第十一讲 图像识别与分割

2019年5月20日　　11:08

1.分割

1.1语义分割

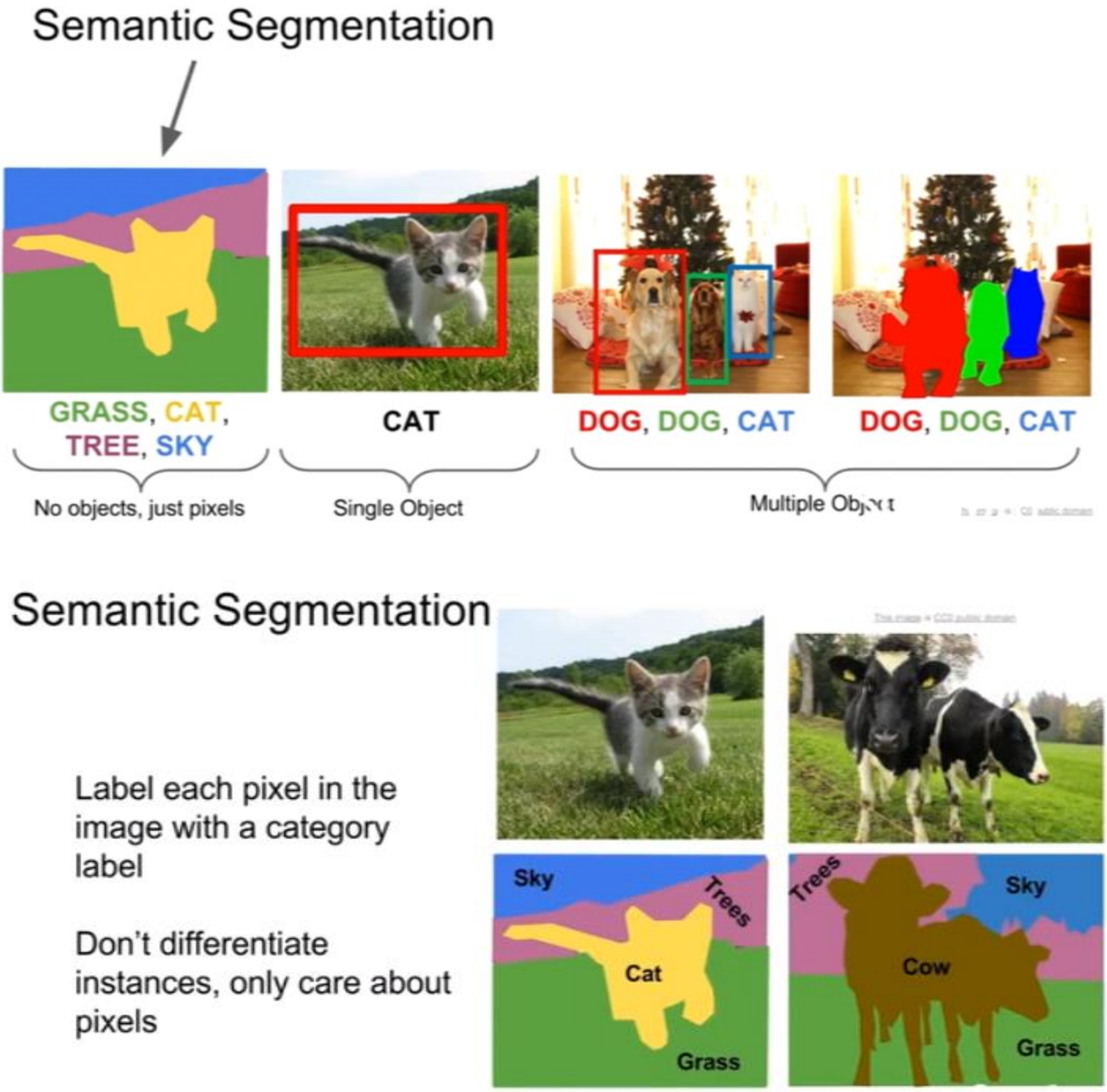


图 11.1.1 语义分割

语义分割定义：判断图像中每个像素点所属的标签如上图。

1.2滑动窗口

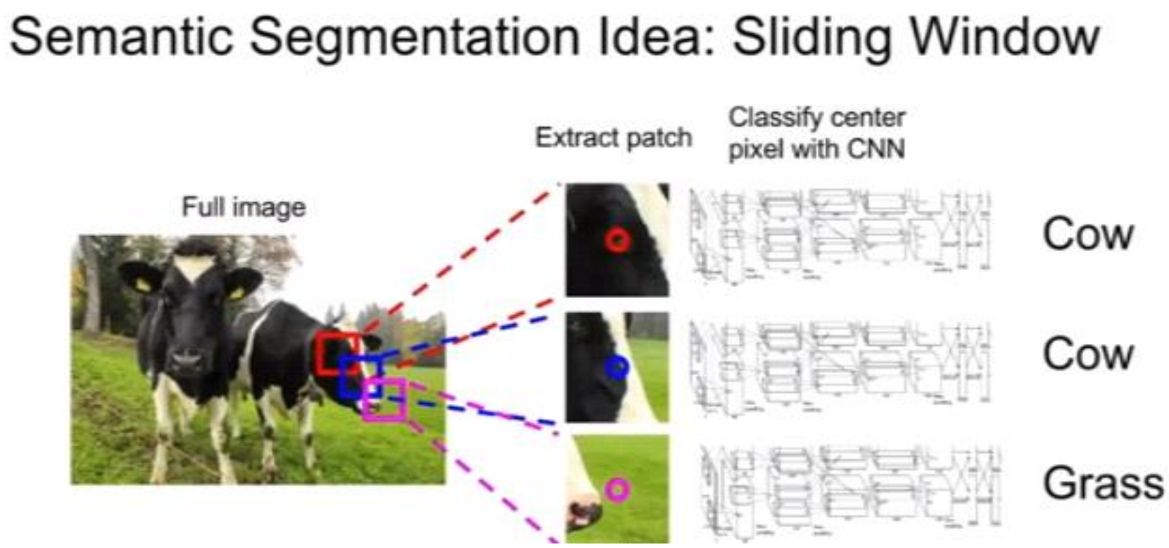
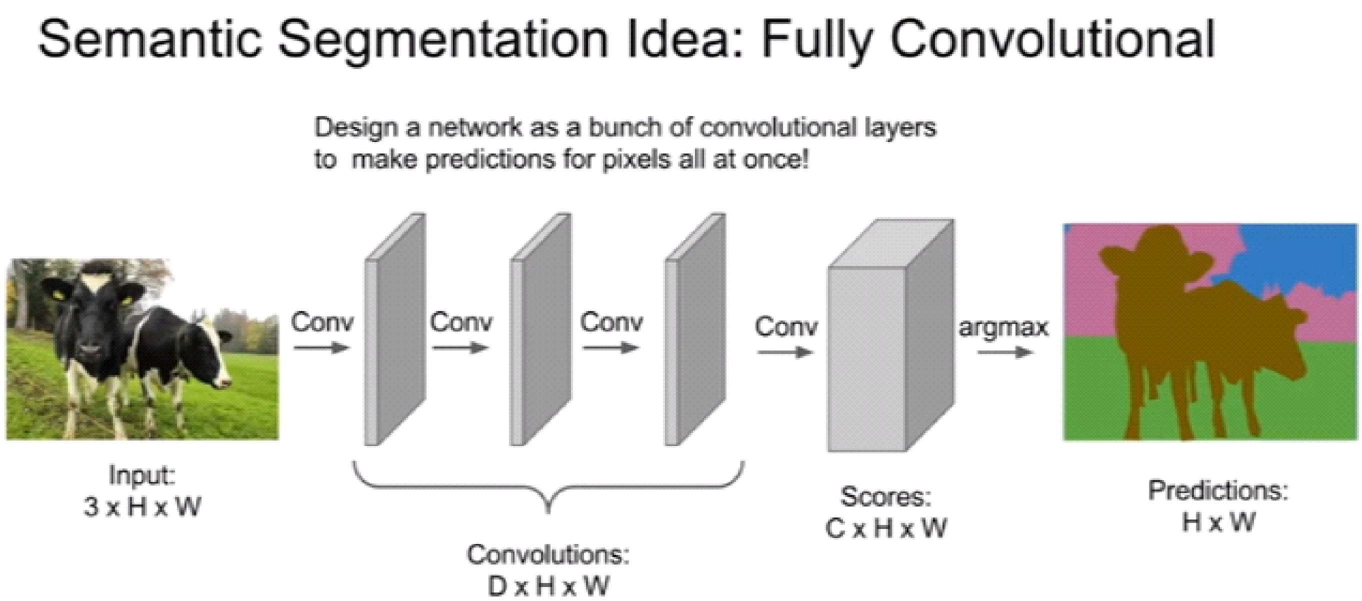


图 11.1.2 滑动窗口

每次在图像上截取一小部分像素，判断这部分像素属于哪个标签。(分类思想)

滑动窗口语义分割缺点：计算复杂度很高

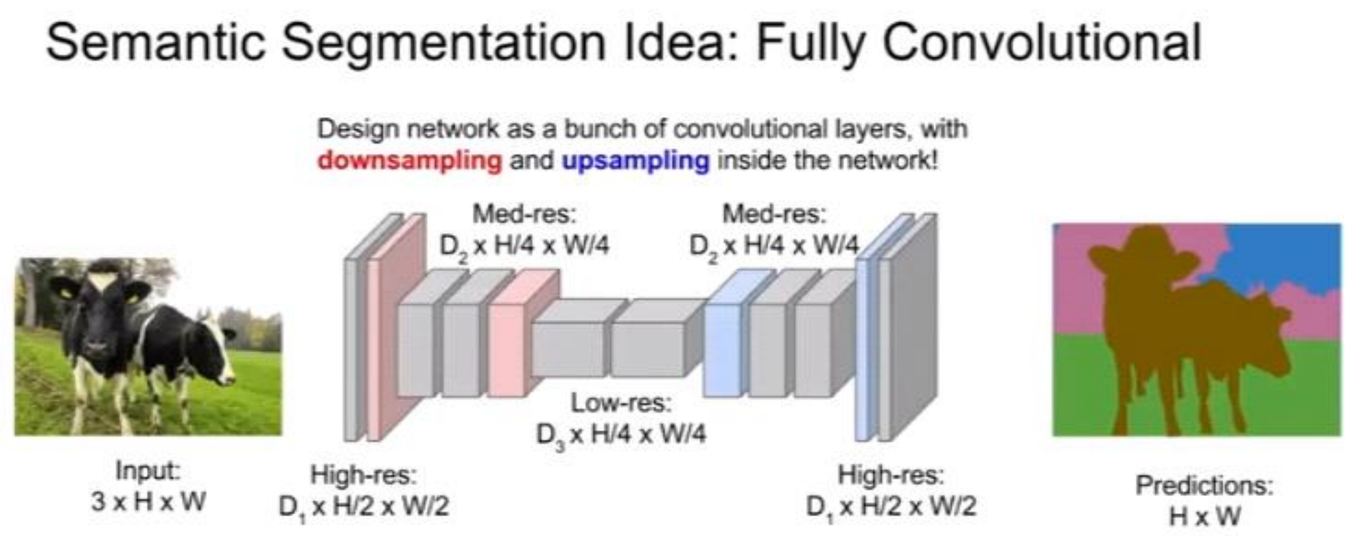
1.3全卷积网络实现语义分割



全卷积网络：顾名思义，CNN传统模型中全连接层去掉改为卷积层。

0填充3x3小卷积，保持图像尺寸不变，最后一层取平均或最大值。

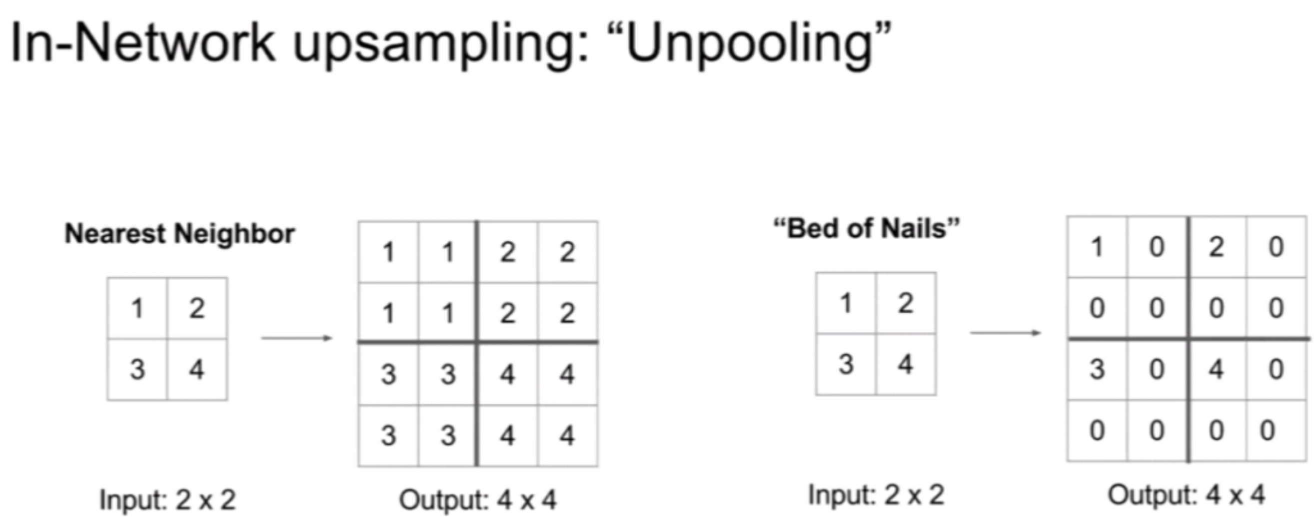
实际上全卷积网络很耗费内存，更常见的作法如下：



跨卷积(strided convoutions)：

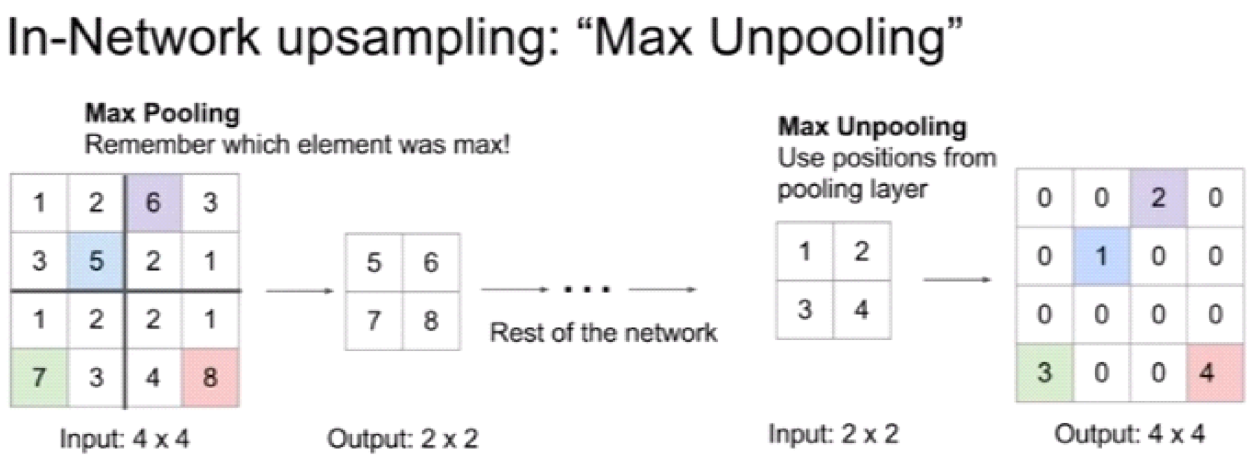
上采样：去池化

1.3上采样



NearestNeighbor:填充当前值。

Bed of Nails：元素填充左上角，其余部分填充0.



Max unpooling：最大池化时记录最大值位置，然后去池化时将元素放置在该位置，其余位置0填充。

1.4Transpose Convolution(Deconvolution)

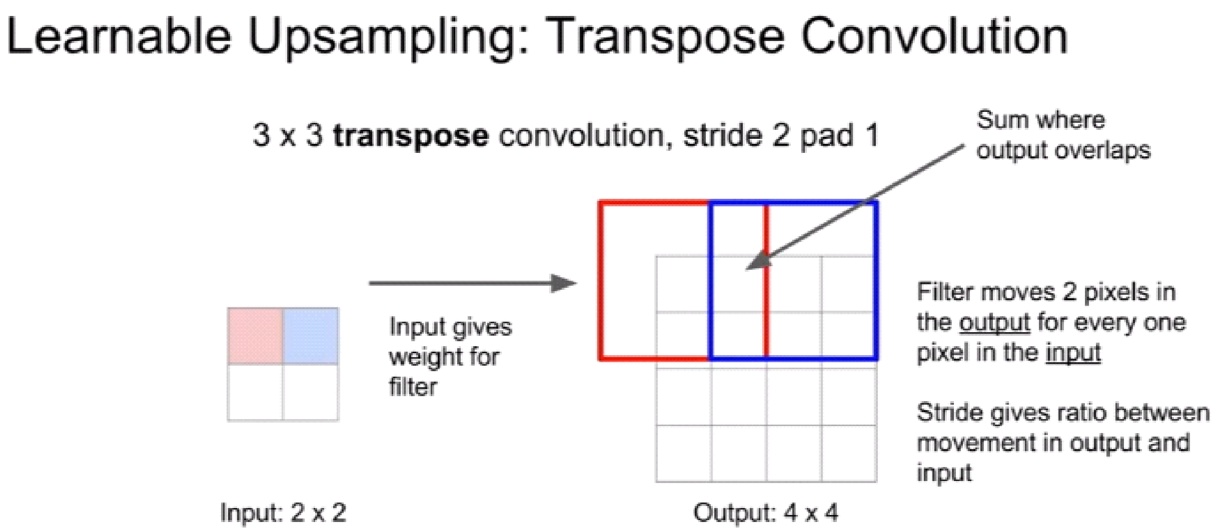
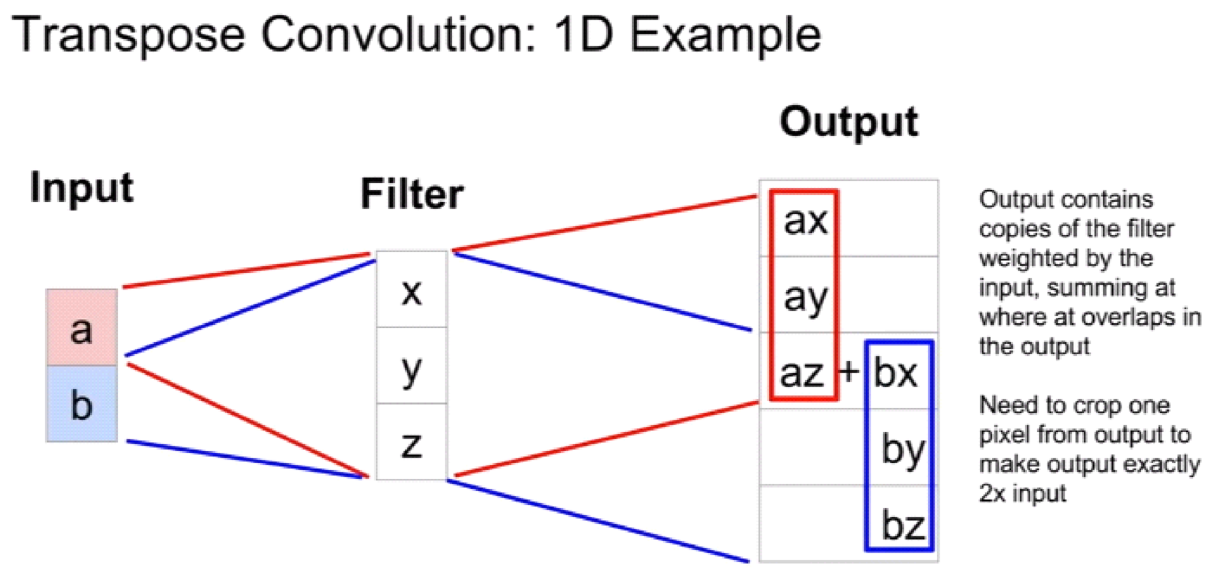
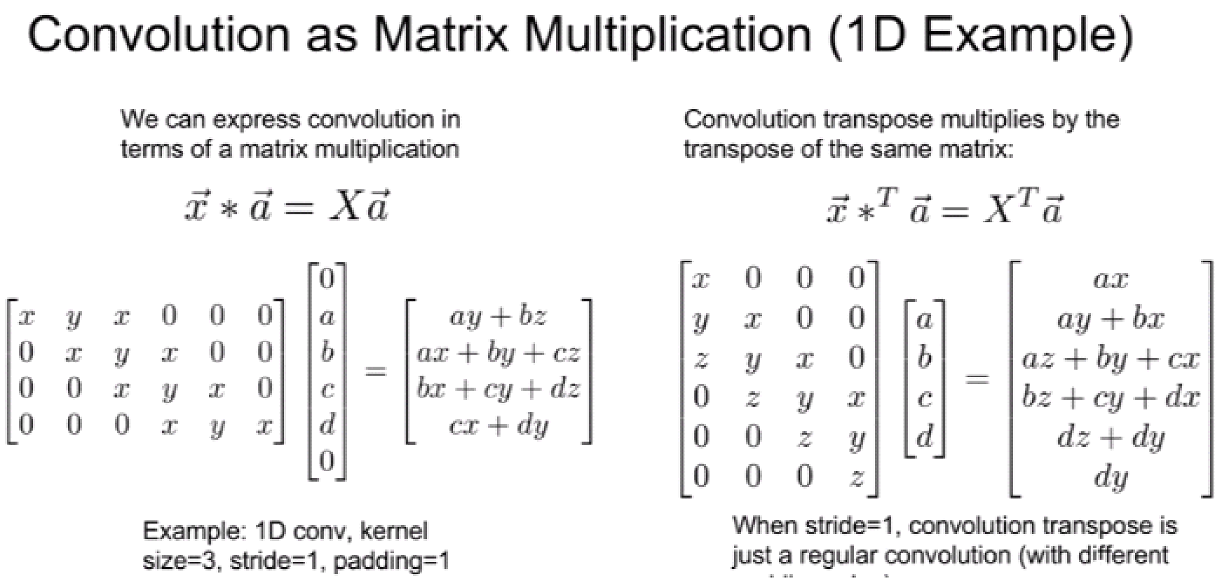


图11.1.7 Transpose Convolution

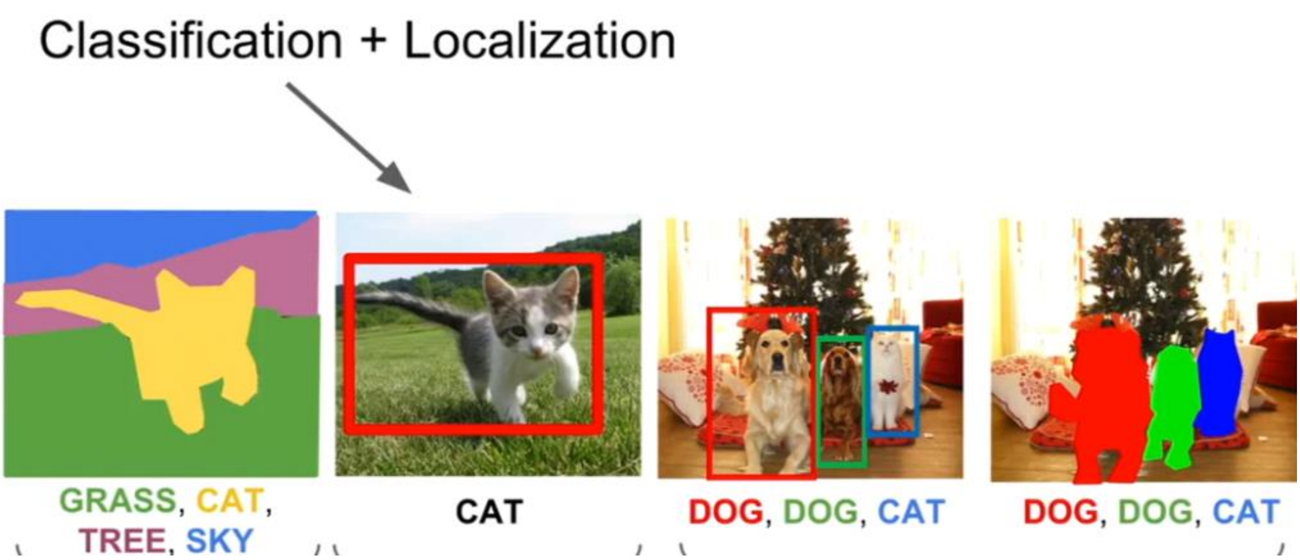
Transpose Convolution(反卷积)：输入与输出与卷积操作相反，计算过程如下。

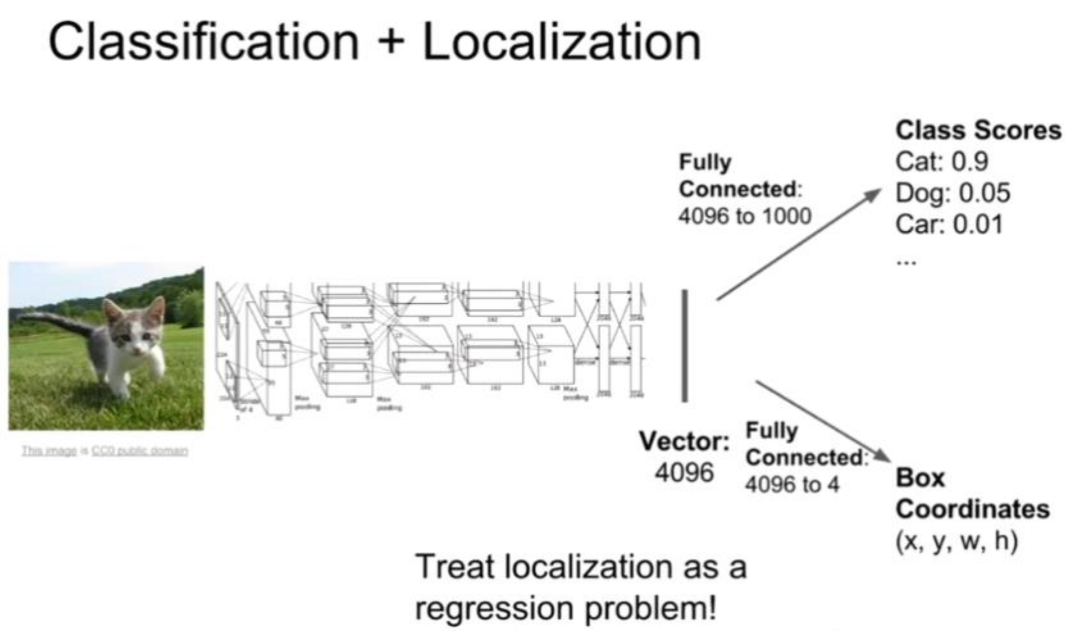

Transpose Convolution: 1D Example
Input
Filter
Output
a
b
x
y
z
ax
ay
az + bx
by
bz
Output contains copies of the filter weighted by the input, summing at where at overlaps in the output
Need to crop one pixel from output to make output exactly 2x input


Convolution as Matrix Multiplication (1D Example)
We can express convolution in terms of a matrix multiplication
$\vec{x} * \vec{a} = X\vec{a}$
$\begin{bmatrix} x & y & x & 0 & 0 & 0 \\ 0 & x & y & x & 0 & 0 \\ 0 & 0 & x & y & x & 0 \\ 0 & 0 & 0 & x & y & x \end{bmatrix} \begin{bmatrix} 0 \\ a \\ b \\ c \\ d \\ 0 \end{bmatrix} = \begin{bmatrix} ay + bz \\ ax + by + cz \\ bx + cy + dz \\ cx + dy \end{bmatrix}$
Example: 1D conv, kernel size=3, stride=1, padding=1
Convolution transpose multiplies by the transpose of the same matrix:
$\vec{x} *^T \vec{a} = X^T\vec{a}$
$\begin{bmatrix} x & 0 & 0 & 0 \\ y & x & 0 & 0 \\ z & y & x & 0 \\ 0 & z & y & x \\ 0 & 0 & z & y \\ 0 & 0 & 0 & z \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} ax \\ ay + bx \\ az + by + cx \\ bz + cy + dx \\ dz + dy \\ dy \end{bmatrix}$
When stride=1, convolution transpose is just a regular convolution (with different

2.定位

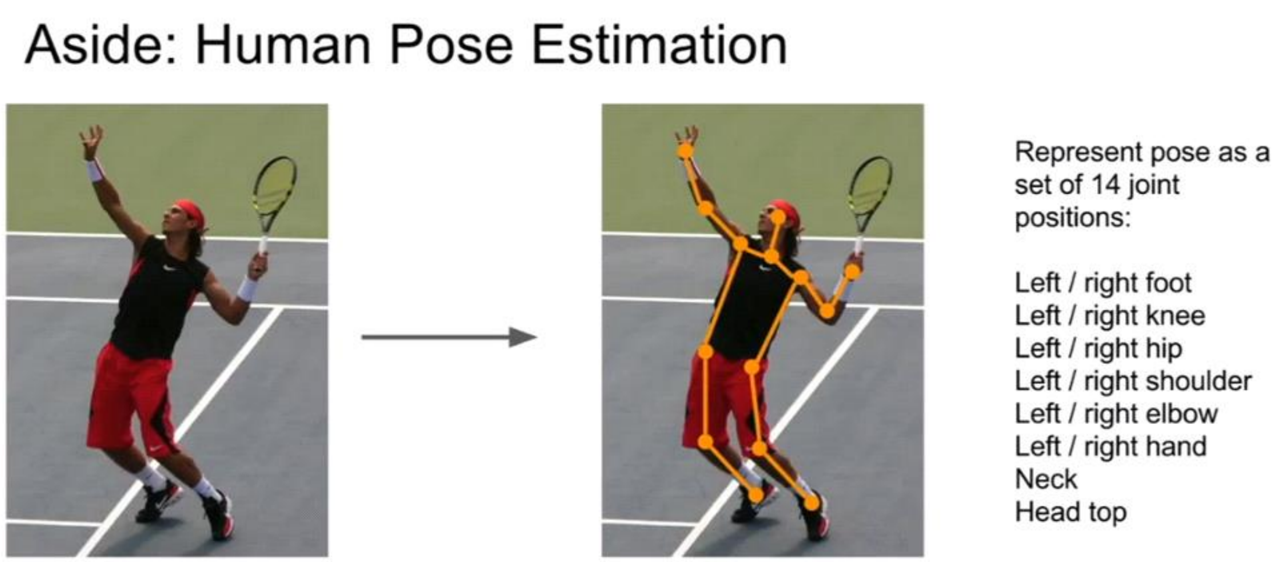
2.1分类与定位

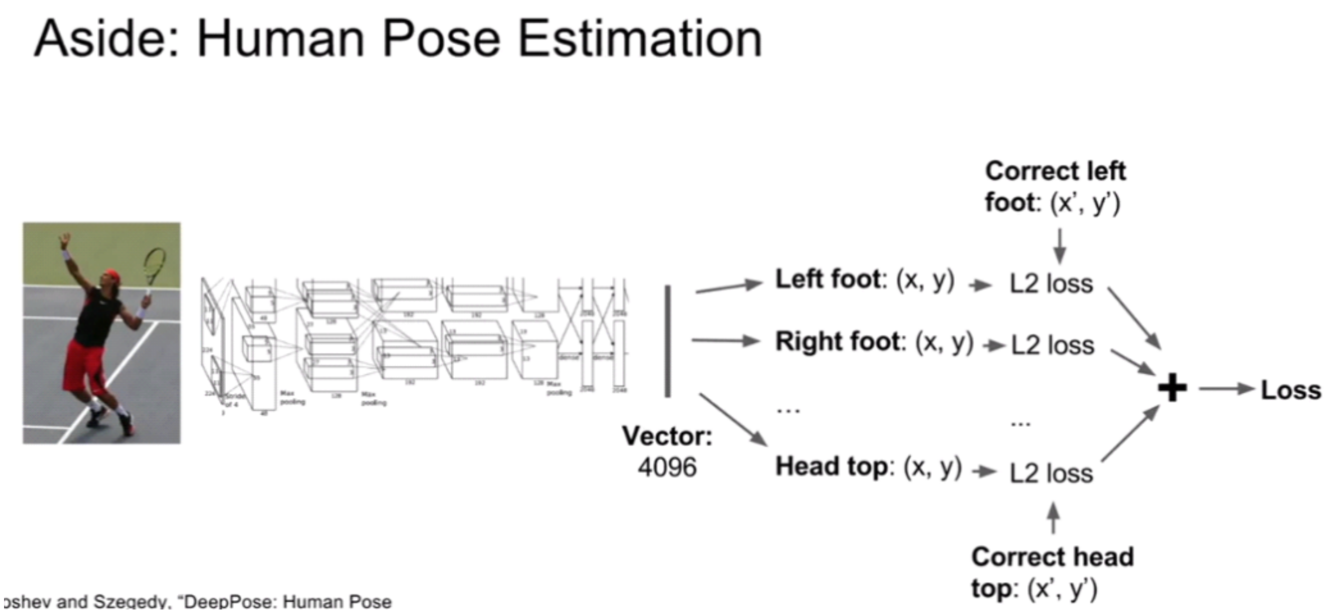

Classification + Localization
GRASS, CAT,
TREE, SKY
CAT
DOG, DOG, CAT
DOG, DOG, CAT


Classification + Localization
Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...
Fully
Connected:
4096 to 1000
Vector:
4096
Fully
Connected:
4096 to 4
Box
Coordinates
(x, y, w, h)
Treat localization as a
regression problem!
This image is CC0 public domain

图

2.2 姿态估计

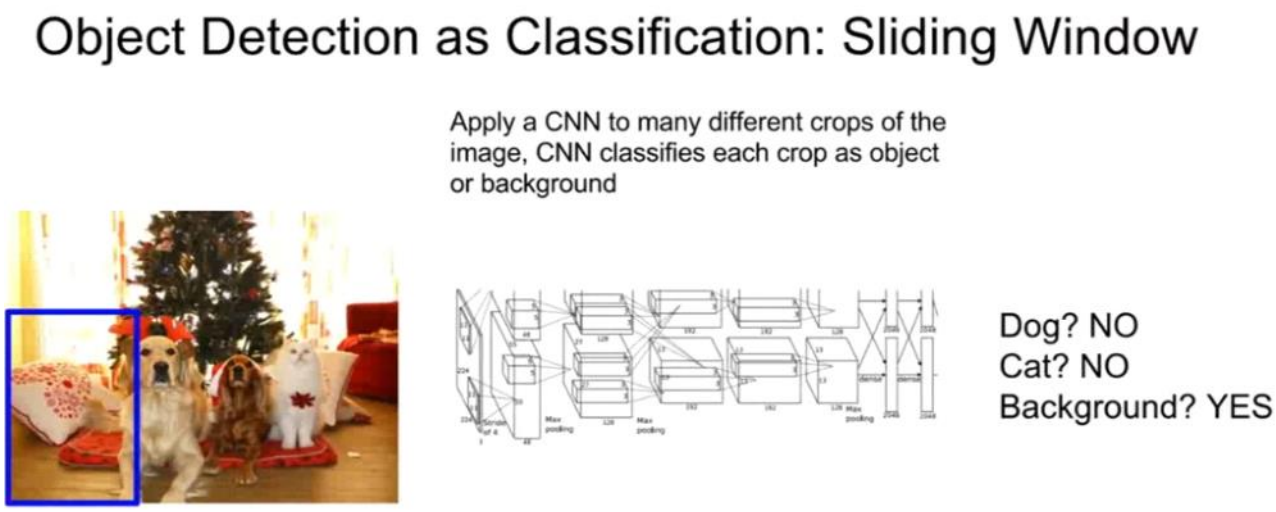

Aside: Human Pose Estimation
Represent pose as a
set of 14 joint
positions:
Left / right foot
Left / right knee
Left / right hip
Left / right shoulder
Left / right elbow
Left / right hand
Neck
Head top



3.识别

3.1 Object Detection

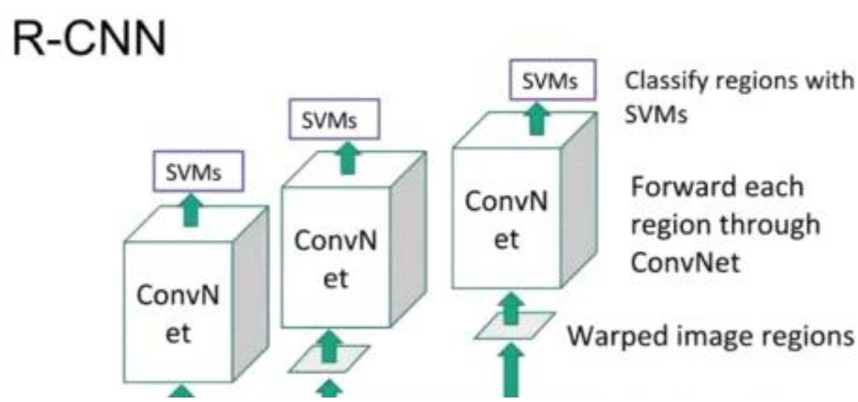
3.1滑动窗口



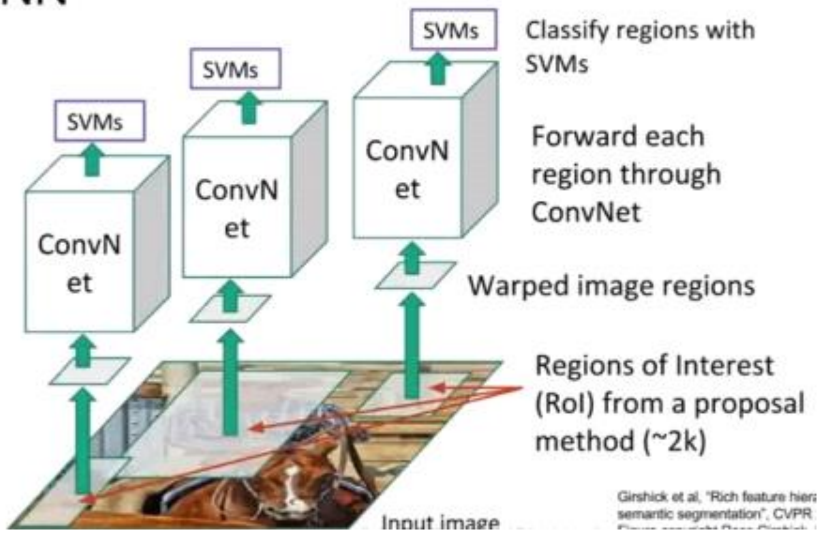
滑动窗口的问题：待选窗口的大小和位置如何选择？一张图里有多少个对象也是不知道！

候选区：随机产生一千个待选区域，然后判断候选区的内容

3.2R-CNN



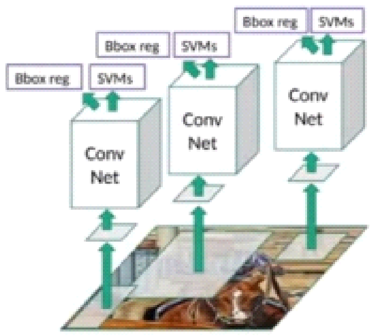
R-CNN



候选区：产生两千个待选区域，对候选区截取的图片做分类。

R-CNN: Problems

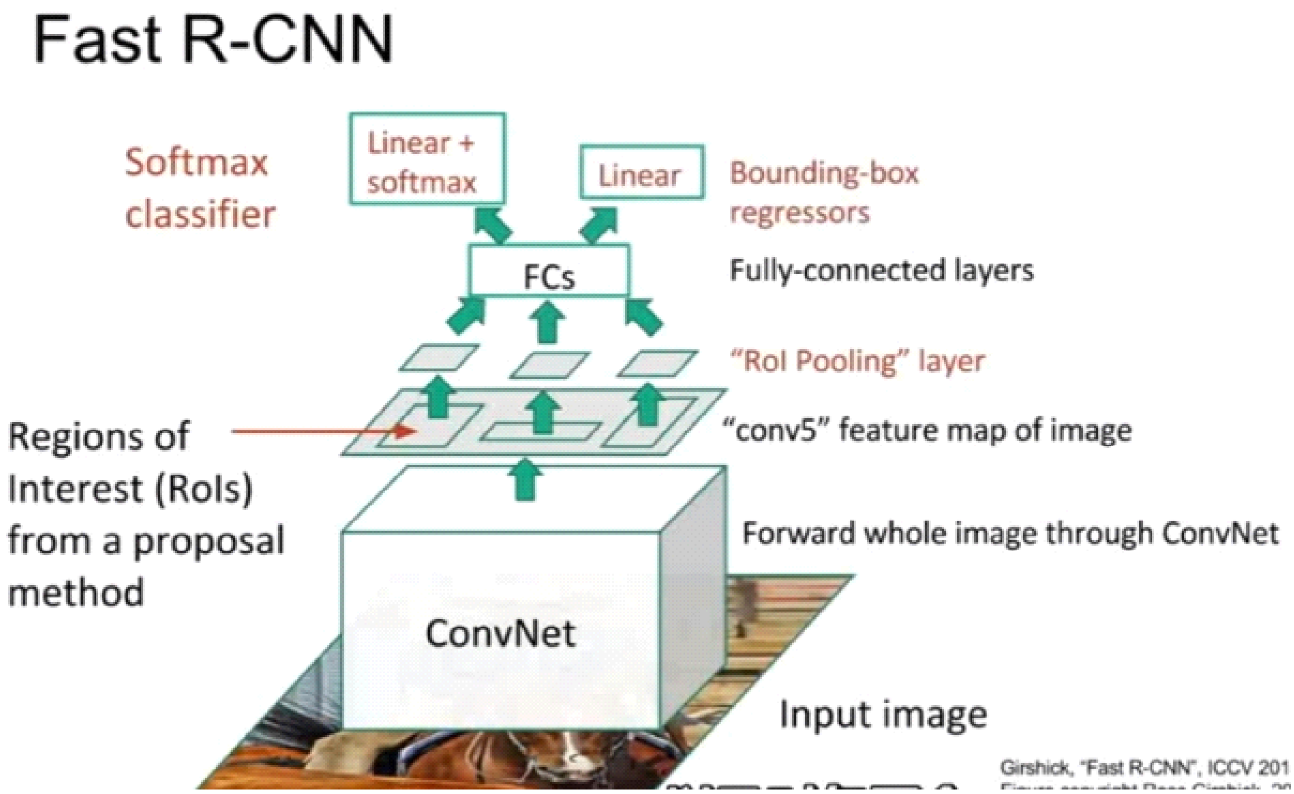
- Ad hoc training objectives
  - Fine-tune network with softmax classifier (log loss)
  - Train post-hoc linear SVMs (hinge loss)
  - Train post-hoc bounding-box regressions (least squares)
- Training is slow (84h), takes a lot of disk space
- Inference (detection) is slow
  - 47s / image with VGG16 [Simonyan & Zisserman. ICLR15]
  - Fixed by SPP-net [He et al. ECCV14]



R-CNN存在的问题：网络训练时间耗费时间很长，并且需要大量的存储空间。

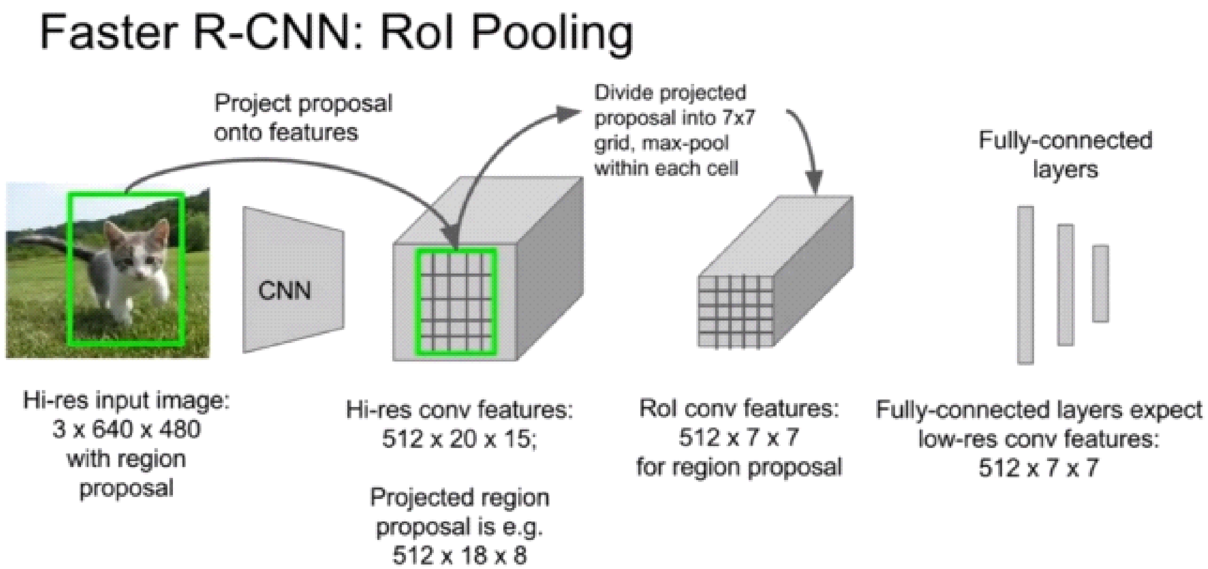
Fixed by SPP-net ECCV14

3. 3 Fast R-CNN



Fast R-CNN：先卷积然后在卷积后的Feature map中选择候选区，然后对候选区进行"RoI Pooling"，再经过全连接层。

RoI Pooling(兴趣池化？)：roi：讲师说不想讲述细节，感兴趣的私下去了解。

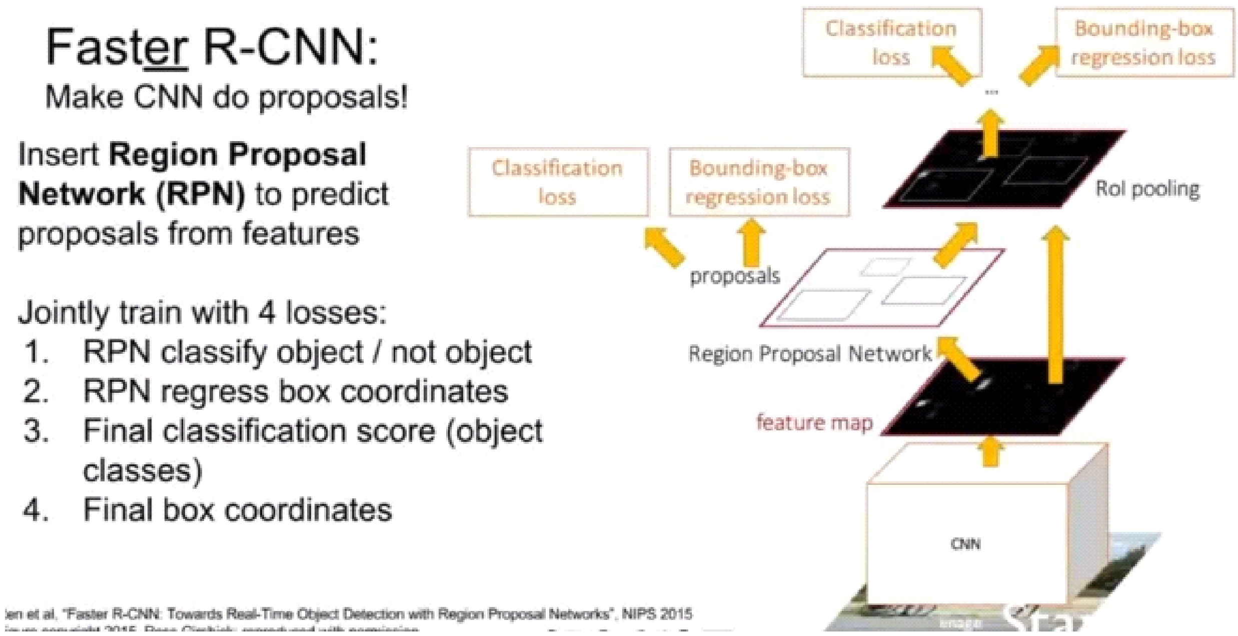


#------------------------------------

#todo roi pooling

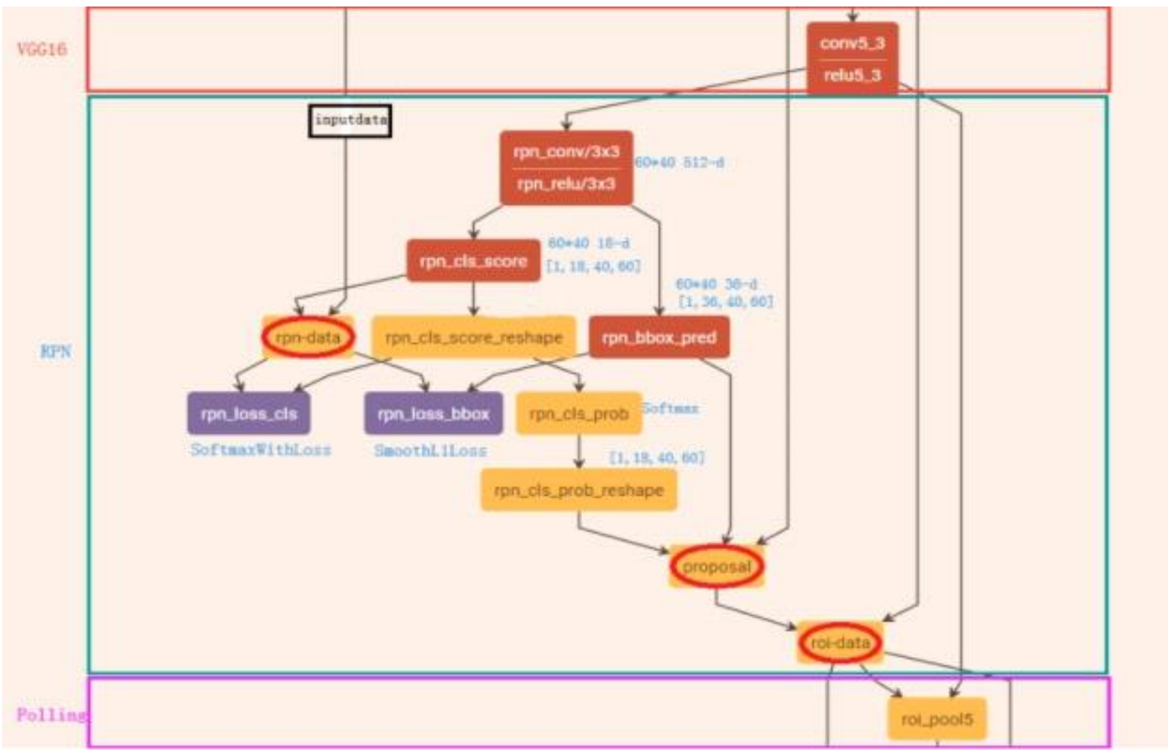
#------------------------------------

3.4Faster R-CNN



在之前的模型中最耗时的操作是产生候选区，（两千个候选区）大约需要1秒多，但在实时物体探测中这个速度显然是不行的，于是有了Faster R-CNN，如上图。

RPN:RPN网络主要用于生成region proposals，首先生成一堆Anchor box，对其进行裁剪过滤后通过softmax判断anchors属于前景(foreground)或者后景(background)，即是物体or不是物体，所以这是一个二分类；同时，另一分支bounding box regression修正anchor box，形成较精确的proposal 。



关于FasterCNN的一篇博客：https://www.cnblogs.com/wangyong/p/8513563.html

更多Object detection的模型：

Base Network
VGG16
ResNet-101
Inception V2
Inception V3
Inception
ResNet
MobileNet

Object Detection architecture
Faster R-CNN
R-FCN
SSD

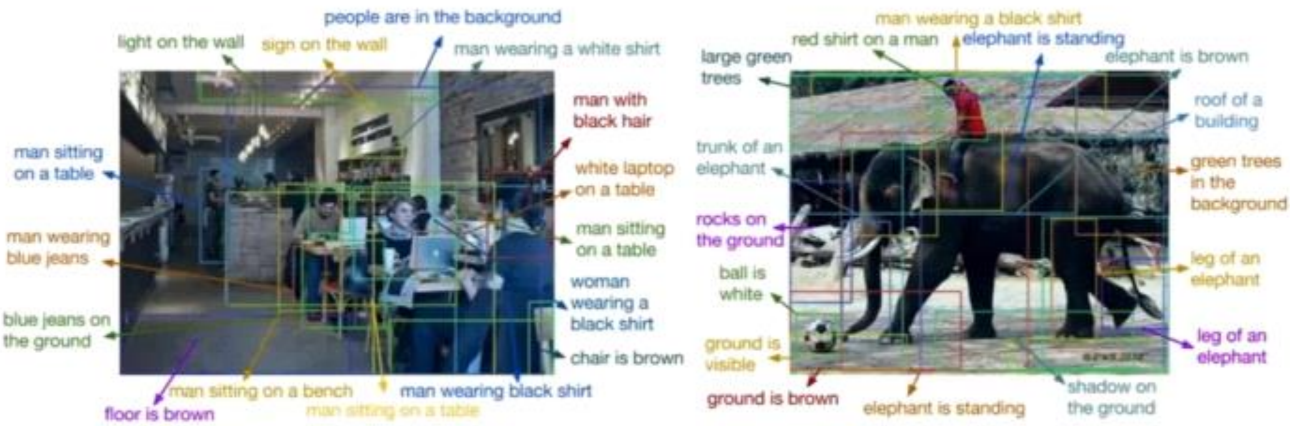
Image Size
 # Region Proposals

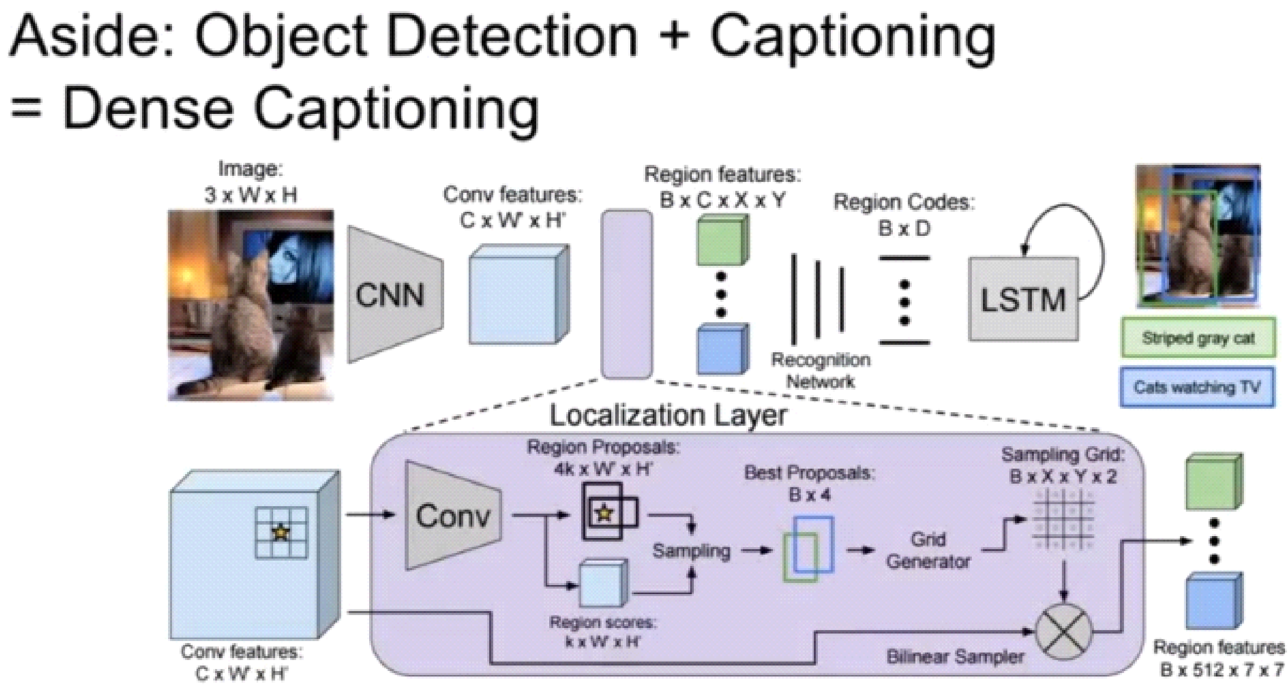
Takeaways
Faster R-CNN is slower but more accurate

SSD is much faster but not as accurate

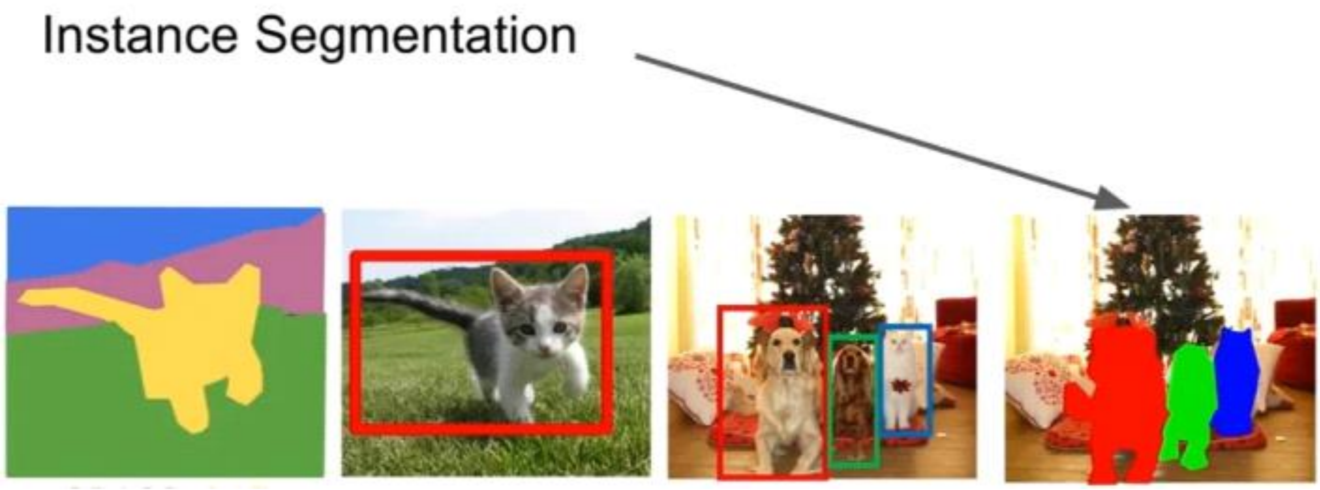
3.5Dense Captioning

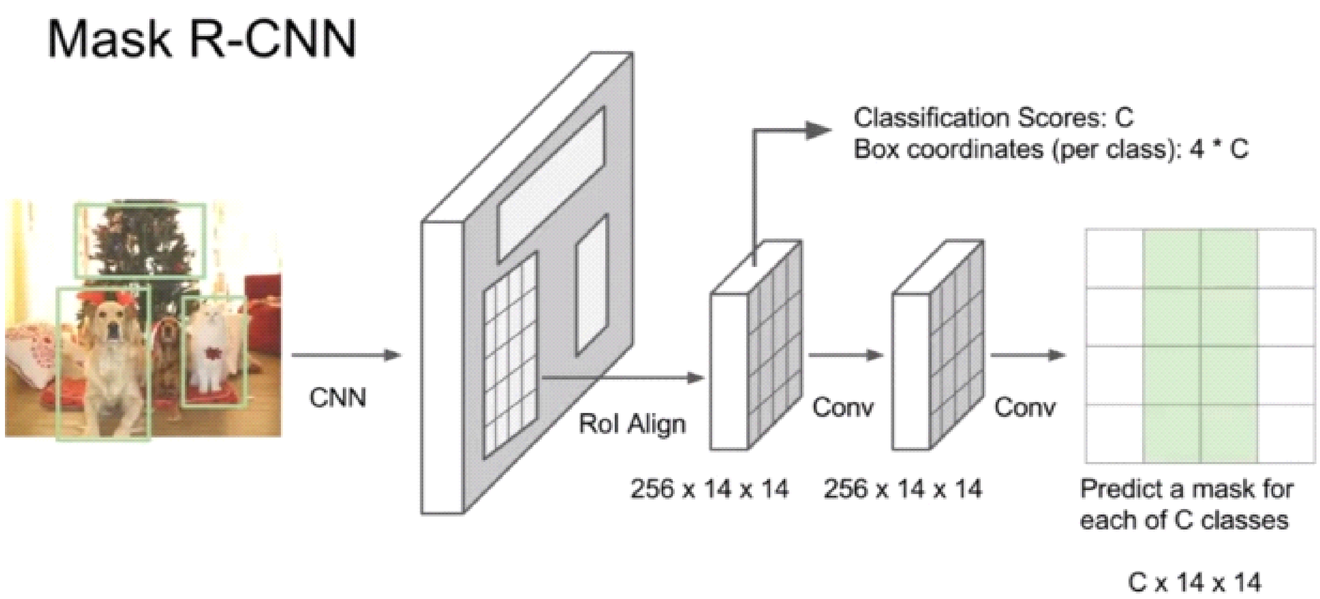
Aside: Object Detection + Captioning
= Dense Captioning



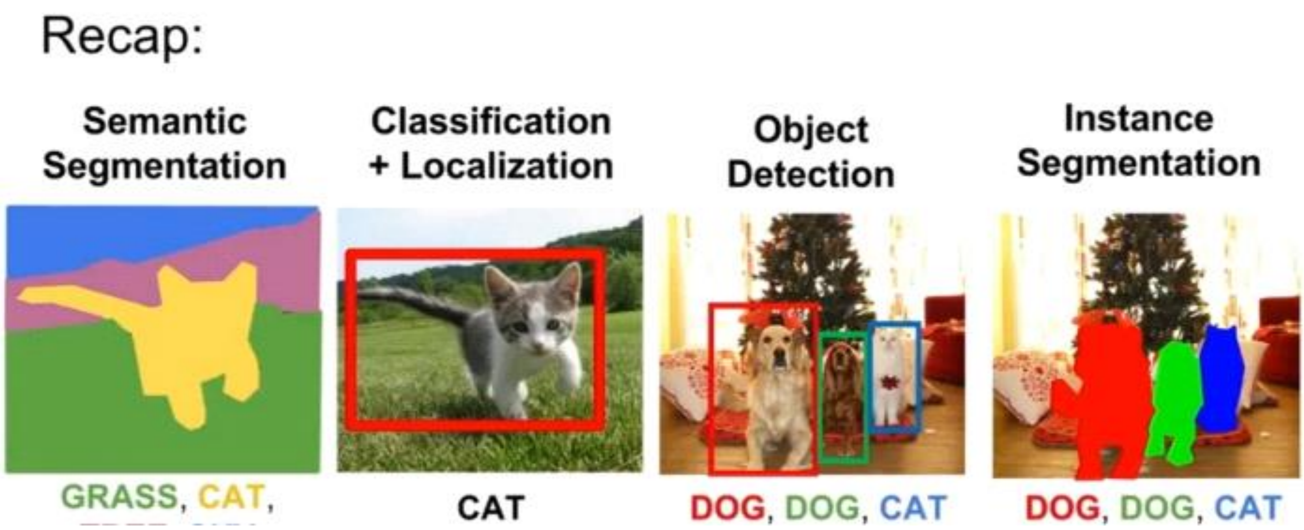

Aside: Object Detection + Captioning
= Dense Captioning
Image:
3 x W x H
CNN
Conv features:
C x W' x H'
Region features:
B x C x X x Y
Region Codes:
B x D
LSTM
Recognition
Network
Striped gray cat
Cats watching TV
Localization Layer
Conv features:
C x W' x H'
Conv
Region Proposals:
4k x W' x H'
Region scores:
k x W' x H'
Sampling
Best Proposals:
B x 4
Grid
Generator
Sampling Grid:
B x X x Y x 2
Bilinear Sampler
Region features
B x 512 x 7 x 7

3.6实例分割


Instance Segmentation


Mask R-CNN
CNN
RoI Align
Classification Scores: C
Box coordinates (per class): 4 * C
Conv
Conv
256 x 14 x 14
256 x 14 x 14
Predict a mask for
each of C classes
C x 14 x 14

3.7小节


Recap:
Semantic
Segmentation
Classification
+ Localization
Object
Detection
Instance
Segmentation
GRASS, CAT,
CAT
DOG, DOG, CAT
DOG, DOG, CAT