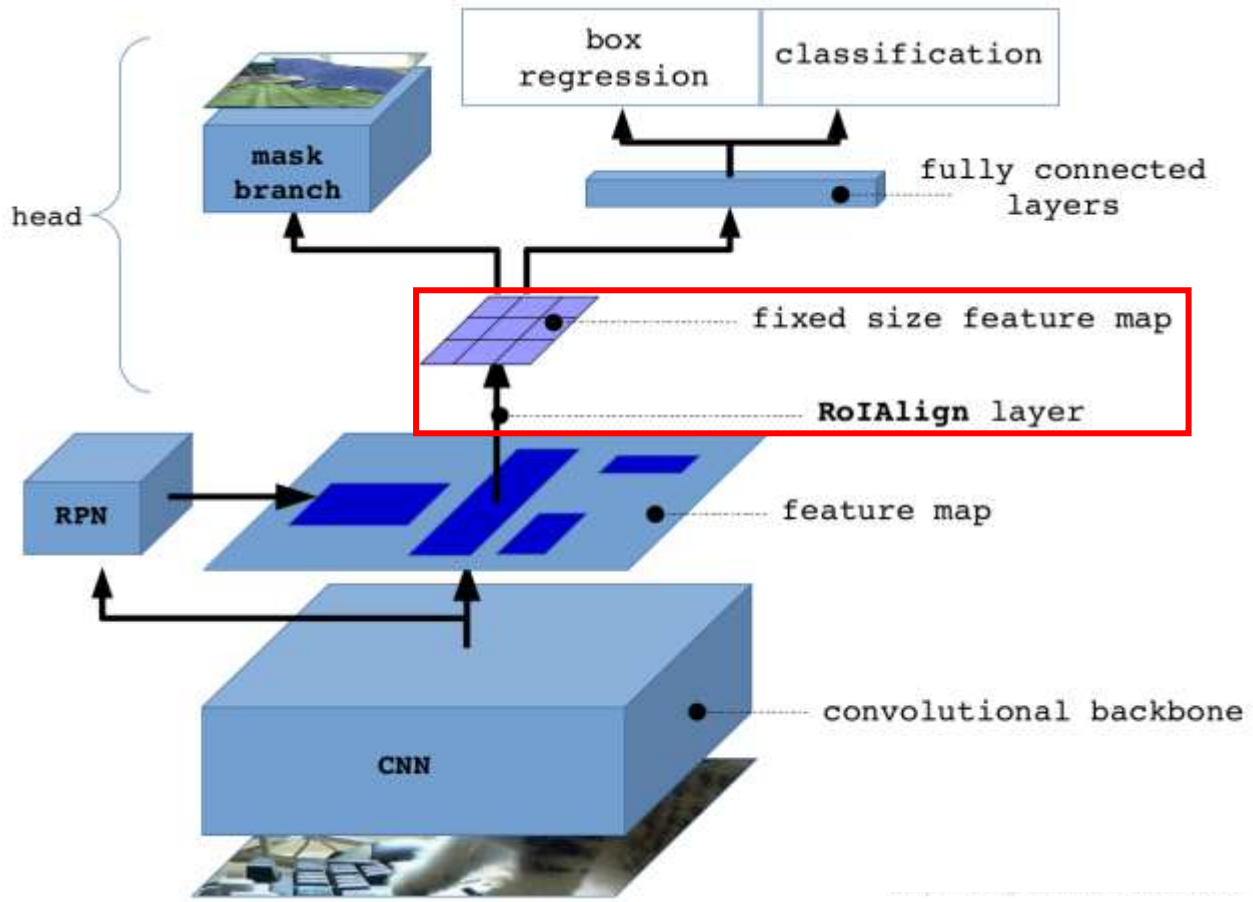
对20W+候选框进行过滤，按照前景得分排序；

取前6000个高分候选框，同时配合其回归值；

Non-maximum suppression（NMS)再过滤：

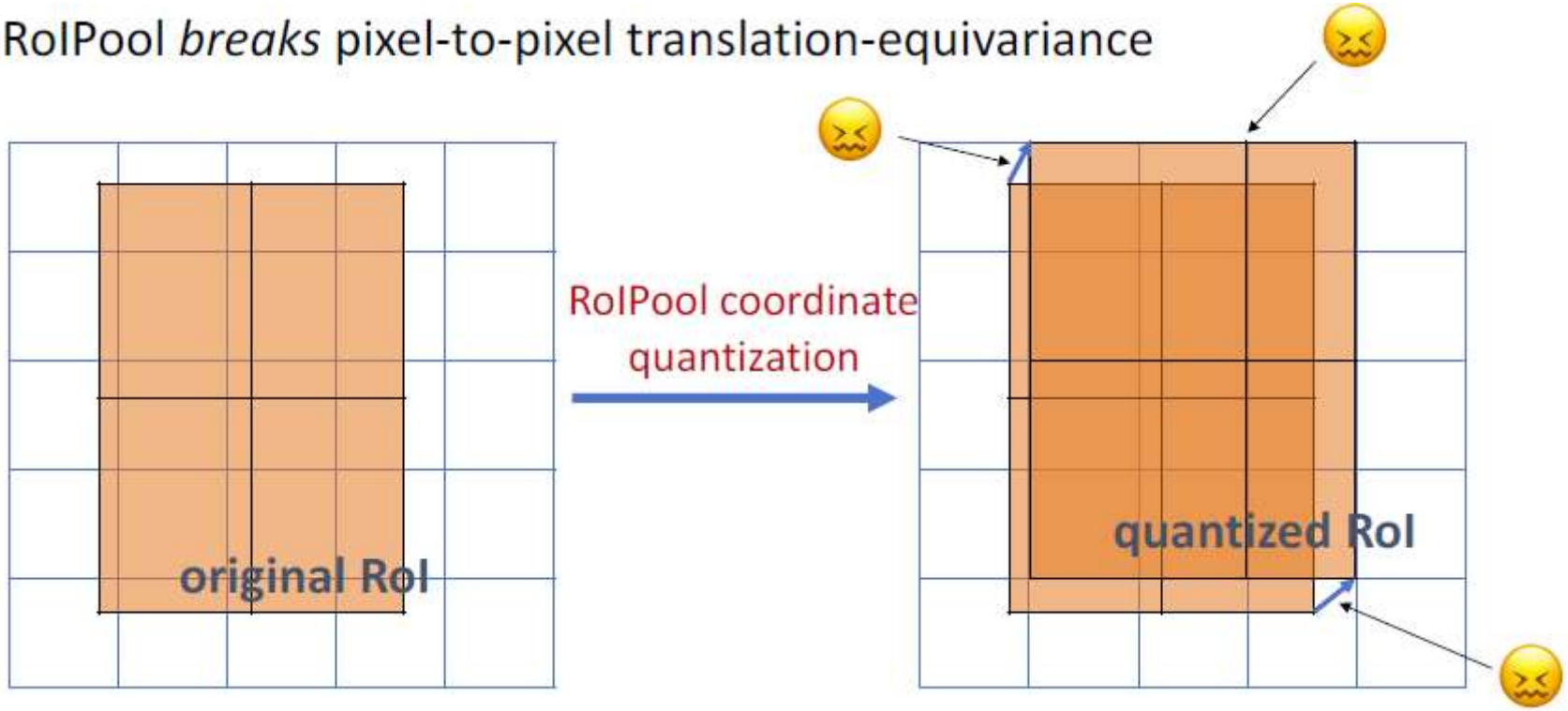


Mask RCNN

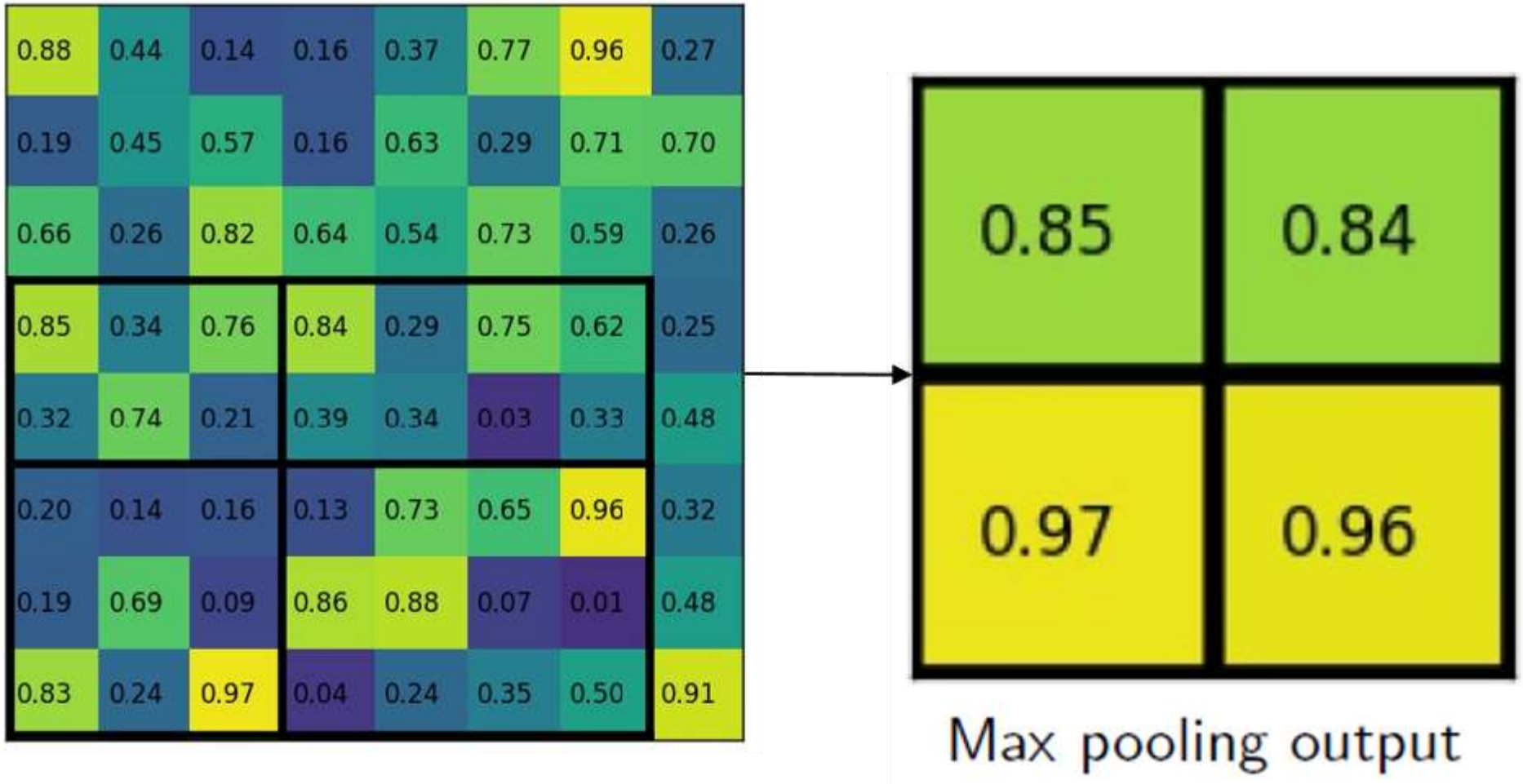

box
regression
classification
mask
branch
head
fully connected
layers
fixed size feature map
RoIAlign layer
RPN
feature map
CNN
convolutional backbone

ROI Pooling

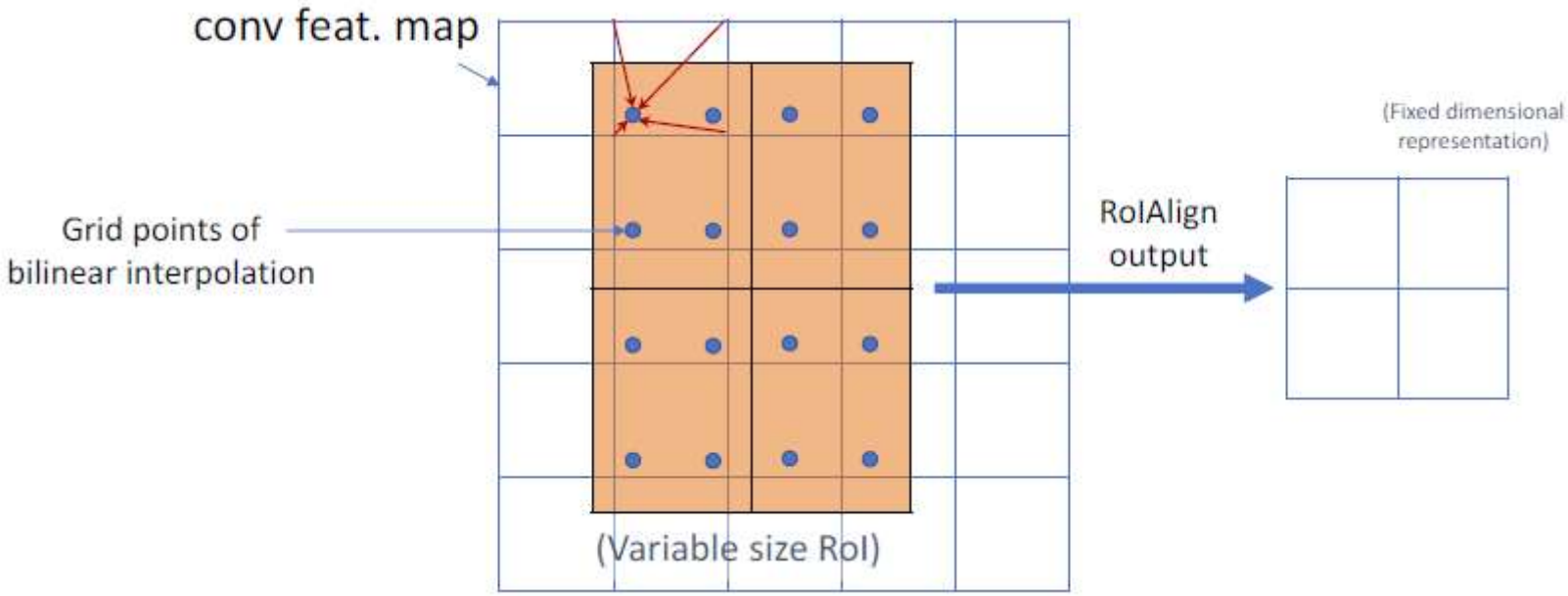
- RoIPool *breaks* pixel-to-pixel translation-equivariance



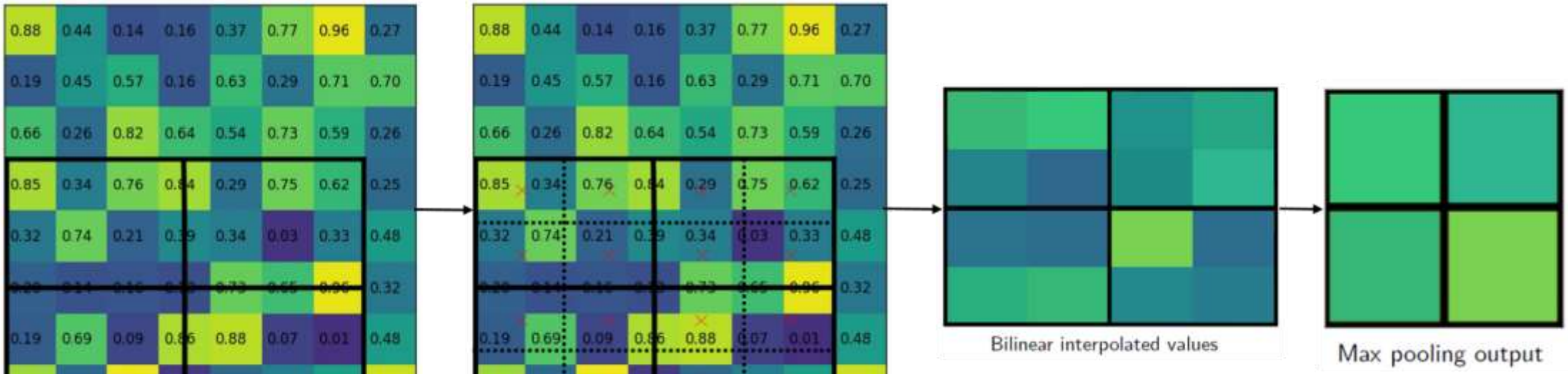
ROI Pooling


0.88 0.44 0.14 0.16 0.37 0.77 0.96 0.27
0.19 0.45 0.57 0.16 0.63 0.29 0.71 0.70
0.66 0.26 0.82 0.64 0.54 0.73 0.59 0.26
0.85 0.34 0.76 0.84 0.29 0.75 0.62 0.25
0.32 0.74 0.21 0.39 0.34 0.03 0.33 0.48
0.20 0.14 0.16 0.13 0.73 0.65 0.96 0.32
0.19 0.69 0.09 0.86 0.88 0.07 0.01 0.48
0.83 0.24 0.97 0.04 0.24 0.35 0.50 0.91
0.85
0.84
0.97
0.96
Max pooling output

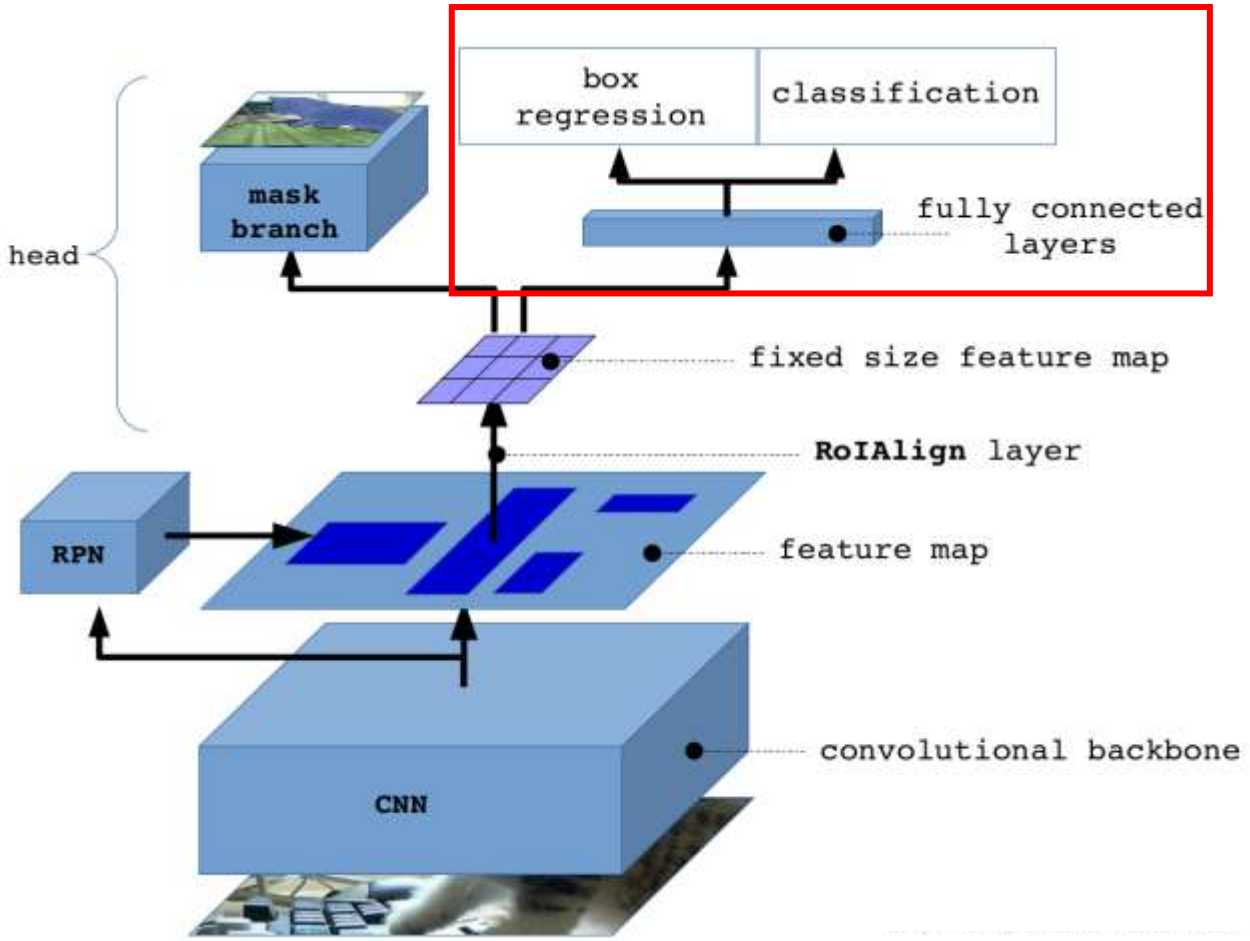
ROI Align


conv feat. map
Grid points of
bilinear interpolation
(Variable size RoI)
RoIAlign
output
(Fixed dimensional
representation)

ROI Align

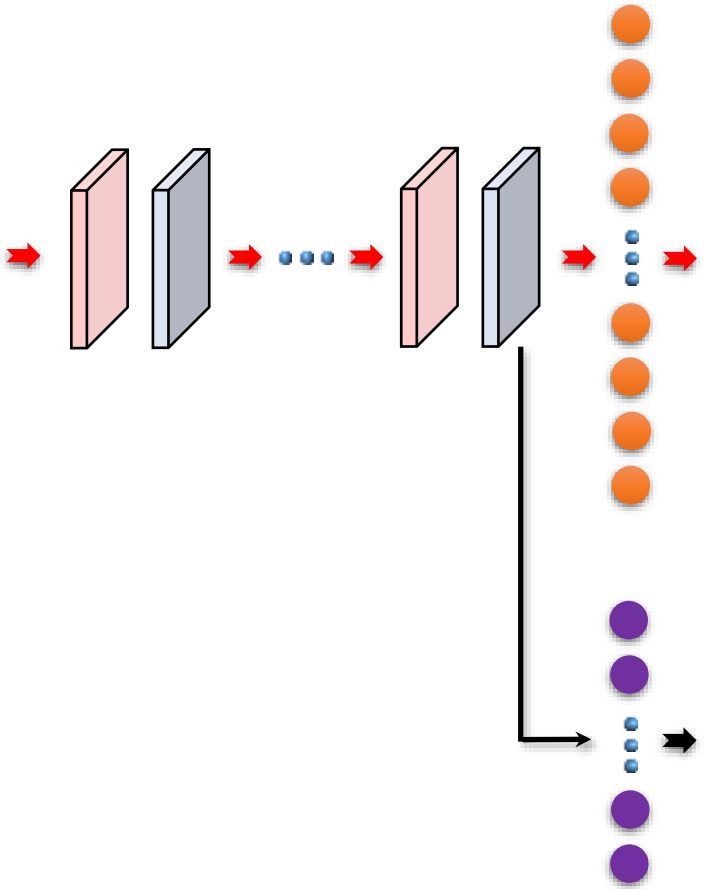

0.88 0.44 0.14 0.16 0.37 0.77 0.96 0.27
0.19 0.45 0.57 0.16 0.63 0.29 0.71 0.70
0.66 0.26 0.82 0.64 0.54 0.73 0.59 0.26
0.85 0.34 0.76 0.84 0.29 0.75 0.62 0.25
0.32 0.74 0.21 0.39 0.34 0.03 0.33 0.48
0.19 0.69 0.09 0.86 0.88 0.07 0.01 0.48
0.83 0.24 0.97 0.04 0.24 0.35 0.50 0.91
Bilinear interpolated values
Max pooling output

Mask RCNN


box
regression
classification
mask
branch
head
fully connected
layers
fixed size feature map
RoIAlign layer
RPN
feature map
CNN
convolutional backbone

Object Localization

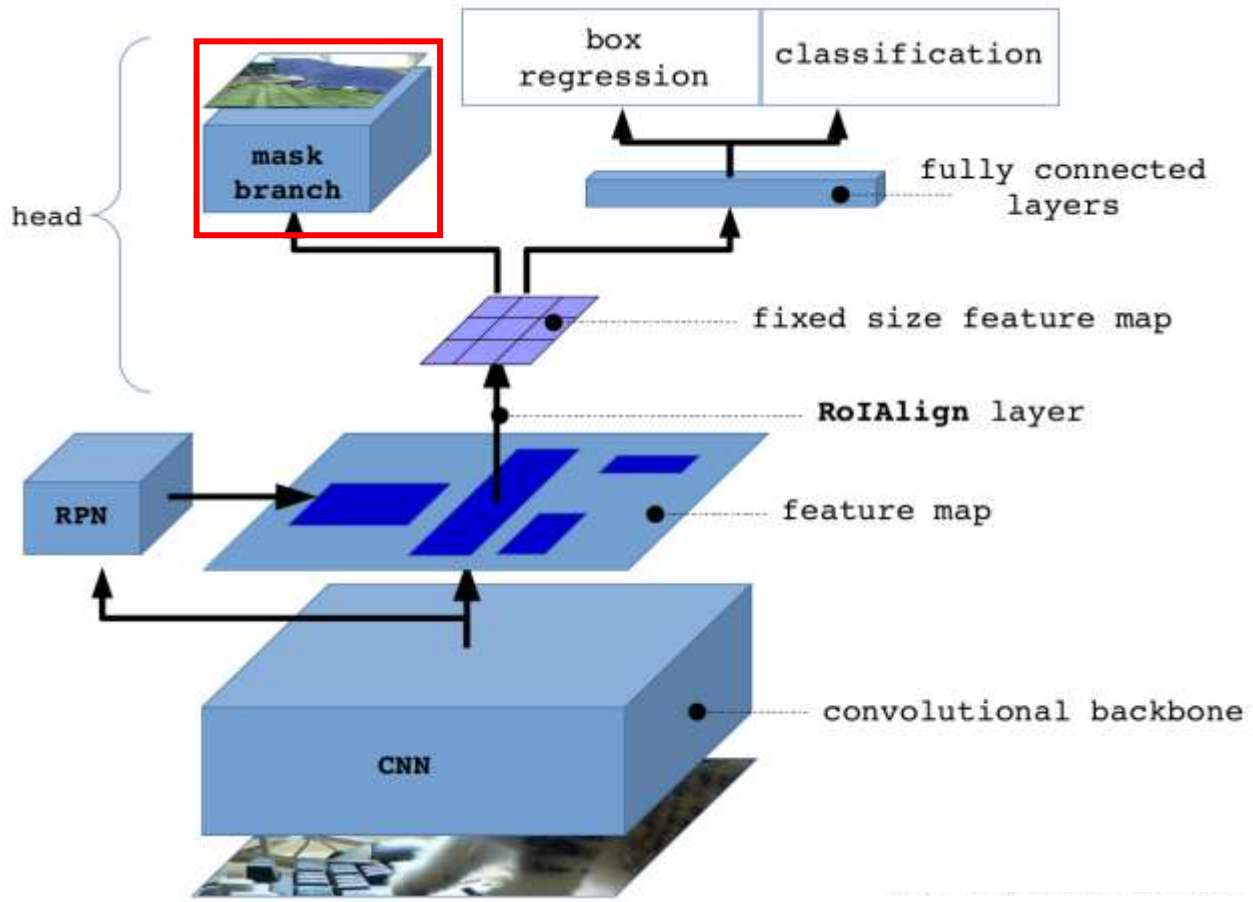
(b_x, b_y, b_w, b_h)
Bounding Box



$$y = \begin{bmatrix} P_c \\ b_x \\ b_y \\ b_w \\ b_h \\ C_1 \\ C_2 \\ C_3 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_w \\ b_h \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad y = \begin{bmatrix} 0 \\ X \\ X \\ X \\ X \\ X \\ X \\ X \end{bmatrix}$$

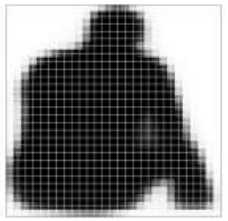
Mask RCNN


box
regression
classification
mask
branch
head
fully connected
layers
fixed size feature map
RoIAlign layer
RPN
feature map
CNN
convolutional backbone

Mask Branch

Mask RCNN：

28x28 soft prediction from Mask R-CNN
(enlarged)



Soft prediction resampled to image coordinates
(bilinear and bicubic interpolation work equally well)

Final prediction (threshold at 0.5)

Mask Scoring RCNN

Figure 1. Demonstrative cases of instance segmentation in which bounding box has a high overlap with ground truth and a high classification score while the mask is not good enough. The scores predicted by both Mask R-CNN and our proposed MS R-CNN are attached above their corresponding bounding boxes. The left four images show good detection results with high classification scores but low mask quality. Our method aims at solving this problem. The rightmost image shows the case of a good mask with a high classification score. Our method will retrain the high score. As can be seen, scores predicted by our model can better interpret the actual mask quality.

https://arxiv.org/pdf/1903.00241.pdf

Mask Scoring RCNN

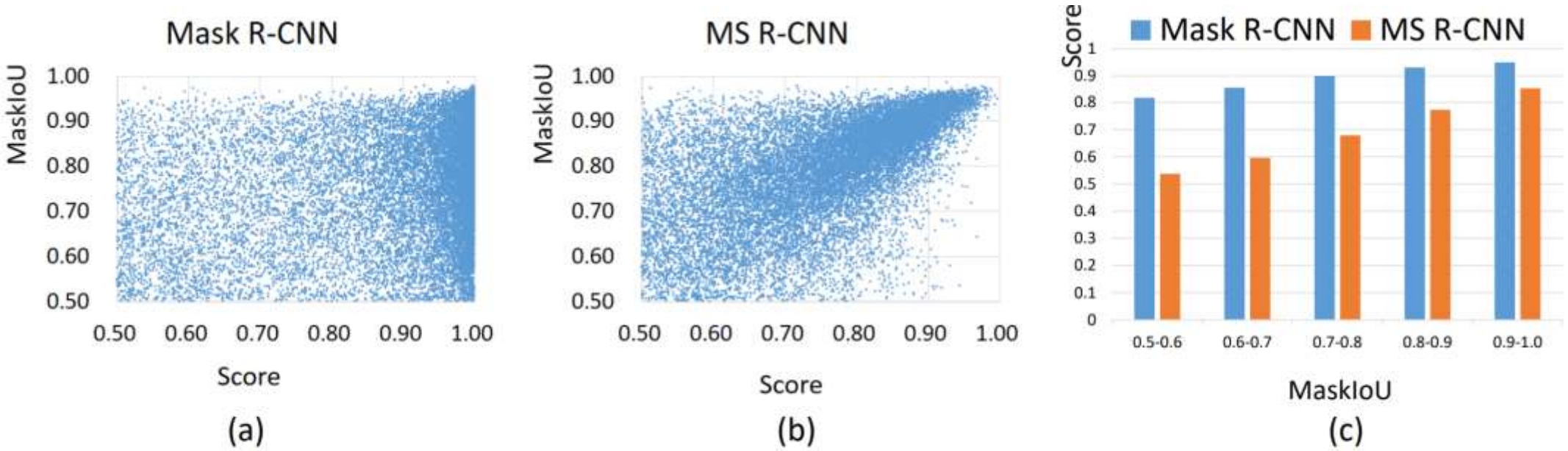


Figure 2. Comparisons of Mask R-CNN and our proposed MS R-CNN. (a) shows the results of Mask R-CNN, the mask score has less relationship with MaskIoU. (b) shows the results of MS R-CNN, we penalize the detection with high score and low MaskIoU, and the mask score can correlate with MaskIoU better. (c) shows the quantitative results, where we average the score between each MaskIoU interval, we can see that our method can have a better correspondence between score and MaskIoU.

Mask Scoring RCNN

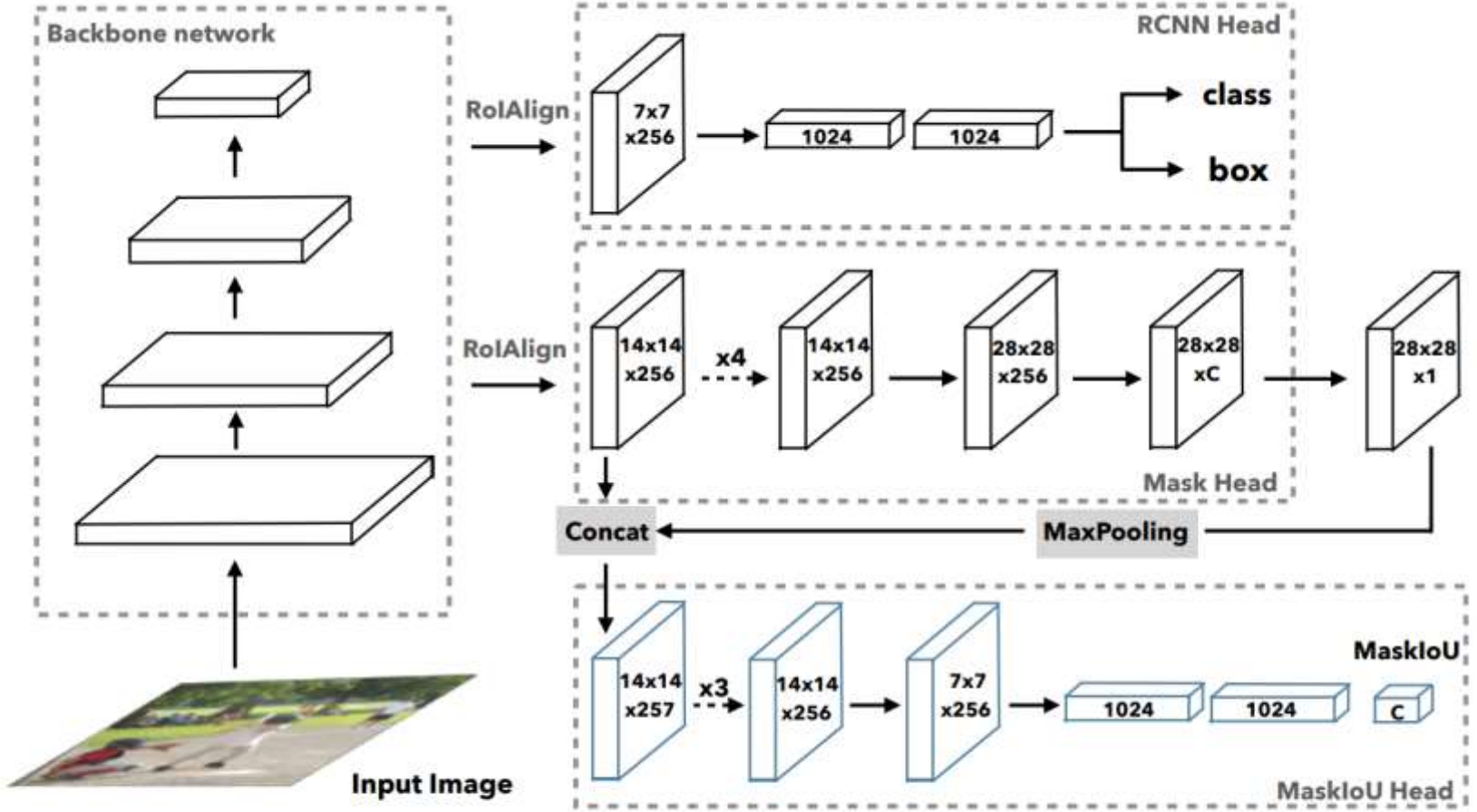


Figure 3. Network architecture of Mask Scoring R-CNN. The input image is fed into a backbone network to generate RoIs via RPN and RoI features via RoIAlign. The RCNN head and Mask head are standard components of Mask R-CNN. For predicting MaskIoU, we use the predicted mask and RoI feature as input. The MaskIoU head has 4 convolution layers (all have kernel=3 and the final one uses stride=2 for downsampling) and 3 fully connected layers (the final one outputs C classes MaskIoU.)

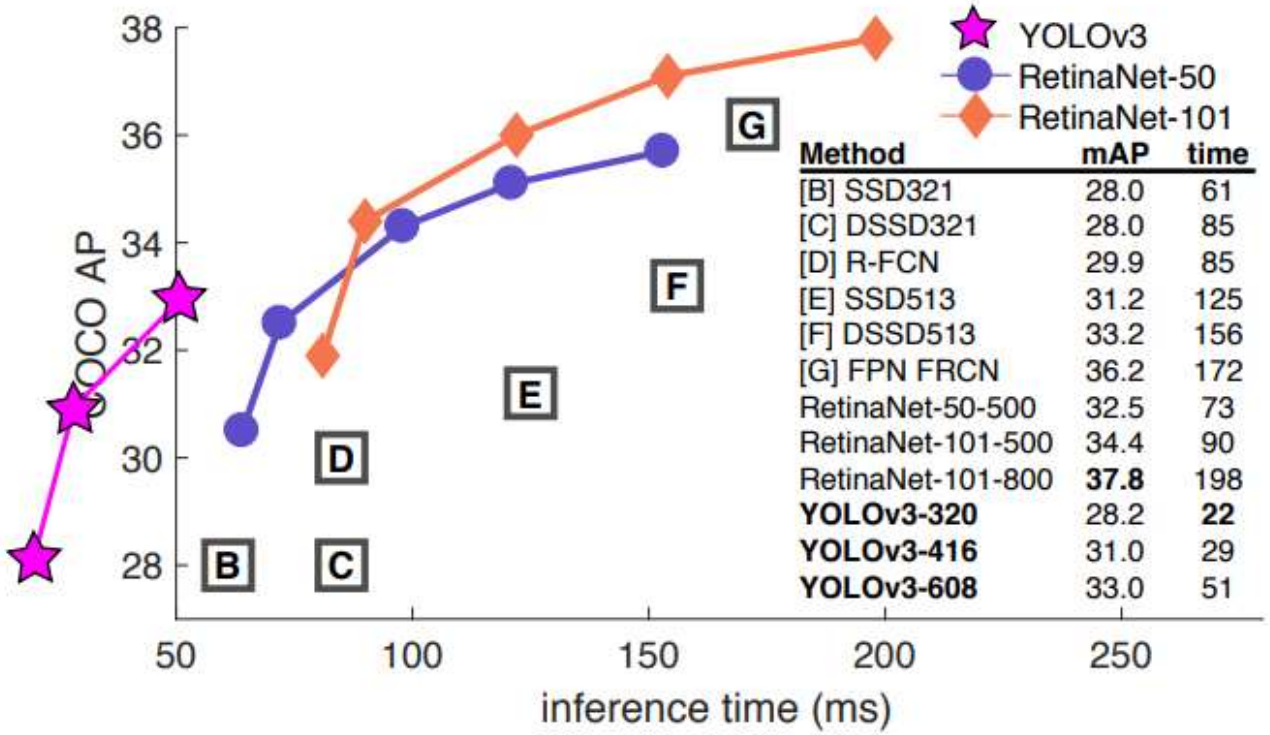
$$S_{mask} = S_{class} \cdot S_{mask\_iou}$$

03

Object Detection: One Stage

一刀流： Yolo v3

Yolo v3: You Onlv Look Once



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

Figure 1. We adapt this figure from the Focal Loss paper [9]. YOLOv3 runs significantly faster than other detection methods with comparable performance. Times from either an M40 or Titan X, they are basically the same GPU.

https://pjreddie.com/darknet/yolo/

In closing, do not @ me. (Because I finally quit Twitter).

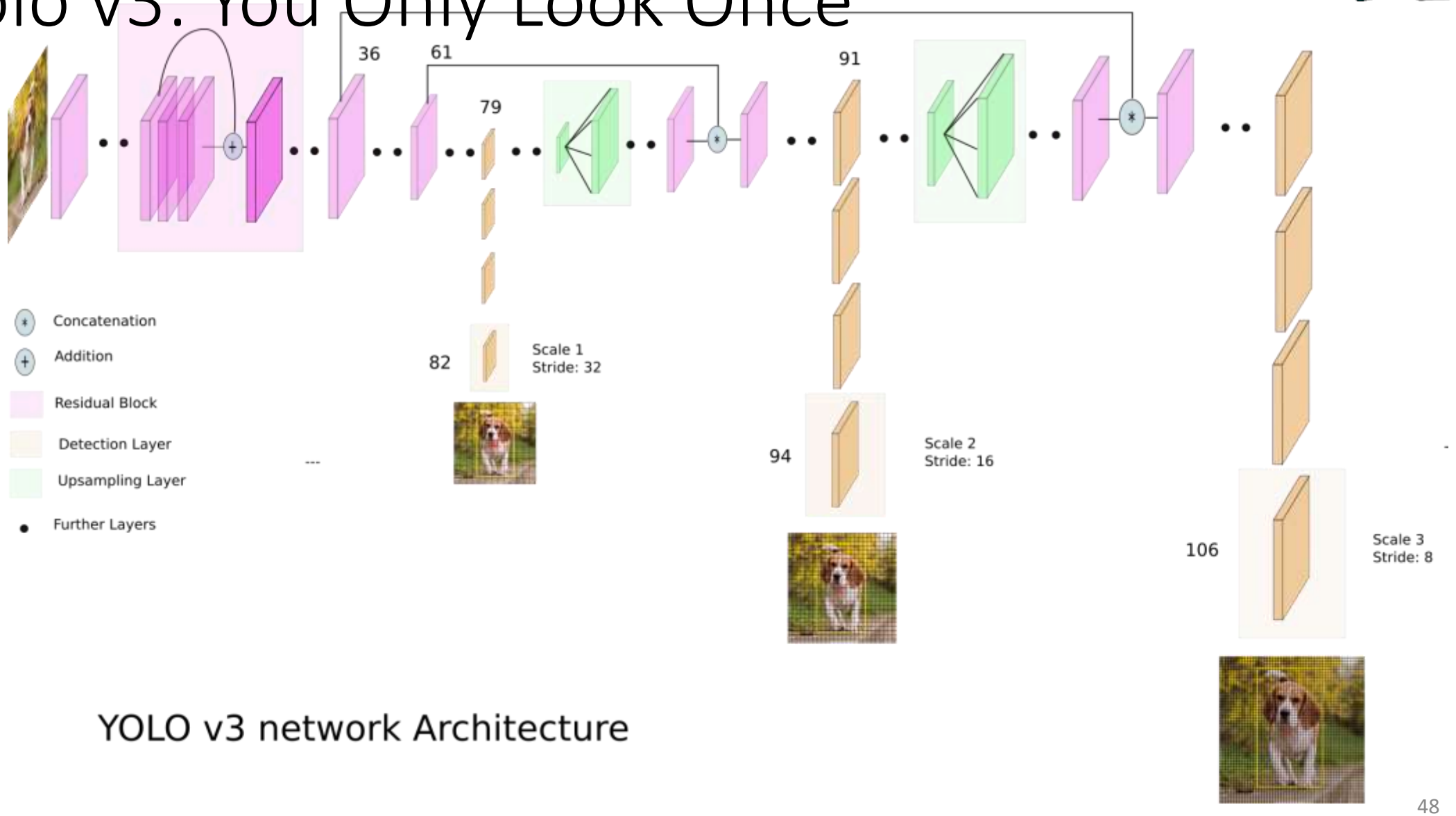
[1]The author is funded by the Office of Naval Research and Google.

Darknet 53

Backbone	Top-1	Top-5	Bn Ops	BFLOP/s	FPS
Darknet-19 [15]	74.1	91.8	7.29	1246	**171**
ResNet-101[5]	77.1	93.7	19.7	1039	53
ResNet-152 [5]	**77.6**	**93.8**	29.4	1090	37
Darknet-53	77.2	**93.8**	18.7	**1457**	78

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

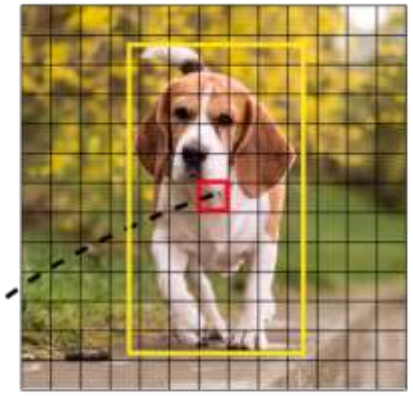
Yolo v3: You Only Look Once



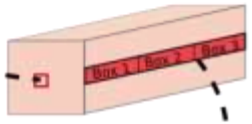
YOLO v3 network Architecture

Yolo v3: You Only Look Once

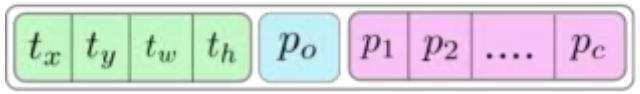
Image Grid. The Red Grid is responsible for detecting the dog



Prediction Feature Map

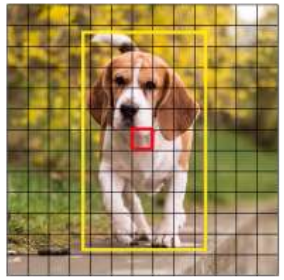


Attributes of a bounding box



x B

Prediction Feature Maps at different Scales



13 x 13

26 x 26

52 x 52

Non-maximum Suppression

Multiple Grids may detect the same object
NMS is used to remove multiple detections

Q&A