视觉识别

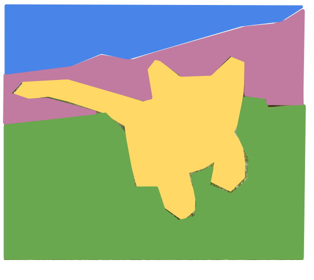
鲁 鹏
北京邮电大学 计算机学院 智能科学与技术中心

主要素材取自于CS231N课程课件

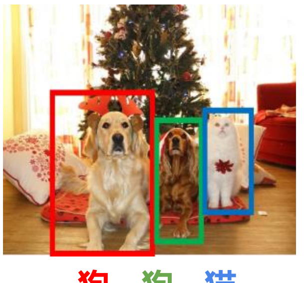
视觉识别任务

分类 | 语义分割 | 目标检测 | 实例分割

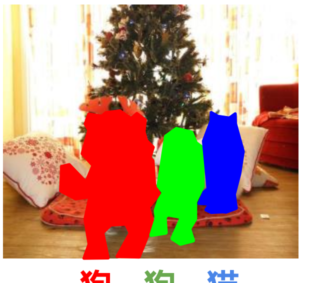
猫



草，猫，树，天空



狗，狗，猫



狗，狗，猫

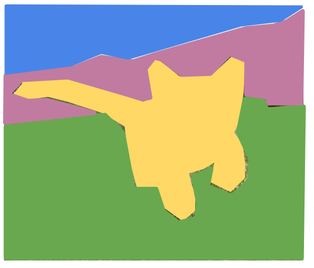
不考虑空间位置

像素的类别

多目标

视觉识别任务

分类 | 语义分割 | 目标检测 | 实例分割



猫 | 草，猫，树，天空 | 狗，狗，猫 | 狗，狗，猫

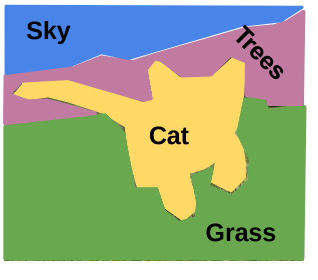
不考虑空间位置 | 像素的类别 | 多目标

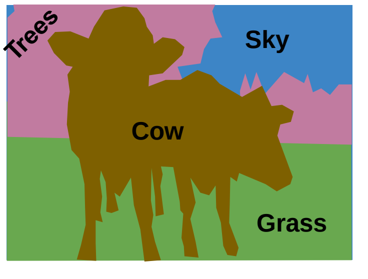
语义分割

给每个像素分配类别标签

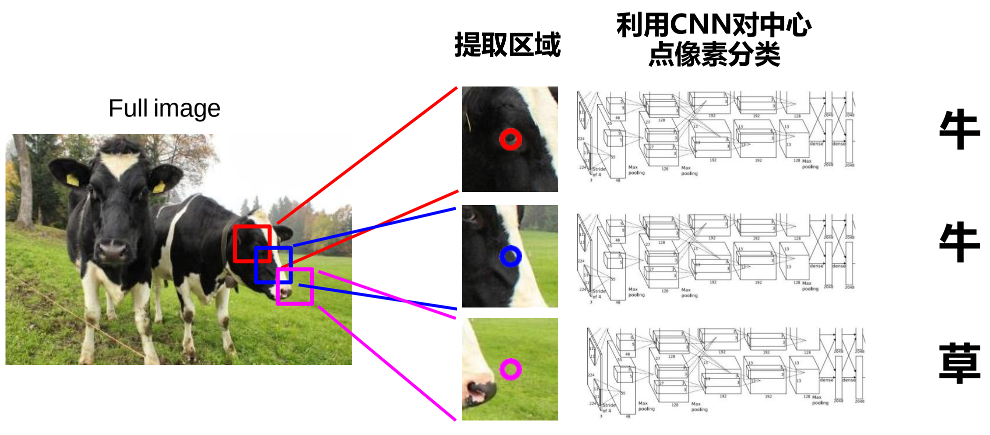
不区分实例，只考虑像素类别





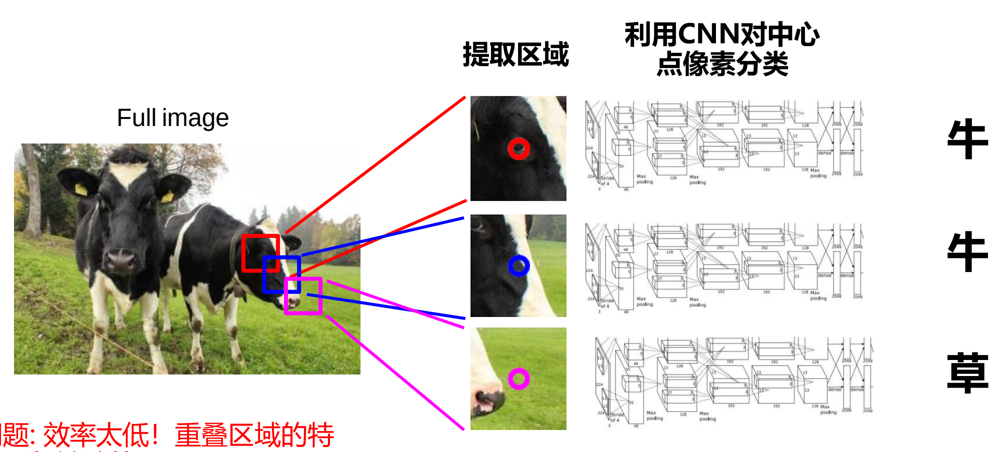


语义分割思路：滑动窗口



Farabet et al, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013
Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, ICML 2014

语义分割思路：滑动窗口

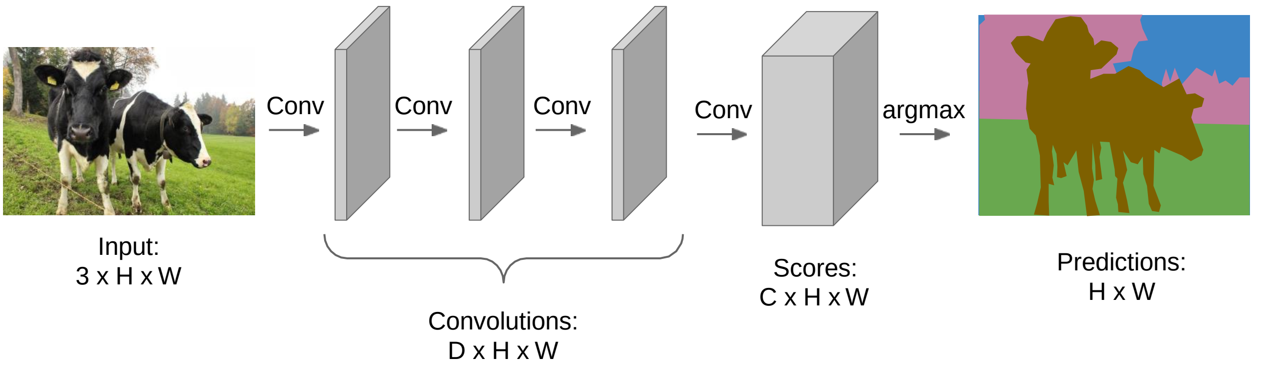


问题: 效率太低！重叠区域的特征反复被计算

Farabet et al, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013
Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, ICML 2014

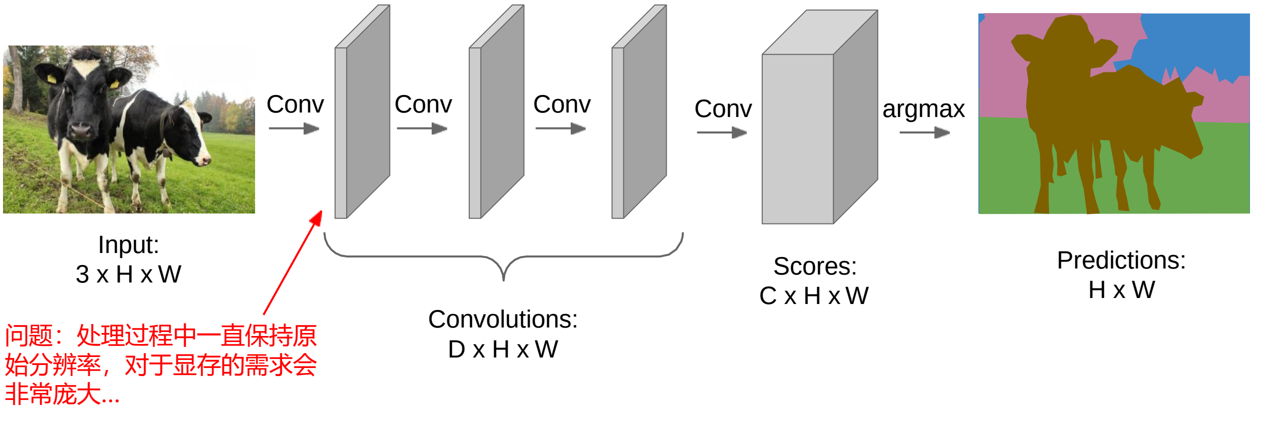
语义分割思路：全卷积

解决方案：让整个网络只包含卷积层，
一次性输出所有像素的类别预测。



语义分割思路：全卷积

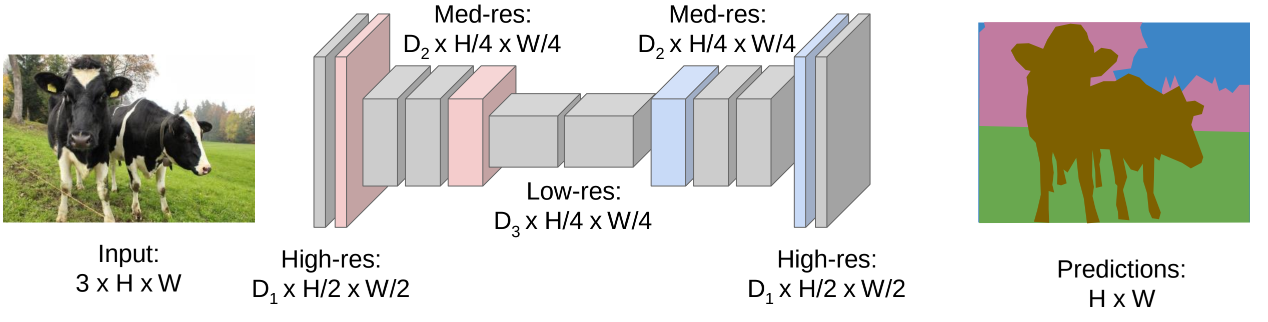
解决方案：让整个网络只包含卷积层，
一次性输出所有像素的类别预测。



语义分割思路：全卷积

解决方案：让整个网络只包含卷积层，

并在网络中嵌入**下采样**与**上采样**过程。



Long, Shelhamer, and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015
Noh et al, “Learning Deconvolution Network for Semantic Segmentation”, ICCV 2015

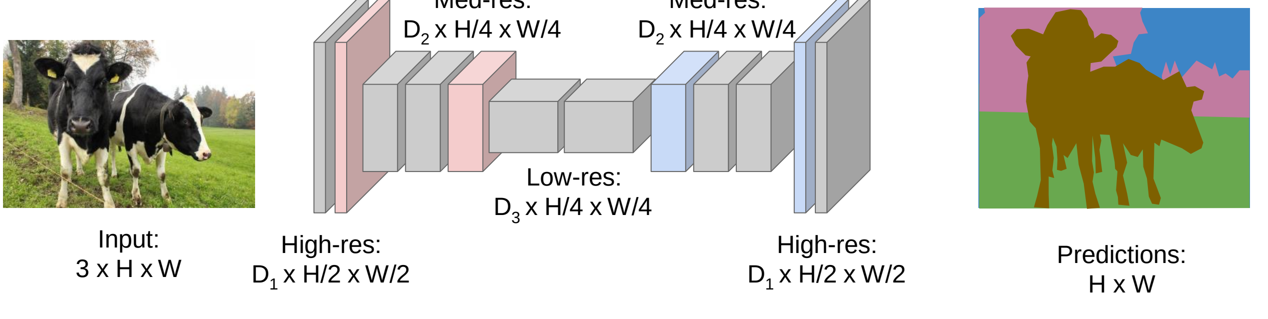
语义分割思路：全卷积

解决方案：让整个网络只包含卷积层，

并在网络中嵌入**下采样**与**上采样**过程。

下采样
Pooling, strided convolution

上采样
???



Long, Shelhamer, and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015
Noh et al, “Learning Deconvolution Network for Semantic Segmentation”, ICCV 2015

反池化操作: “Unpooling”

Nearest Neighbor

1	2
3	4

Input: 2 x 2

1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4

Output: 4 x 4

“Bed of Nails”

1	2
3	4

Input: 2 x 2

1	0	2	0
0	0	0	0
3	0	4	0
0	0	0	0

Output: 4 x 4

反池化操作: “Max Unpooling”

Max Pooling
Remember which element was max!

1	2	6	3
3	5	2	1
1	2	2	1
7	3	4	8

Input: 4 x 4

5	6
7	8

Output: 2 x 2

...

Rest of the network

Max Unpooling
Use positions from pooling layer

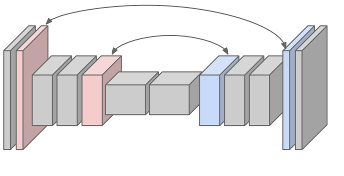
1	2
3	4

Input: 2 x 2

0	0	2	0
0	1	0	0
0	0	0	0
3	0	0	4

Output: 4 x 4

Corresponding pairs of downsampling and upsampling layers



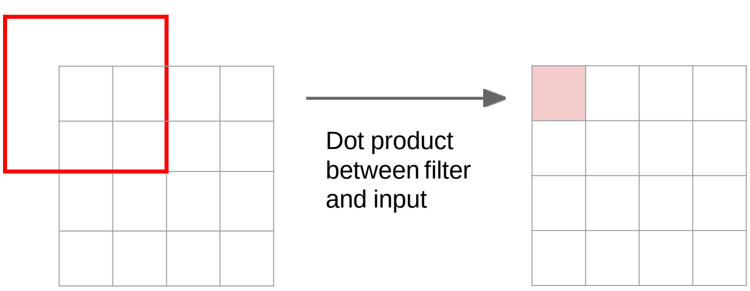
可学习的上采样: 转置卷积（Transpose Convolution)

回顾: 3 x 3 卷积, 步长（stride） 1， 零填充（pad） 1

Input: 4 x 4　　Output: 4 x 4

可学习的上采样: 转置卷积（Transpose Convolution)

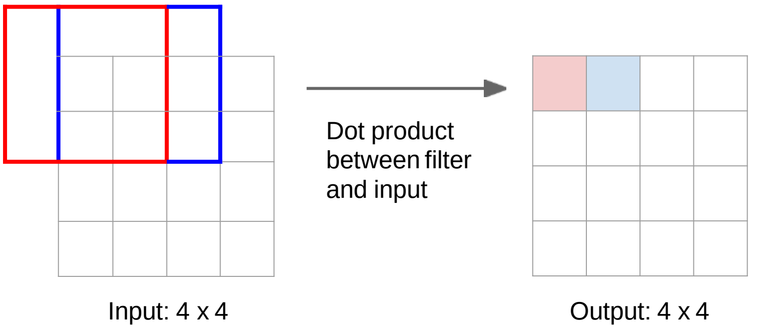
回顾: 3 x 3 卷积, 步长（stride） 1， 零填充（pad） 1



Input: 4 x 4　　Output: 4 x 4

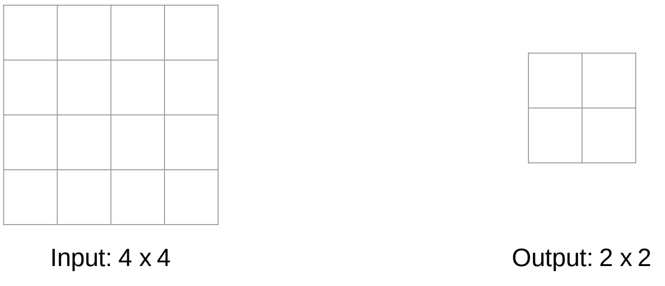
可学习的上采样: 转置卷积 (Transpose Convolution)

回顾: 3 x 3 卷积, 步长（stride） 1， 零填充（pad） 1



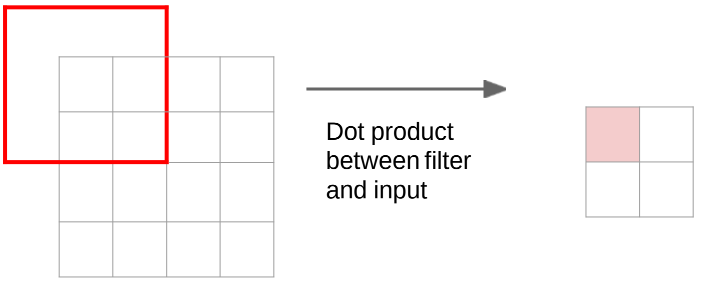
可学习的上采样: 转置卷积 (Transpose Convolution)

回顾: 3 x 3 卷积, 步长（stride） 2， 零填充（pad） 1



可学习的上采样: 转置卷积（Transpose Convolution)

回顾: 3 x 3 卷积, 步长（stride） 2， 零填充（pad） 1

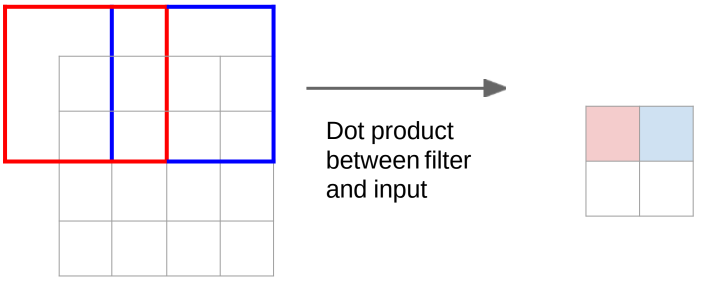


Input: 4 x 4

Output: 2 x 2

可学习的上采样: 转置卷积（Transpose Convolution)

回顾: 3 x 3 卷积, 步长（stride） 2， 零填充（pad） 1



Input: 4 x 4

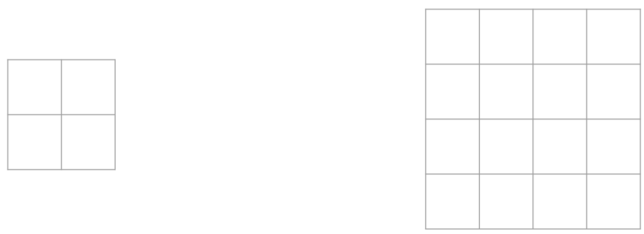
Output: 2 x 2

Filter moves 2 pixels in the input for every one pixel in the output

Stride gives ratio between movement in input and output

可学习的上采样: 转置卷积（Transpose Convolution)

3 x 3 转置卷积（transpose convolution）, stride 2 pad 1

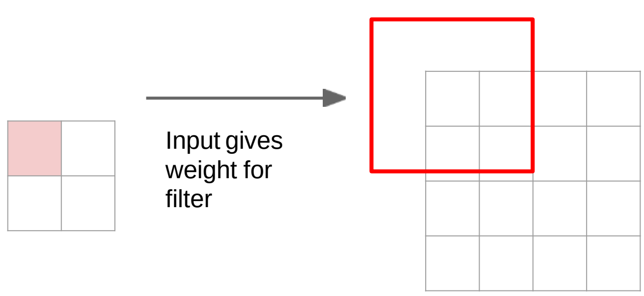


Input: 2 x 2

Output: 4 x 4

可学习的上采样: 转置卷积（Transpose Convolution)

3 x 3 转置卷积（transpose convolution）, stride 2 pad 1

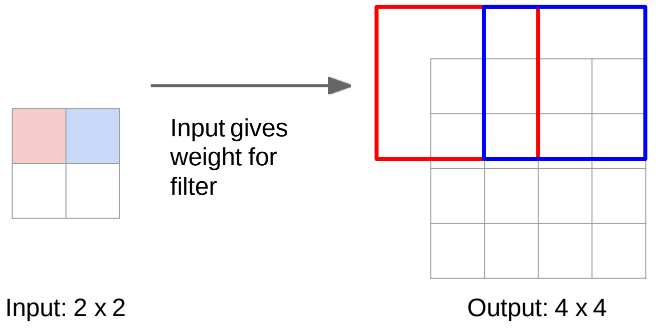


Input: 2 x 2

Output: 4 x 4

可学习的上采样: 转置卷积（Transpose Convolution)

3 x 3 转置卷积（transpose convolution），stride 2 pad 1

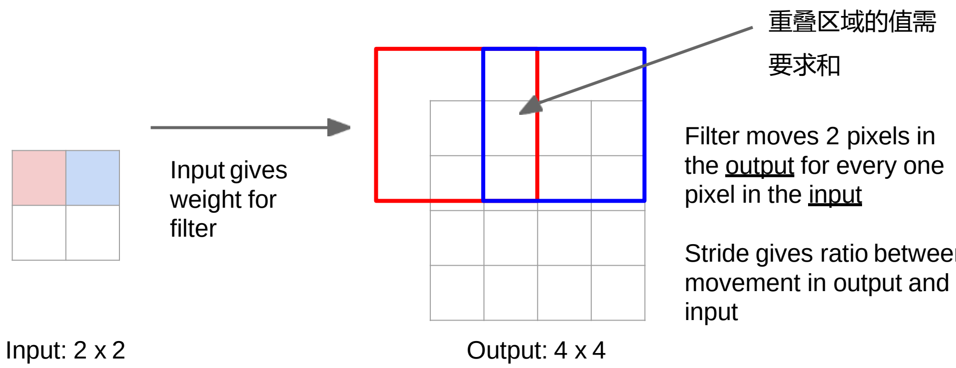


Filter moves 2 pixels in the output for every one pixel in the input

Stride gives ratio between movement in output and input

可学习的上采样: 转置卷积（Transpose Convolution)

3 x 3 转置卷积（transpose convolution），stride 2 pad 1



Filter moves 2 pixels in the output for every one pixel in the input

Stride gives ratio between movement in output and input

可学习的上采样: 转置卷积（Transpose Convolution）

3 x 3 转置卷积（transpose convolution），stride 2 pad 1

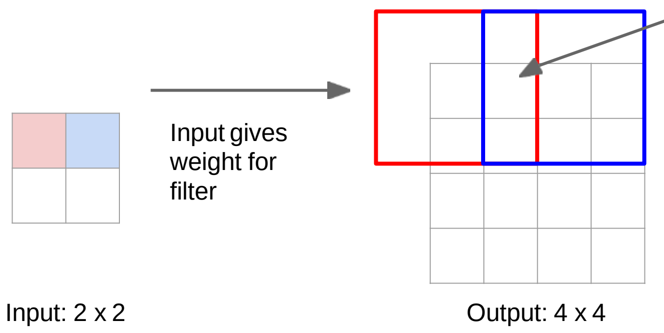
其他叫法:

-Deconvolution (bad)

-Upconvolution

-Fractionally strided convolution

-Backward strided convolution

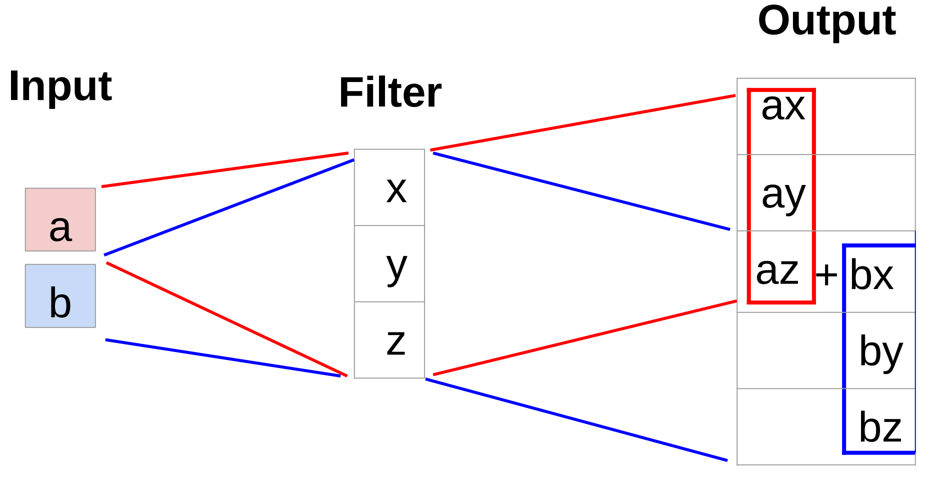


重叠区域的值需要求和

Filter moves 2 pixels in the <u>output</u> for every one pixel in the <u>input</u>

Stride gives ratio between movement in output and input

可学习的上采样: 一维例子



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

卷积与矩阵相乘 (一维例子)

将卷积写为矩阵乘法

$$\vec{x} * \vec{a} = X\vec{a}$$

$$\begin{bmatrix} x & y & x & 0 & 0 & 0 \\ 0 & x & y & x & 0 & 0 \\ 0 & 0 & x & y & x & 0 \\ 0 & 0 & 0 & x & y & x \end{bmatrix} \begin{bmatrix} 0 \\ a \\ b \\ c \\ d \\ 0 \end{bmatrix} = \begin{bmatrix} ay + bz \\ ax + by + cz \\ bx + cy + dz \\ cx + dy \end{bmatrix}$$

例子: 1D 卷积, 卷积核尺寸=3, 步长=1, 零填充=1

卷积与矩阵相乘 (一维例子)

将卷积写为矩阵乘法

$$\vec{x} * \vec{a} = X\vec{a}$$

$$\begin{bmatrix} x & y & x & 0 & 0 & 0 \\ 0 & x & y & x & 0 & 0 \\ 0 & 0 & x & y & x & 0 \\ 0 & 0 & 0 & x & y & x \end{bmatrix} \begin{bmatrix} 0 \\ a \\ b \\ c \\ d \\ 0 \end{bmatrix} = \begin{bmatrix} ay + bz \\ ax + by + cz \\ bx + cy + dz \\ cx + dy \end{bmatrix}$$

例子: 1D 卷积, 卷积核尺寸=3, 步长=1, 零填充=1

Convolution transpose multiplies by the transpose of the same matrix:

$$\vec{x} *^T \vec{a} = X^T\vec{a}$$

$$\begin{bmatrix} x & 0 & 0 & 0 \\ y & x & 0 & 0 \\ z & y & x & 0 \\ 0 & z & y & x \\ 0 & 0 & z & y \\ 0 & 0 & 0 & z \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} ax \\ ay + bx \\ az + by + cx \\ bz + cy + dx \\ cz + dy \\ dz \end{bmatrix}$$

卷积与矩阵相乘 (一维例子)

将卷积写为矩阵乘法

$$\vec{x} * \vec{a} = X\vec{a}$$

$$\begin{bmatrix} x & y & x & 0 & 0 & 0 \\ 0 & 0 & x & y & x & 0 \end{bmatrix} \begin{bmatrix} 0 \\ a \\ b \\ c \\ d \\ 0 \end{bmatrix} = \begin{bmatrix} ay + bz \\ bx + cy + dz \end{bmatrix}$$

例子: 1D 卷积, 卷积核尺寸=3, 步长=2, 零填充=1

卷积与矩阵相乘 (一维例子)

将卷积写为矩阵乘法

$$\vec{x} * \vec{a} = X\vec{a}$$

$$\begin{bmatrix} x & y & x & 0 & 0 & 0 \\ 0 & 0 & x & y & x & 0 \end{bmatrix} \begin{bmatrix} 0 \\ a \\ b \\ c \\ d \\ 0 \end{bmatrix} = \begin{bmatrix} ay + bz \\ bx + cy + dz \end{bmatrix}$$

例子: 1D 卷积, 卷积核尺寸=3, 步长=2, 零填充=1

Convolution transpose multiplies by the transpose of the same matrix:

$$\vec{x} *^T \vec{a} = X^T\vec{a}$$

$$\begin{bmatrix} x & 0 \\ y & 0 \\ z & x \\ 0 & y \\ 0 & z \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} ax \\ ay \\ az + bx \\ by \\ bz \\ 0 \end{bmatrix}$$

语义分割：全卷积神经网络

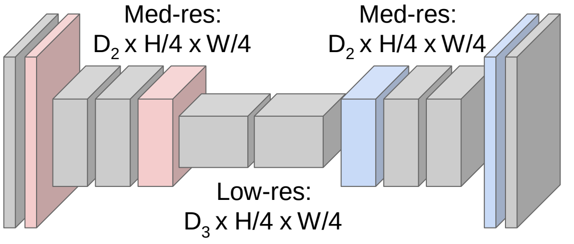
解决方案：让整个网络只包含卷积层，

并在网络中嵌入**下采样**与**上采样**过程。

下采样
Pooling, strided convolution

上采样
Unpooling or strided transpose convolution

Input:
3 x H x W



Predictions:
H x W

Long, Shelhamer, and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015
Noh et al, “Learning Deconvolution Network for Semantic Segmentation”, ICCV 2015

计算机视觉任务

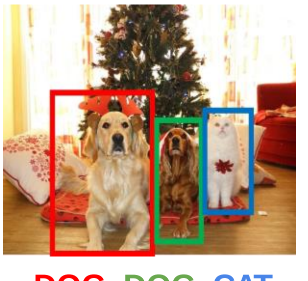
分类	语义分割	目标检测	实例分割

CAT

No spatial extent

GRASS, CAT, TREE, SKY

No objects, just pixels



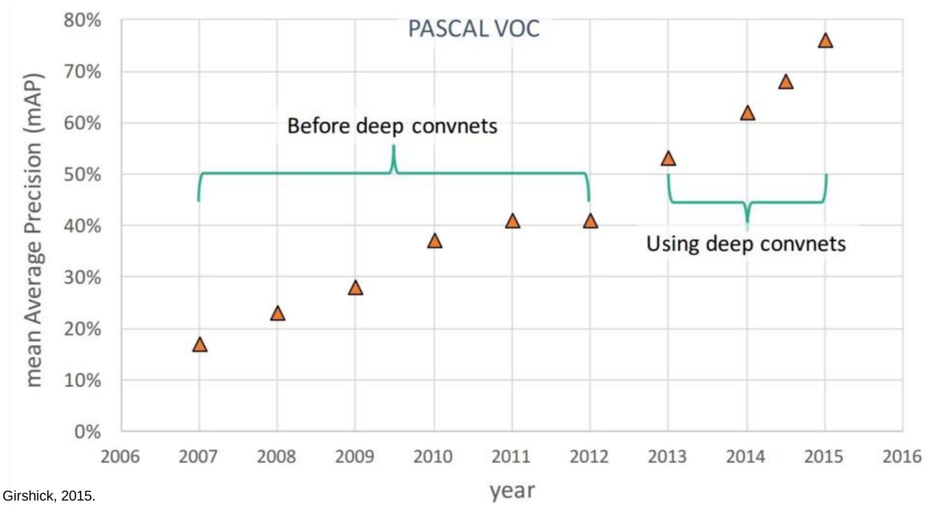
DOG, DOG, CAT

DOG, DOG, CAT

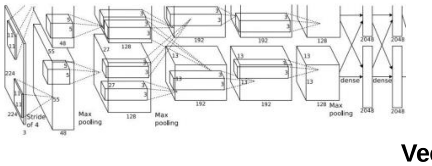
Multiple Object

This image is CC0 public domain

深度学习带来的飞跃



目标检测: 单目标(分类+定位)



Fully Connected: 4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Vector: 4096

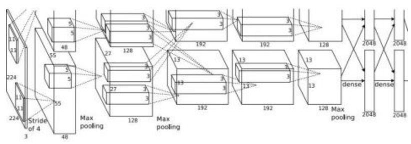
Fully Connected: 4096 to 4

Box Coordinates
(x, y, w, h)

将定位任务建模为回归问题!

目标检测: 单目标(分类+定位)

This image is CC0 public domain



Correct label:
Cat

Fully
Connected:
4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Softmax
Loss

Vector:
4096

Fully
Connected:
4096 to 4

Box
Coordinates
(x, y, w, h)

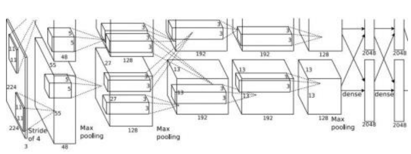
L2 Loss

Correct box:
(x', y', w', h')

将定位任务建模为回归问题!

目标检测: 单目标(分类+定位)

This image is CC0 public domain



Correct label:
Cat

Fully
Connected:
4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Softmax
Loss

多任务损失

\+ → Loss

Vector:
4096

Fully
Connected:
4096 to 4

Box
Coordinates
(x, y, w, h)

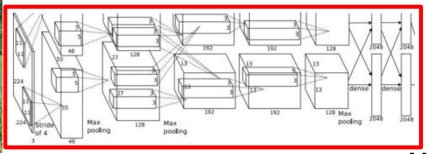
L2 Loss

Correct box:
(x', y', w', h')

将定位任务建模为回归问题!

目标检测: 单目标(分类+定位)

This image is CC0 public domain



常使用在ImageNet上预训练的模型（迁移学习）

Vector: 4096

Fully Connected: 4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Correct label: Cat

Softmax Loss

Fully Connected: 4096 to 4

Box Coordinates (x, y, w, h)

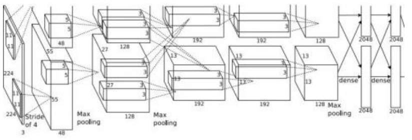
L2 Loss

Correct box: (x', y', w', h')

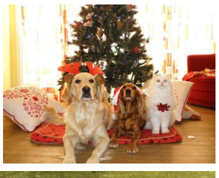
+ → Loss

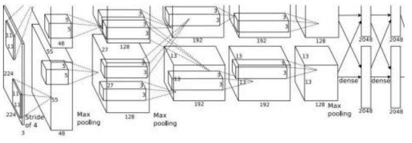
将定位任务建模为回归问题!

目标检测: 多目标

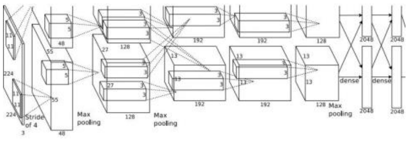


CAT: (x, y, w, h)





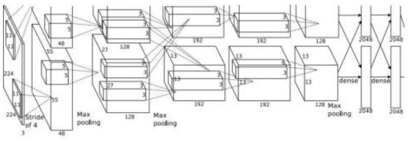
DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)



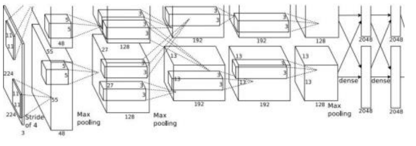
DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
….

目标检测: 多目标

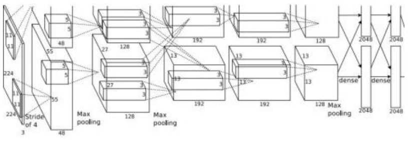
困境：每张图像期望输出的维度都不一样!



CAT: (x, y, w, h)　　4个实数



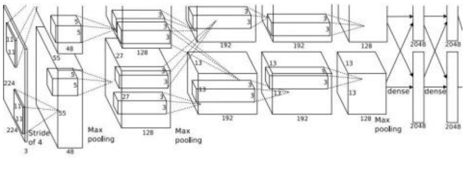
DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)　　16个实数



DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
….　　多少个实数?

目标检测: 多目标

利用CNN对图像中的区域进行多分类，以确定当前区域是背景还是哪个类别的目标。



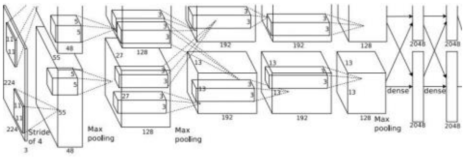
狗? 不是

猫? 不是

背景? 是

目标检测: 多目标

利用CNN对图像中的区域进行多分类，以确定当前区域是背景还是哪个类别的目标。



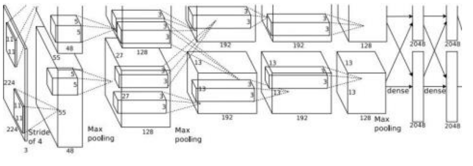
狗? 是

猫? 不是

背景? 不是

目标检测: 多目标

利用CNN对图像中的区域进行多分类，以确定当前区域是背景还是哪个类别的目标。



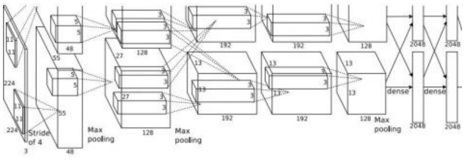
狗? 是

猫? 不是

背景? 不是

目标检测: 多目标

利用CNN对图像中的区域进行多分类，以确定当前区域是背景还是哪个类别的目标。

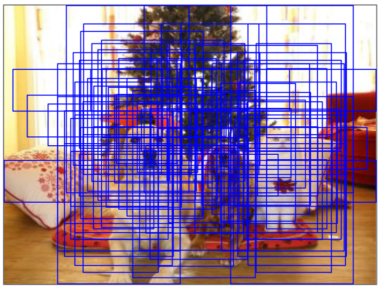


狗? 不是

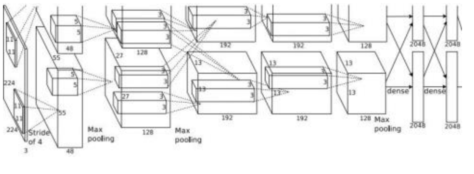
猫? 是

背景? 不是

目标检测: 多目标



利用CNN对图像中的区域进行多分类，以确定当前区域是背景还是哪个类别的目标。



狗? 不是

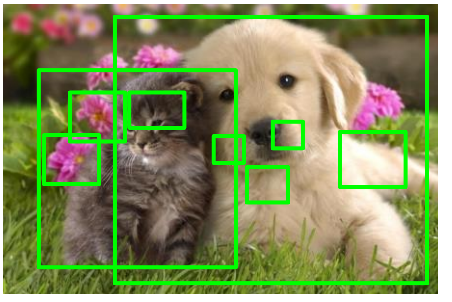
猫? 是

背景? 不是

困境： CNN需要对图像中所有可能的区域（不同位置、尺寸、长宽比）进行分类，计算量巨大！

区域建议: Selective Search

- 找出所有潜在可能包含目标的区域；
- 运行速度需要相对较快；比如， Selective Search在CPU上仅需要运行几秒钟就可以产生2000个候选区域。



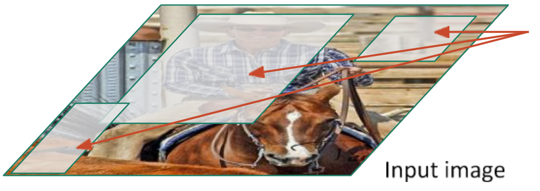
Alexe et al, "Measuring the objectness of image windows", TPAMI 2012
Uijlings et al, "Selective Search for Object Recognition", IJCV 2013
Cheng et al, "BING: Binarized normed gradients for objectness estimation at 300fps", CVPR 2014
Zitnick and Dollar, "Edge boxes: Locating object proposals from edges", ECCV 2014

R-CNN

Input image

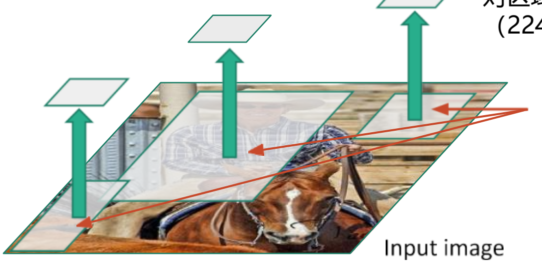
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

R-CNN



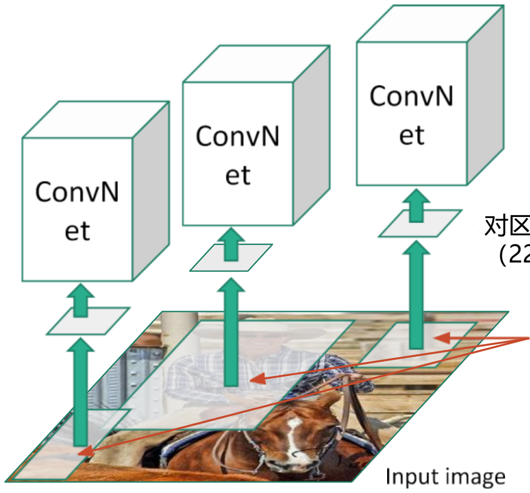
Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

R-CNN



Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

R-CNN



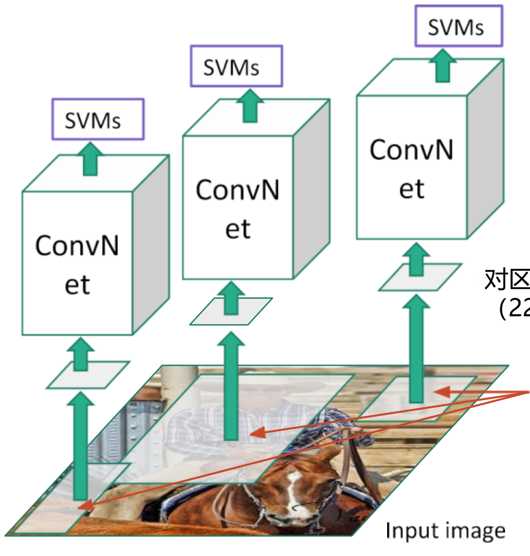
将图像区域送入卷积网络进行特征提取

对区域进行缩放（224x224）

利用区域建议方法产生的感兴趣区域（~2k）

Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

R-CNN



使用支持向量机对区域进行分类

将图像区域送入卷积网络进行特征提取

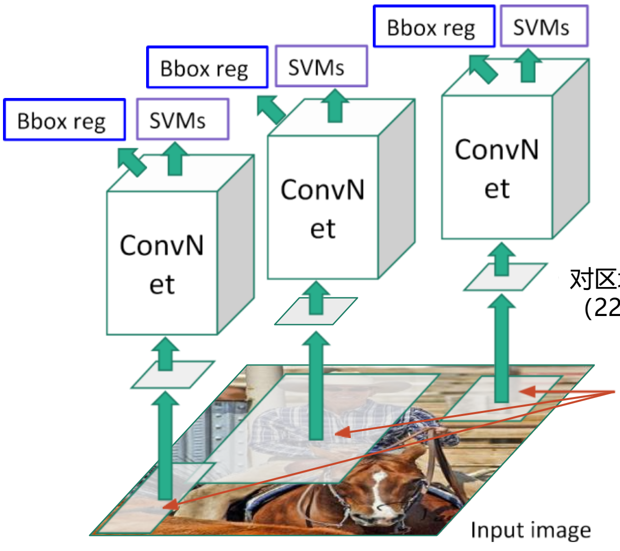
对区域进行缩放（224x224）

利用区域建议方法产生的感兴趣区域（~2k）

Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

R-CNN

Predict “corrections” to the RoI: 4 numbers: (dx, dy, dw, dh)



使用支持向量机
对区域进行分类

将图像区域送入
卷积网络进行特
征提取

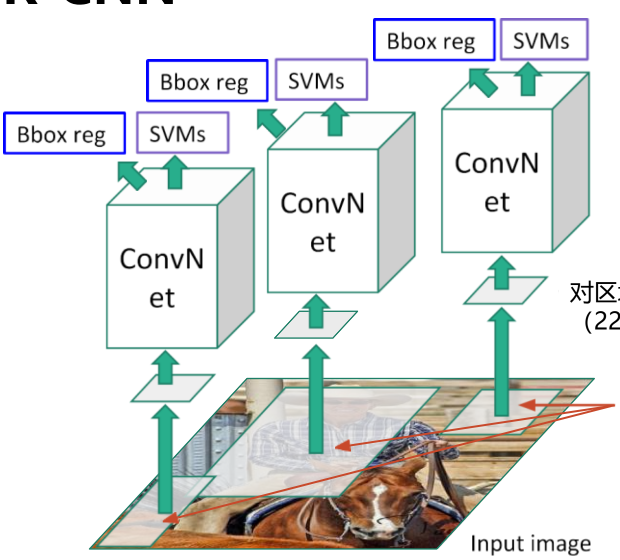
对区域进行缩放
（224x224）

利用区域建议方
法产生的感兴趣
区域（~2k）

Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

R-CNN

Predict “corrections” to the RoI: 4 numbers: (dx, dy, dw, dh)



使用支持向量机
对区域进行分类

将图像区域送入
卷积网络进行特
征提取

对区域进行缩放
（224x224）

利用区域建议方
法产生的感兴趣
区域（~2k）

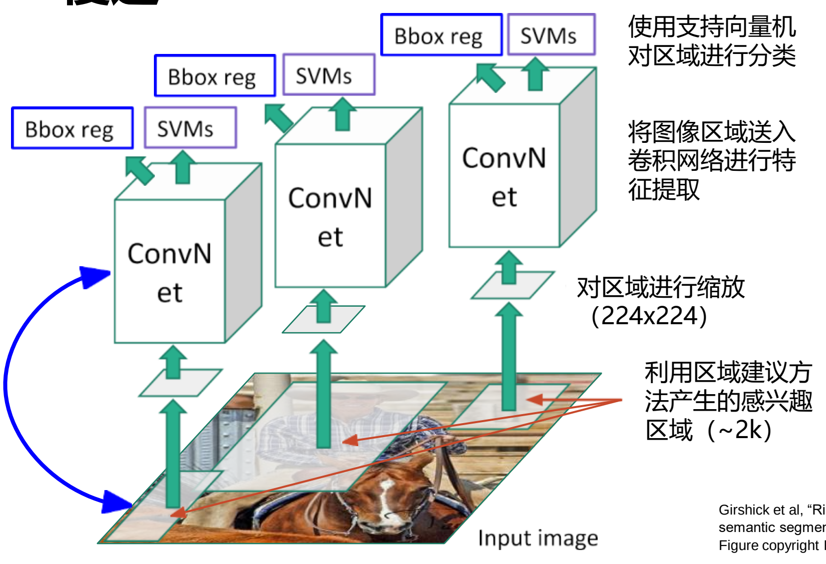
问题:计算效率低下! 每一张图像大约有2k个区域需要卷积网络进行特征提取，重叠区域反复计算。

想法：在特征图上进行区域扣取

Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

“慢速” R-CNN

Predict “corrections” to the RoI: 4 numbers: (dx, dy, dw, dh)

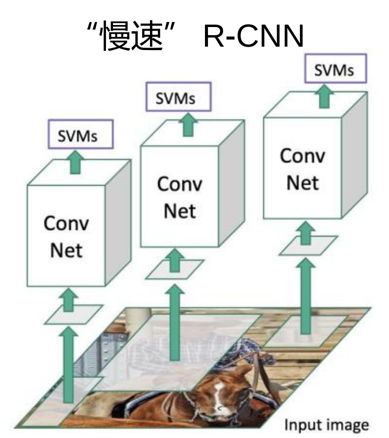


问题:计算效率低下! 每一张图像大约有2k个区域需要卷积网络进行特征提取，重叠区域反复计算。

想法：在特征图上进行区域扣取

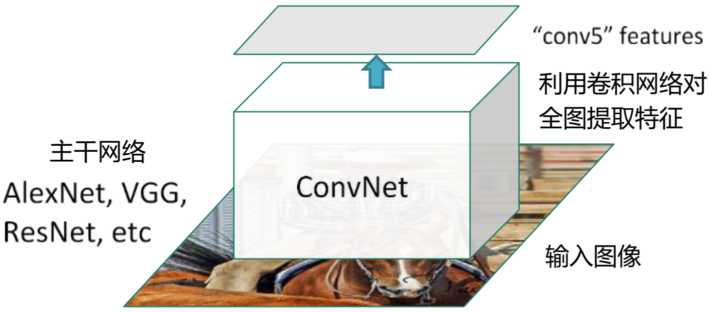
Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Fast R-CNN

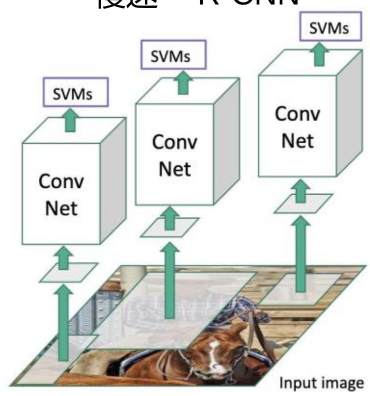


Girshick, “Fast R-CNN”, ICCV 2015. Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Fast R-CNN

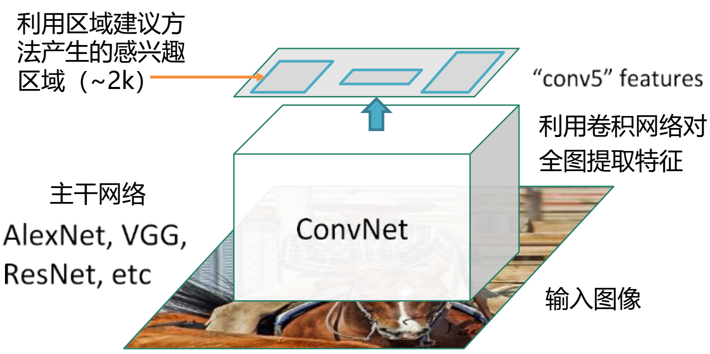


"慢速" R-CNN

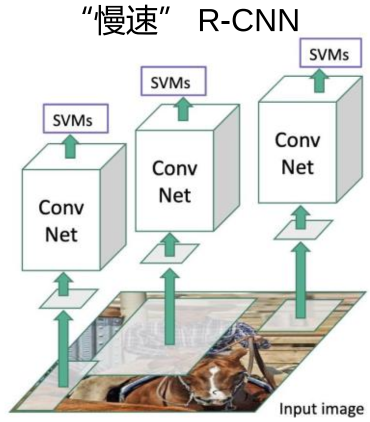


Girshick, "Fast R-CNN", ICCV 2015. Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Fast R-CNN



"慢速" R-CNN



Girshick, "Fast R-CNN", ICCV 2015. Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

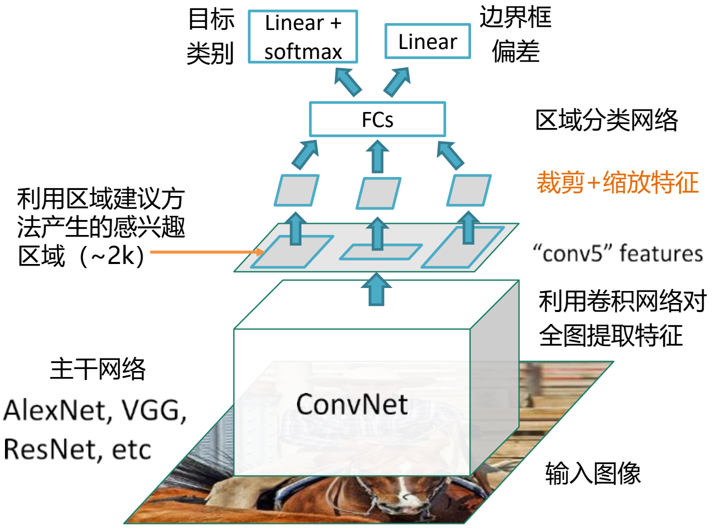
Fast R-CNN

Girshick, "Fast R-CNN", ICCV 2015. Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

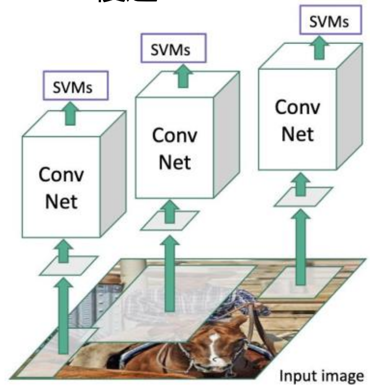
Fast R-CNN

Girshick, "Fast R-CNN", ICCV 2015. Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Fast R-CNN

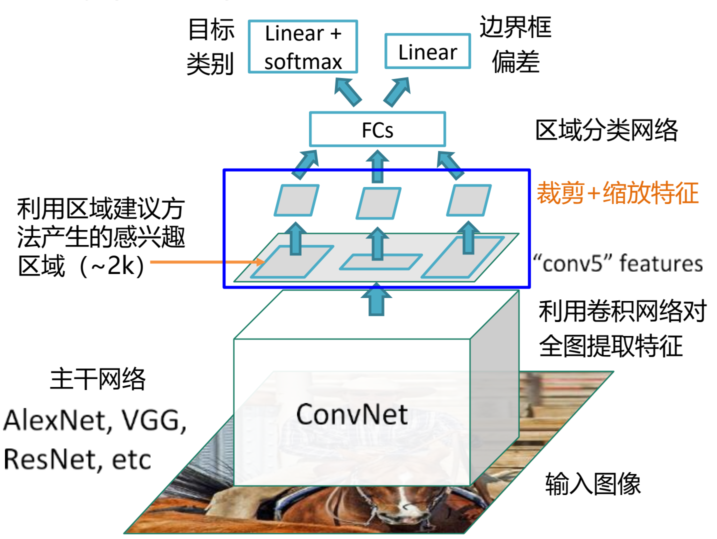


"慢速" R-CNN

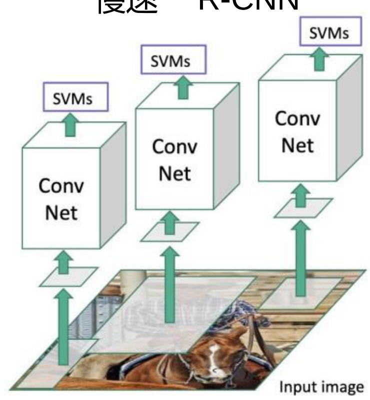


Girshick, "Fast R-CNN", ICCV 2015. Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

Fast R-CNN

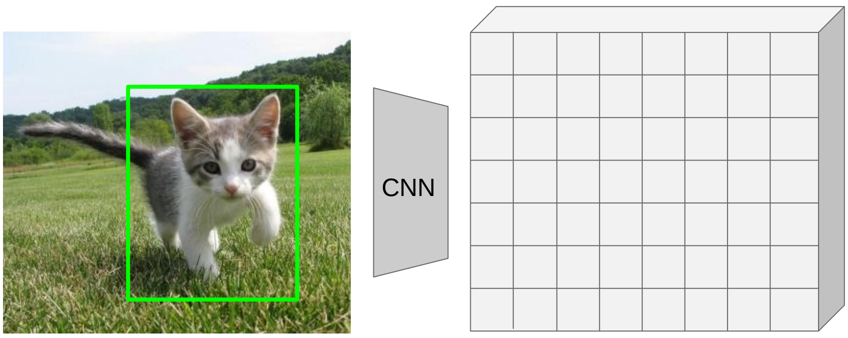


"慢速" R-CNN



Girshick, "Fast R-CNN", ICCV 2015. Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

区域裁剪: RoI Pool



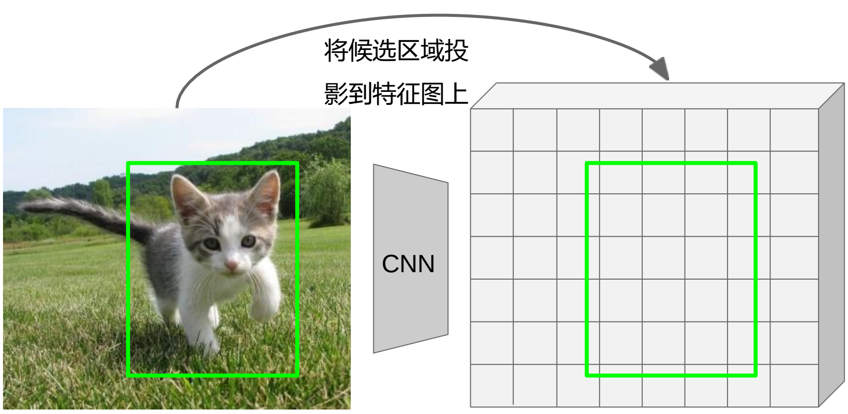
Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

Girshick, "Fast R-CNN", ICCV 2015.

Girshick, "Fast R-CNN", ICCV 2015.

区域裁剪: RoI Pool



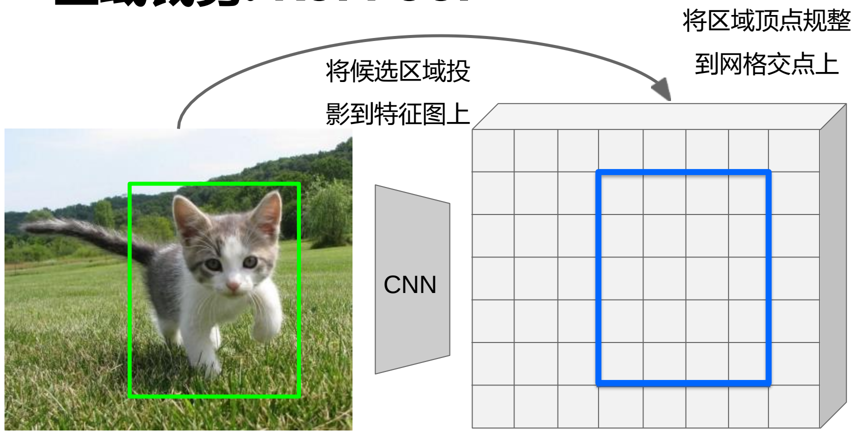
Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

Girshick, "Fast R-CNN", ICCV 2015.

Girshick, "Fast R-CNN", ICCV 2015.

区域裁剪: RoI Pool



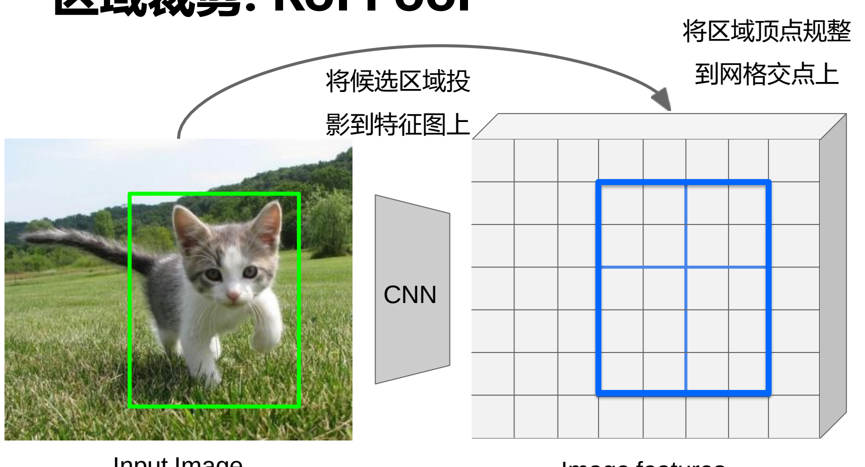
Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

Girshick, "Fast R-CNN", ICCV 2015.

区域裁剪: RoI Pool

将其粗略的分成面积
相等的2x2个子区域。

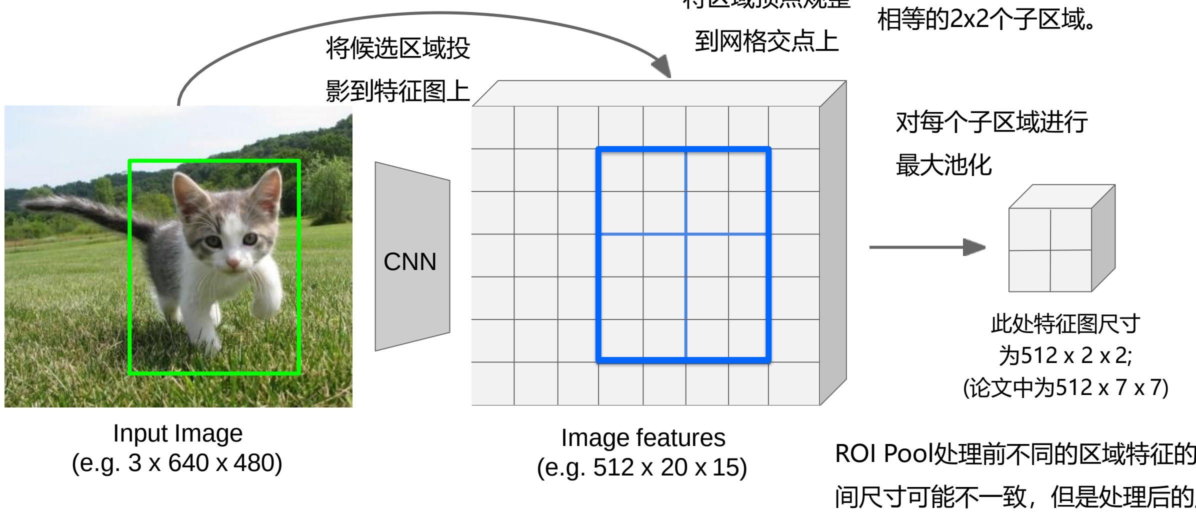


Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

Girshick, "Fast R-CNN", ICCV 2015.

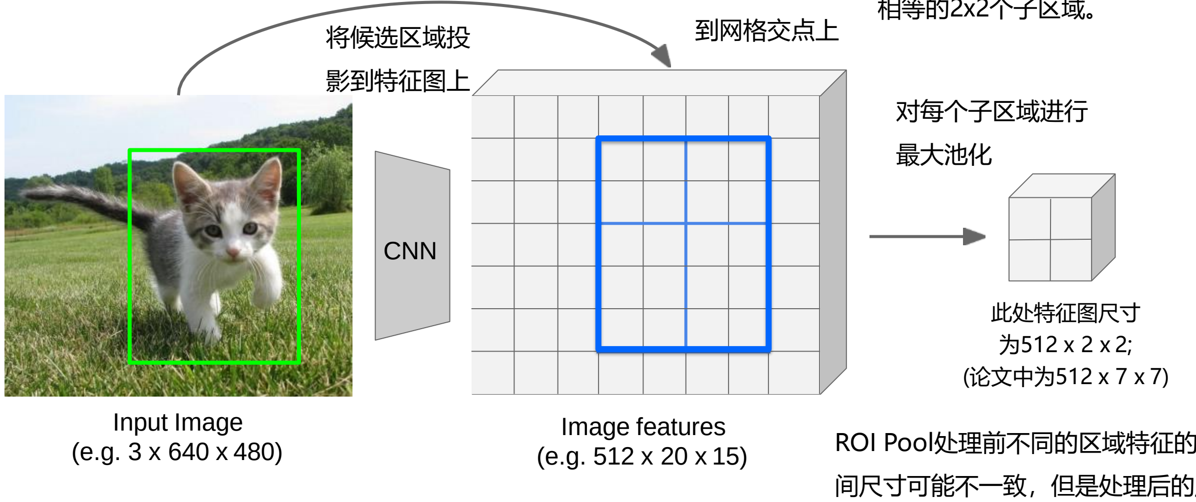
区域裁剪: RoI Pool



ROI Pool处理前不同的区域特征的空间尺寸可能不一致，但是处理后的所有区域特征尺寸都是一样的。

Girshick, “Fast R-CNN”, ICCV 2015.

区域裁剪: RoI Pool

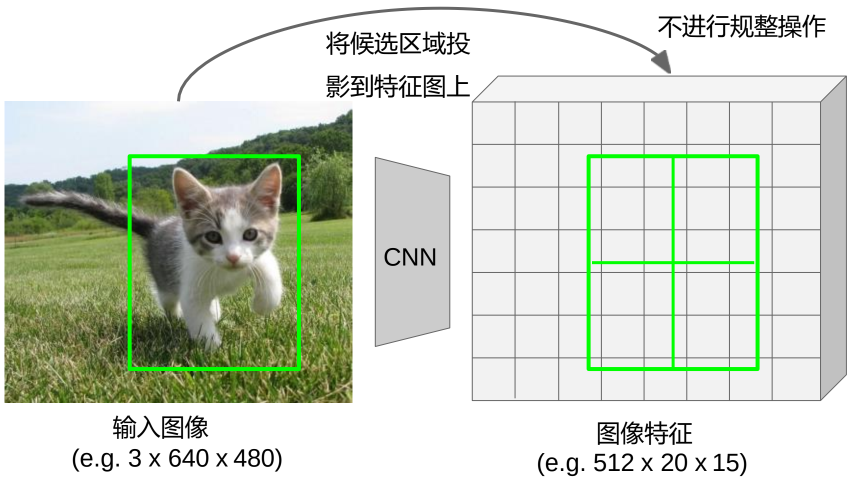


ROI Pool处理前不同的区域特征的空间尺寸可能不一致，但是处理后的所有区域特征尺寸都是一样的。

Girshick, “Fast R-CNN”, ICCV 2015.

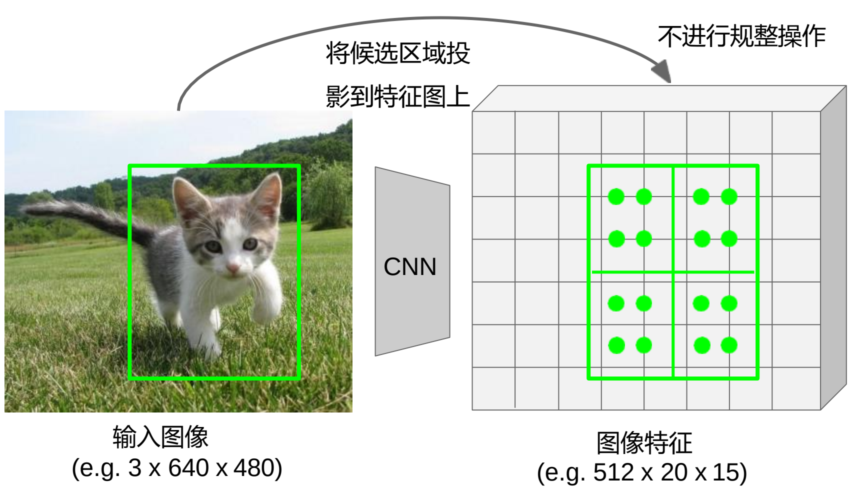
问题: 处理后的区域特征会有轻微的对不齐!

区域裁剪: RoI Align



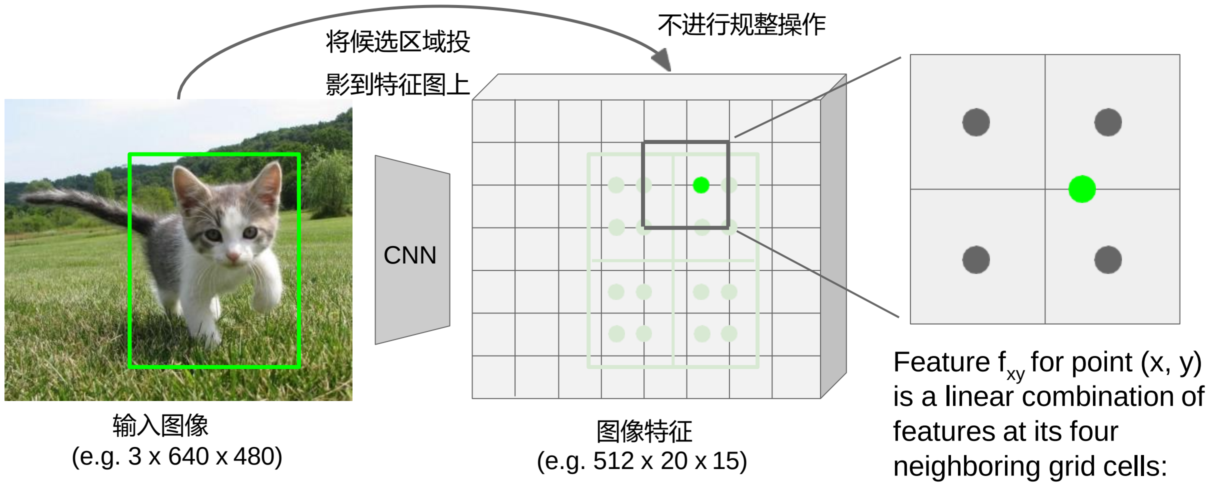
He et al, "Mask R-CNN", ICCV 2017

区域裁剪: RoI Align



He et al, "Mask R-CNN", ICCV 2017 **Sample at regular points in each subregion using bilinear interpolation**

区域裁剪: RoI Align



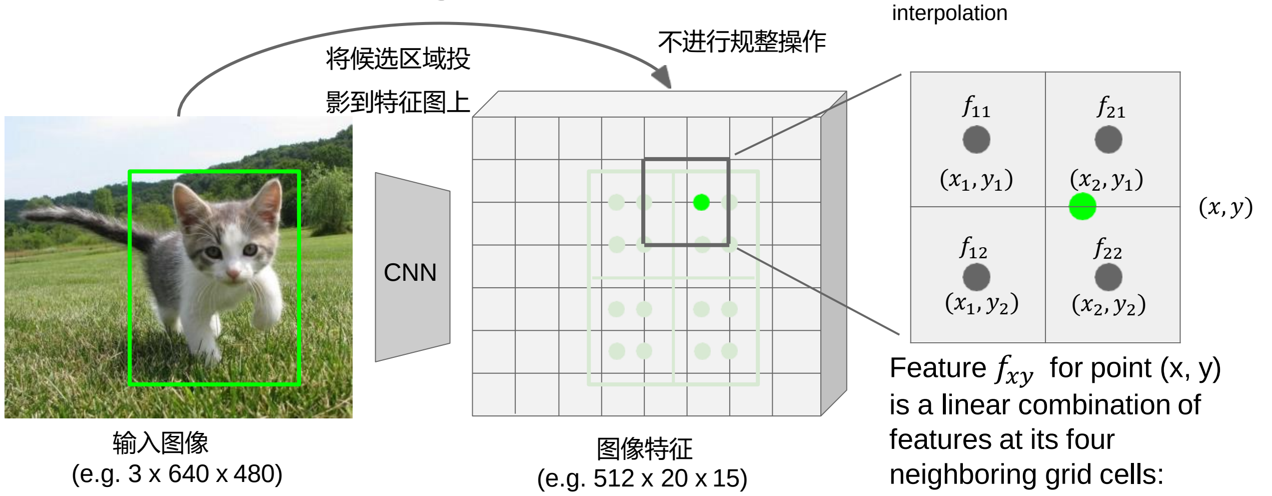
Feature f_{xy} for point (x, y) is a linear combination of features at its four neighboring grid cells:

He et al, "Mask R-CNN", ICCV 2017

Sample at regular points in each subregion using bilinear interpolation

区域裁剪: RoI Align

Sample at regular points in each subregion using bilinear interpolation

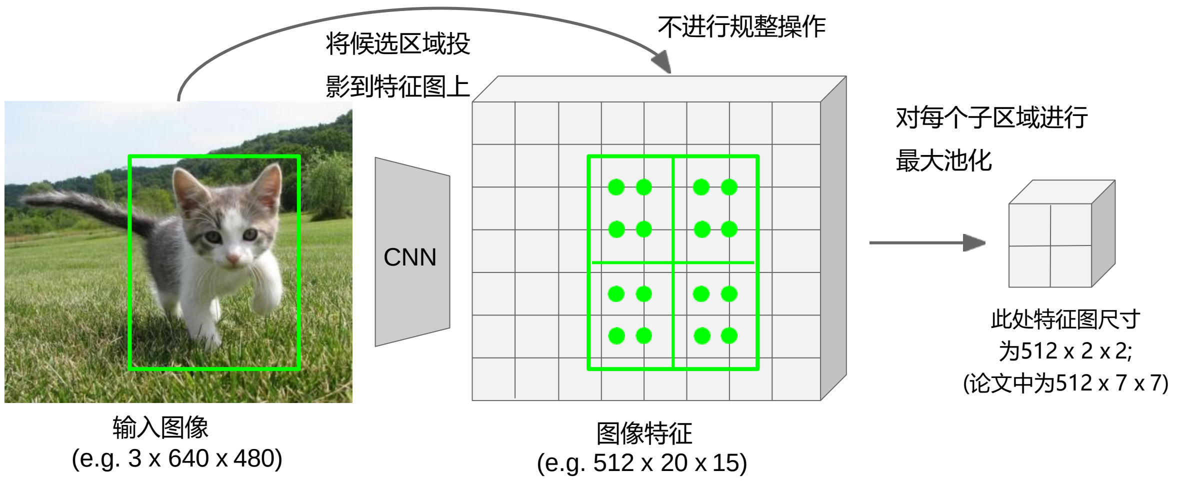


Feature f_{xy} for point (x, y) is a linear combination of features at its four neighboring grid cells:

$$f_{xy} = \sum_{i,j=1}^{2} f_{i,j} \max(0, 1 - |x - x_i|) \max(0, 1 - |y - y_j|)$$

He et al, "Mask R-CNN", ICCV 2017

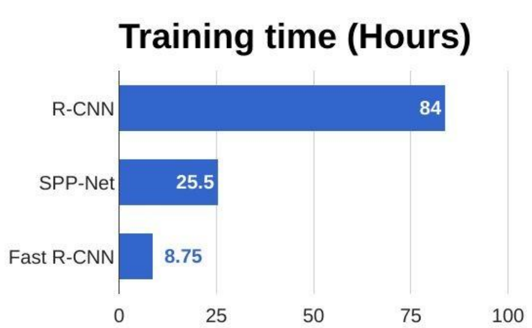
区域裁剪: RoI Align

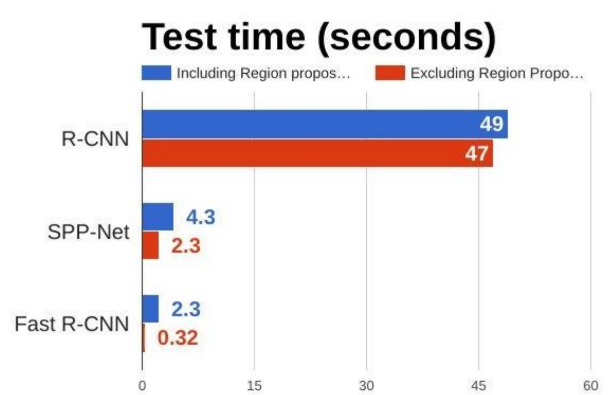


He et al, "Mask R-CNN", ICCV 2017

Sample at regular points in each subregion using bilinear interpolation

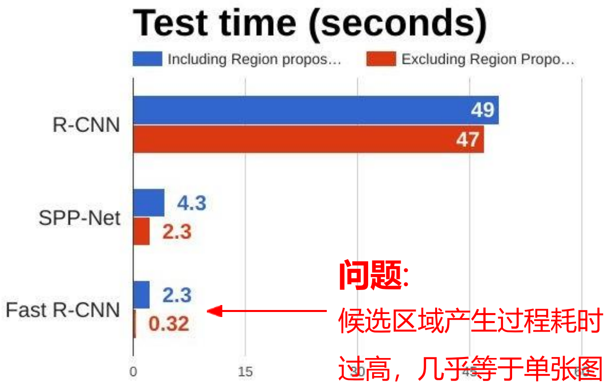
R-CNN vs Fast R-CNN





Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
He et al, "Spatial pyramid pooling in deep convolutional networks for visual recognition", ECCV 2014
Girshick, "Fast R-CNN", ICCV 2015

R-CNN vs Fast R-CNN



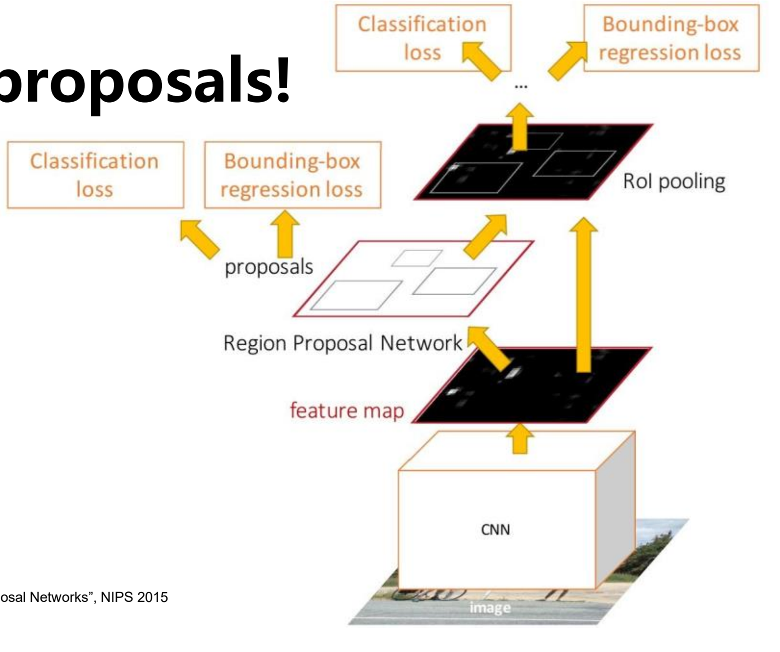
问题:
候选区域产生过程耗时过高，几乎等于单张图片的检测时间

Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
He et al, "Spatial pyramid pooling in deep convolutional networks for visual recognition", ECCV 2014
Girshick, "Fast R-CNN", ICCV 2015

Faster R-CNN: Make CNN do proposals!

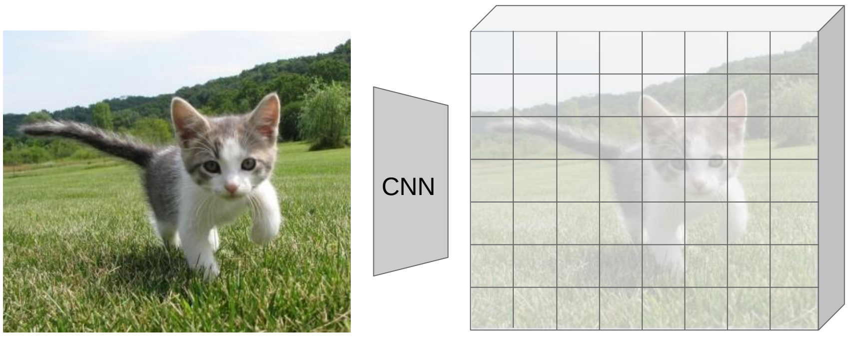
在中间特征层后加入区域建议网络**RPN (Region Proposal Network)** 产生候选区域

其他部分保持与Fast R-CNN一致，即扣取每个候选区域的特征，然后对其进行分类。



Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015
Figure copyright 2015, Ross Girshick; reproduced with permission

区域建议（Region Proposal Network）

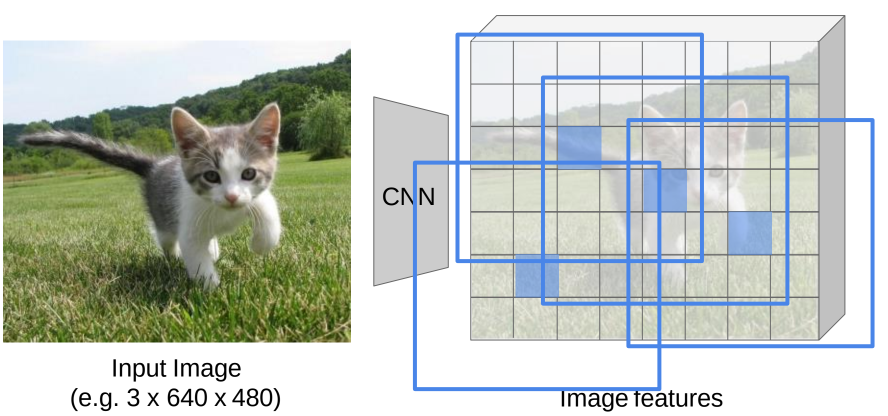


Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

区域建议（Region Proposal Network）

Imagine an **anchor box** of fixed size at each point in the feature map

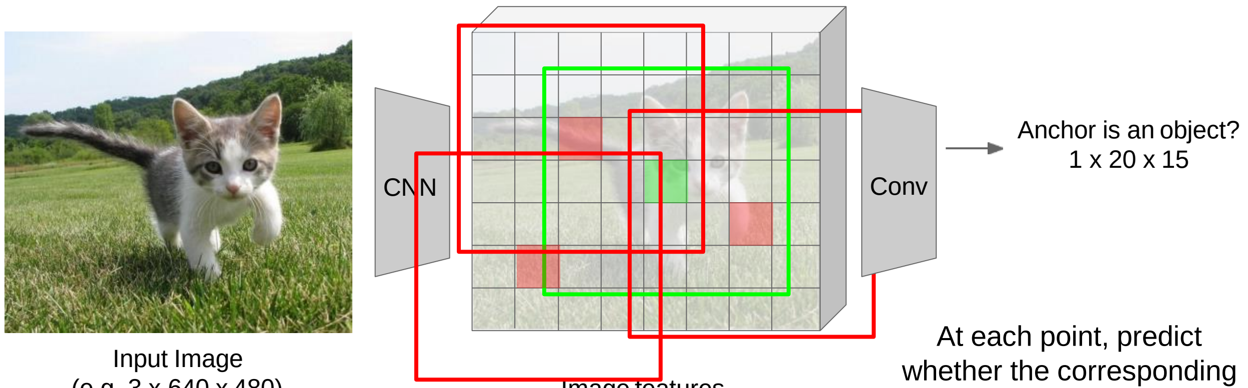


Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

区域建议（Region Proposal Network）

Imagine an **anchor box** of fixed size at each point in the feature map



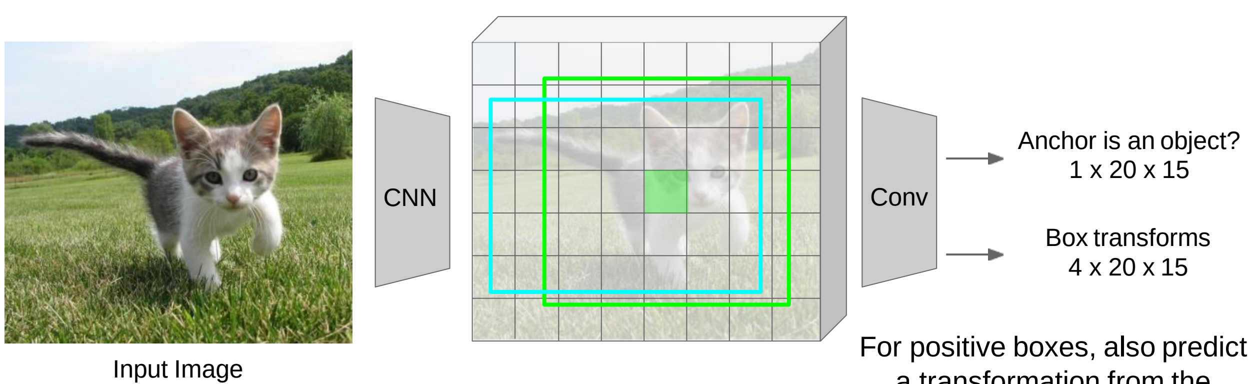
Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

At each point, predict whether the corresponding anchor contains an object (per-pixel logistic regression)

区域建议（Region Proposal Network）

Imagine an **anchor box** of fixed size at each point in the feature map



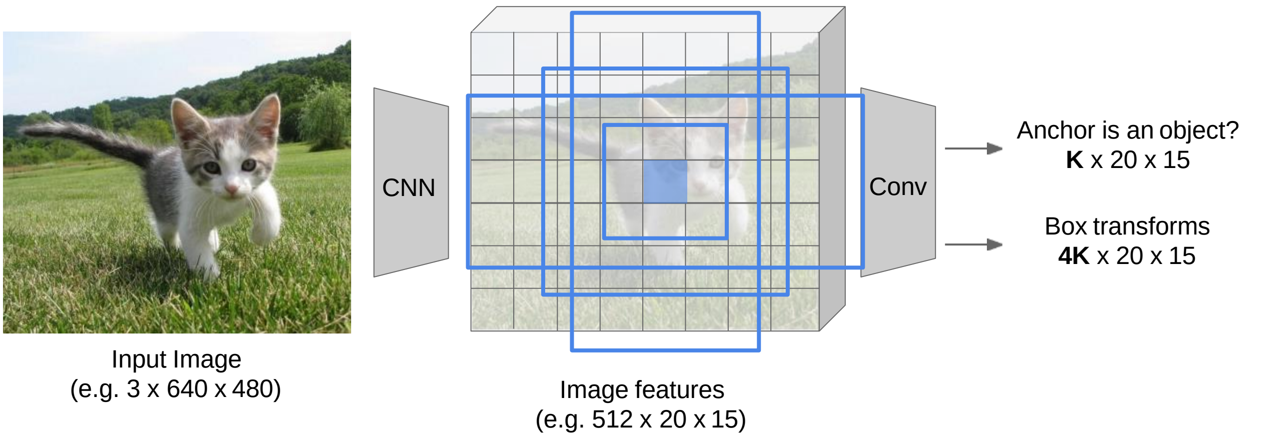
Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

For positive boxes, also predict a transformation from the anchor to the ground-truth box (regress 4 numbers per pixel)

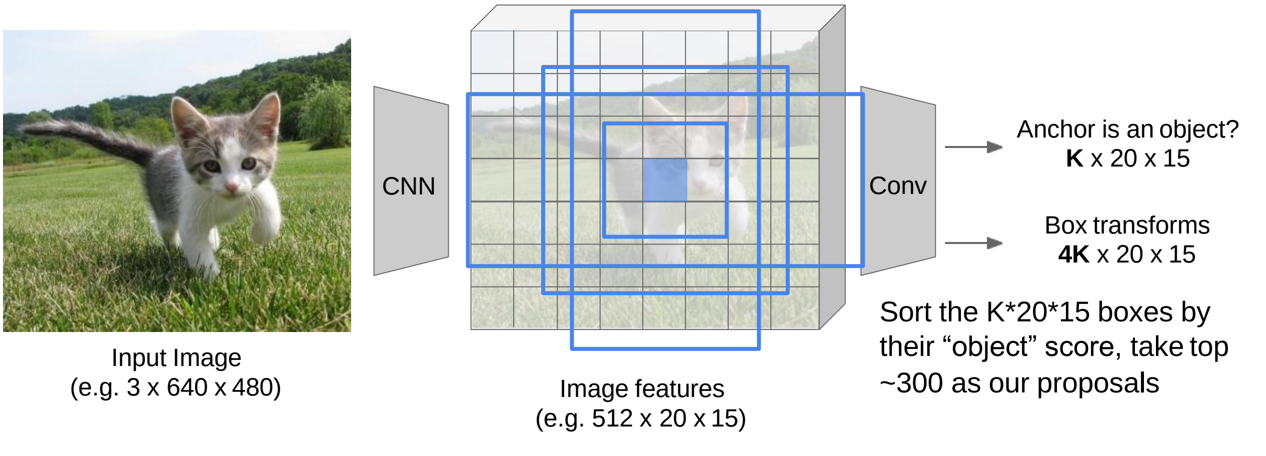
区域建议（Region Proposal Network）

实际使用中，对于每个特征图上的每个位置，我们通常会采用k个不同尺寸和分辨率的锚点区域（anchor boxes）



区域建议（Region Proposal Network）

实际使用中，对于每个特征图上的每个位置，我们通常会采用k个不同尺寸和分辨率的锚点区域（anchor boxes）



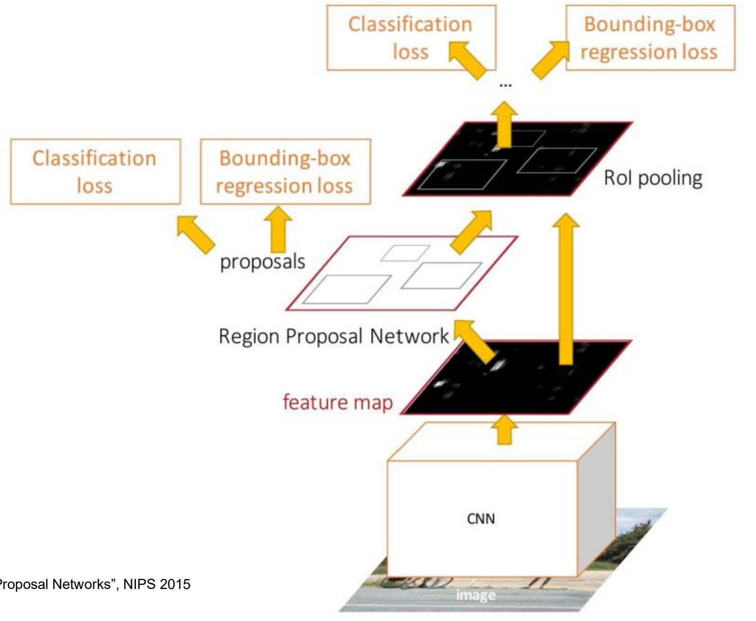
Sort the K*20*15 boxes by their “object” score, take top ~300 as our proposals

Faster R-CNN

利用卷积网络产生候选区域!

四种损失联合训练:

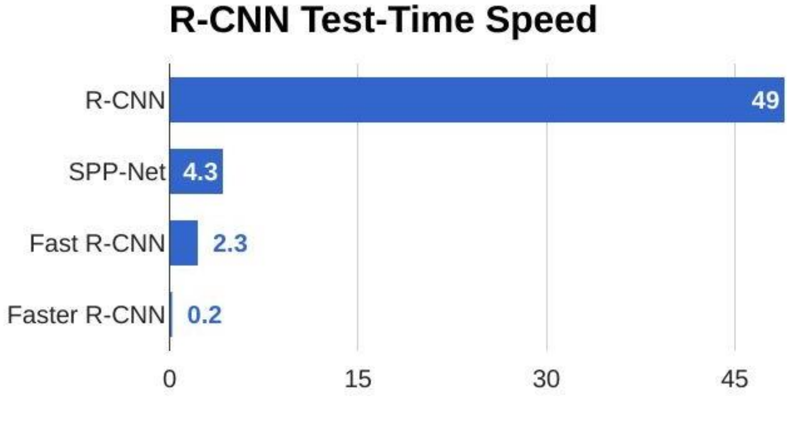
- RPN分类损失(目标/非目标)
- RPN边界框坐标回归损失
- 候选区域分类损失
- 最终边界框坐标回归损失



Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015
Figure copyright 2015, Ross Girshick; reproduced with permission

Faster R-CNN

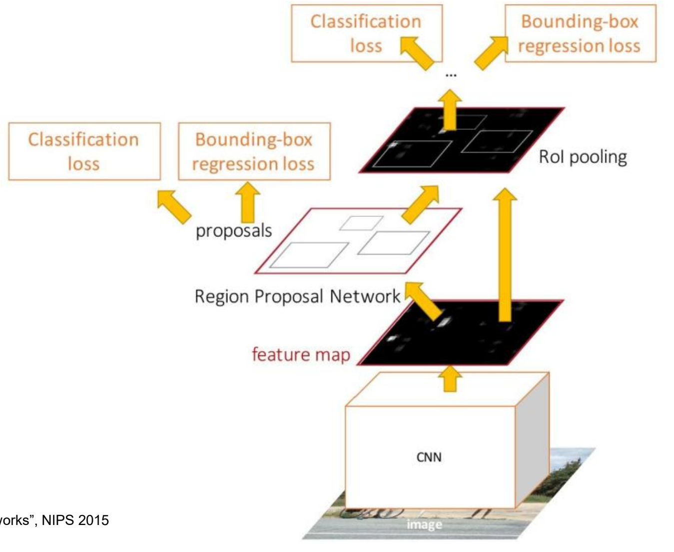
利用卷积网络产生候选区域!



Faster R-CNN

利用卷积网络产生候选区域!

Glossing over many details:
- Ignore overlapping proposals with **non-max suppression**
- How to determine whether a proposal is positive or negative?
- How many positives / negatives to send to second stage?
- How to parameterize bounding box regression?



Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015
Figure copyright 2015, Ross Girshick; reproduced with permission

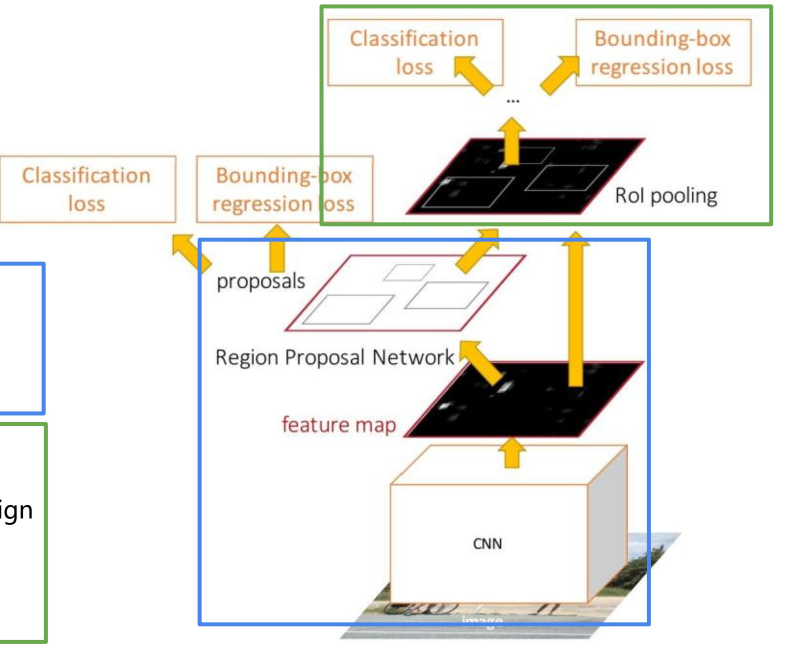
Faster R-CNN

利用卷积网络产生候选区域!

Faster R-CNN是一个两阶段目标检测器

第一阶段: 每张图运行一次
- 主干网络(Backbone)
- 区域建议网络(RPN)

第二阶段: 每个区域运行一次
- 扣取区域特征: RoI pool / align
- 预测目标类别
- 预测边界框偏移量



Faster R-CNN

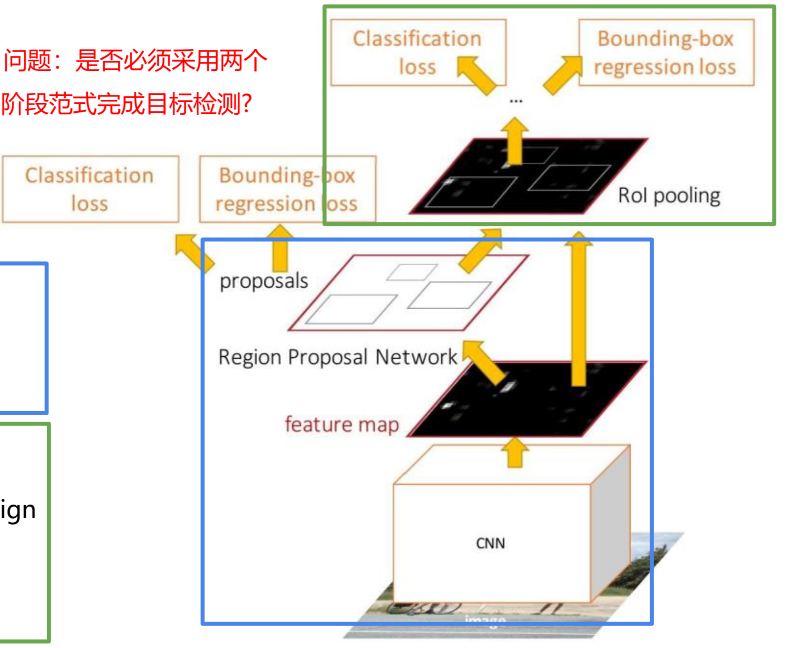
利用卷积网络产生候选区域!

Faster R-CNN是一个两阶段目标检测器

第一阶段: 每张图运行一次
- 主干网络(Backbone)
- 区域建议网络(RPN)

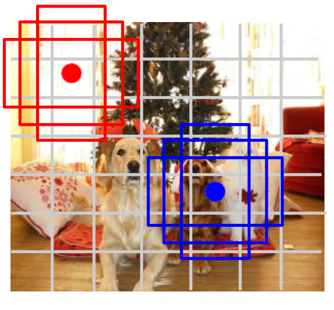
第二阶段: 每个区域运行一次
- 扣取区域特征: RoI pool / align
- 预测目标类别
- 预测边界框偏移量

问题：是否必须采用两个阶段范式完成目标检测?



一阶段目标检测: YOLO / SSD / RetinaNet

Input image
3 x H x W



Divide image into grid
7 x 7

Image a set of **base boxes** centered at each grid cell
Here B = 3

Within each grid cell:
- Regress from each of the B base boxes to a final box with 5 numbers: (dx, dy, dh, dw, confidence)
- Predict scores for each of C classes (including background as a class)
- Looks a lot like RPN, but category-specific!

Output:
7 x 7 x (5 * B + C)

Redmon et al, "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016
Liu et al, "SSD: Single-Shot MultiBox Detector", ECCV 2016
Lin et al, "Focal Loss for Dense Object Detection", ICCV 2017

目标检测: 影响精度的因素 ...

主干网络:

VGG16
ResNet-101
Inception V2
Inception V3
Inception
ResNet
MobileNet

基础架构:

- 两阶段: Faster R-CNN
- 一阶段: YOLO / SSD
- 混合： R-FCN

图像尺寸
区域建议个数
...

一些经验性的结论:

- Faster R-CNN速度偏慢，但精度高
- SSD速度快，相对于Faster R-CNN精度有所欠缺
- 主干网越宽、深度越深，对性能的帮助就越大

Huang et al, “Speed/accuracy trade-offs for modern convolutional object detectors”, CVPR 2017

R-FCN: Dai et al, “R-FCN: Object Detection via Region-based Fully Convolutional Networks”, NIPS 2016
Inception-V2: Ioffe and Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”, ICML 2015
Inception V3: Szegedy et al, “Rethinking the Inception Architecture for Computer Vision”, arXiv 2016
Inception ResNet: Szegedy et al, “Inception-V4, Inception-ResNet and the Impact of Residual Connections on Learning”, arXiv 2016
MobileNet: Howard et al, “Efficient Convolutional Neural Networks for Mobile Vision Applications”, arXiv 2017

视觉识别任务

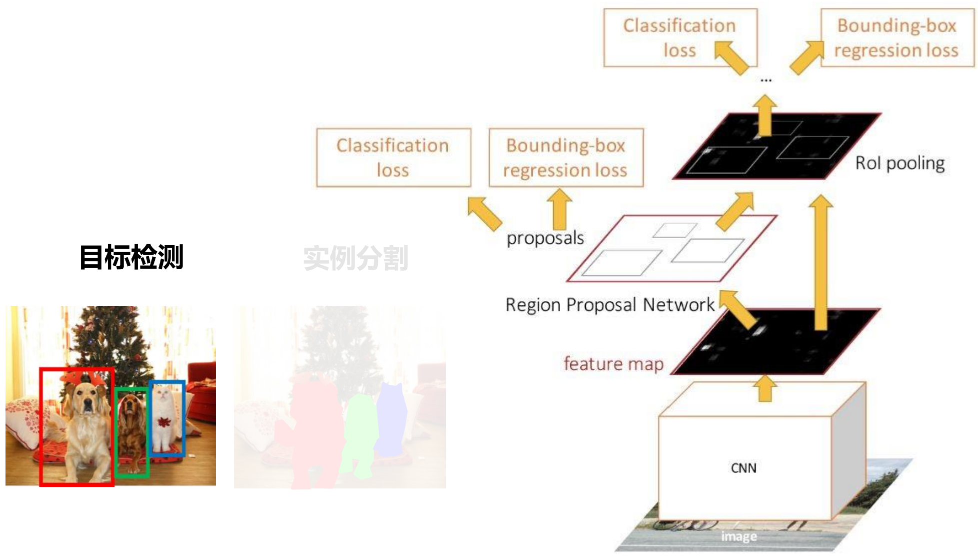
分类 语义分割 目标检测 **实例分割**

猫 | 草，猫，树，天空 | 狗，狗，猫 | 狗，狗，猫

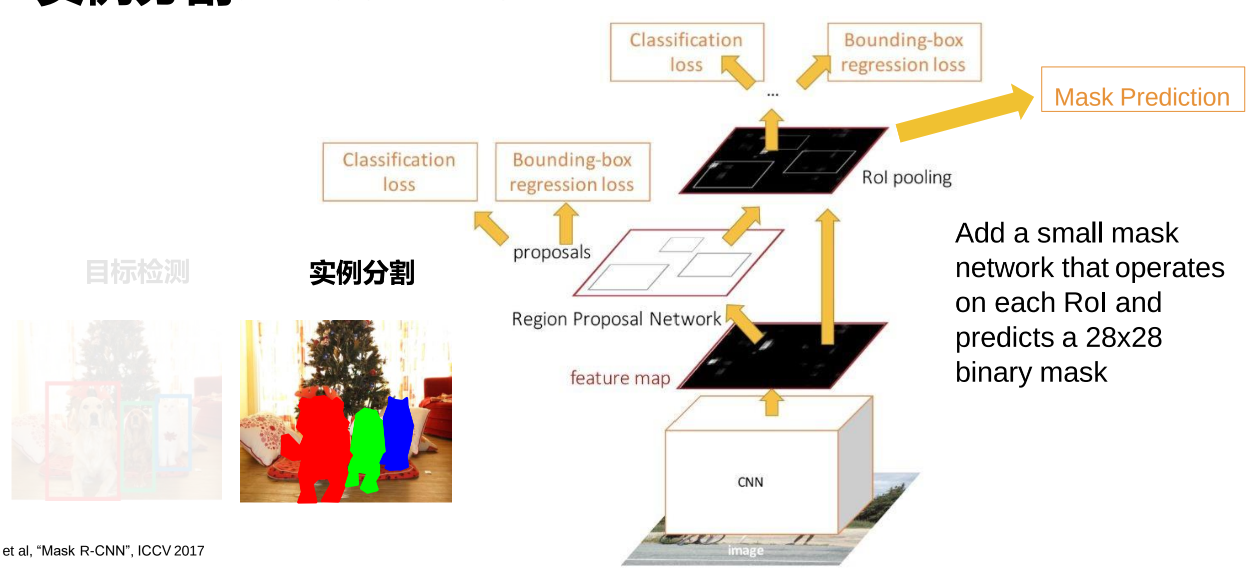
不考虑空间位置 | 像素的类别 | **多目标**

This image is CC0 public domain

目标检测：Faster R-CNN



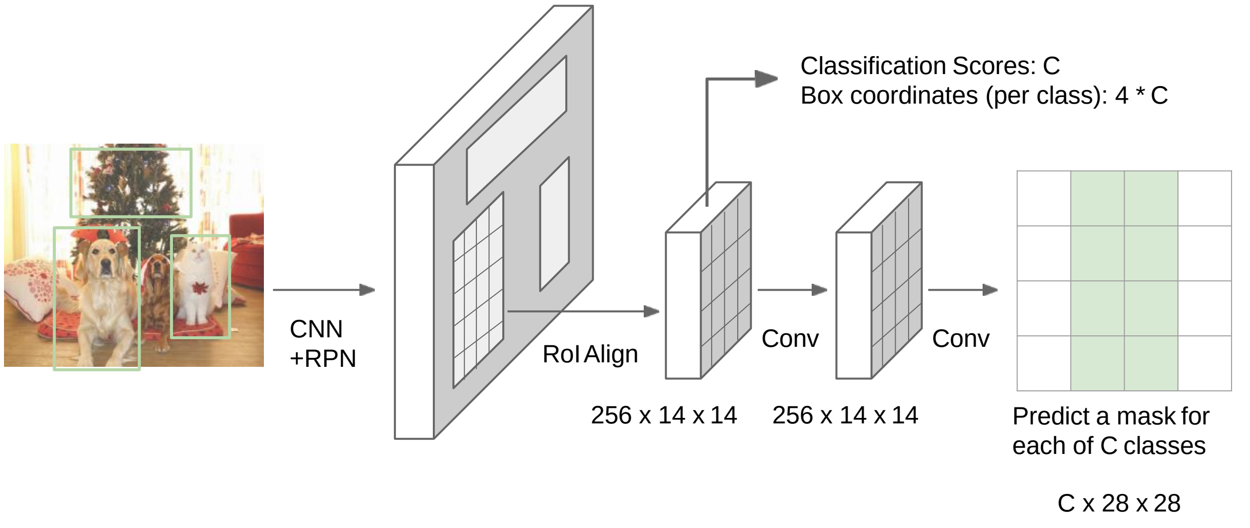
实例分割：Mask R-CNN



Add a small mask network that operates on each RoI and predicts a 28x28 binary mask

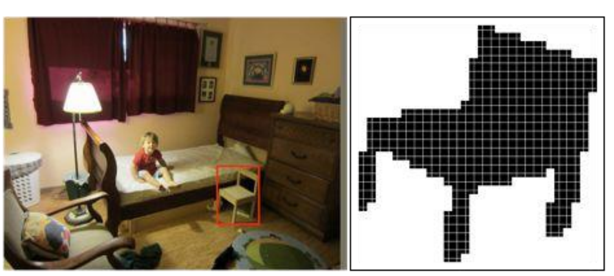
He et al, “Mask R-CNN”, ICCV 2017

Mask R-CNN

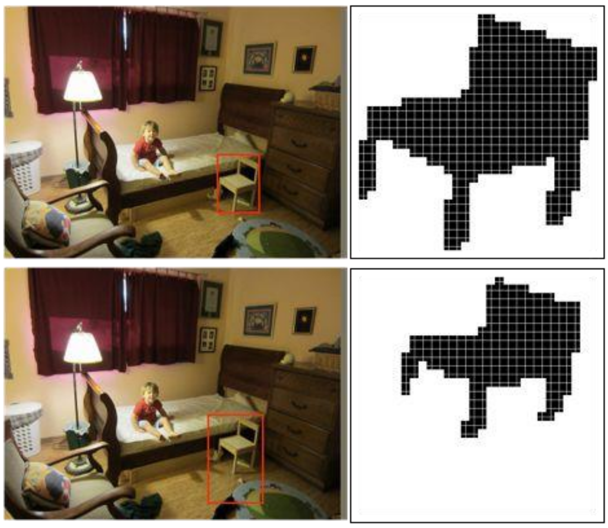


He et al, "Mask R-CNN", arXiv 2017

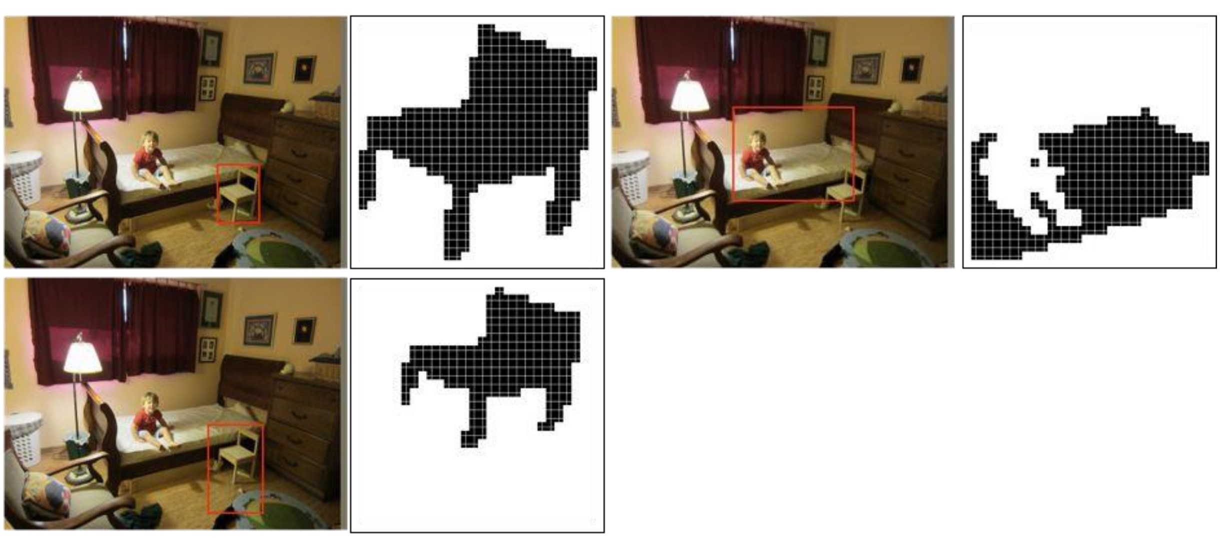
Mask R-CNN训练阶段使用的Mask样例



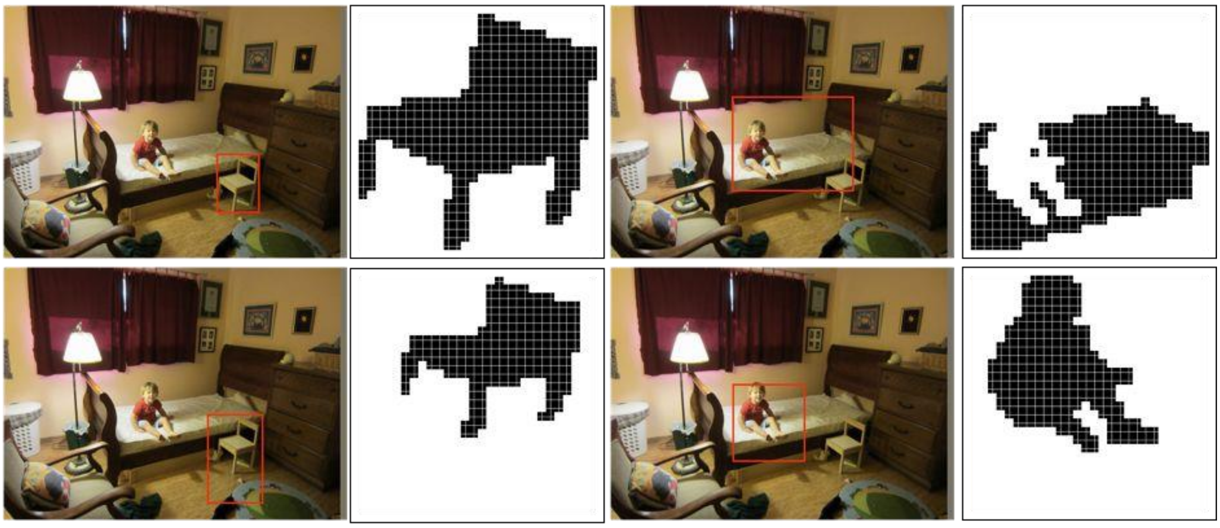
Mask R-CNN训练阶段使用的Mask样例



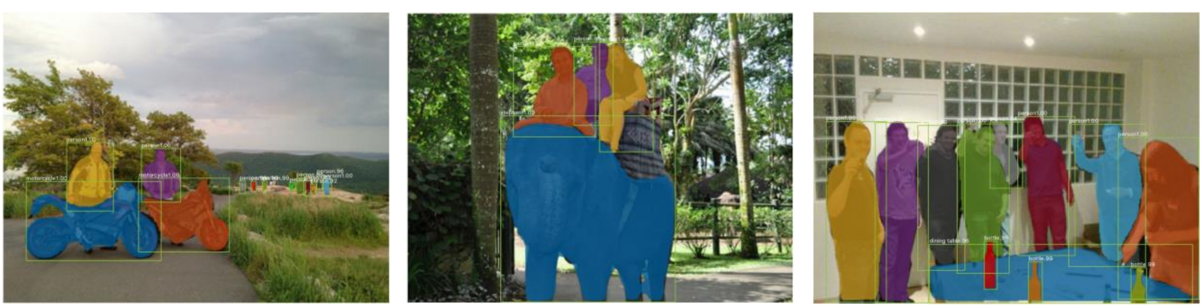
Mask R-CNN训练阶段使用的Mask样例



Mask R-CNN训练阶段使用的Mask样例



Mask R-CNN实例分割结果



He et al, “Mask R-CNN”, ICCV 2017

Mask R-CNN检测姿态

He et al, "Mask R-CNN", ICCV 2017

Open Source Frameworks

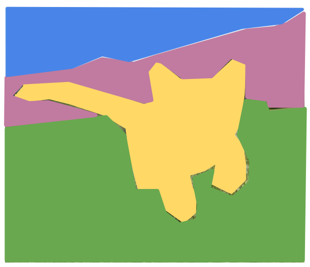
Lots of good implementations on GitHub!

TensorFlow Detection API:
https://github.com/tensorflow/models/tree/master/research/object_detection
Faster RCNN, SSD, RFCN, Mask R-CNN

Caffe2 Detectron:
https://github.com/facebookresearch/Detectron
Mask R-CNN, RetinaNet, Faster R-CNN, RPN, Fast R-CNN, R-FCN

Finetune on your own dataset with pre-trained models

视觉识别任务

分类	语义分割	目标检测	实例分割
			
猫	草，猫，树，天空	狗，狗，猫	狗，狗，猫

不考虑空间位置

像素的类别

多目标

This image is CC0 public domain