

## 深度学习框架Caffe学习与应用 第7课

# 本节课内容

---

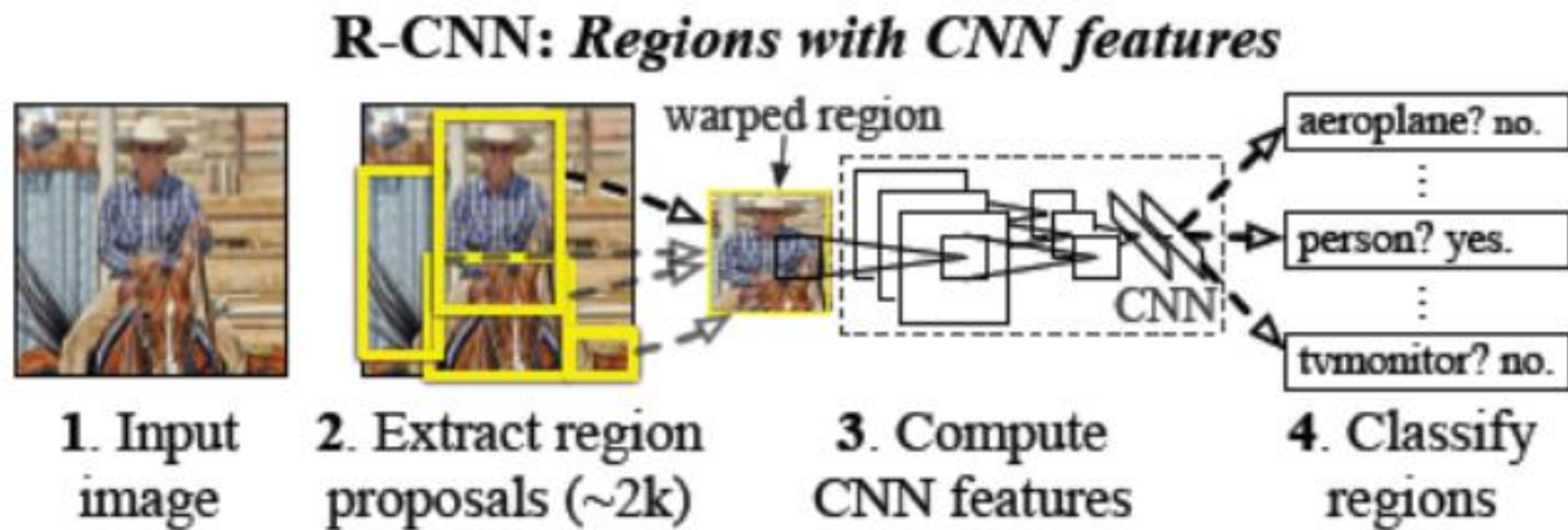
## ■ 1. 【2012-2016年】深度学习的一些新成果

AlexNet->GoogLeNet,VGG->RCNN -> SPPNET -> Fast-RCNN ->

Faster-RCNN->YOLO->SSD

## ■ 2. 使用PYTHON LAYER不改动CAFFE源码下添加新层，使用py-faster-rcnn为例。

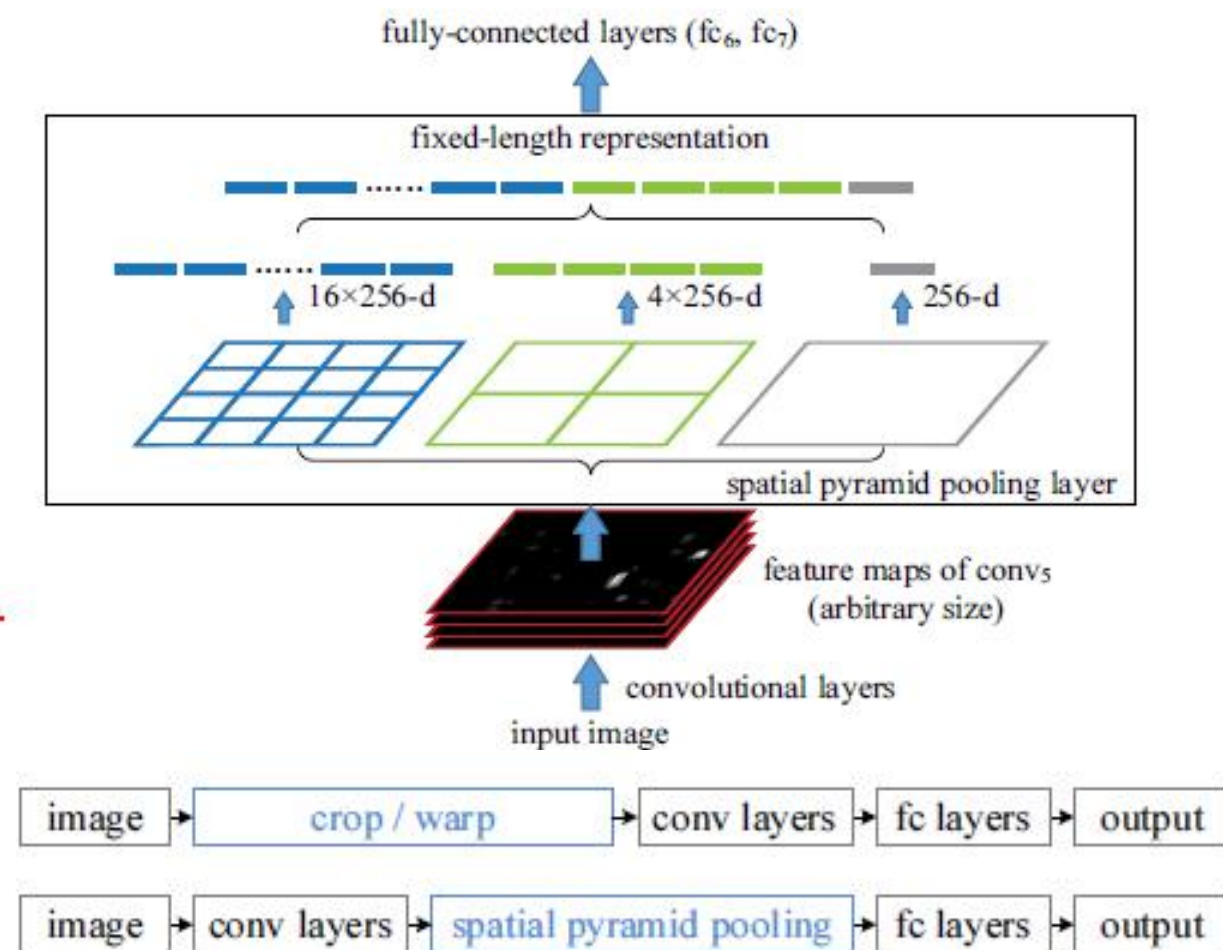
# R-CNN



- 基本思想是首先用一个非深度的方法，在图像中提取可能是物体的图形块，然后深度学习算法根据这些图像块，判断属性和一个具体物体的位置。
- Selective Search：先把完全不可能是物体的图像块去除，只剩2000左右的图像块放到深度网络里面判断。

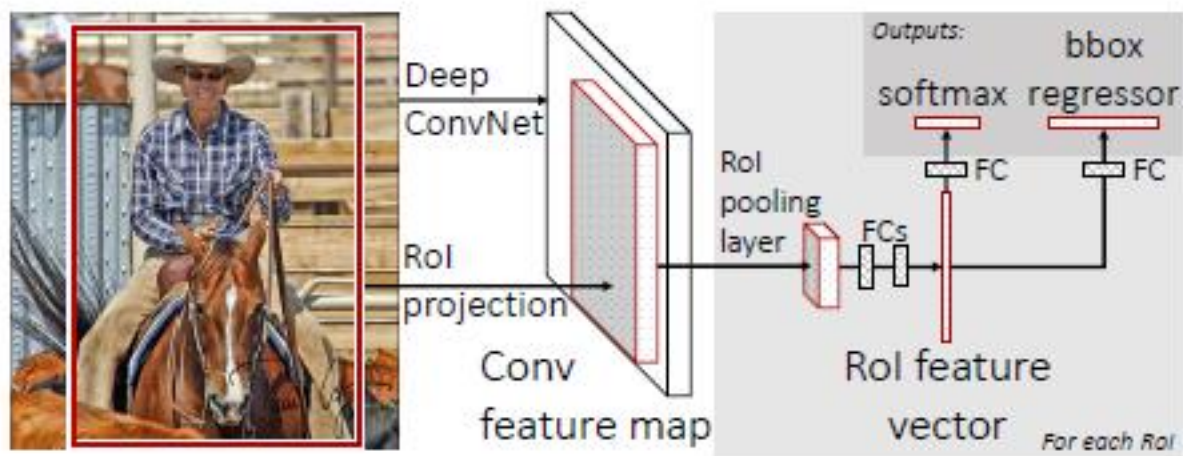
- 意义：RegionProposal+CNN的开山之作，在imagenet/voc/mscoco基本所有top的方法都是与这个框架。
- 缺点：速度慢，重复计算多，wrapping造成的图像变形(FC层的输入需要确定的size)。
- 速度：超慢，10s~15s处理一张图片
- 总结：多个regions+多次CNN+单个pooling

# SPPNET



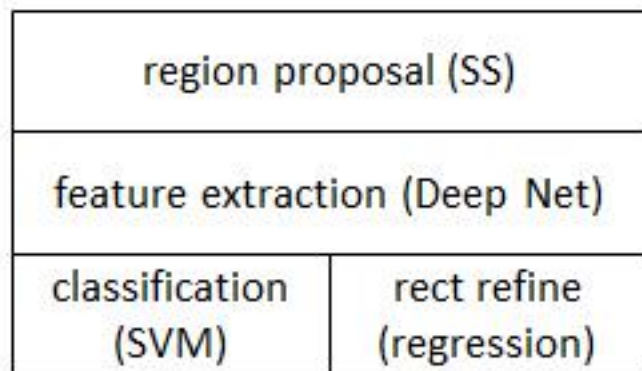
- spatial pyramid pooling:金字塔式池化
- 可运用不同大小的pooling窗口。
- 优点：保留了图像的特征，没有变形
- 速度：有进步，比RCNN速度提升了24-102倍。
- 缺点：和RCNN一样，训练过程是一个多阶段过程；用到的微调技术只能更新FC层，限制了深度CNN的潜力。
- 总结：单个图像+单次CNN+多个region+多个pooling

# Fast-R-CNN

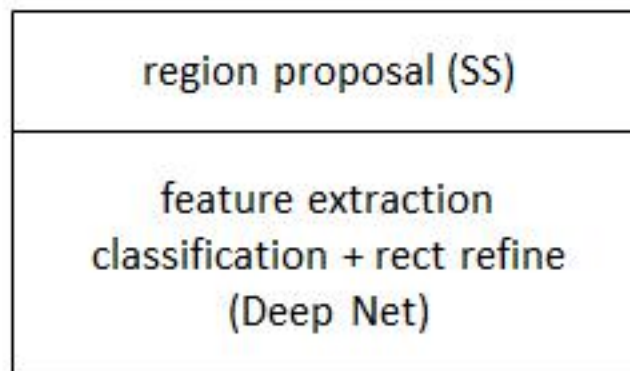


- 1. RoI pooling layer的方法：一方面是将image中的roI定位到feature map中对应patch，另一个是用一个单层的SPP layer将这个feature map patch下采样为大小固定的feature再传入全连接层。
- 2.训练过程通过运用多任务损失，实现单步骤完成。
- 3.在训练过程中所有层都可以得到更新。
- 速度：比SPPNet训练时间快了3倍，测试时间快了10倍。
- 缺点：proposal的计算瓶颈依然存在

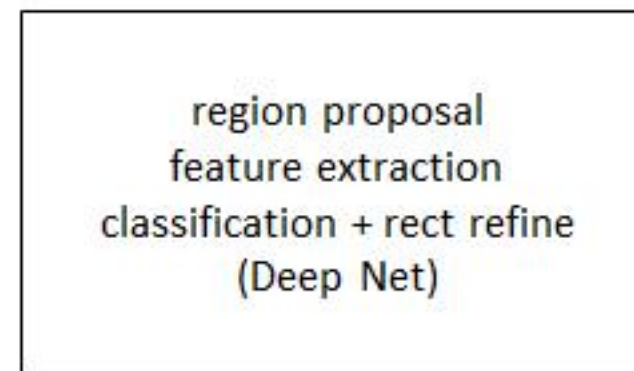




RCNN



fast RCNN

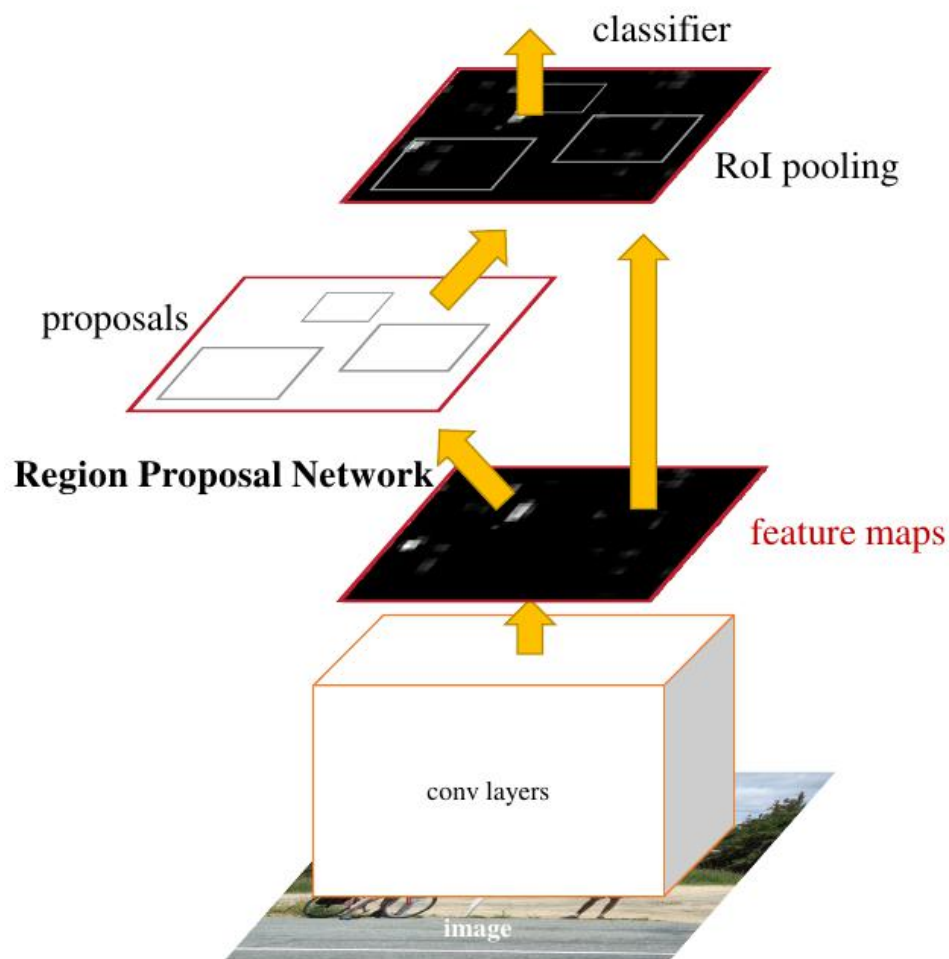


faster RCNN

三个问题：

1. 如何设计区域生成网络
2. 如何训练区域生成网络
3. 如何让区域生成网络和fast RCNN网络共享特征提取网络

# Faster-R-CNN

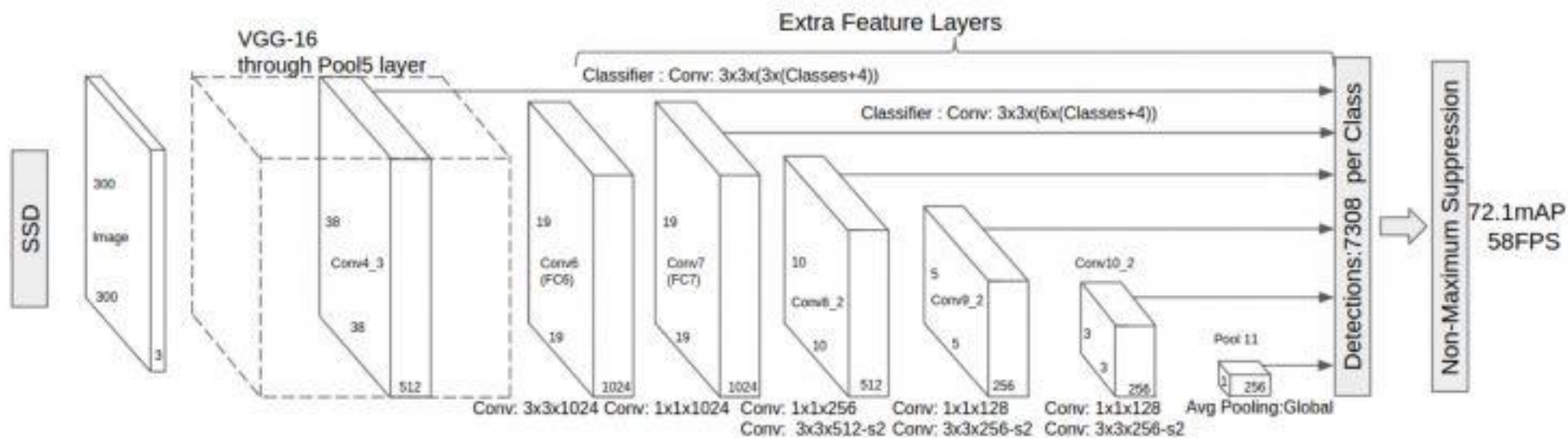


- 改进：不是用图像块来判断是物体还是背景，而把整张图像一起扔进深度网络里，让深度网络自行判断哪里有物体，物体的方块在哪里，种类是什么。
- 速度：比RCNN提升了两三百倍





# SSD



- 它是YOLO的超级改进版，吸取了YOLO的精度下降的教训，同时保留速度快的特点。它能达到58帧每秒，精度有72.1。速度超过Faster R-CNN 有8倍，但达到类似的精度。

# Python Layer

---

- 源码：[https://github.com/BVLC/caffe/blob/master/include/caffe/layers/python\\_layer.hpp](https://github.com/BVLC/caffe/blob/master/include/caffe/layers/python_layer.hpp)
- 1. 类PythonLayer继承自Layer，并且新增私有变量boost::python::object self来表示我们自己定义的python layer的内存对象。
- 2. 类PythonLayer类的成员函数LayerSetUp, Reshape, Forward\_cpu和Backward\_cpu分别是对我们自己定义的python layer中成员函数setup, reshape, forward和backward的封装调用。
- 参考项目：<https://github.com/rbgirshick/py-faster-rcnn> RBG大神的py-faster-rcnn

- 检查是否安装：
- 如果编译Caffe是使用cmake编译的，则默认安装了。

- CMakeLists.txt:

```
36 | caffe_option(BUILD_python_layer "Build the Caffe Python layer" ON)
```

- 如果使用make编译：
- 如果是首次编译，修改Caffe根目录下的Makefile.cinfig，uncomment
- WITH\_PYTHON\_LAYER:=1

```

layer {
  name: 'data'
  type: 'Python'
  top: 'data'
  top: 'rois'
  top: 'labels'
  top: 'bbox_targets'
  top: 'bbox_inside_weights'
  top: 'bbox_outside_weights'
  python_param {
    module: 'roi_data_layer.layer'
    layer: 'RoIDataLayer'
    param_str: "'num_classes': 21"
  }
}

```

module的名字，通常是定义Layer的.py文件的文件名，  
需要在\$PYTHONPATH

module中的类名

# parse the layer parameter string, which must be valid YAML  
layer\_params = yaml.load(self.param\_str\_)

self.\_num\_classes = layer\_params['num\_classes']