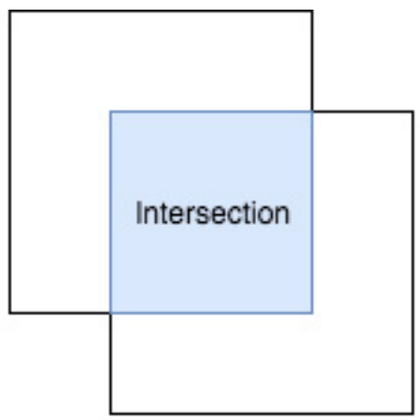
Object Detection

向王涛

Outline

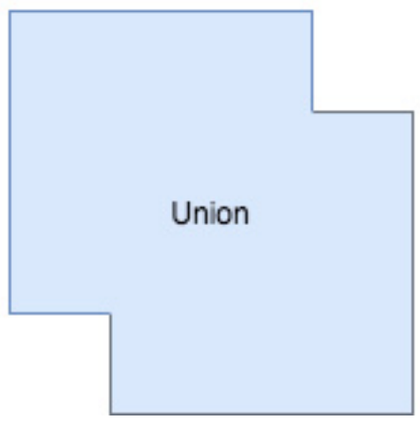
- Prerequisites
- R-CNN/FAST-RCNN/FASTER-RCNN VS. YOLO/YOLOv2/ YOLOv3 VS. SSD ……
- Summary

IoU(Intersection over Union)



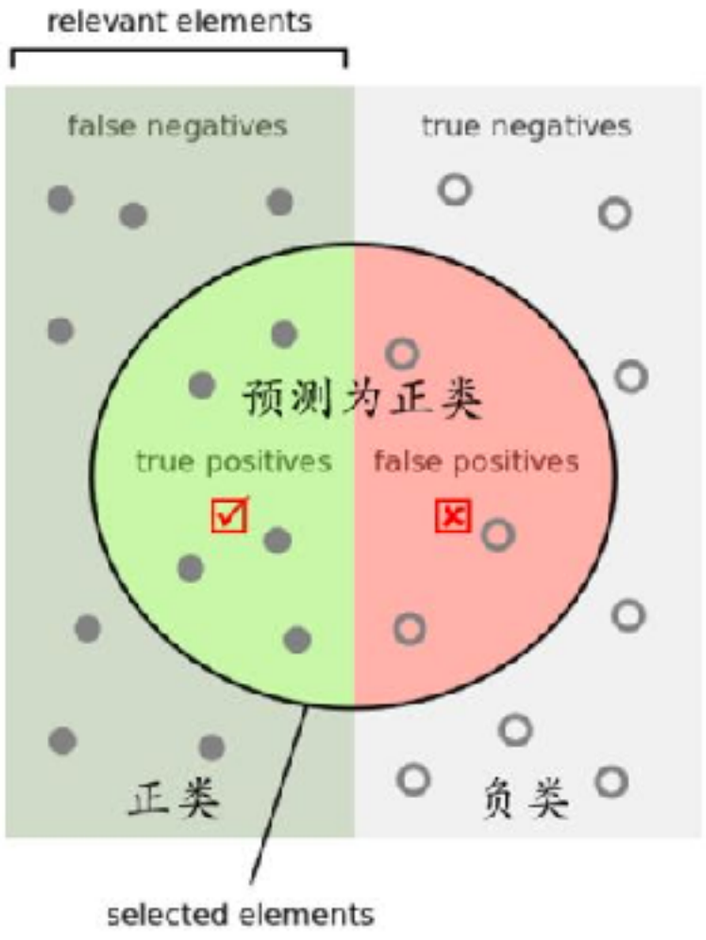
$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

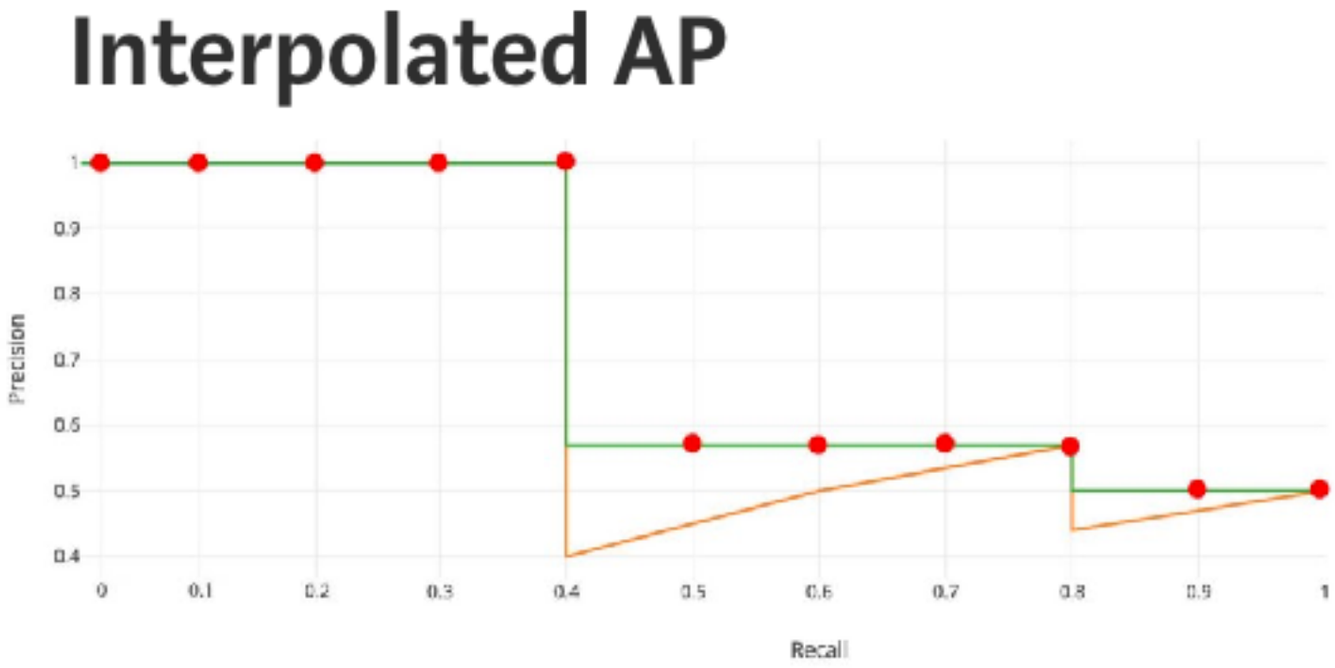
IoU =



一般来说，这个 score ＞ 0.5 可以被认为一个不错的结果

mAP(Mean Average Precision)





$$\mathrm{AP} = \int_0^1 p(r)dr$$

$$AP = \frac{1}{11} \sum_{r \in \{0.0, \ldots, 1.0\}} AP_r$$

$$= \frac{1}{11} \sum_{r \in \{0.0, \ldots, 1.0\}} p_{interp}(r)$$

How many selected items are relevant?

$$\text{Precision} = \frac{\text{green}}{\text{green} + \text{red}}$$

How many relevant items are selected?

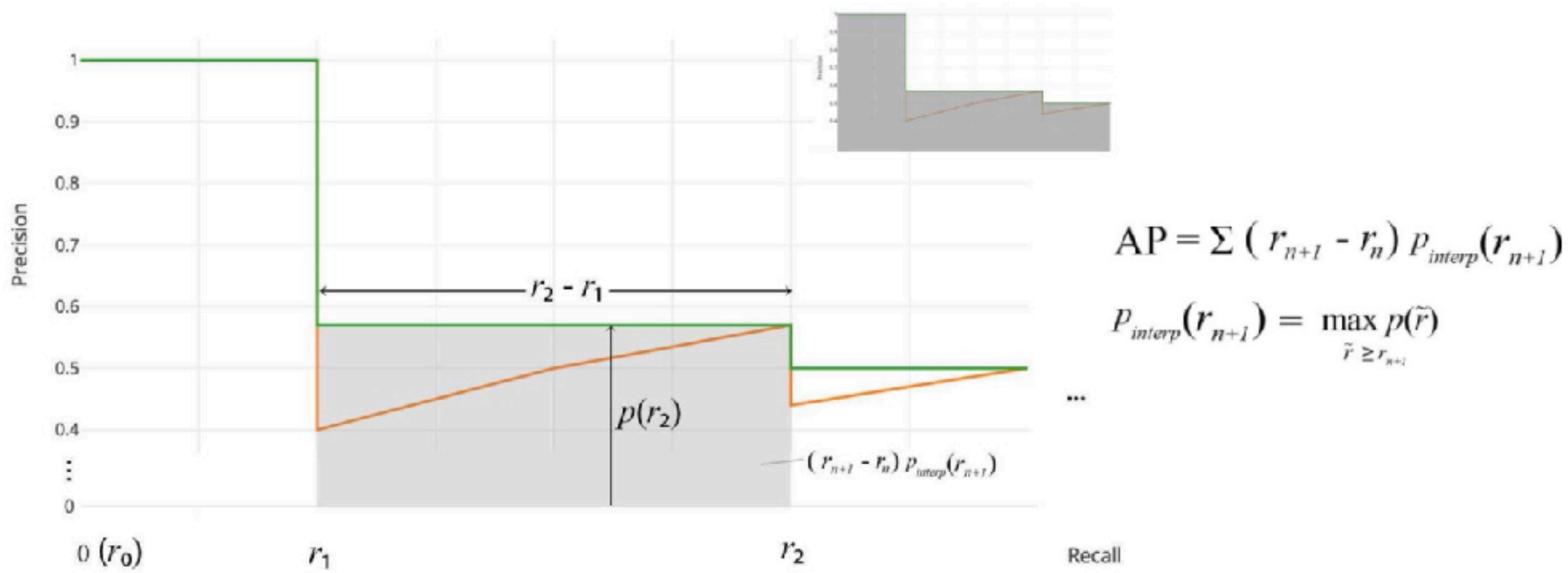
$$\text{Recall} = \frac{\text{green}}{\text{green} + \text{dark green}}$$

$$AP = \frac{1}{11} \times \left(AP_r(0) + AP_r(0.1) + \ldots + AP_r(1.0) \right)$$

$$p_{interp}(r) = \max_{\tilde{r} \geq r} p(\tilde{r})$$

$$\mathrm{AP} = (5 \times 1.0 + 4 \times 0.57 + 2 \times 0.5)/11$$

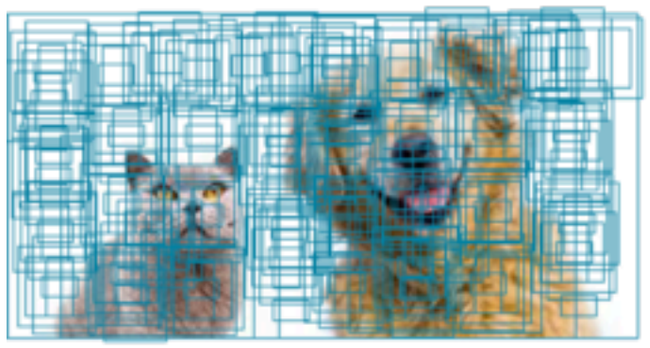
mAP（after VOC2010）



mAP (mean average precision) is the average of AP. In some context, we compute the AP for each class and average them. But in some context, they mean the same thing.

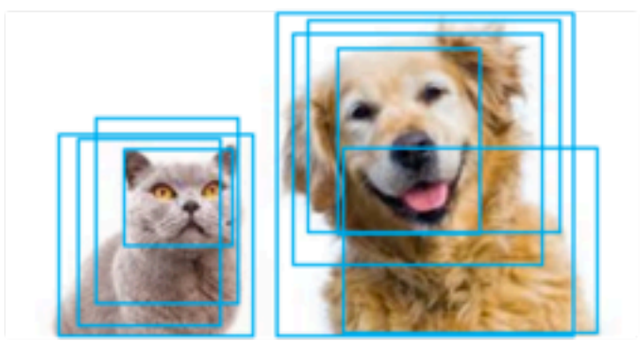
Non-Maximum Suppression, NMS

All candidates



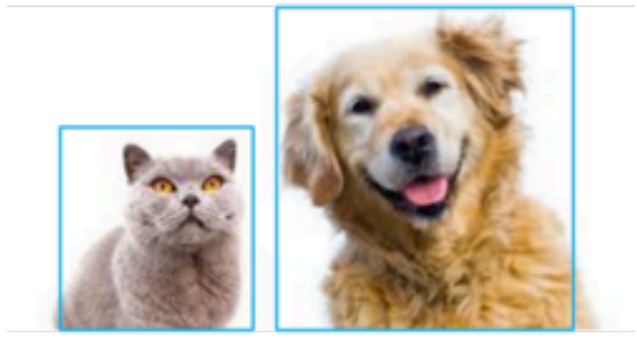
Confidence score > threshold

Possible candidates

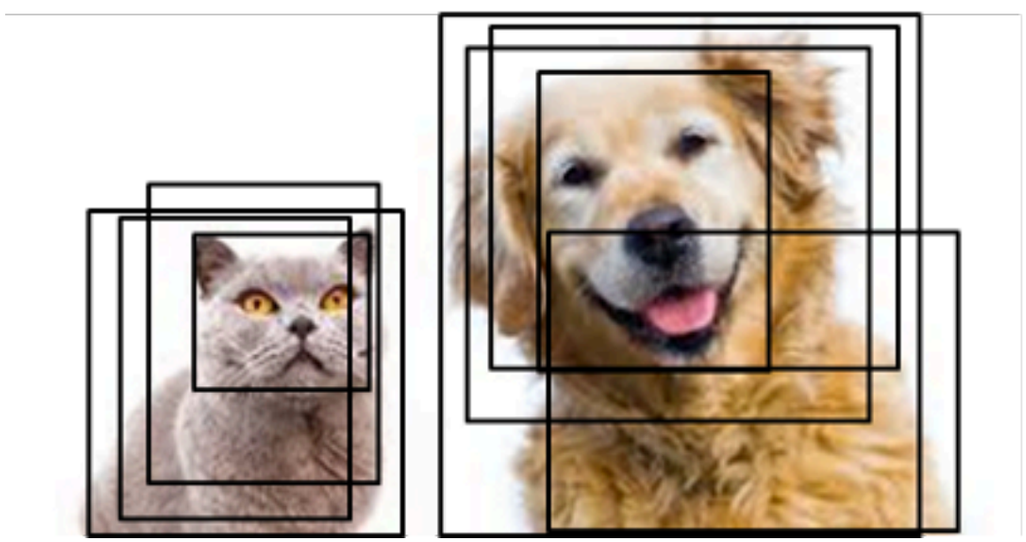


Non-Maximum Suppression (NMS)

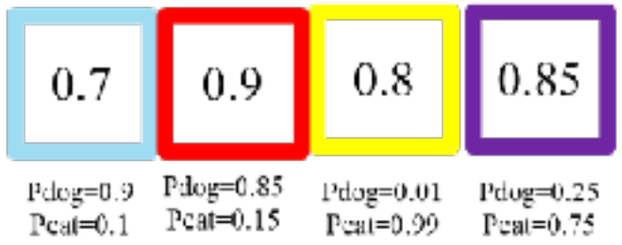
Selected Objects



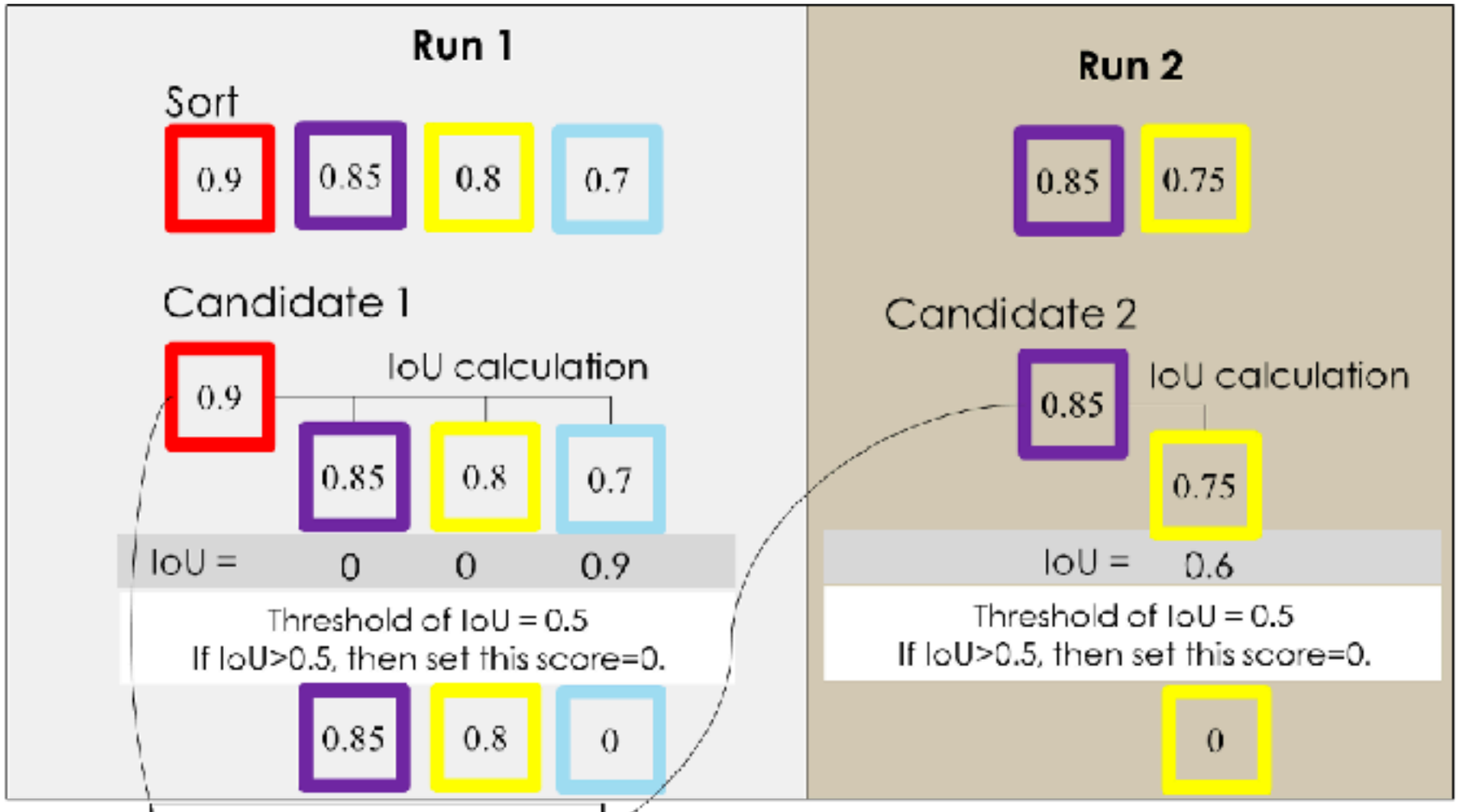
Non-Maximum Suppression (NMS)



Score for 4 candidates



NMS

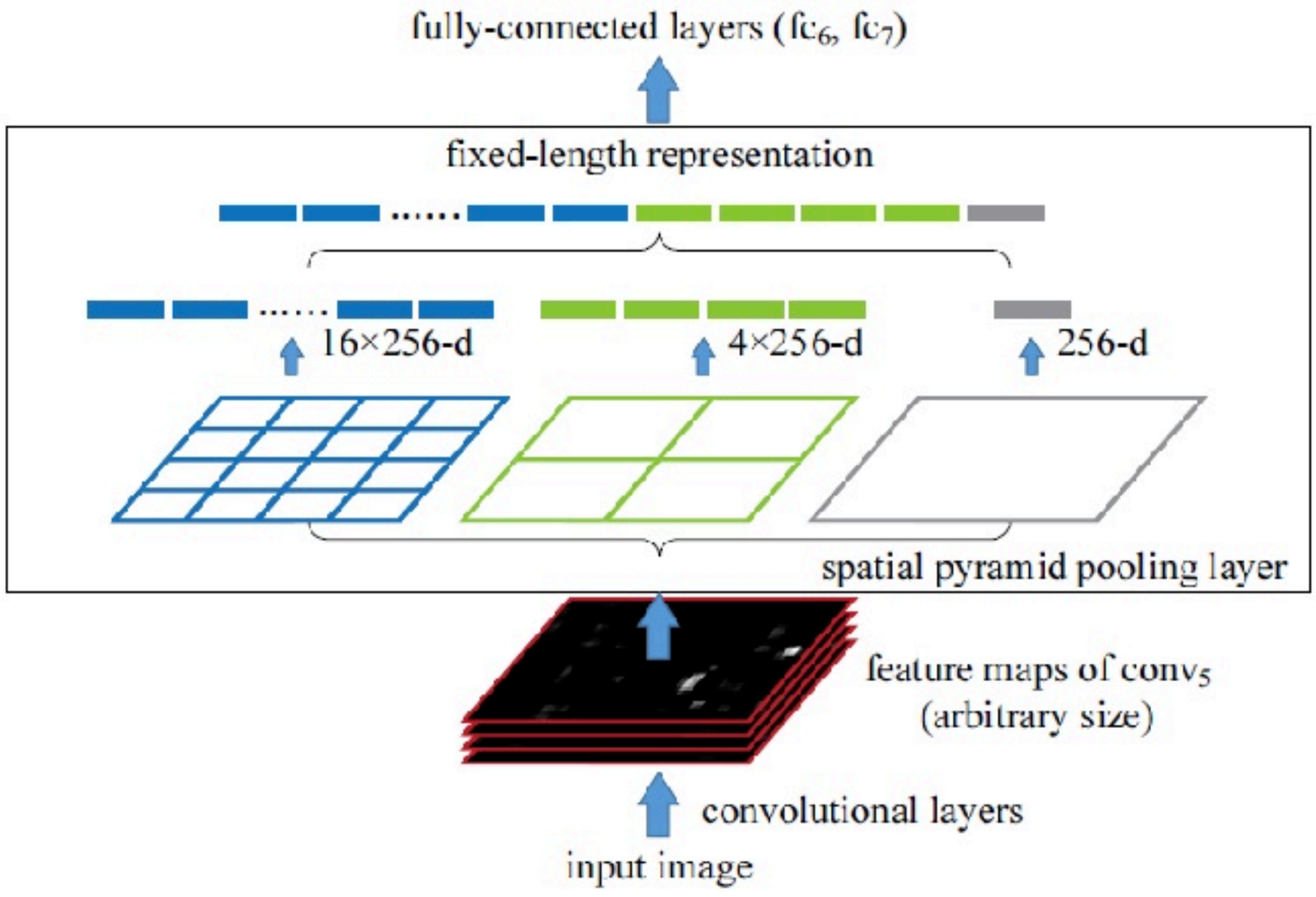


There are no objects, so NMS is done.

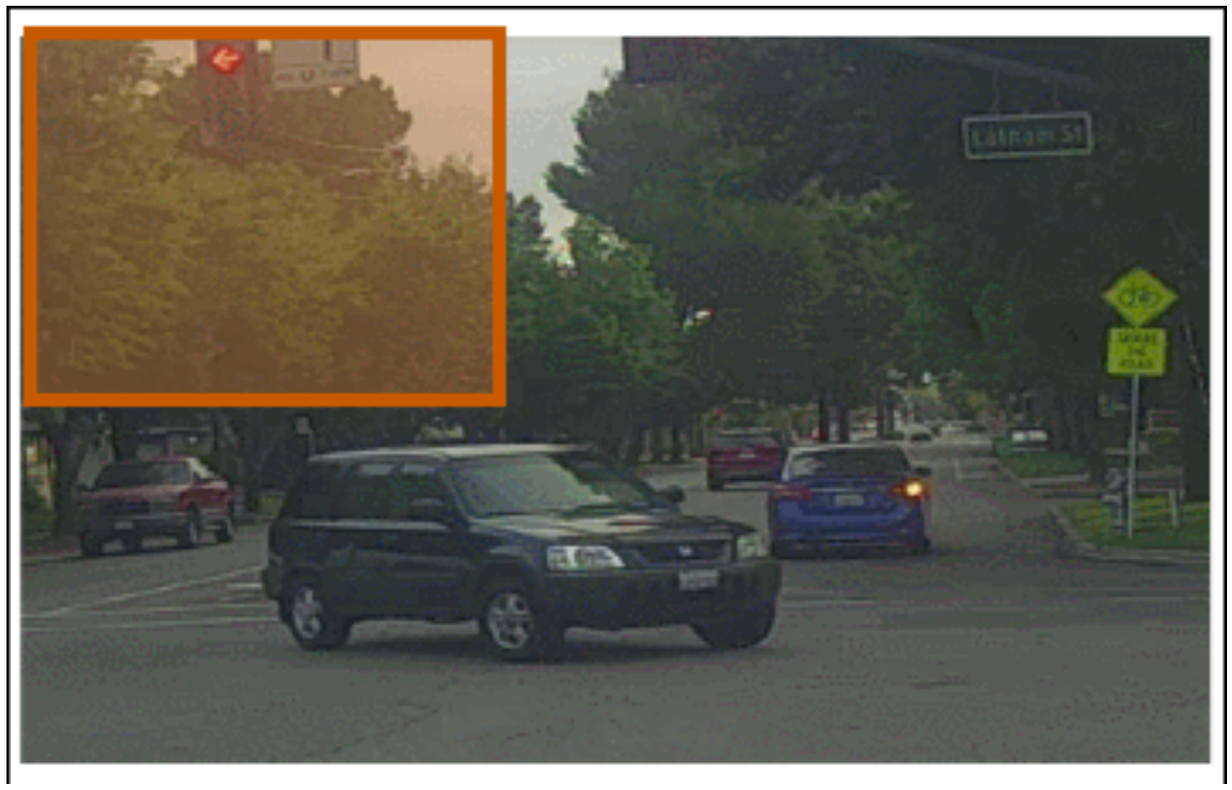
SPPnet

Figure 1: Top: cropping or warping to fit a fixed size. Middle: a conventional CNN. Bottom: our spatial pyramid pooling network structure.

SPPnet


fully-connected layers (fc_6, fc_7)
fixed-length representation
16×256-d
4×256-d
256-d
spatial pyramid pooling layer
feature maps of $conv_5$
(arbitrary size)
convolutional layers
input image

sliding window



expensive computation

not share convolution computation

Prerequisites

- Classification
- Localization
- Detection

Sermanet P, Eigen D, Zhang X, et al. Overfeat: Integrated recognition, localization and detection using convolutional networks[J]. arXiv preprint arXiv:1312.6229, 2013.

Classification

1. 多尺度。
2. 全卷积
3. 全卷积的好处

Localization

1.使用分类的做预训练。

2.训练的时候固定特征提取网络，根据box和真实box之间的l2损失进行训练。

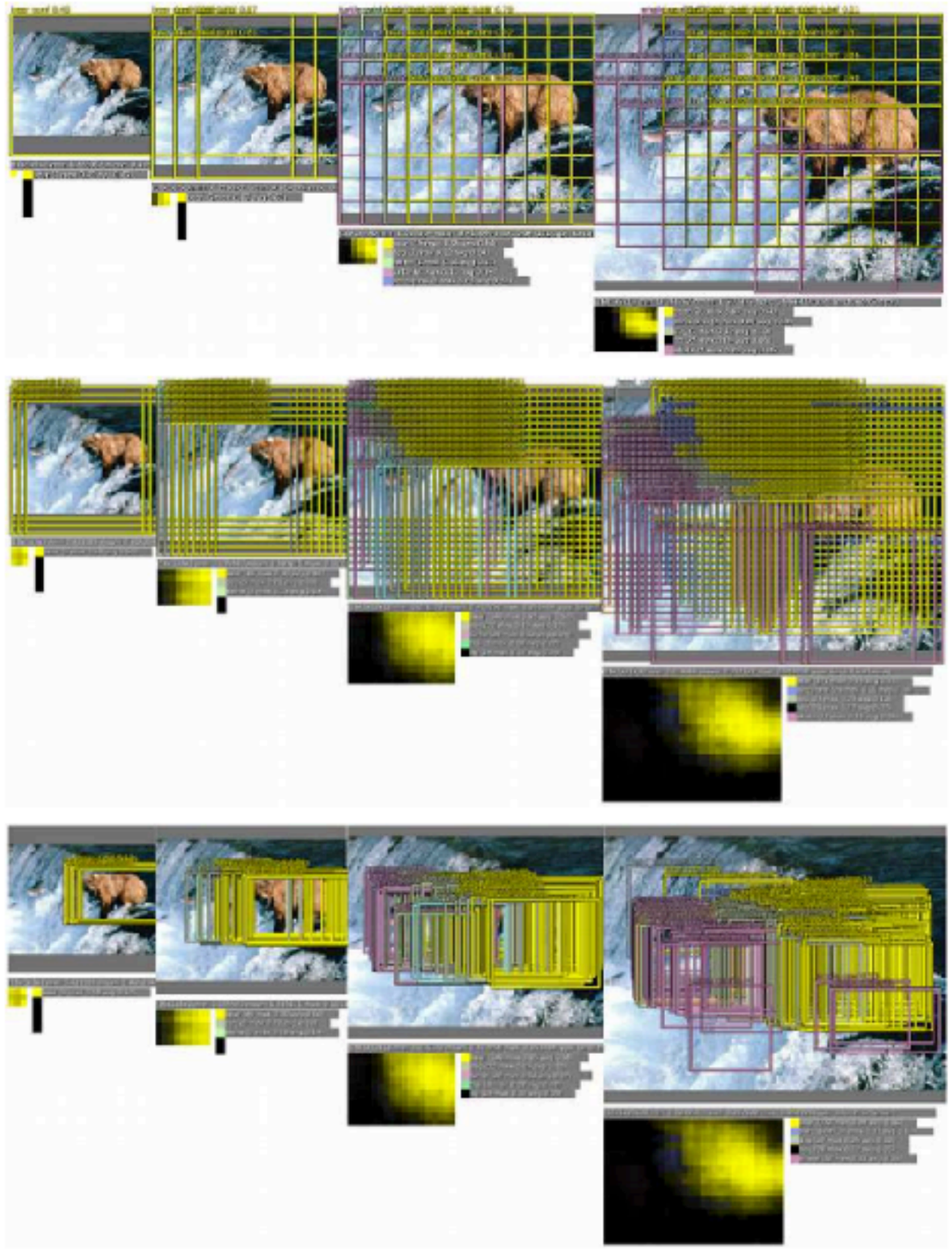
3.回归层最后是1000个类，每个类训练一个回归网络。

预测阶段

累积预测

1. 对于回归网络得到的一系列bounding box，该论文不是通过传统的非极大值抑制，而是使用了累积预测的方法。
2. 首先对于每个scale计算出前k个类别，对每个类别计算出所有的bouding box。
3. 然后合并所有scale的bounding box得到集合B，重复以下步骤
4. $(b_1^*, b_2^*) = argmin_{b_1 \neq b_2 \in B} match\_score(b_1, b_2)$
5. 假如，$match\_score(b_1, b_2) > t$，则停止
6. 否则，$B \leftarrow B \setminus \{b_1^*, b_2^*\} \cup box\_merge(b_1^*, b_2^*)$

这种方法可以淘汰那些低置信度以及低连续（多个box相差很远）的类别



RCNN（two-stage）

步骤一：训练（或者下载）一个分类模型（比如AlexNet）
步骤二：对该模型做fine-tuning

将分类数从1000改为20
去掉最后一个全连接层

步骤三：
提取图像的所有候选框（selective search）
对于每一个区域：修正区域大小以适合CNN的输入，做一次前向运算，将第五个池化层的输出（就是对候选框提取到的特征）存到硬盘

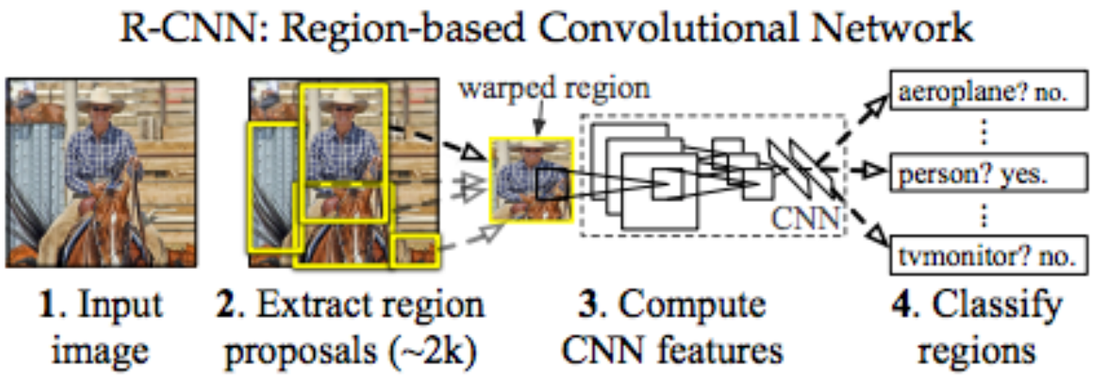
步骤四：
训练一个SVM分类器（二分类）来判断这个候选框里物体的类别

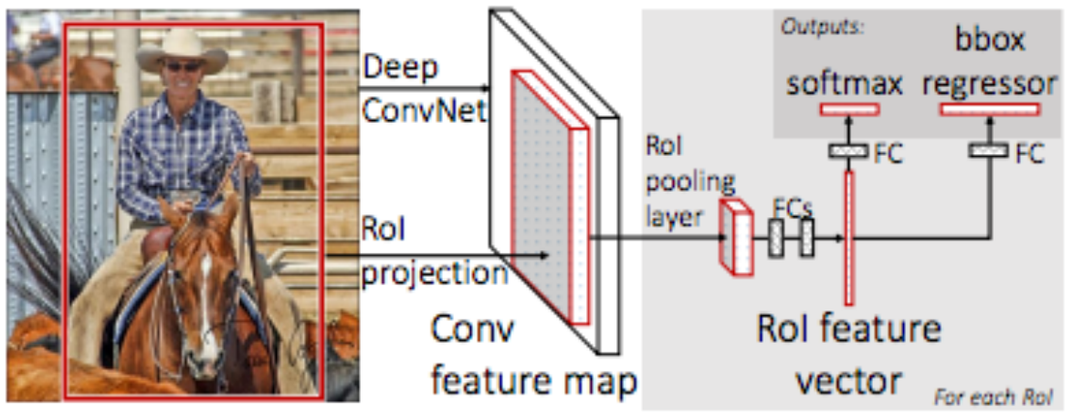
步骤五：使用回归器精细修正候选框位置：对于每一个类，训练一个线性回归模型去判定这个框是否框得完美。

Fast-RCNN

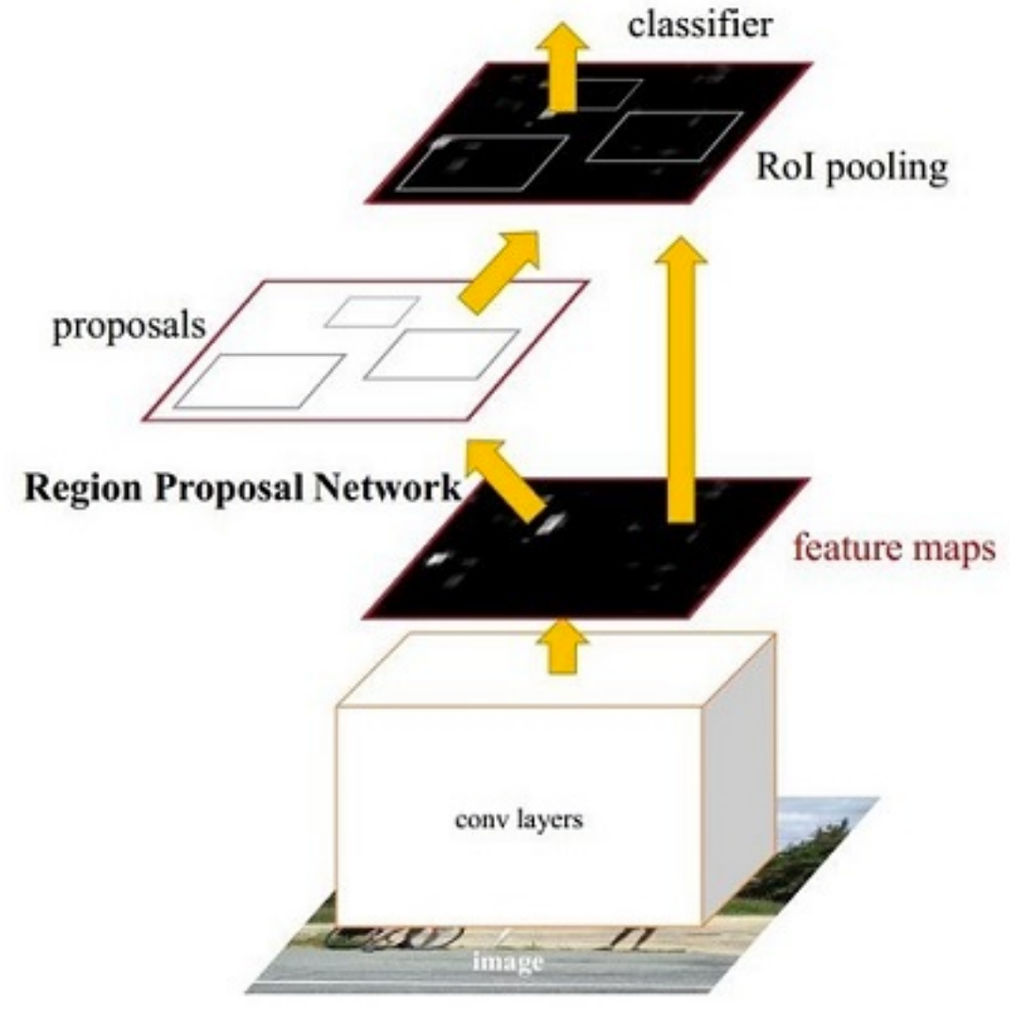
原来的方法：许多候选框（比如两千个）-->CNN-->得到每个候选框的特征-->分类+回归

现在的方法：一张完整图片-->CNN-->得到每张候选框的特征-->分类+回归



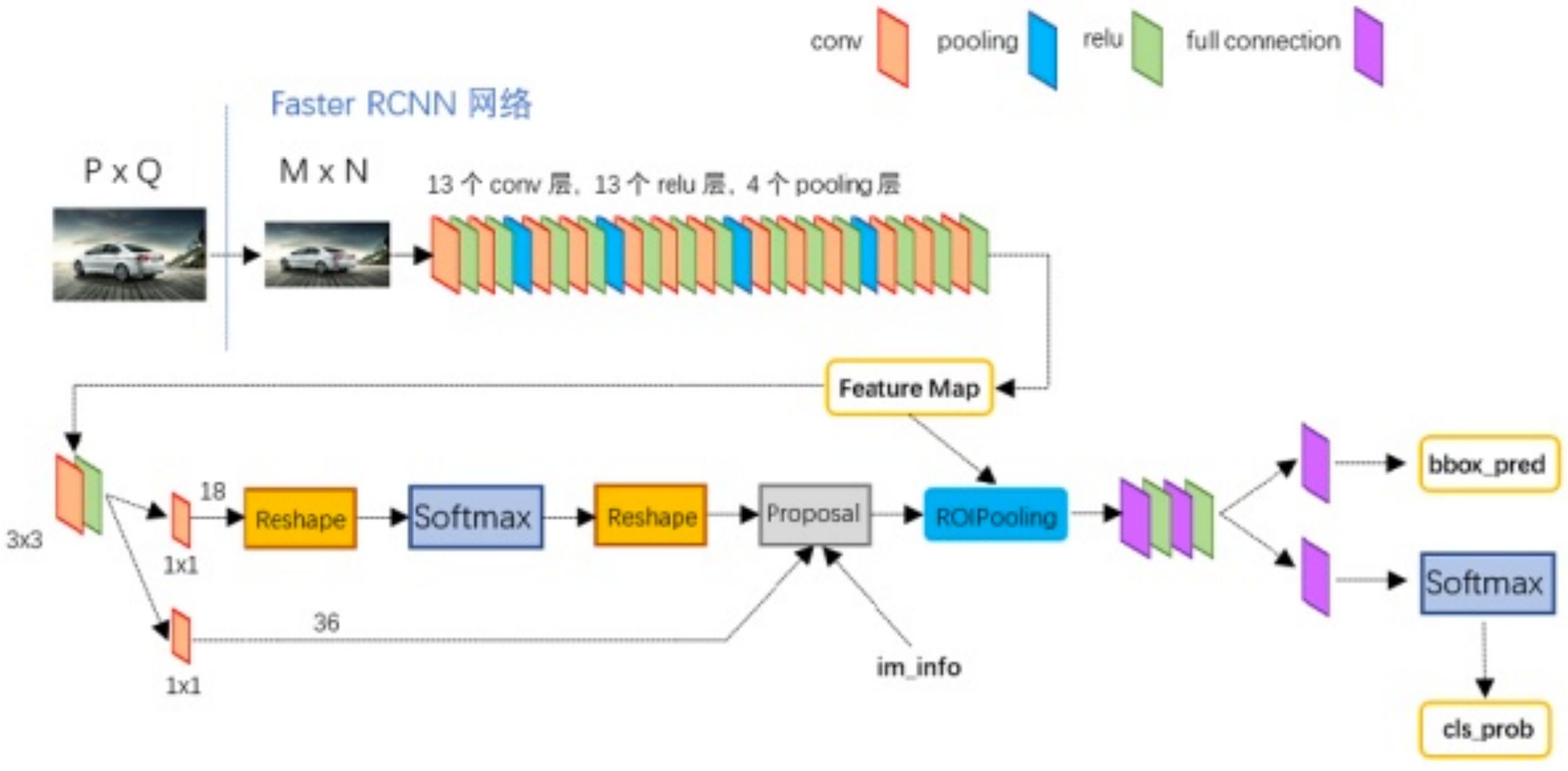


Faster-RCNN

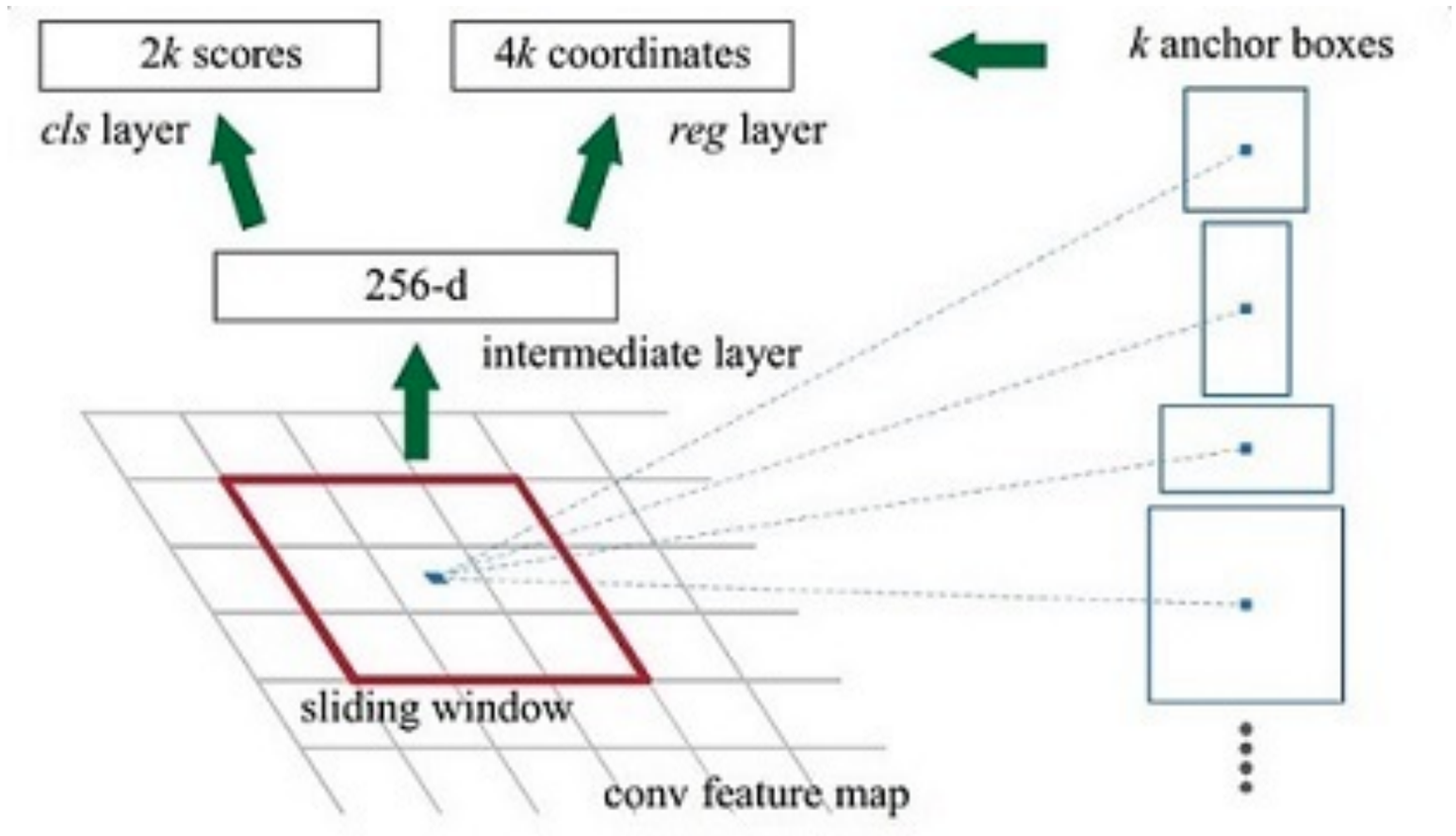


用网络的方法提出了一种proposals方法，RPN网络，
后面RoI pooling跟Fast RCNN差不多，但是不是多个svm分类器
而是一个多分类，通过之前分类信息去确定的

Faster-RCNN


conv
pooling
relu
full connection
Faster RCNN 网络
P x Q
M x N
13 个 conv 层，13 个 relu 层，4 个 pooling 层
Feature Map
3x3
18
1x1
Reshape
Softmax
Reshape
Proposal
ROIPooling
bbox_pred
Softmax
cls_prob
36
1x1
im_info

Faster-RCNN

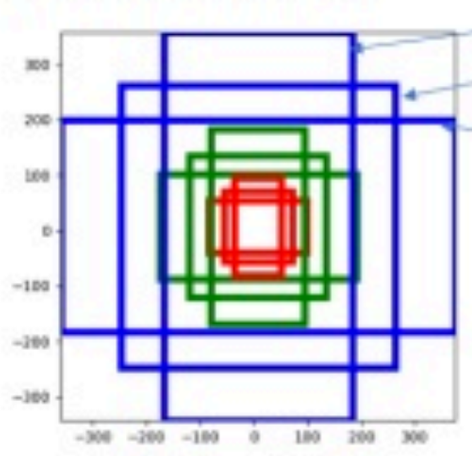

2k scores
4k coordinates
k anchor boxes
cls layer
reg layer
256-d
intermediate layer
sliding window
conv feature map

Faster-RCNN

Generate Anchors

Given:

- Set of aspect ratios (0.5, 1, 2)
- Stride length (downscaling performed by resnet head: 16)
- Anchor Scales (8, 16, 32)



# of anchors: 9

Add anchors to every grid location

16,16

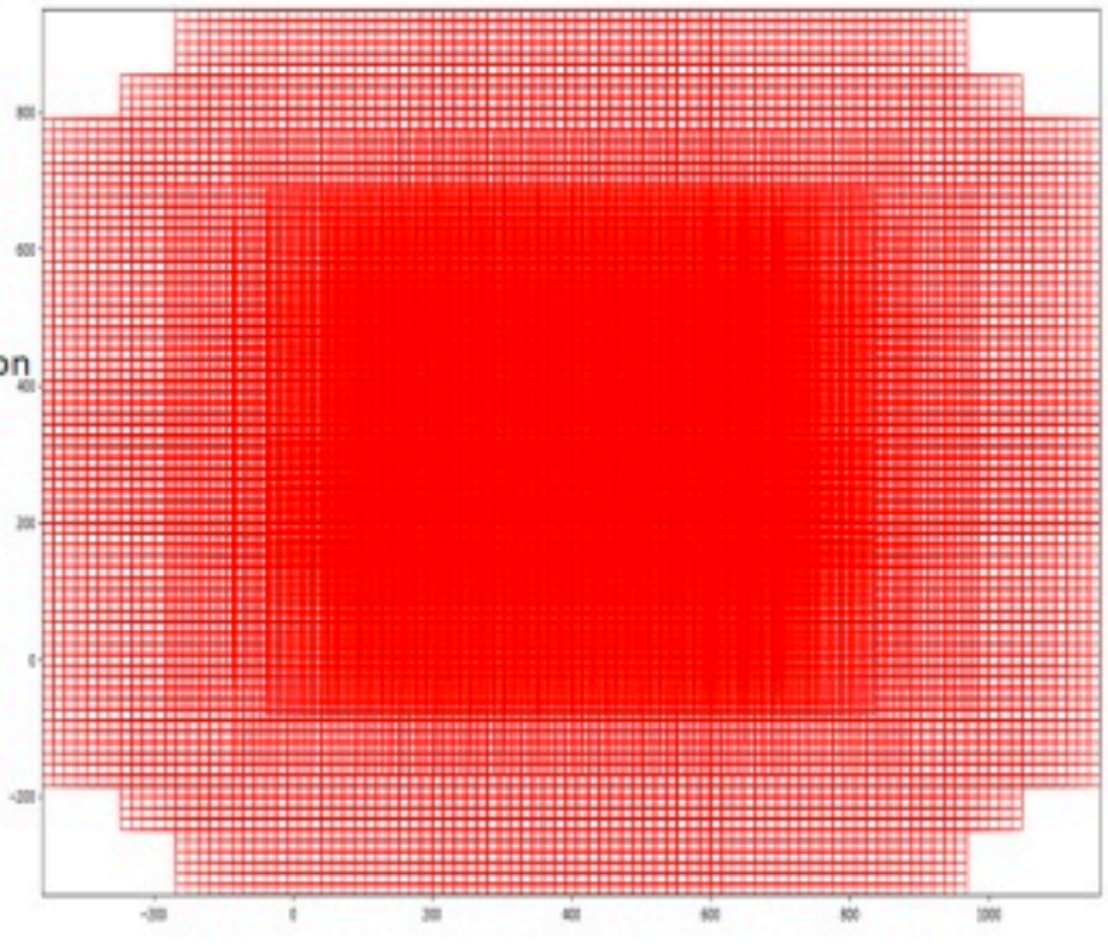
800

600

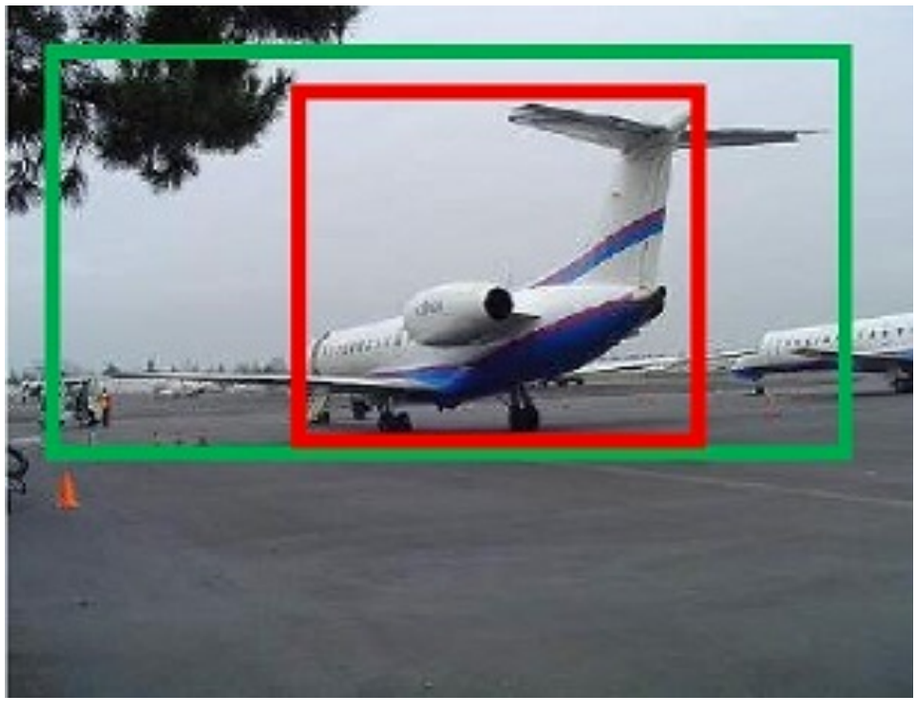
# of grid locations: $\frac{800}{16} * \frac{600}{16} = 1900$

Create uniformly spaced grid with spacing = stride length

Total number of anchors: 1900*9 = 17100
Some boxes lie outside the image boundary



Faster-RCNN



- 给定：anchor $A=(A_x,A_y,A_w,A_h)$ 和 $GT=[G_x,G_y,G_w,G_h]$
- 寻找一种变换F，使得：$F(A_x,A_y,A_w,A_h)=(G'_x,G'_y,G'_w,G'_h)$，其中 $(G'_x,G'_y,G'_w,G'_h)\approx(G_x,G_y,G_w,G_h)$
  - 先做平移

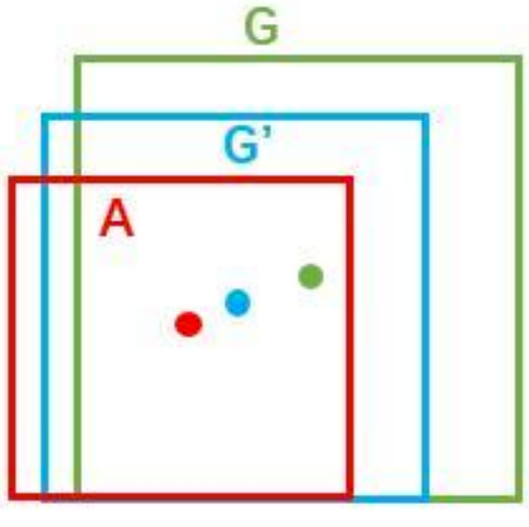
$$G'_x=A_w\cdot d_x(A)+A_x$$

$$G'_y=A_h\cdot d_y(A)+A_y$$

  - 再做缩放

$$G'_w=A_w\cdot \exp(d_w(A))$$

$$G'_h=A_h\cdot \exp(d_h(A))$$



$$\hat{W}_*=\text{argmin}_{W_*}\sum_i^n(t_*^i-W_*^T\cdot\phi(A^i))^2+\lambda||W_*||^2$$

Faster-RCNN

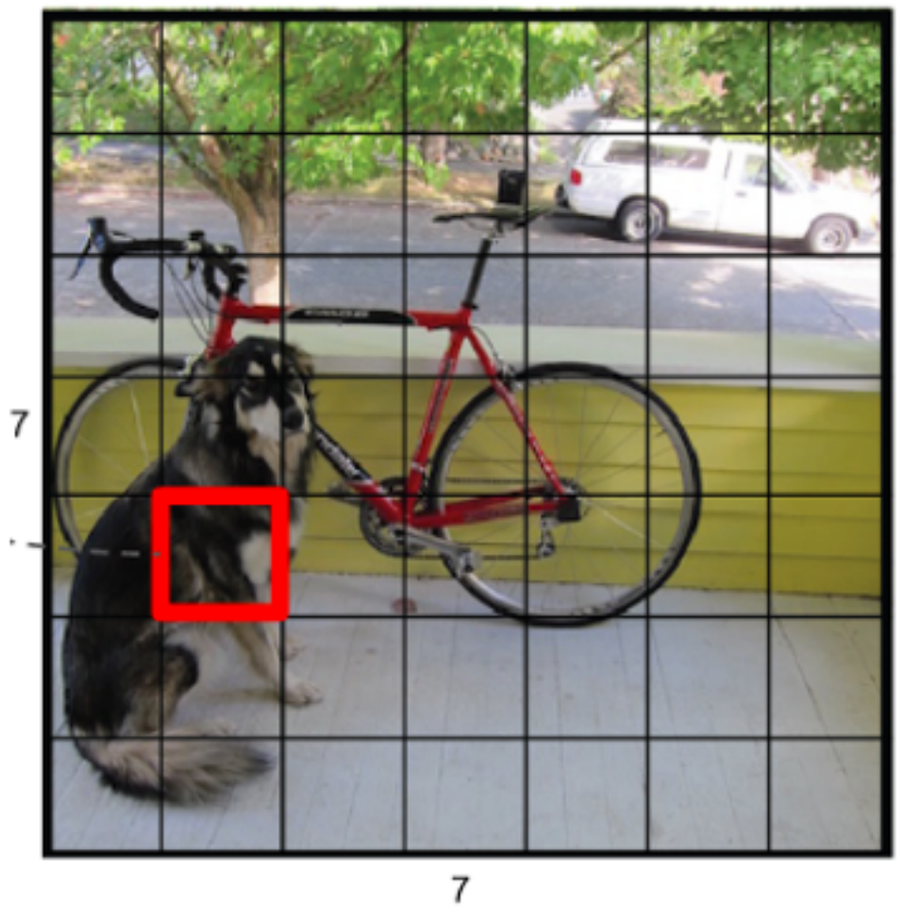
$$\mathrm{L}(\{p_i\},\{t_i\}) = \frac{1}{N_{\mathrm{cls}}}\sum_i \mathrm{L}_{\mathrm{cls}}(p_i, p_i^*) + \lambda \frac{1}{N_{\mathrm{reg}}}\sum_i p_i^* \mathrm{L}_{\mathrm{reg}}(t_i, t_i^*)$$

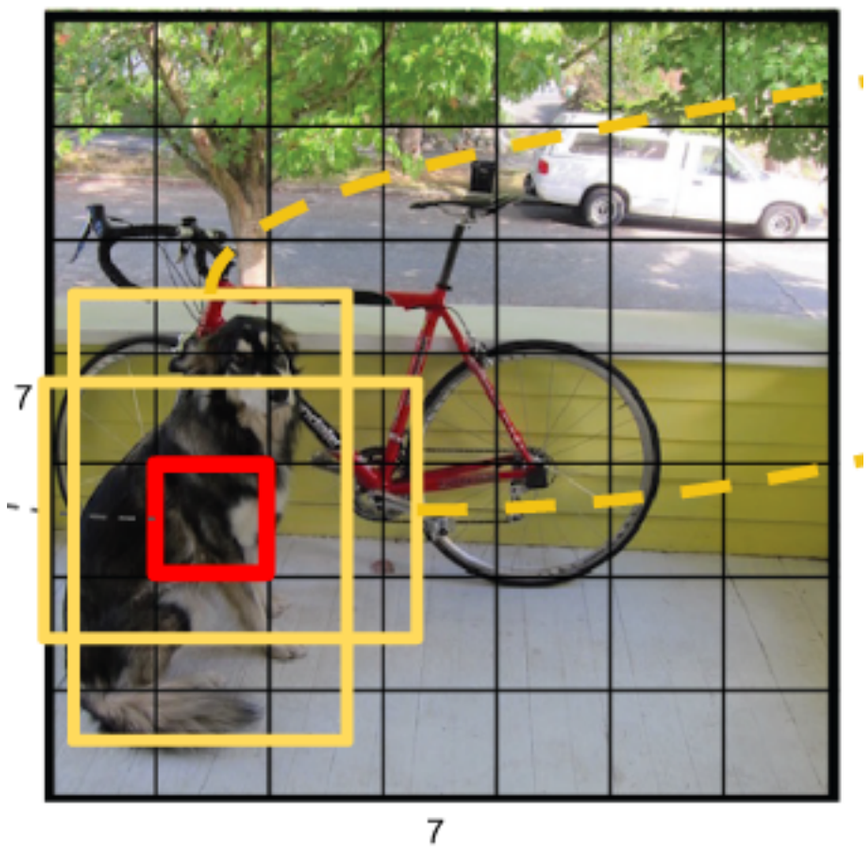
$$\mathrm{L}_{\mathrm{reg}}(t_i, t_i^*) = \sum_{i\in\{x,y,w,h\}} \mathrm{smooth}_{\mathrm{L1}}(t_i - t_i^*)$$

$$\mathrm{soomth}_{\mathrm{L1}}(x) = \begin{cases} 0.5x^2 & \text{if } |\mathrm{x}|<1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

YOLO(one-stage)

You Only Look Once





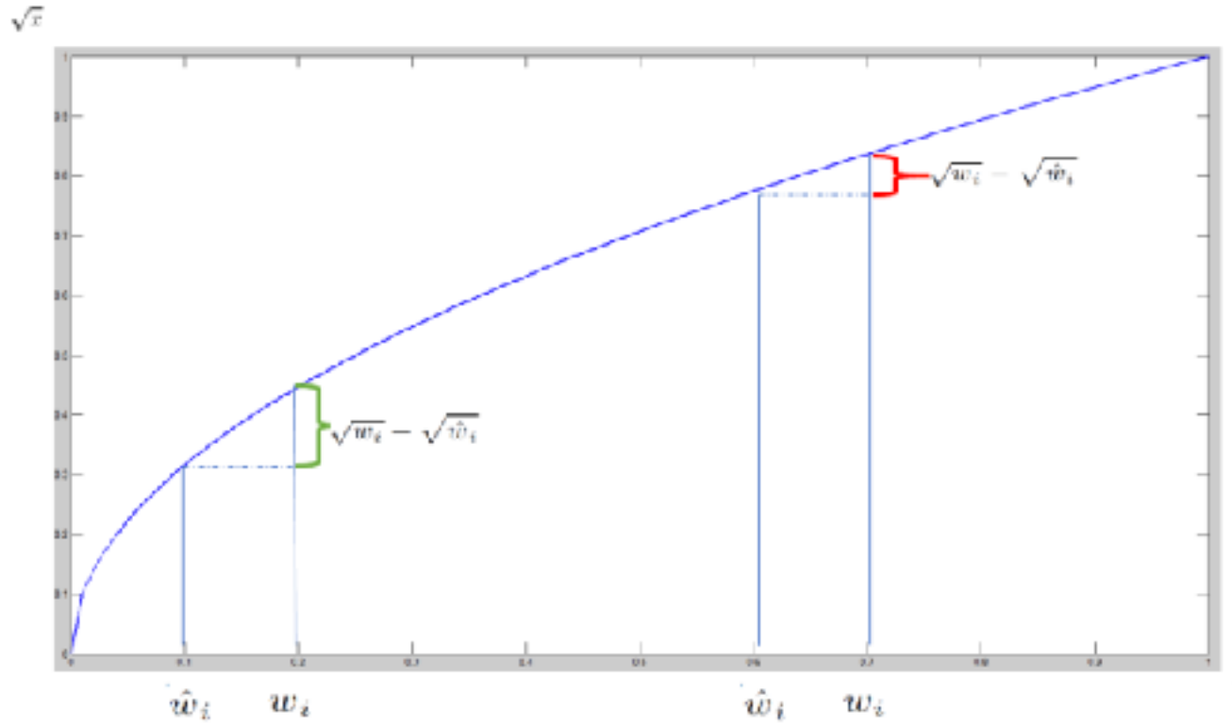
$S \times S \times (B * 5 + C)$

x, y, w, h, confidence.

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}} \quad (1)$$

YOLO(one-stage)

$$
\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2 \qquad (3)
\end{aligned}
$$

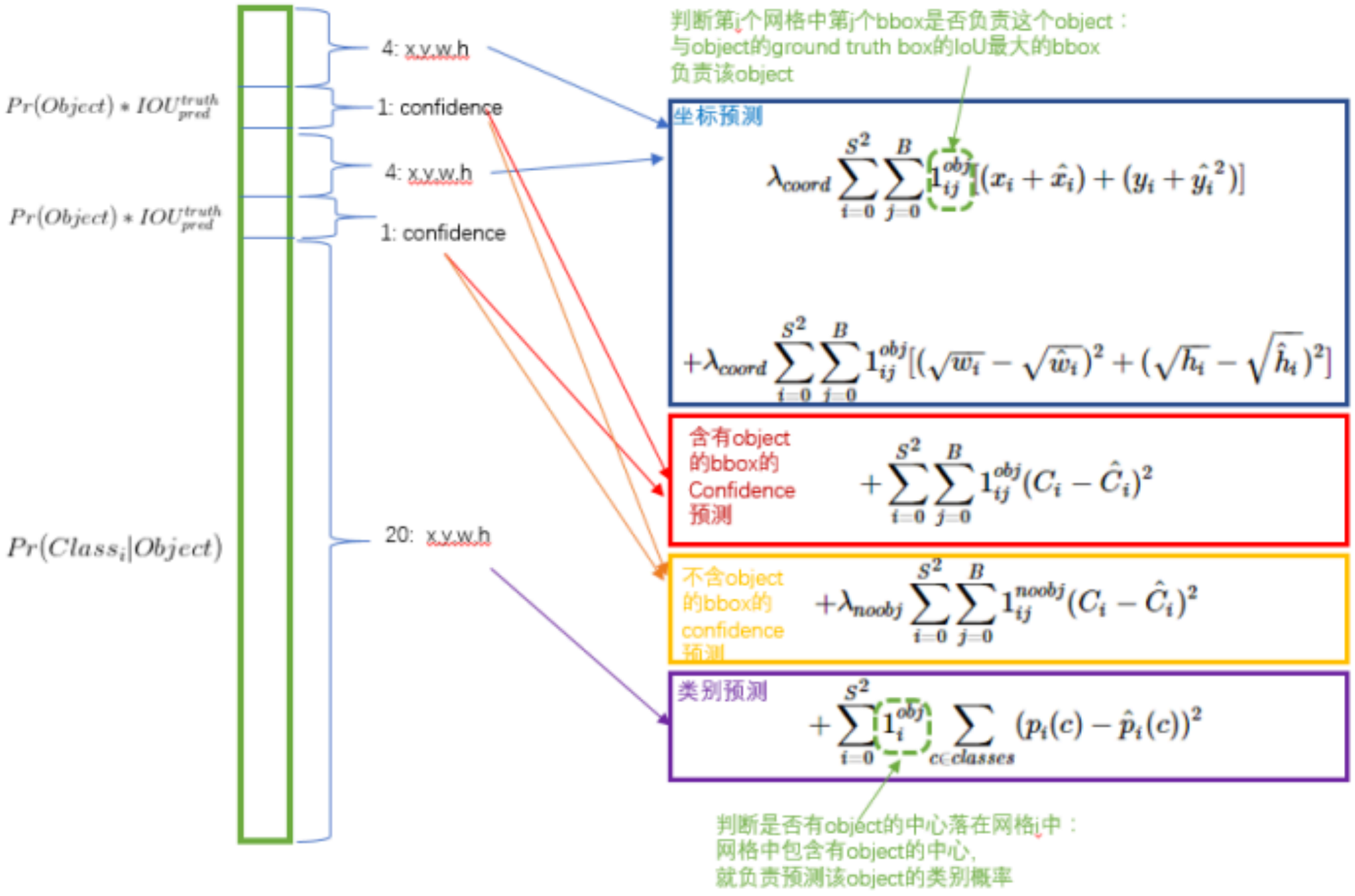


Small bbox

Big bbox

where $\mathbb{1}_i^{\text{obj}}$ denotes if object appears in cell i and $\mathbb{1}_{ij}^{\text{obj}}$ denotes that the jth bounding box predictor in cell i is "responsible" for that prediction.

YOLO(one-stage)



YOLO(one-stage)

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(one-stage)

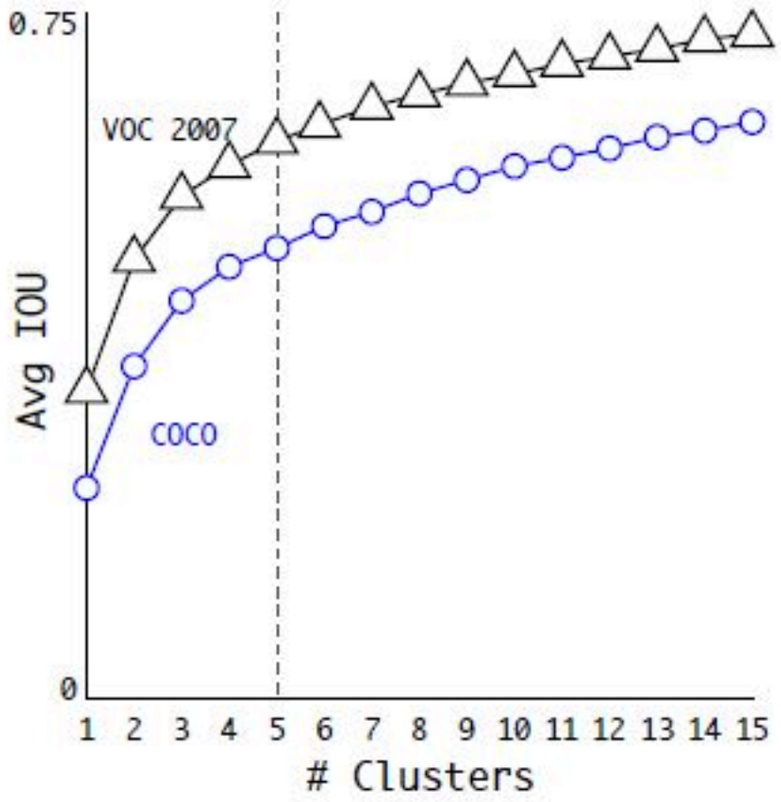
Limitations of YOLO

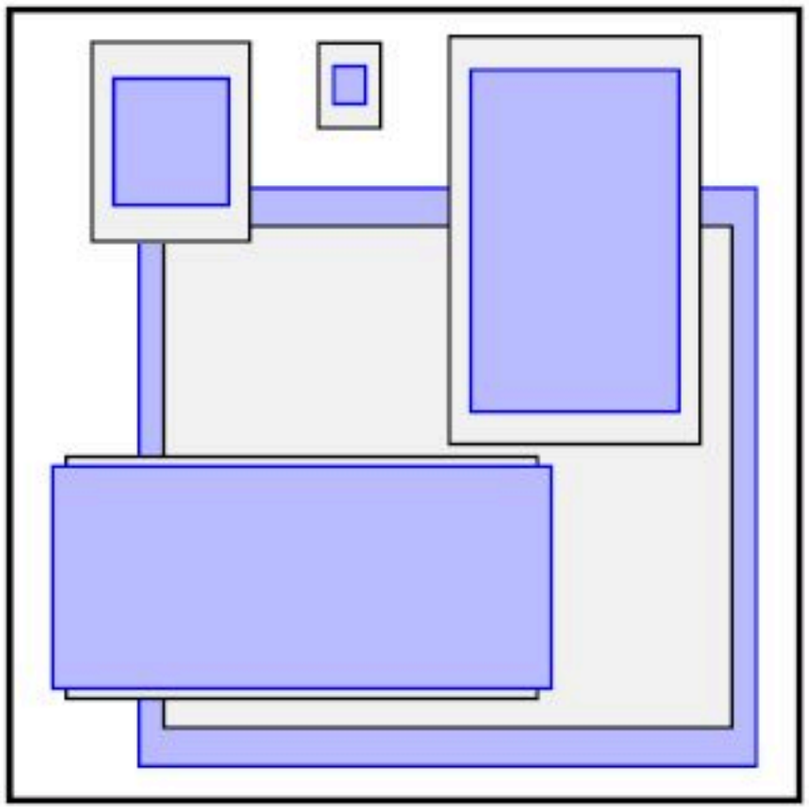
1. each grid cell only predicts two boxes and can only have one class.
2. Our model struggles with small objects that appear in groups, such as flocks of birds.

YOLOv2

	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**

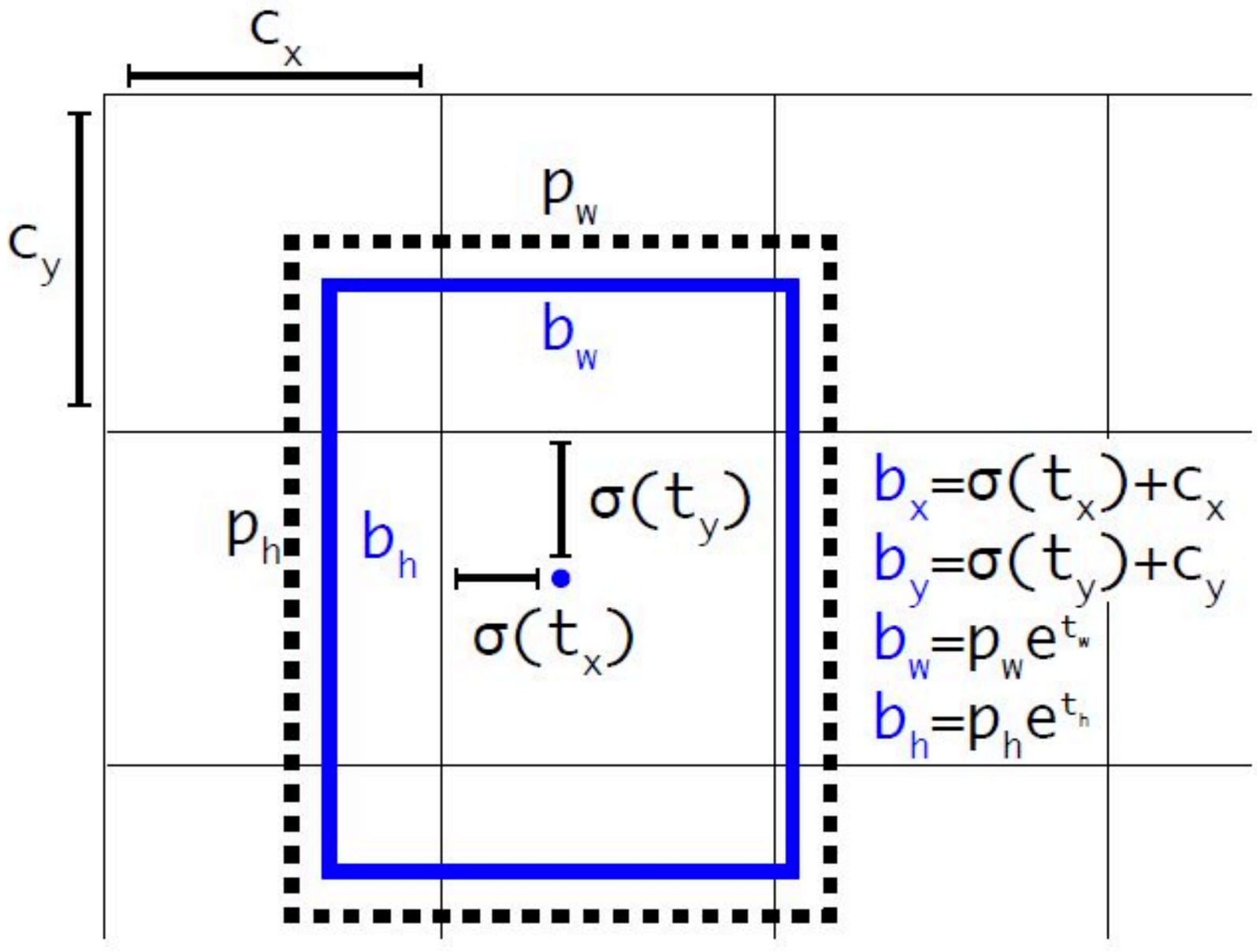
YOLOv2



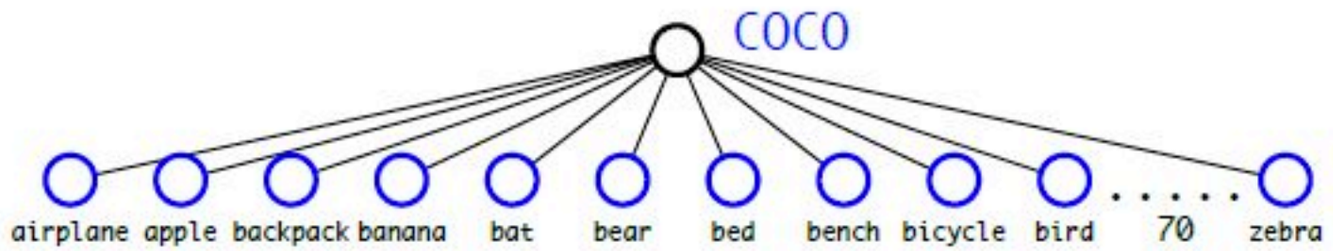


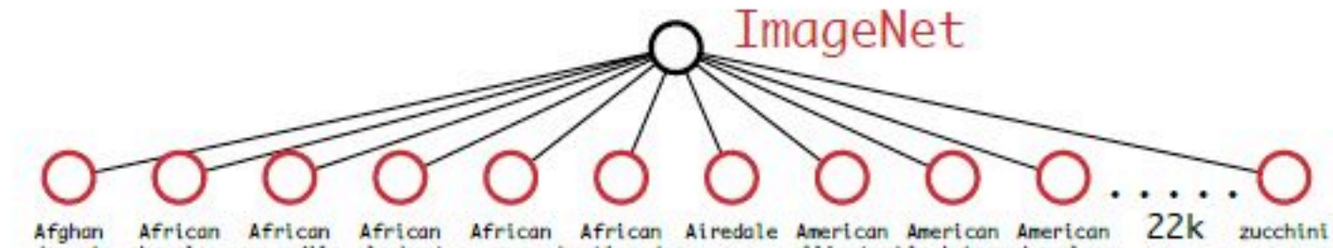
$$d(box, centroid) = 1 - IOU(box, centroid)$$

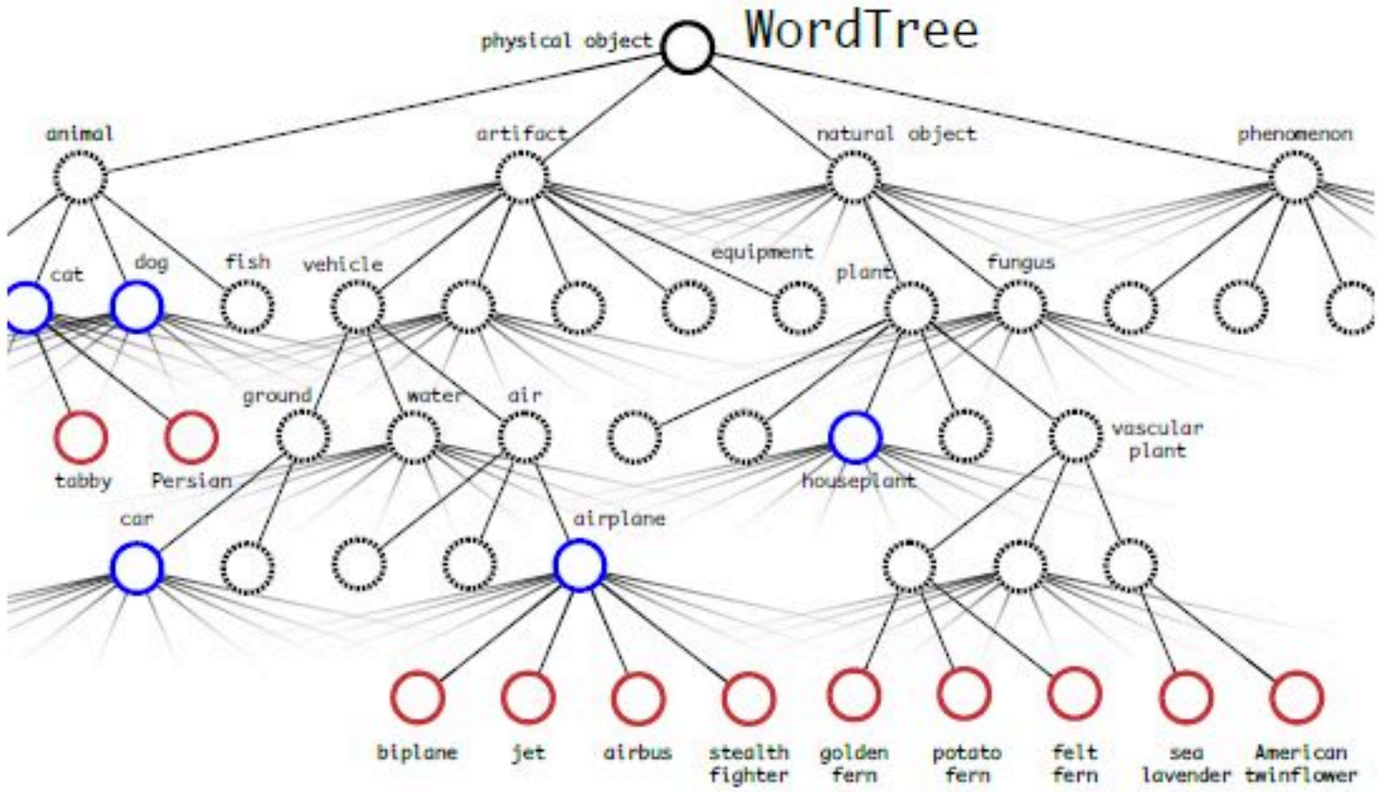
YOLOv2


c_x
c_y
p_w
p_h
b_w
b_h
$\sigma(t_y)$
$\sigma(t_x)$
$b_x=\sigma(t_x)+c_x$
$b_y=\sigma(t_y)+c_y$
$b_w=p_w e^{t_w}$
$b_h=p_h e^{t_h}$

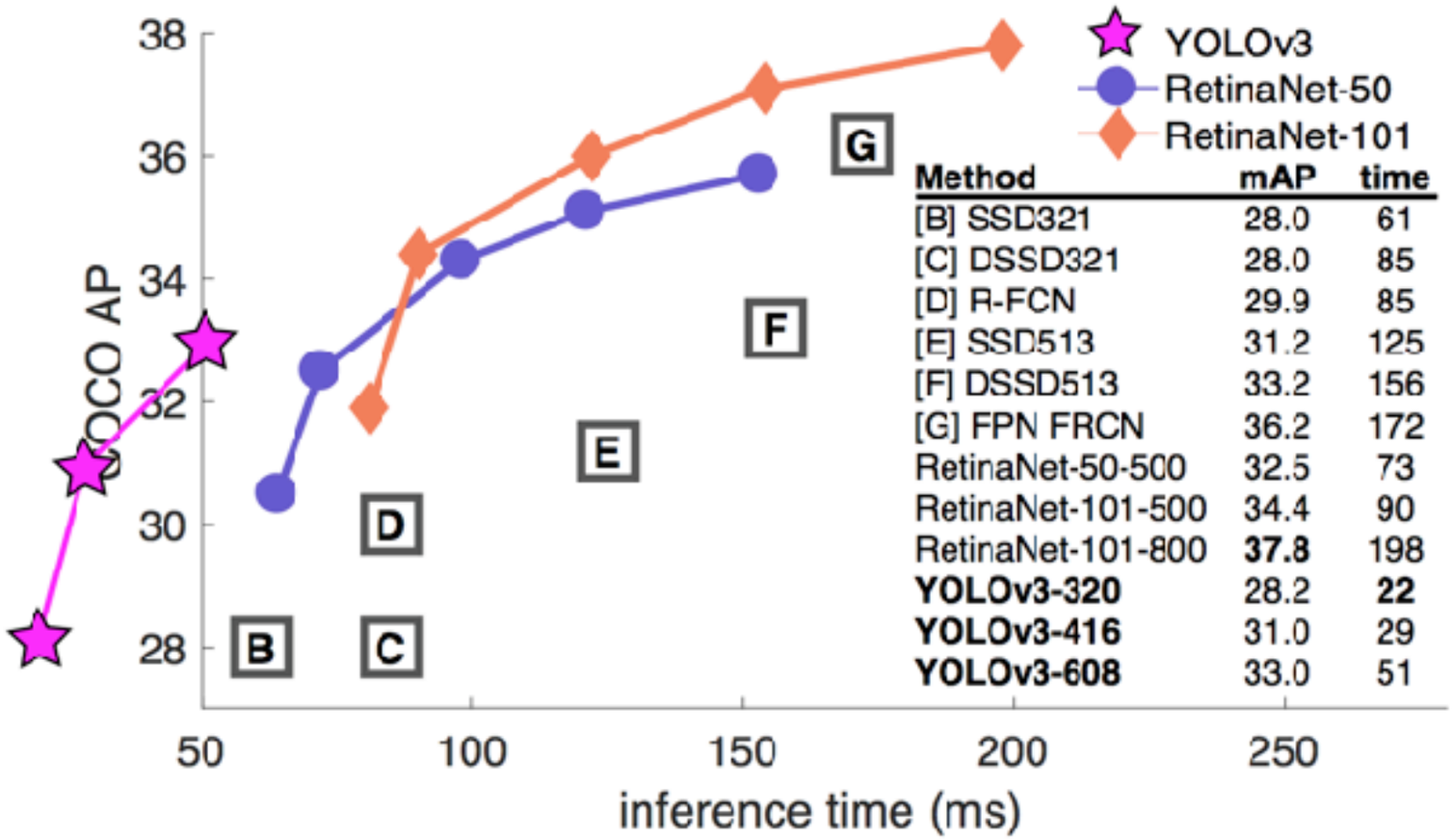
YOLO9000


COCO
airplane
apple
backpack
banana
bat
bear
bed
bench
bicycle
bird
70
zebra


ImageNet
Afghan hound
African chameleon
African crocodile
African elephant
African grey
African hunting dog
Airedale
American alligator
American black bear
American chameleon
22k
zucchini


WordTree
physical object
animal
artifact
natural object
phenomenon
cat
dog
fish
vehicle
equipment
plant
fungus
ground
water
air
vascular plant
tabby
Persian
houseplant
car
airplane
biplane
jet
airbus
stealth fighter
golden fern
potato fern
felt fern
sea lavender
American twinflower

YOLOv3



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

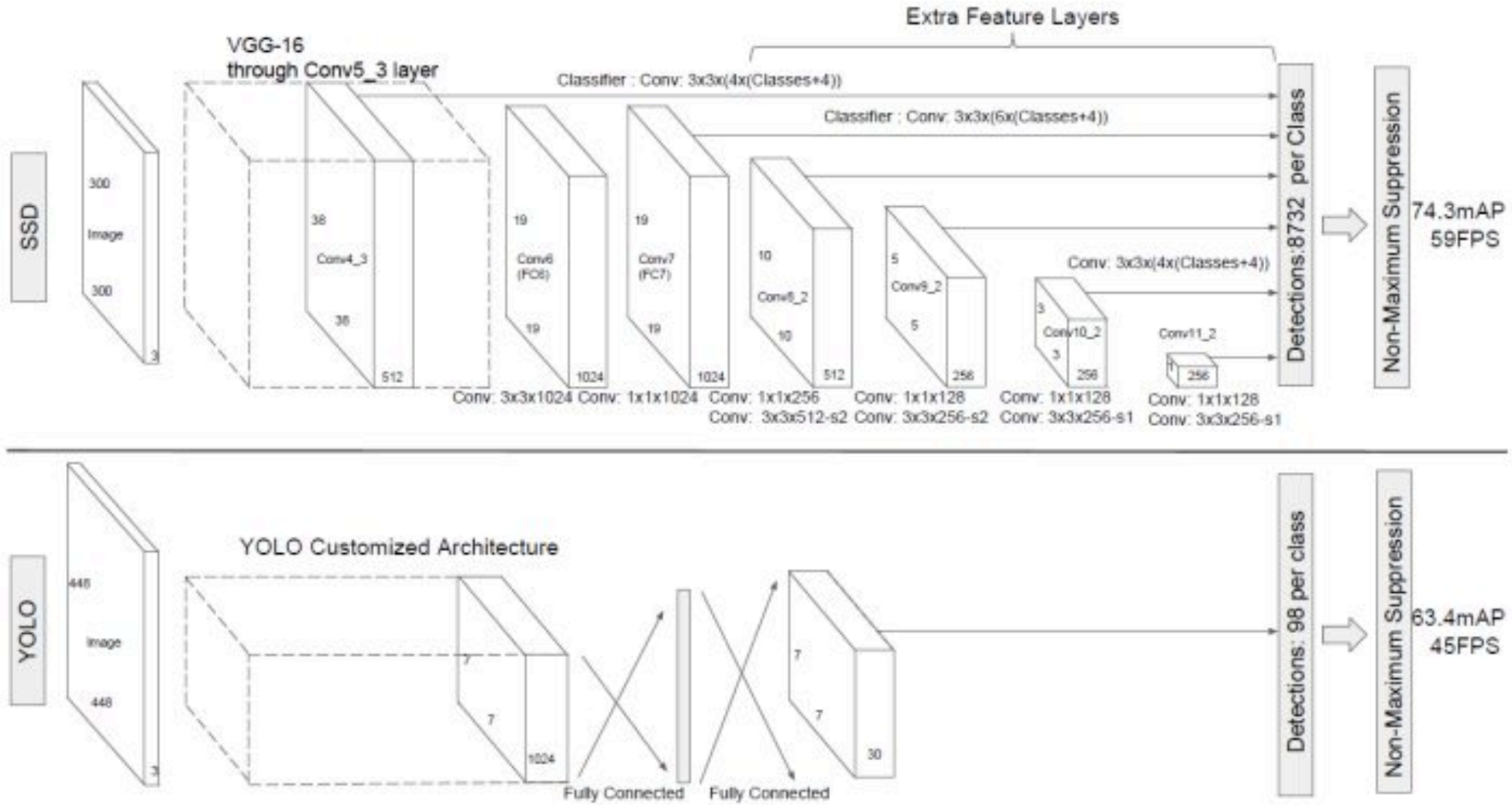
加入了残差网络

说了自己尝试了什么失败了

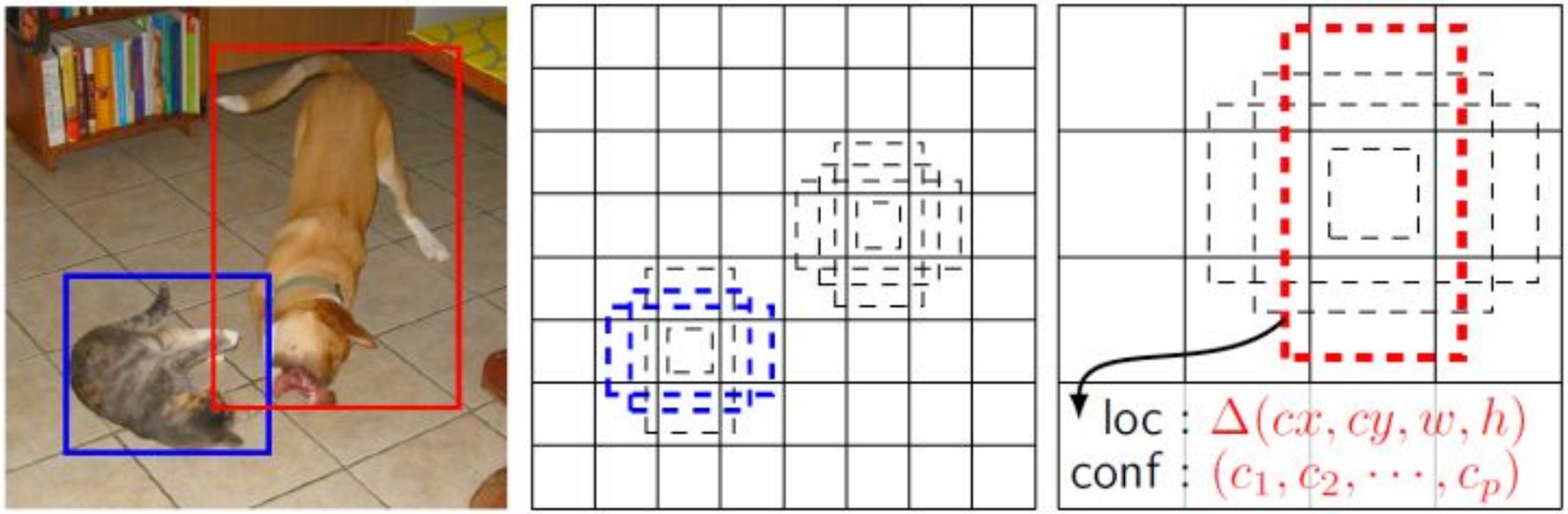
然后总结了一下自己这一年的学术实践，说自己投身到GAN的领域当中

Darknet

SSD(one-stage)

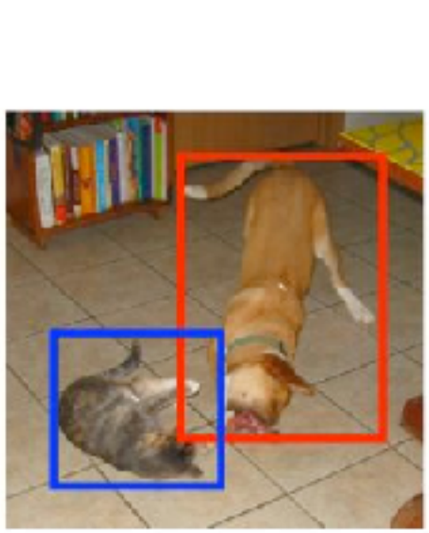


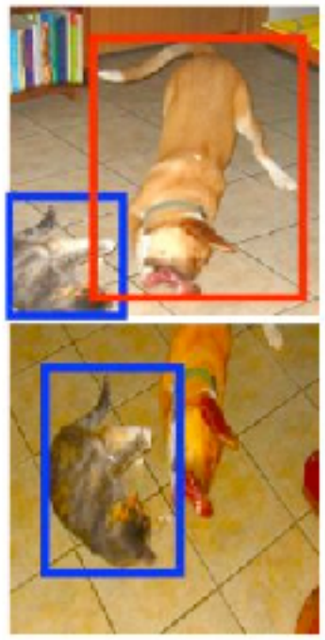
SSD(one-stage)

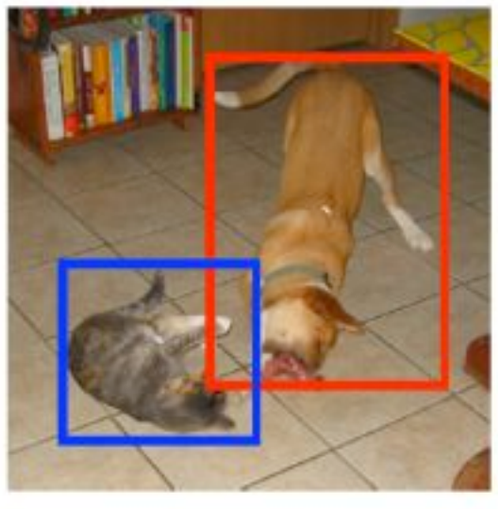


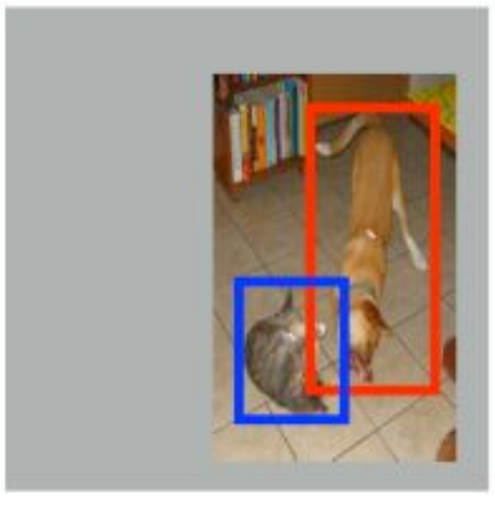
(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

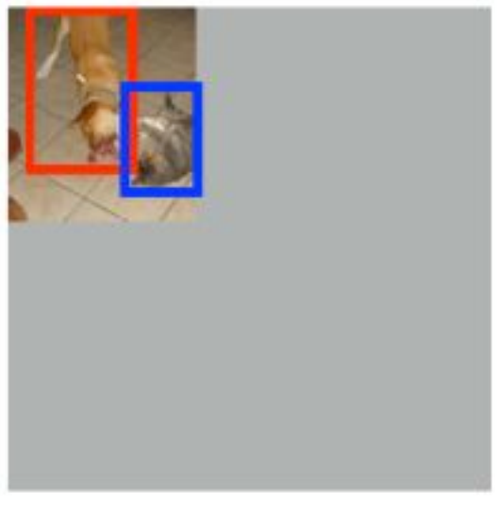
SSD(one-stage)





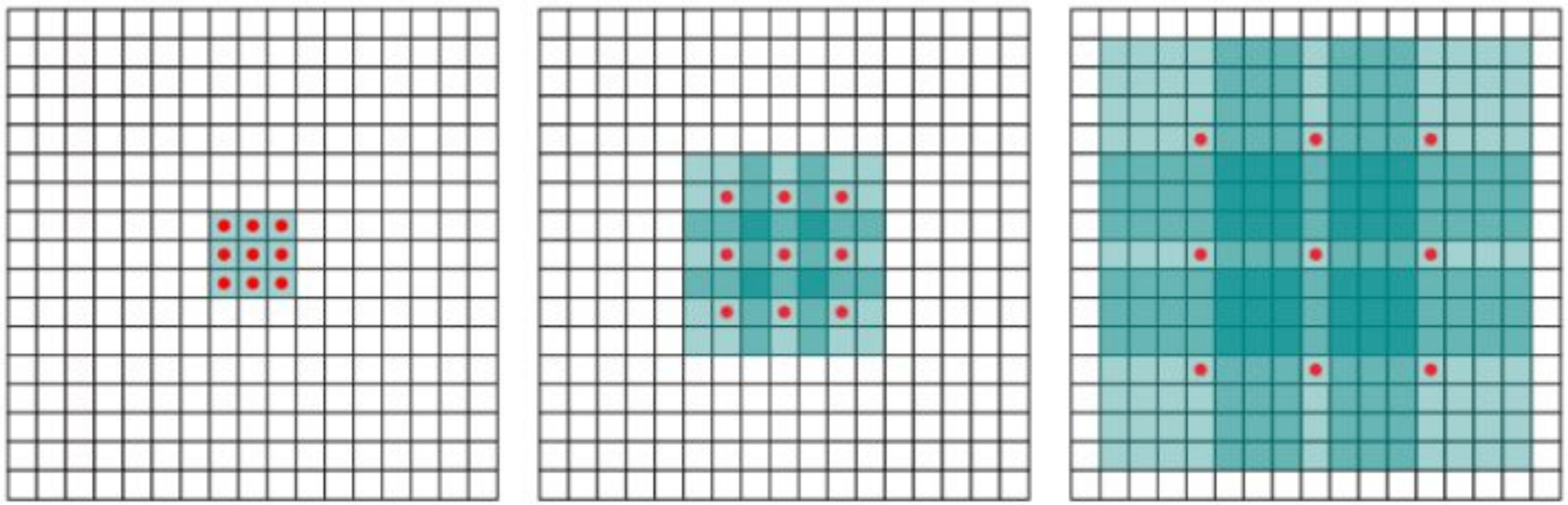






Random expansion creates more **small** training examples

SSD(one-stage)



(a) (b) (c)