



TensorFlow



bilibili : 霹雳吧啦WZ

PYTORCH

# 深度学习-目标检测篇

bilibili : 霹雳吧啦Wz

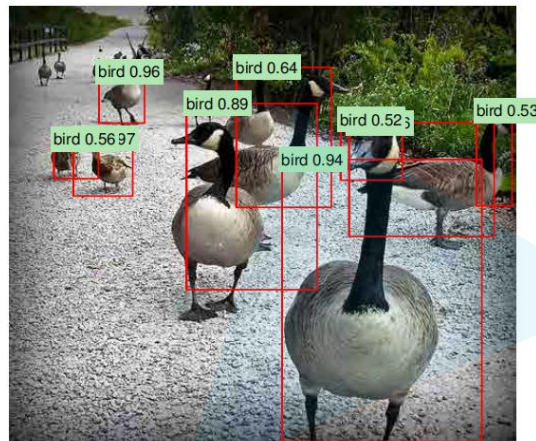
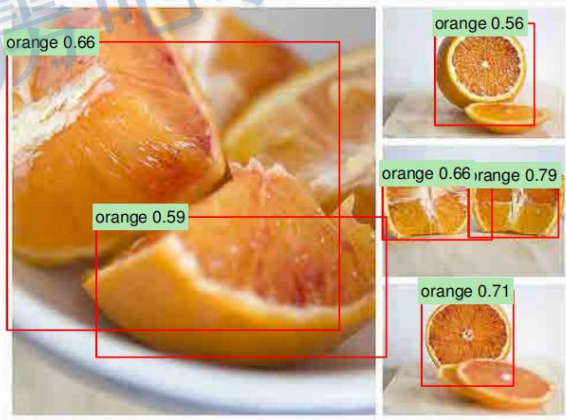
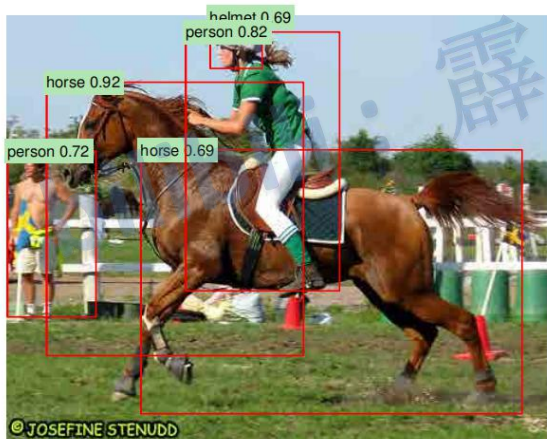
作者：神秘的wz

# R-CNN (Region with CNN feature)

## 原论文名称

## Rich feature hierarchies for accurate object detection and semantic segmentation

R-CNN可以说是利用深度学习进行目标检测的开山之作。作者Ross Girshick多次在PASCAL VOC的目标检测竞赛中折桂，曾在2010年带领团队获得终身成就奖。



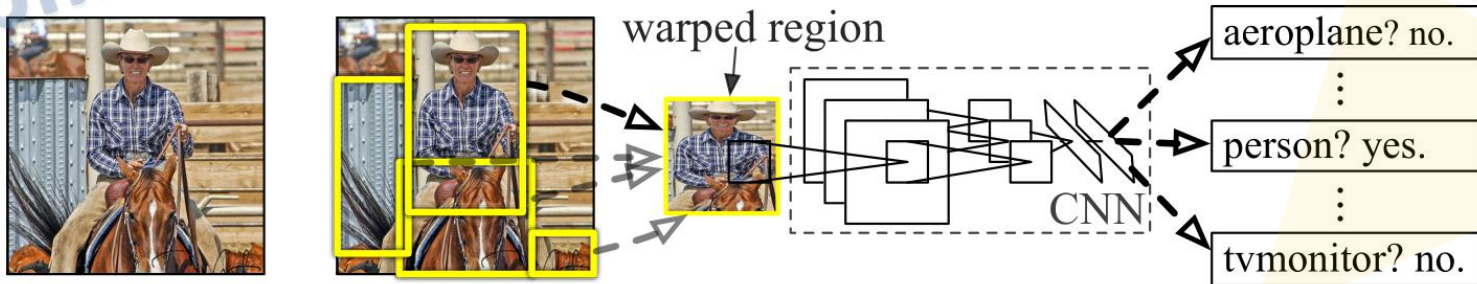
# R-CNN (Region with CNN feature)

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

输入网络之前, 需要 Warp 调整尺寸

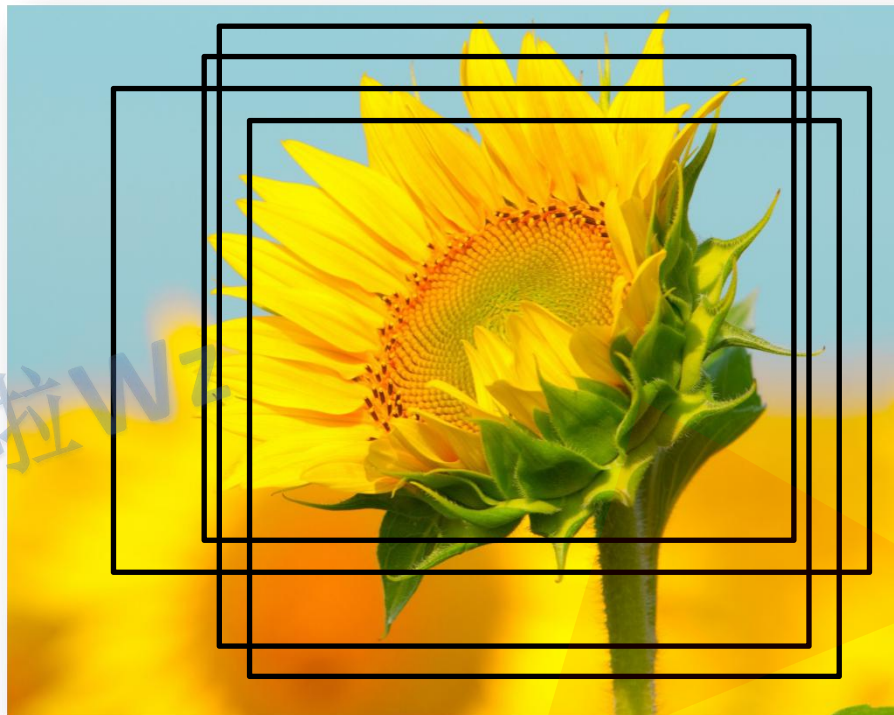
## R-CNN: *Regions with CNN features*



# R-CNN (Region with CNN feature)

## 1. 候选区域的生成

利用Selective Search算法通过图像分割的方法得到一些原始区域，然后使用一些合并策略将这些区域合并，得到一个层次化的区域结构，而这些结构就包含着可能需要的物体。



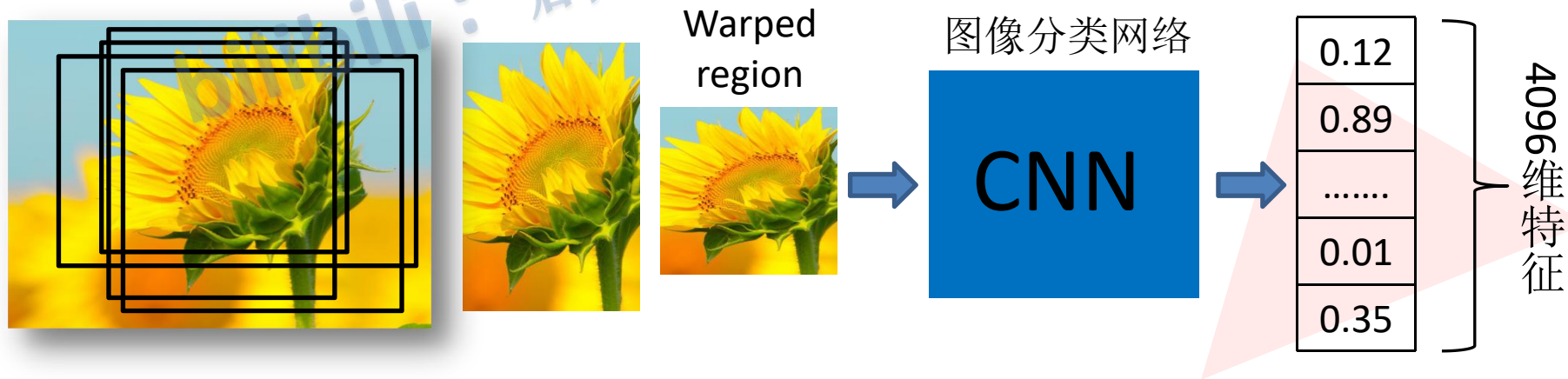
bilibili: 霹雳吧啦WZ



# R-CNN (Region with CNN feature)

## 2. 对每个候选区域，使用深度网络提取特征

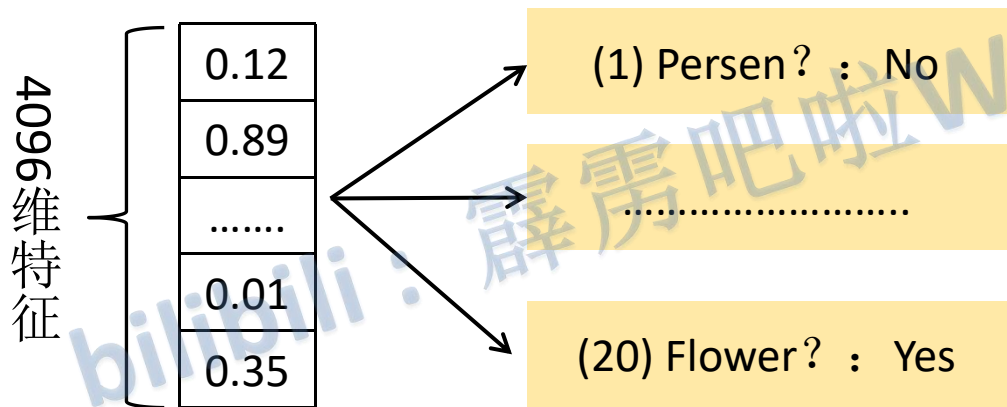
将2000候选区域缩放到227x227pixel，接着将候选区域输入事先训练好的AlexNet CNN网络获取4096维的特征得到 $2000 \times 4096$ 维矩阵。



# R-CNN (Region with CNN feature)

## 3. 特征送入每一类的SVM分类器，判定类别

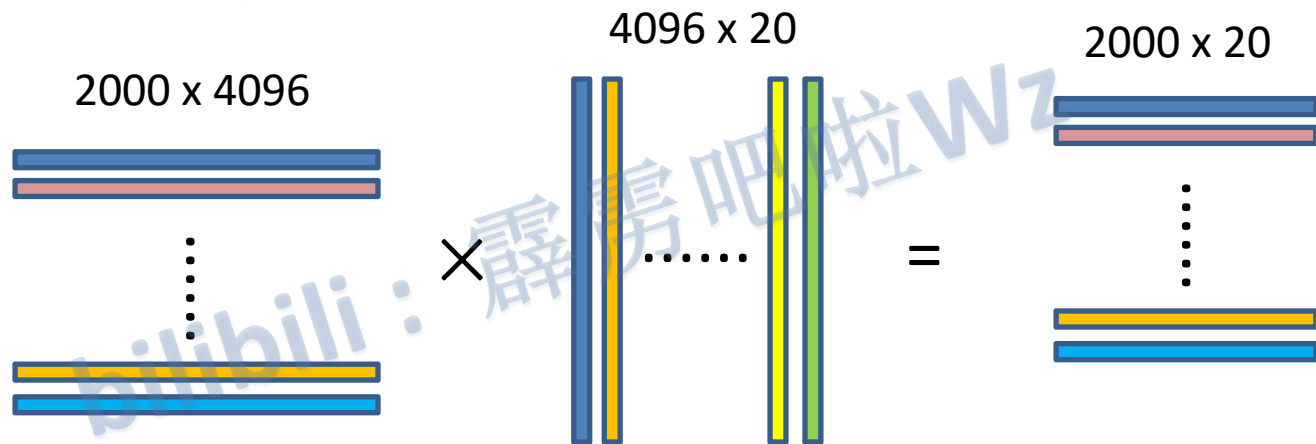
将 $2000 \times 4096$ 维特征与20个SVM组成的权值矩阵 $4096 \times 20$ 相乘，获得 $2000 \times 20$ 维矩阵表示每个建议框是某个目标类别的得分。分别对上述 $2000 \times 20$ 维矩阵中每一列即每一类进行非极大值抑制剔除重叠建议框，得到该列即该类中得分最高的一些建议框。



# R-CNN (Region with CNN feature)

## 3. 特征送入每一类的SVM分类器，判定类别

将 $2000 \times 4096$ 的特征矩阵与20个SVM组成的权值矩阵 $4096 \times 20$ 相乘，获得 $2000 \times 20$ 的概率矩阵，每一行代表一个建议框归于每个目标类别的概率。分别对上述 $2000 \times 20$ 维矩阵中每一列即每一类进行非极大值抑制剔除重叠建议框，得到该列即该类中得分最高的一些建议框。



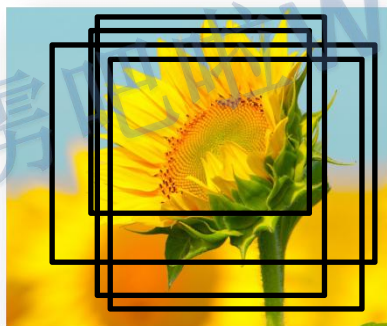
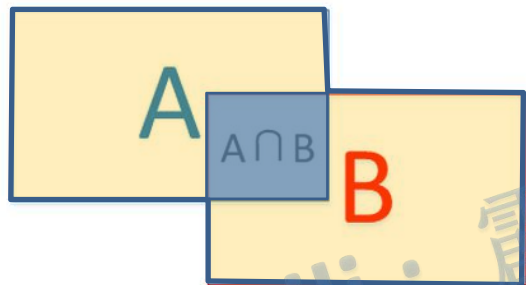


# R-CNN (Region with CNN feature)

非极大值抑制剔除重叠建议框

NMS

IoU (Intersection over Union)  
表示  $(A \cap B) / (A \cup B)$



A: 0.98



B: 0.86

通过IoU计算  
为同一物体。  
删除B保留A

寻找得分最高的目标

计算其他目标与该目标的iou值

删除所有iou值大于给定阈值的目标

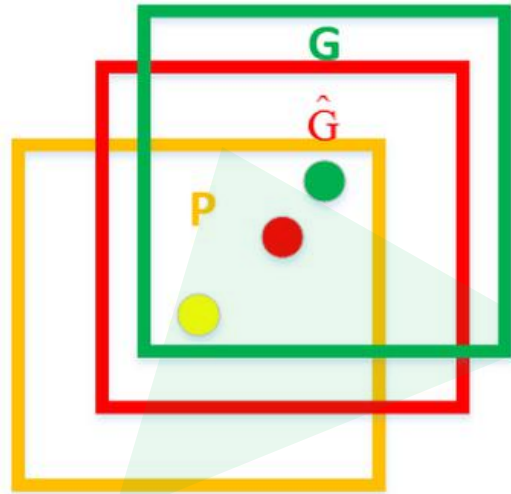
# R-CNN (Region with CNN feature)

## 4. 使用回归器精细修正候选框位置

对NMS处理后剩余的候选框进一步筛选。接着分别用20个回归器对上述20个类别中剩余的候选框进行回归操作，最终得到每个类别的修正后的得分最高的 bounding box。

如图，黄色框口P表示建议框Region Proposal，绿色窗口G表示实际框Ground Truth，红色窗口  $\hat{G}$  表示Region Proposal进行回归后的预测窗口，可以用最小二乘法解决的线性回归问题。

依旧针对CNN输出的特征向量进行预测

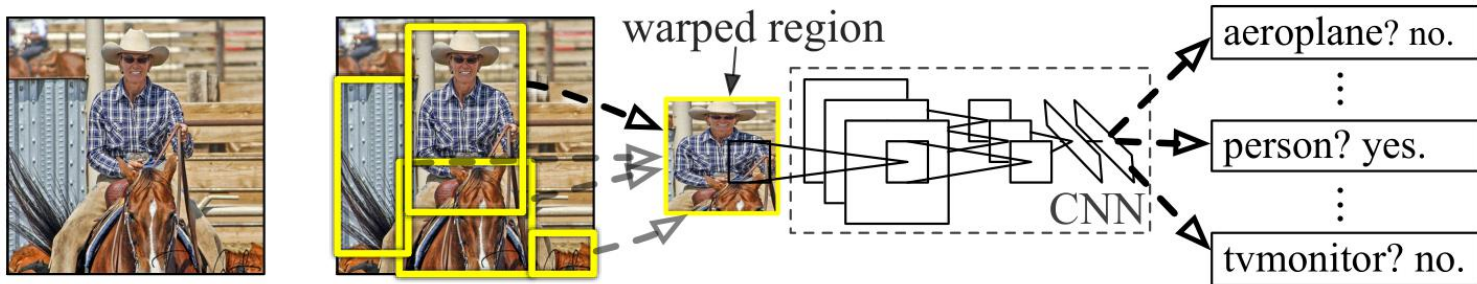


# R-CNN (Region with CNN feature)

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
- 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*



# R-CNN (Region with CNN feature)

## R-CNN框架

<b>Region proposal(Selective Search)</b>	
<b>Feature extraction(CNN)</b>	
<b>Classification (SVM)</b>	<b>Bounding-box regression (regression)</b>

# R-CNN (Region with CNN feature)

## R-CNN存在的问题:

### 1.测试速度慢:

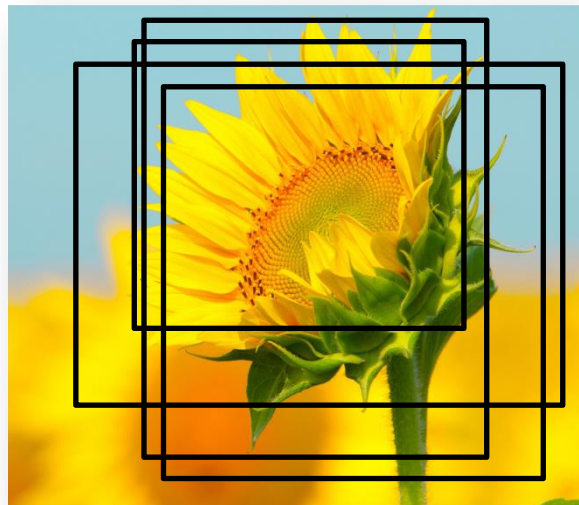
测试一张图片约53s(CPU)。用Selective Search算法提取候选框用时约2秒，一张图像内候选框之间存在大量重叠，提取特征操作冗余。

### 2.训练速度慢:

过程及其繁琐

### 3.训练所需空间大:

对于SVM和bbox回归训练，需要从每个图像中的每个目标候选框提取特征，并写入磁盘。对于非常深的网络，如VGG16，从VOC07训练集上的5k图像上提取的特征需要数百GB的存储空间。



# 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](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