



TensorFlow



bilibili : 霹雳吧啦WZ

PYTORCH

深度学习-目标检测篇

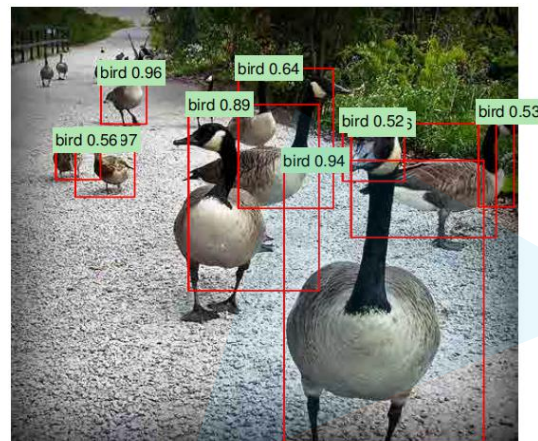
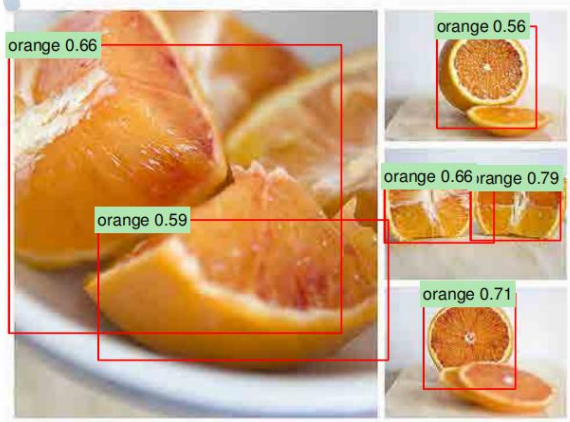
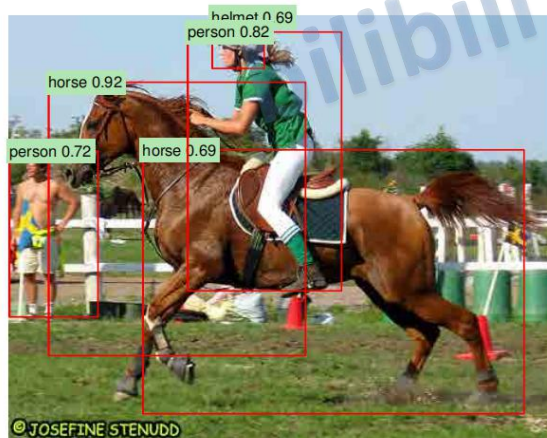
bilibili : 霹雳吧啦Wz

作者：神秘的wz

Fast R-CNN

原论文名称 Fast R-CNN

Fast R-CNN是作者Ross Girshick继R-CNN后的又一力作。同样使用VGG16作为网络的backbone，与R-CNN相比训练时间快9倍，测试推理时间快213倍，准确率从62%提升至66%(再Pascal VOC数据集上)。

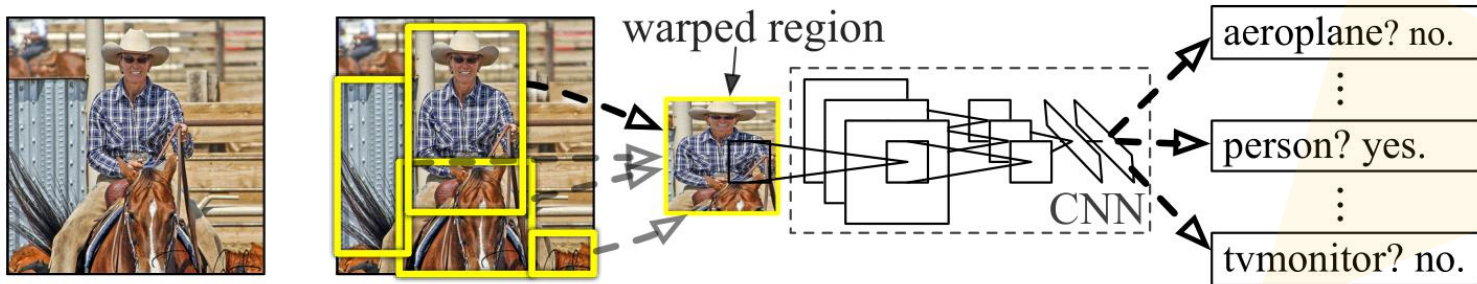


Fast R-CNN

RCNN算法流程可分为4个步骤

- 一张图像生成1K~2K个**候选区域**(使用Selective Search方法)
- 对每个候选区域，使用深度网络**提取特征**
- 特征送入每一类的**SVM 分类器**，判别是否属于该类
- 使用回归器**精细修正**候选框位置

R-CNN: *Regions with CNN features*

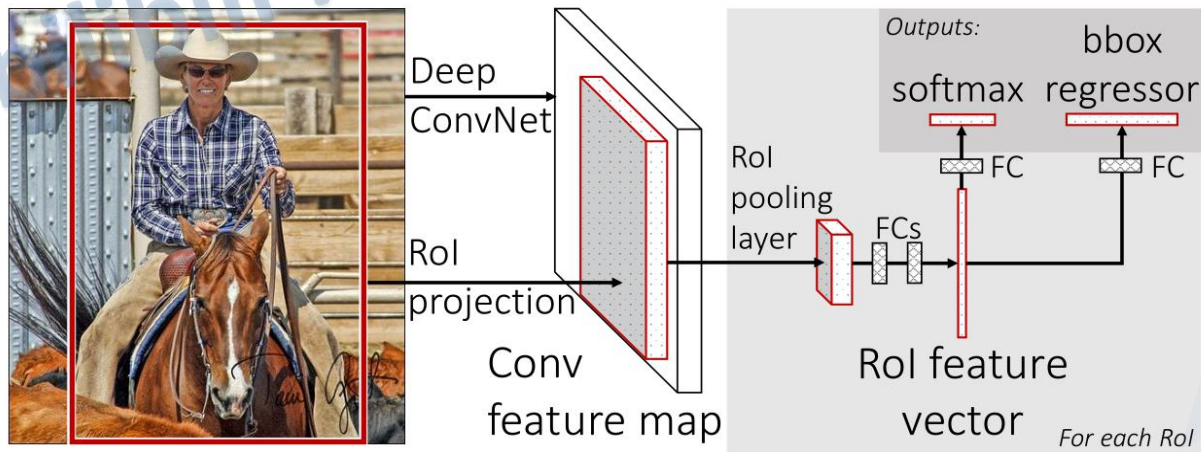


Fast R-CNN

Fast R-CNN算法流程可分为3个步骤

- 一张图像生成1K~2K个候选区域(使用Selective Search方法)
- 将图像输入网络得到相应的特征图, 将SS算法生成的候选框投影到特征图上获得相应的特征矩阵
- 将每个特征矩阵通过ROI pooling层缩放到7x7大小的特征图, 接着将特征图展平通过一系列全连接层得到预测结果

Region of Interest

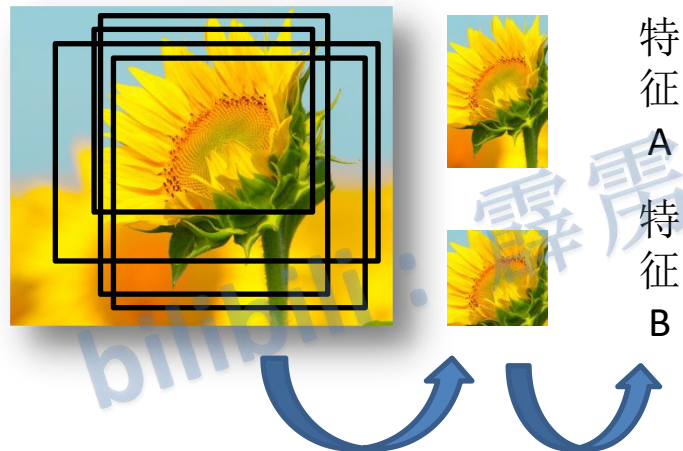


Fast R-CNN

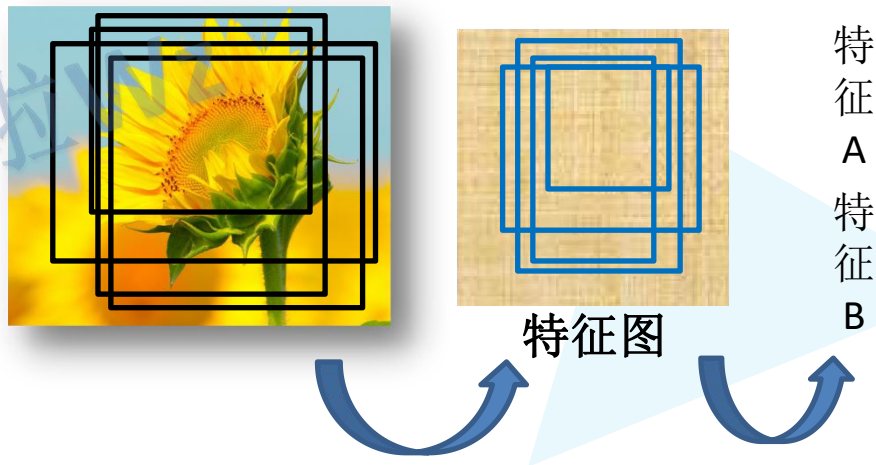
一次性计算整张图像特征

不限制输入
图像的尺寸

R-CNN依次将候选框区域输入卷积神经网络得到特征。



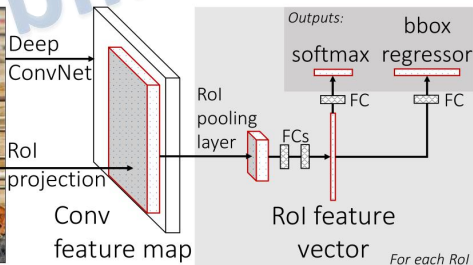
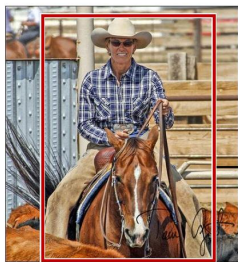
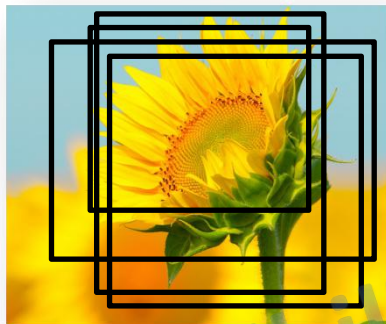
Fast-RCNN将整张图像送入网络，紧接着从特征图像上提取相应的候选区域。这些候选区域的特征不需要再重复计算。



Fast R-CNN

我们将训练数据分为正负样本是为了网络能够分辨不同的事物

训练数据的采样
(正样本, 负样本)

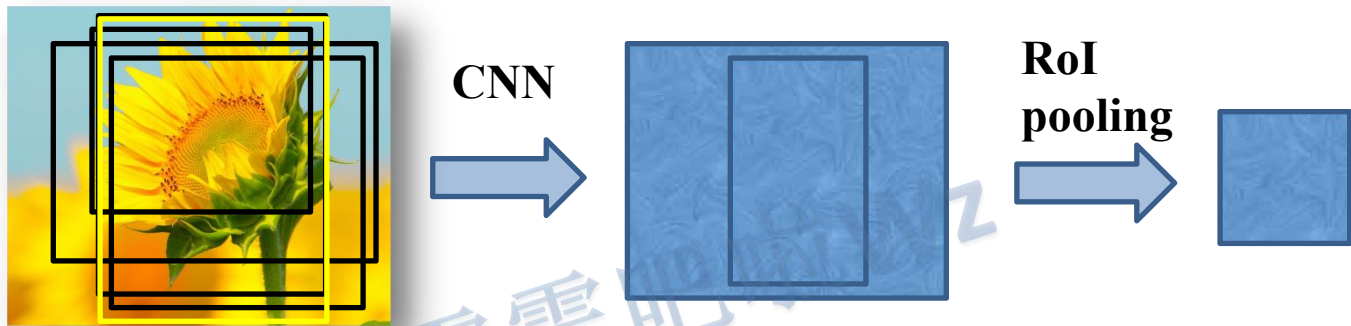


Mini-batch sampling. During fine-tuning, each SGD mini-batch is constructed from $N = 2$ images, chosen uniformly at random (as is common practice, we actually iterate over permutations of the dataset). We use mini-batches of size $R = 128$, sampling 64 RoIs from each image. As in [9], we take 25% of the RoIs from object proposals that have intersection over union (IoU) overlap with a ground-truth bounding box of at least 0.5. These RoIs comprise the examples labeled with a foreground object class, *i.e.* $u \geq 1$. The remaining RoIs are sampled from object proposals that have a maximum IoU with ground truth in the interval $[0.1, 0.5)$, following [11]. These are the background examples and are labeled with $u = 0$. The lower threshold of 0.1 appears to act as a heuristic for hard example mining [8]. During training, images are horizontally flipped with probability 0.5. No other data augmentation is used.

Fast R-CNN

RoI Pooling Layer

不限制输入
图像的尺寸



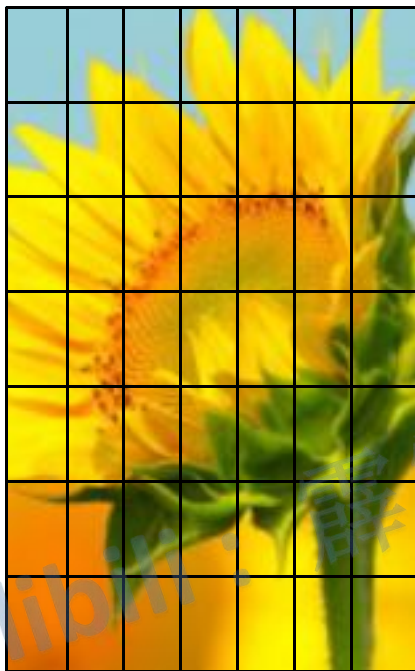
bilibili: 霹雳吧

Fast R-CNN

我们将此特征图分成 $7 \times 7 = 49$ 个特征块，对每个特征块进行maxpooling操作，这样我们就可以得到 7×7 大小的特征图

RoI Pooling Layer

不限制输入
图像的尺寸



0.1	0.1	0.2	0	0.3	0.2	0.3
0.3	0.4	0.5	0.2	0.1	0.1	0
0.4	0.5	0.2	0.1	0.1	0.2	0.1
0.2	0	0.3	0.2	0.3	0.2	0.1
0.2	0.1	0.4	0.5	0.2	0.1	0.1
0.3	0.2	0	0.3	0.2	0.2	0.1
0.3	0.4	0.5	0.2	0.1	0.1	0.1

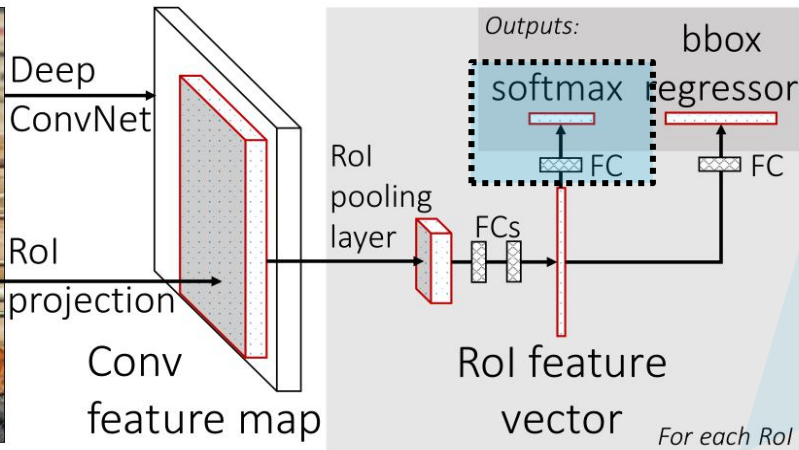
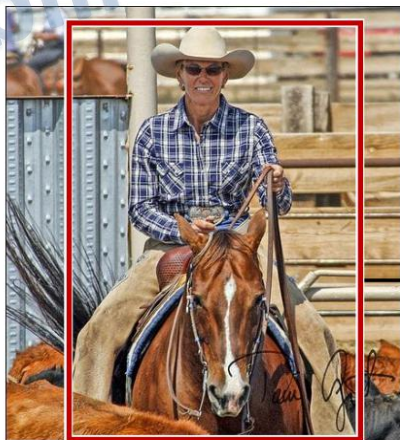
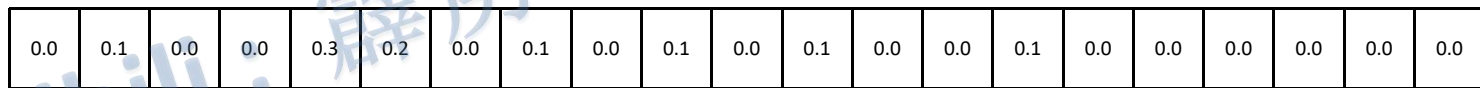
注意：这里忽略了深度channel

Fast R-CNN

分类器

输出 $N+1$ 个类别的概率（ N 为检测目标的种类, 1为背景）共 $N+1$ 个节点

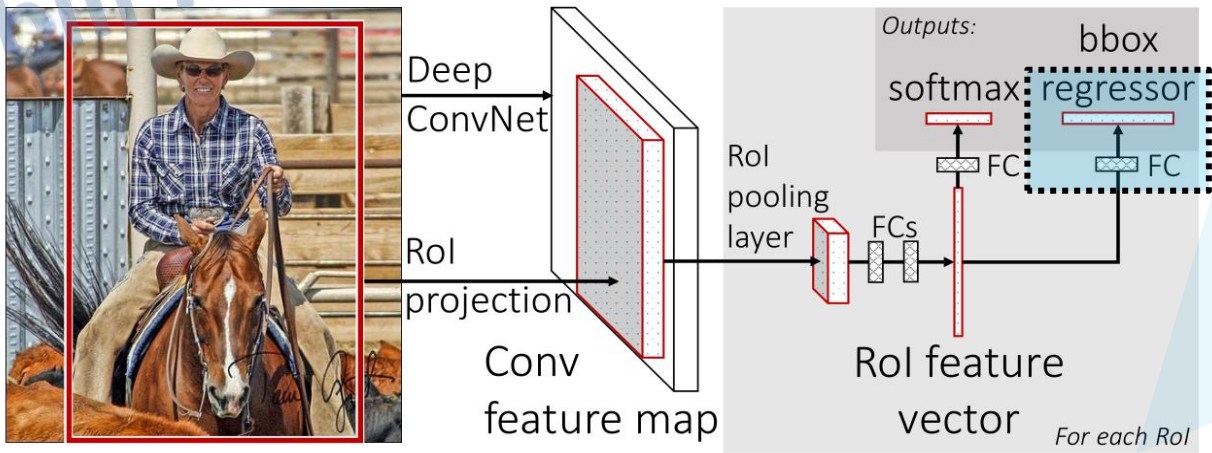
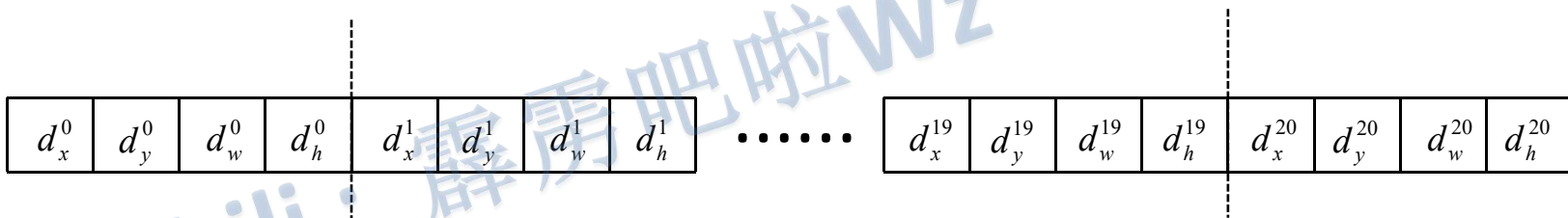
背景概率



Fast R-CNN

边界框回归器

输出对应 $N+1$ 个类别的候选边界框回归参数(d_x, d_y, d_w, d_h),共 $(N+1) \times 4$ 个节点



Fast R-CNN

边界框回归器

输出对应 $N+1$ 个类别的候选边界框回归参数 (d_x, d_y, d_w, d_h) ,共 $(N+1) \times 4$ 个节点

$$\hat{G}_x = P_w d_x(P) + P_x$$

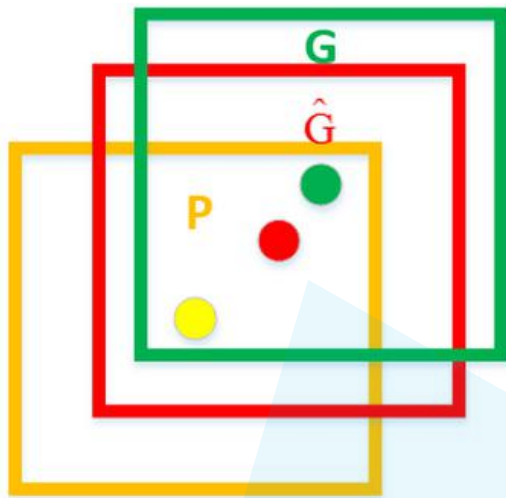
$$\hat{G}_y = P_h d_y(P) + P_y$$

$$\hat{G}_w = P_w \exp(d_w(P))$$

$$\hat{G}_h = P_h \exp(d_h(P))$$

P_x, P_y, P_w, P_h 分别为候选框的中心 x, y 坐标, 以及宽高

$\hat{G}_x, \hat{G}_y, \hat{G}_w, \hat{G}_h$ 分别为最终预测的边界框中心 x, y 坐标, 以及宽高



Fast R-CNN

Multi-task loss

分类损失

边界框回归损失

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v)$$

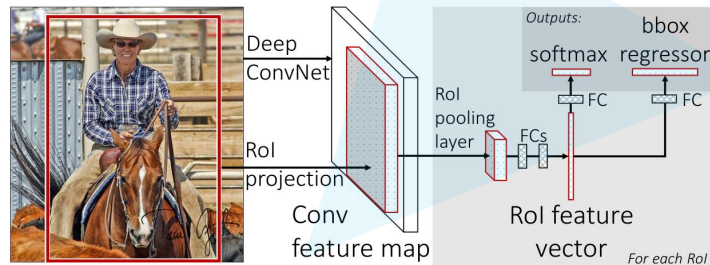
p 是分类器预测的softmax概率分布 $p = (p_0, \dots, p_k)$

u 对应目标真实类别标签

t^u 对应边界框回归器预测的对应类别 u 的回归参数 $(t_x^u, t_y^u, t_w^u, t_h^u)$

v 对应真实目标的边界框回归参数 (v_x, v_y, v_w, v_h)

$v_x = (G_x - P_x) / P_w$ ，其它也是一样反向计算



Fast R-CNN

Multi-task loss

分类损失

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v)$$

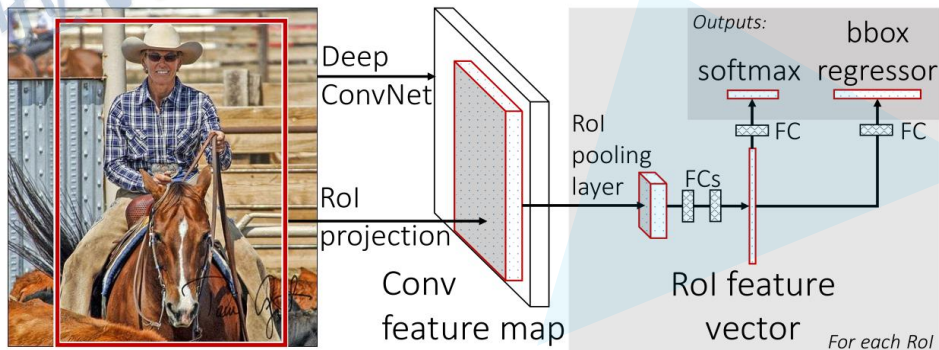
p 是分类器预测的softmax概率分布 $p = (p_0, \dots, p_k)$

u 对应目标真实类别标签

交叉熵损失

分类损失

$$L_{cls}(p, u) = -\log p_u$$



误差的计算

Cross Entropy Loss 交叉熵损失

1. 针对多分类问题（softmax输出，所有输出概率和为1）

$$H = -\sum_i o_i^* \log(o_i)$$

2. 针对二分类问题（sigmoid输出，每个输出节点之间互不相干）

$$H = -\frac{1}{N} \sum_{i=1}^N [o_i^* \log o_i + (1 - o_i^*) \log(1 - o_i)]$$

其中 o_i^* 为真实标签值， o_i 为预测值，默认log以e为底等于ln

Fast R-CNN

Multi-task loss

边界框回归损失

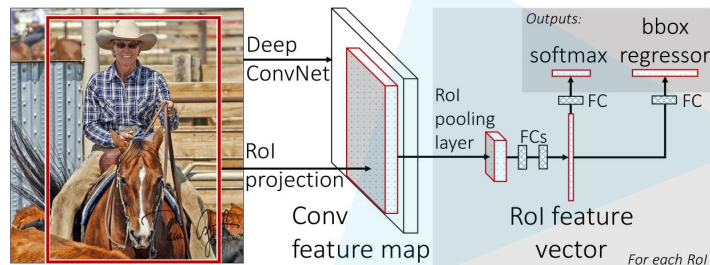
$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v)$$

$[u \geq 1]$ 是艾弗森括号

t^u 对应边界框回归器预测的对应类别 u 的回归参数($t_x^u, t_y^u, t_w^u, t_h^u$)

v 对应真实目标的边界框回归参数(v_x, v_y, v_w, v_h)

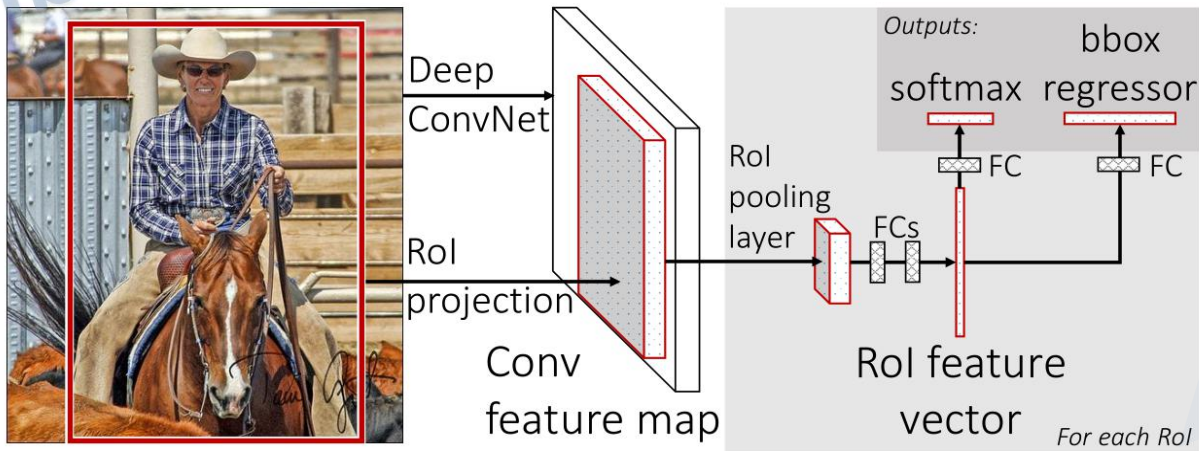
$$L_{loc}(t^u, v) = \sum_{i \in \{x, y, w, h\}} smooth_{L_1}(t_i^u - v_i)$$
$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$



Fast R-CNN

Fast R-CNN算法流程可分为3个步骤

- 一张图像生成1K~2K个候选区域(使用Selective Search方法)
- 将图像输入网络得到相应的特征图，将SS算法生成的候选框投影到特征图上获得相应的特征矩阵
- 将每个特征矩阵通过ROI pooling层缩放到7x7大小的特征图，接着将特征图展平通过一系列全连接层得到预测结果



Fast R-CNN

Fast R-CNN框架

Region proposal(Selective Search)
Feature extraction Classification Bounding-box regression (CNN)

bilibili : 霹雳吧啦WZ

Fast R-CNN

R-CNN框架

Region proposal(Selective Search)	
Feature extraction(CNN)	
Classification (SVM)	Bounding-box regression (regression)

沟通方式

1.github

<https://github.com/WZMIAOMIAO/deep-learning-for-image-processing>

2.CSDN

https://blog.csdn.net/qq_37541097/article/details/103482003

3.bilibili

<https://space.bilibili.com/18161609/channel/index>

尽可能每周更新