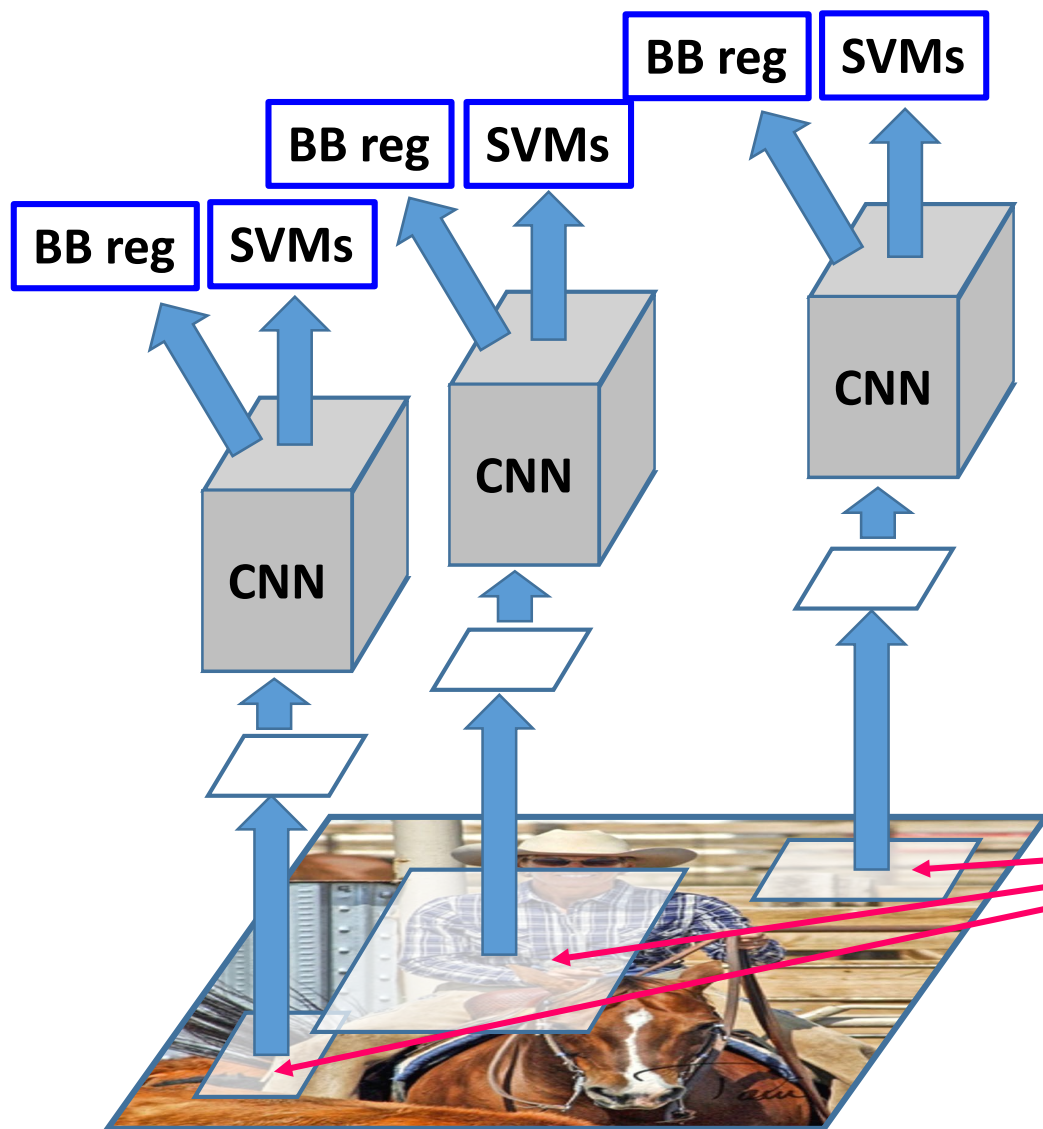


# 上节课回顾

- RCNN系列
  - RCNN
  - SPPNet
  - Fast RCNN
  - Faster RCNN
- YOLO

## 2.RCNN



先使用**SVMs**分类，再使用  
边界框回归算法精修边界框

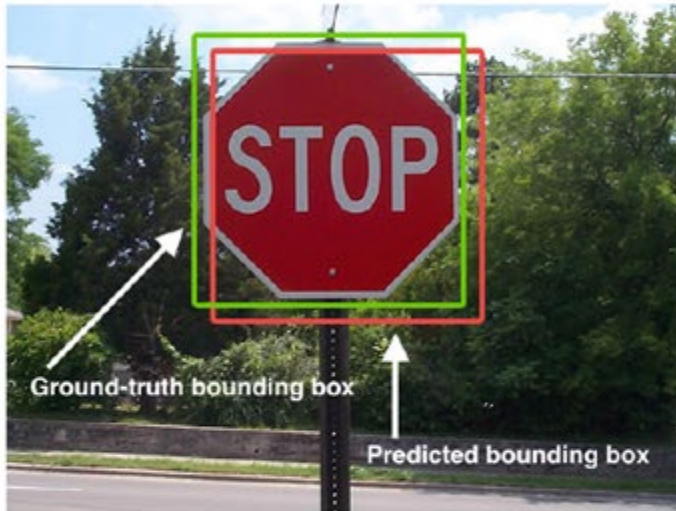
每一个图像区域输入到**CNN**中

变形图像区域

使用**ss**算法提取大  
约**2000**个推荐区域

输入图像

# IOU(Intersection over Union)

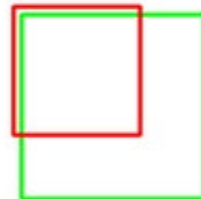


$$\text{IoU}(A,B)=\frac{|A\cap B|}{|A\cup B|}$$

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

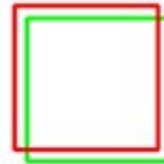


IoU: 0.4034



Poor

IoU: 0.7330



Good

IoU: 0.9264



Excellent

# Non-Maximum Suppression(NMS)

---

## NMS的伪代码

**Input :**  $\mathcal{B} = \{b_1, \dots, b_N\}$ ,  $\mathcal{S} = \{s_1, \dots, s_N\}$ ,  $N_t$

$\mathcal{B}$  is the list of initial detection bounding-boxes

$\mathcal{S}$  contains corresponding detection confidence

$N_t$  is the NMS threshold

**Begin**

$\mathcal{D} \leftarrow \{\}$

**while**  $\mathcal{B} \neq \text{empty}$  **do**

$m \leftarrow \operatorname{argmax} \mathcal{S}$

$\mathcal{M} \leftarrow b_m$

$\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{M}; \mathcal{B} \leftarrow \mathcal{B} - \mathcal{M}$

**for**  $b_i$  **in**  $\mathcal{B}$  **do**

**if**  $\operatorname{iou}(\mathcal{M}, b_i) \geq N_t$  **then**

$\mathcal{B} \leftarrow \mathcal{B} - b_i; \mathcal{S} \leftarrow \mathcal{S} - s_i$

**end**

**end**

**end**

**return**  $\mathcal{D}, \mathcal{S}$

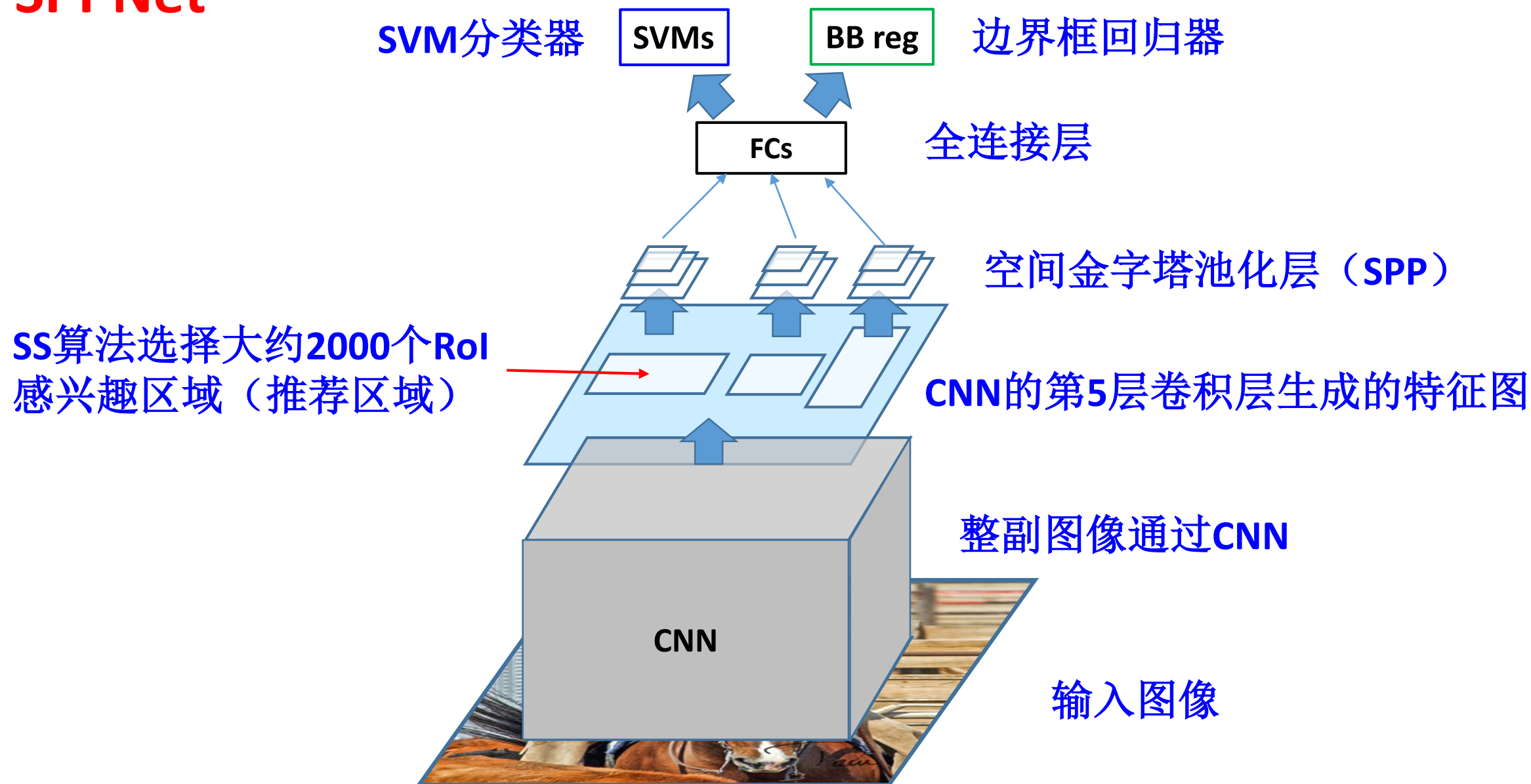
**end**

D为最终留下（输出）的bounding box

m为当前confidence最高的bounding box的编号

M为当前confidence最高的bounding box

# SPPNet



# 3.Fast RCNN

Softmax分类器

Log loss + smooth L1 loss

多任务损失函数

softmax

BB reg

边界框回归器

FC

FC

全连接层

FCs

RoI池化层

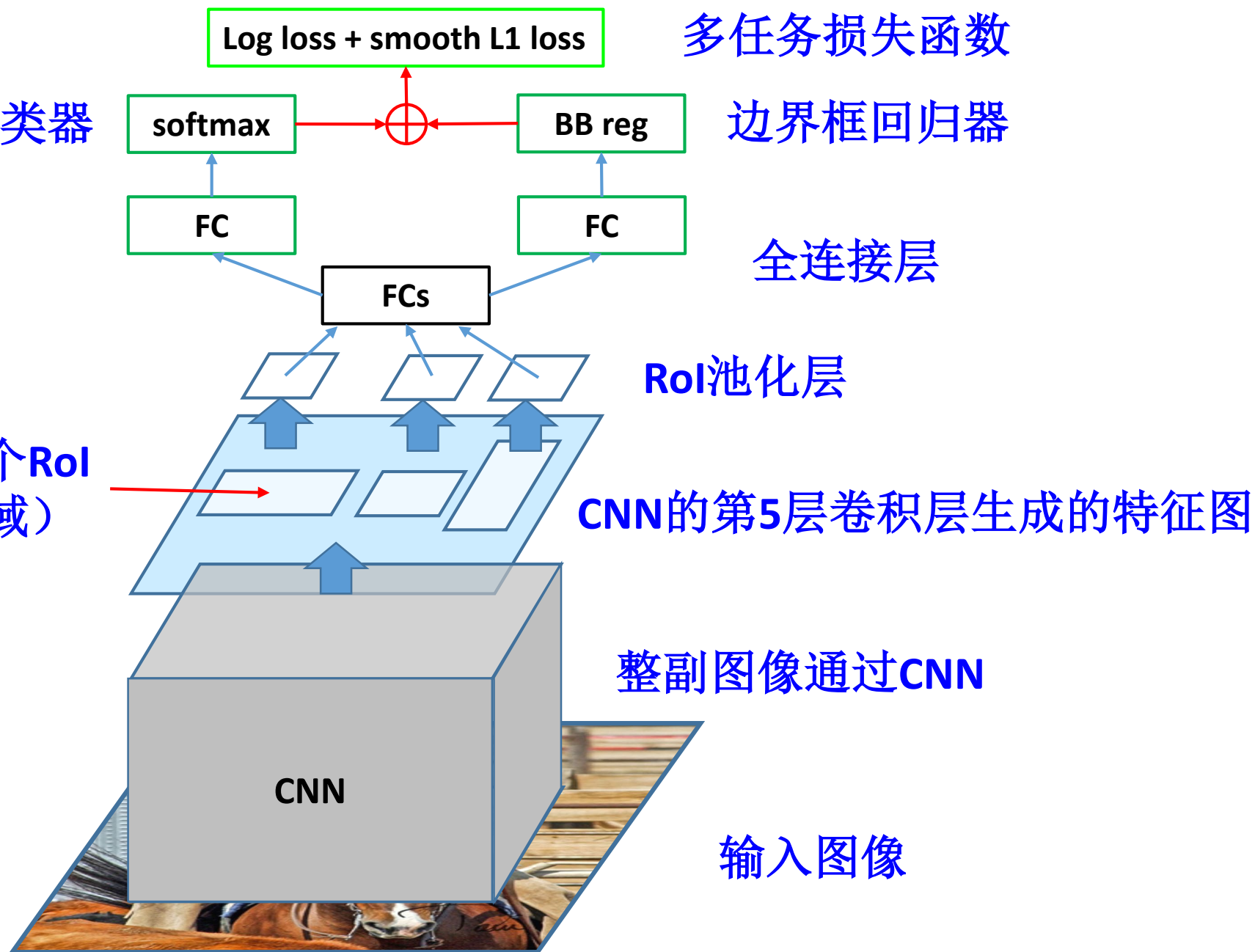
SS算法选择大约2000个RoI  
感兴趣区域（推荐区域）

CNN的第5层卷积层生成的特征图

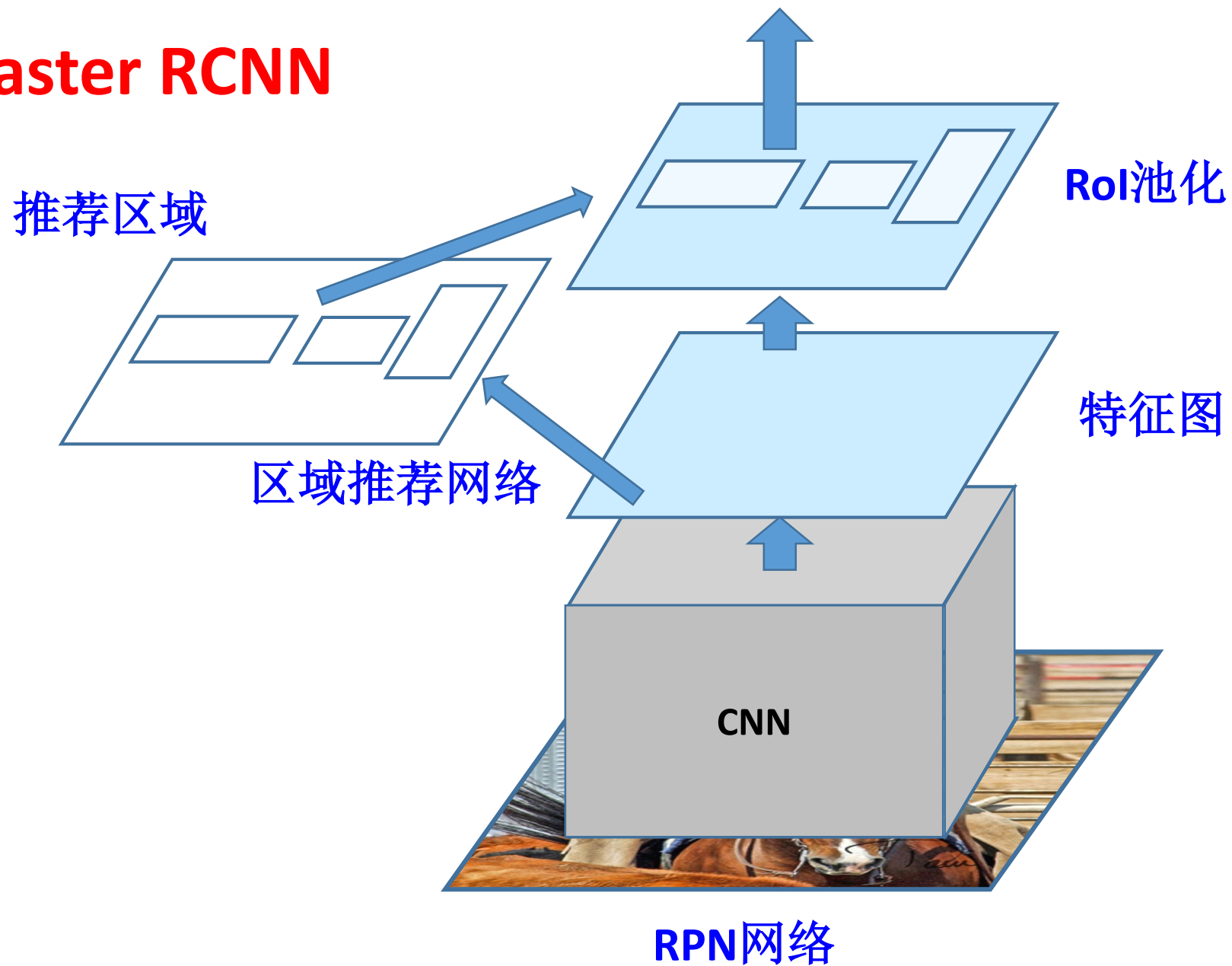
整副图像通过CNN

CNN

输入图像



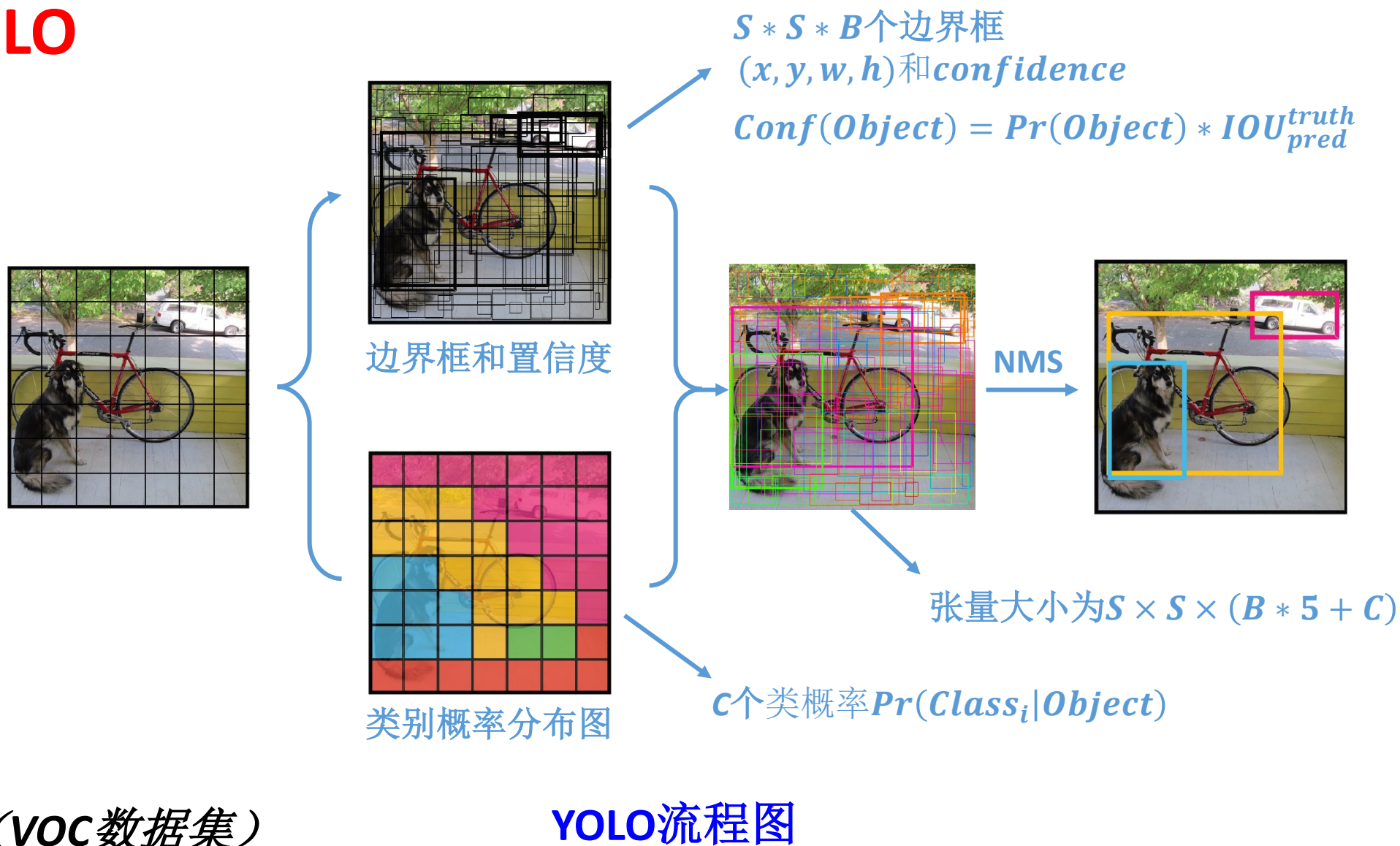
## 4.Faster RCNN



Ren S., He K., Girshick R., Sun J. (2017) Faster RCNN: Towards real time object detection with region proposal networks.  
IEEE TPAMI.

# 5.YOLO

$S = 7$   
 $B = 2$   
 $C = 20$  (VOC数据集)





# 5.YOLO

$$\text{loss} = \sum_{i=0}^{S^2} \text{coordError} + \text{iouError} + \text{classError}$$

判断第i个网格中的  
第j个box是否负责这  
个object

1

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[ \left( \sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left( \sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

坐标误差

2

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2$$

含object的box的置信度预测

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2$$

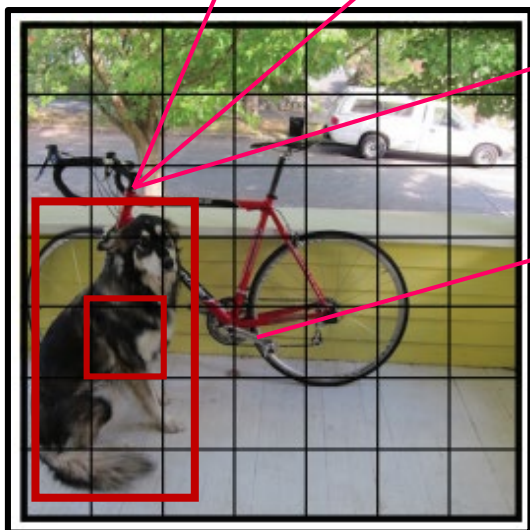
不含object的box的置信度预测

IOU误差  
分类误差

3

$$+ \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

判断是否有object中心落在网格i中



## 5.YOLO

	PASCAL2007 mAP	Speed FPS
RCNN	66.0	0.05
Fast RCNN	70.0	0.5
Faster RCNN	73.2	7
YOLO	63.4	45

实验结果

# YOLO的不足

- YOLO 对相互靠的很近的物体，还有很小的群体检测效果不好，这是因为一个网格中只预测了两个框，并且只属于一类。
- 由于损失函数的问题，定位误差是影响检测效果的主要原因。尤其是小物体的处理上，还有待加强

# YOLO v2

- YOLOv2: 速度要快过其他检测系统（FasterRCNN，SSD），可以在速度与精度之间进行权衡。
- YOLO9000: 可以实时地检测超过 9000 种物体分类，归功于 WordTree，通过 WordTree 来混合检测数据集与识别数据集之中的数据。
- 对小物体的检测效果仍不理想。

YOLO9000: Better, Faster, Stronger

# YOLO v3

- 多尺度预测
  - 三种不同尺度的特征图上进行预测任务
  - 提升小物体检测准确度
- 更好的基础分类网络（类ResNet）darknet-53
  - 53个卷积层，没有使用池化层