Artificial Neural Networks
人工神经网络

权小军教授
中山大学计算机学院

quanxj3@mail.sysu.edu.cn

2023 年 5 月 4 日

Lecture 8 - Semantic Segmentation and Object Detection 语义分割和目标检测

(*Part of the slides are adapted from Stanford CS231N*)

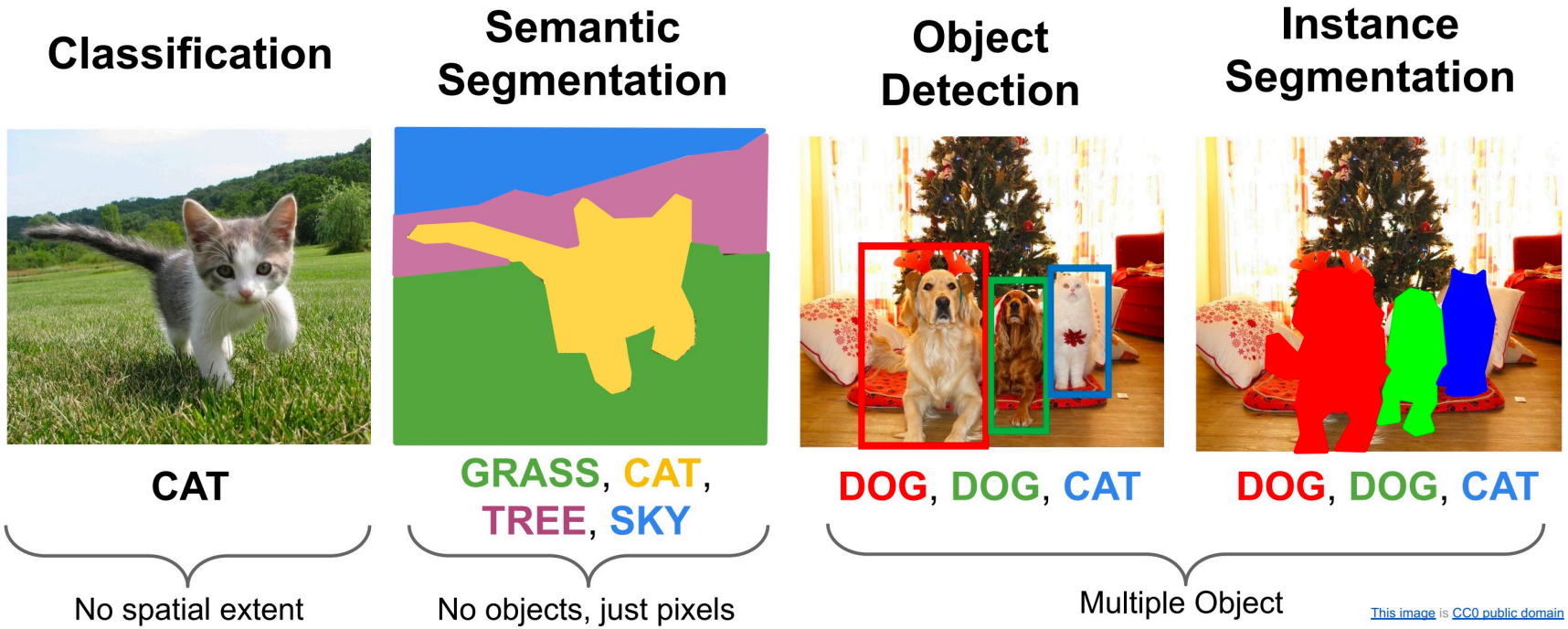
Image Classification

Classification: A core task in Computer Vision

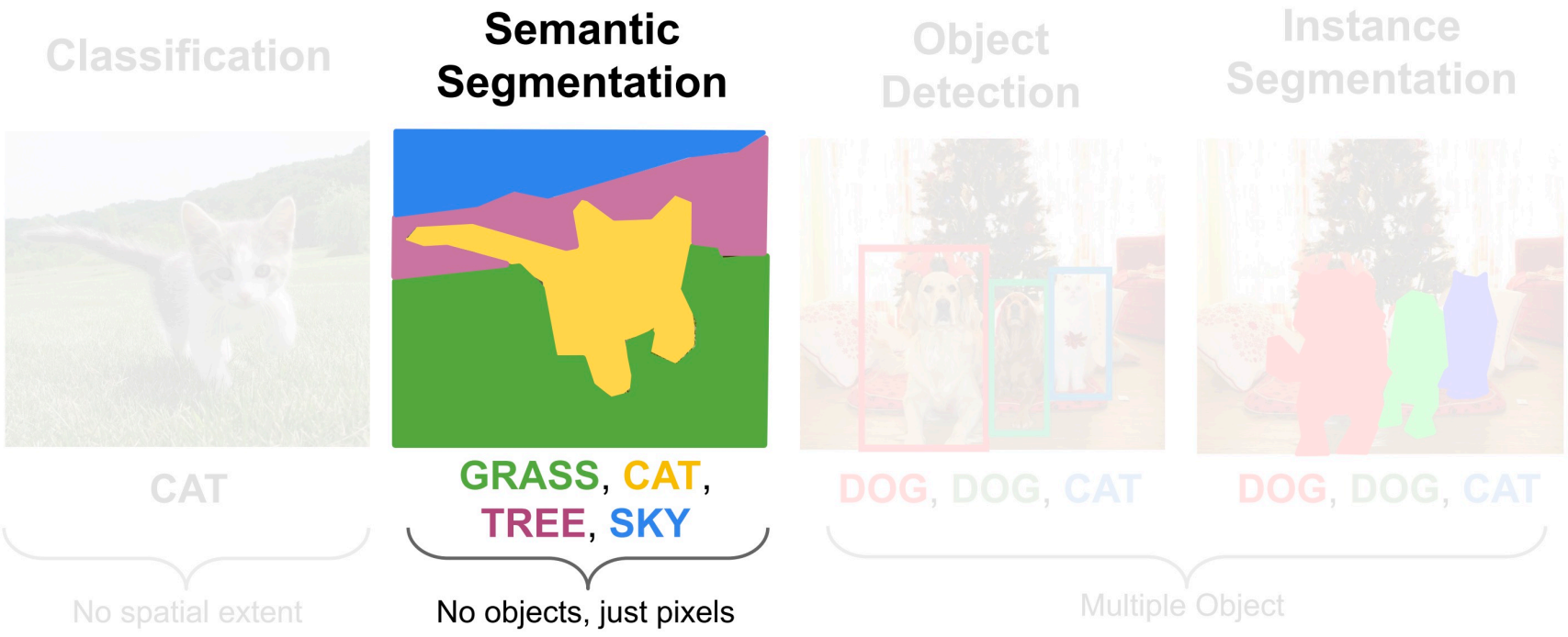
(assume given a set of possible labels)
{dog, cat, truck, plane, ...}

cat

Computer Vision Tasks



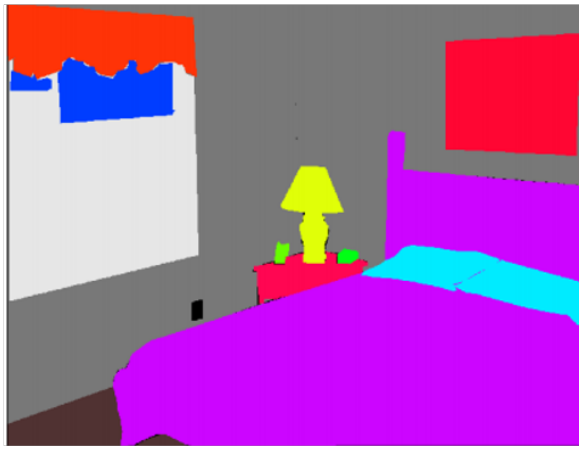
Semantic Segmentation



Lecture 8.1 Semantic Segmentation 语义分割

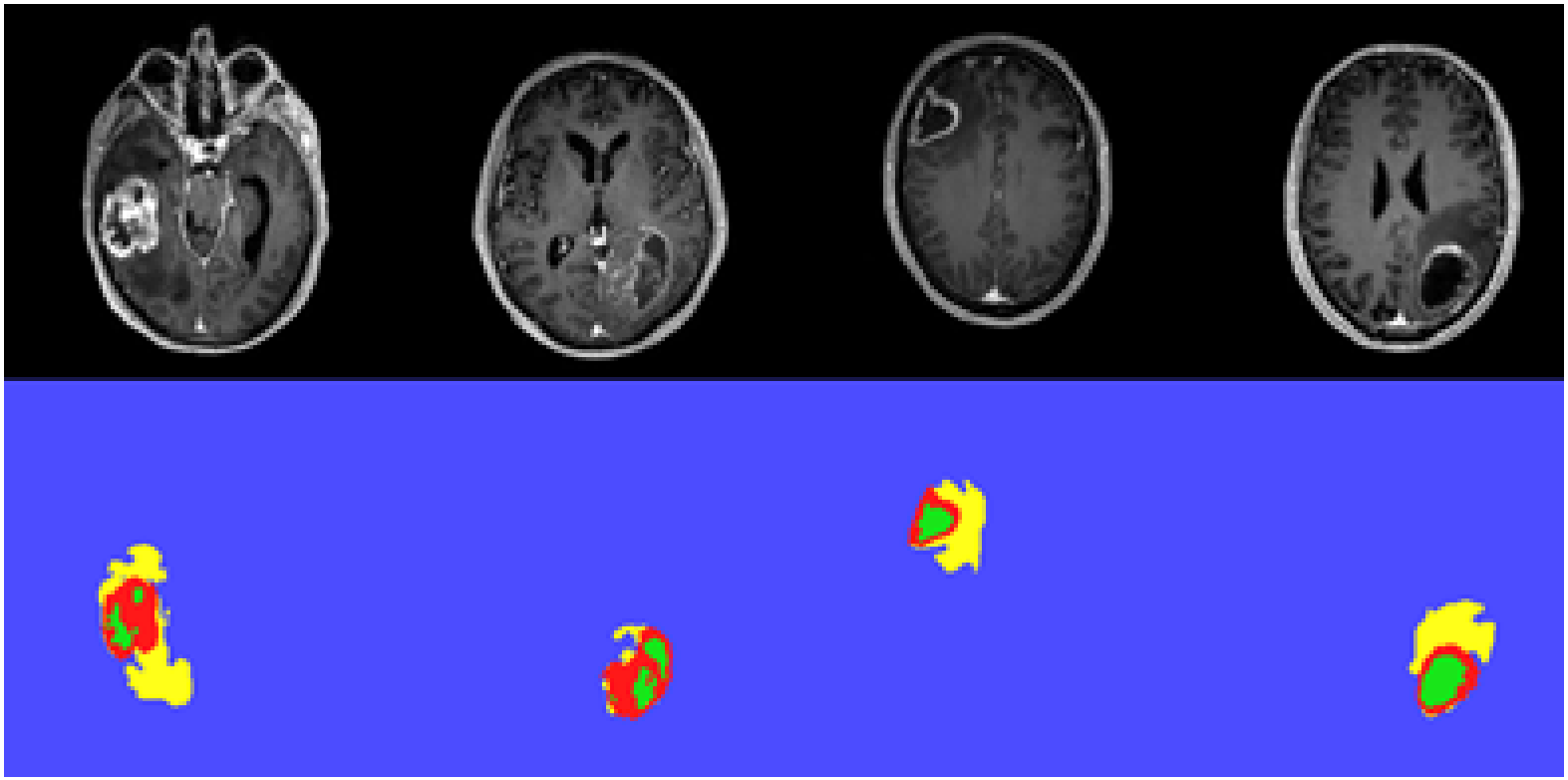
Semantic Segmentation

- Applications: self-driving, smart home



Semantic Segmentation

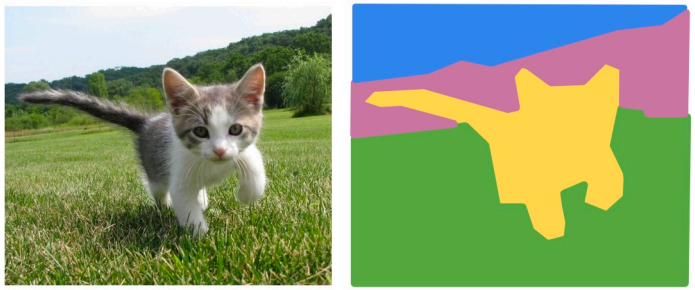
- More applications: intelligent medical analysis, etc.



Find any common characteristics from these segmentation tasks?

Semantic Segmentation: The Problem

- Semantic segmentation (语义分割): classifying each pixel



GRASS, CAT,
TREE, SKY, ...

Paired training data: for each training image, each pixel is labeled with a semantic category.

At test time, classify each pixel of a new image.

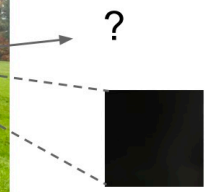
CNNs for Semantic Segmentation

How are CNNs applied to semantic segmentation?

Semantic Segmentation Idea: Sliding Window

Semantic Segmentation Idea: Sliding Window

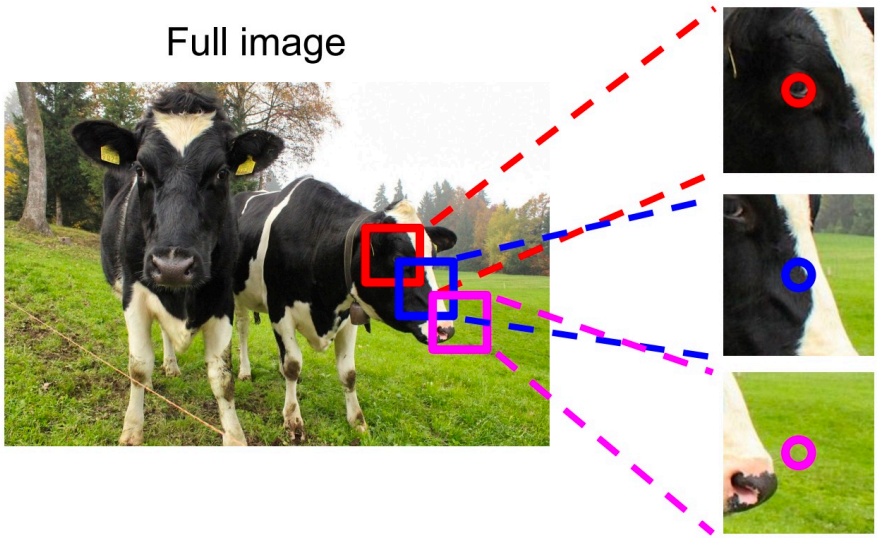
Full image



Impossible to classify without context

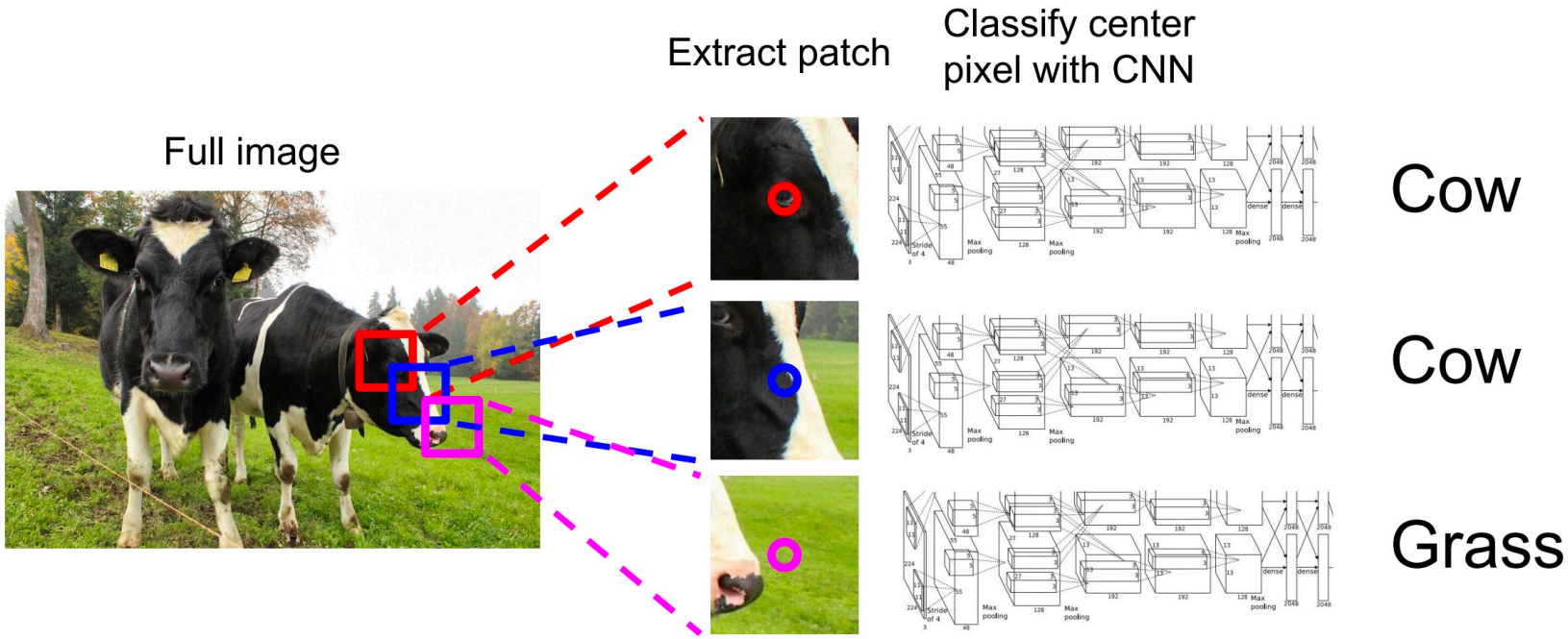
Q: how do we include context?

Semantic Segmentation Idea: Sliding Window

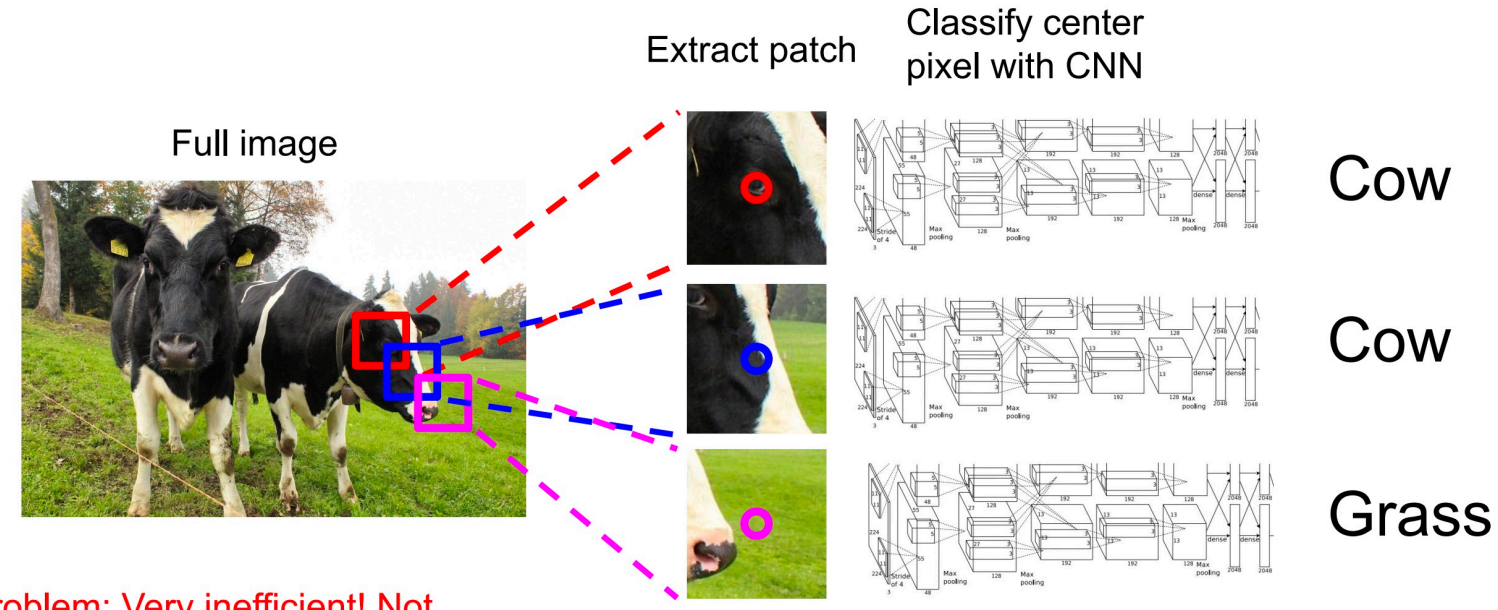


Q: how do we model this?

Semantic Segmentation Idea: Sliding Window



Semantic Segmentation Idea: Sliding Window

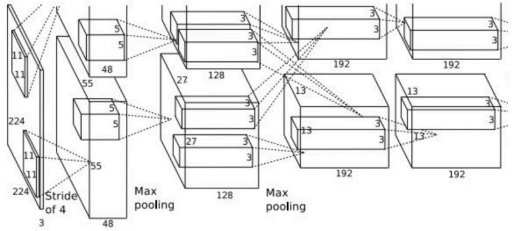


Problem: Very inefficient! Not reusing shared features between overlapping patches

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

Semantic Segmentation Idea: Convolution

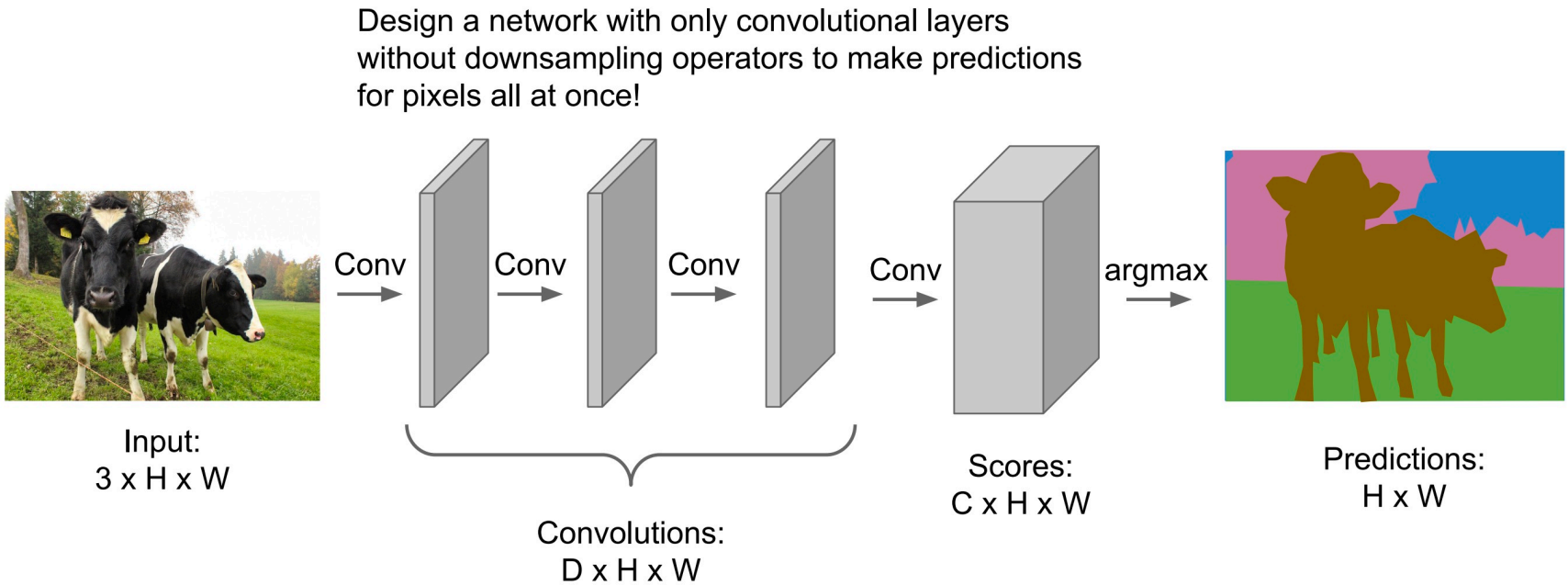
Full image



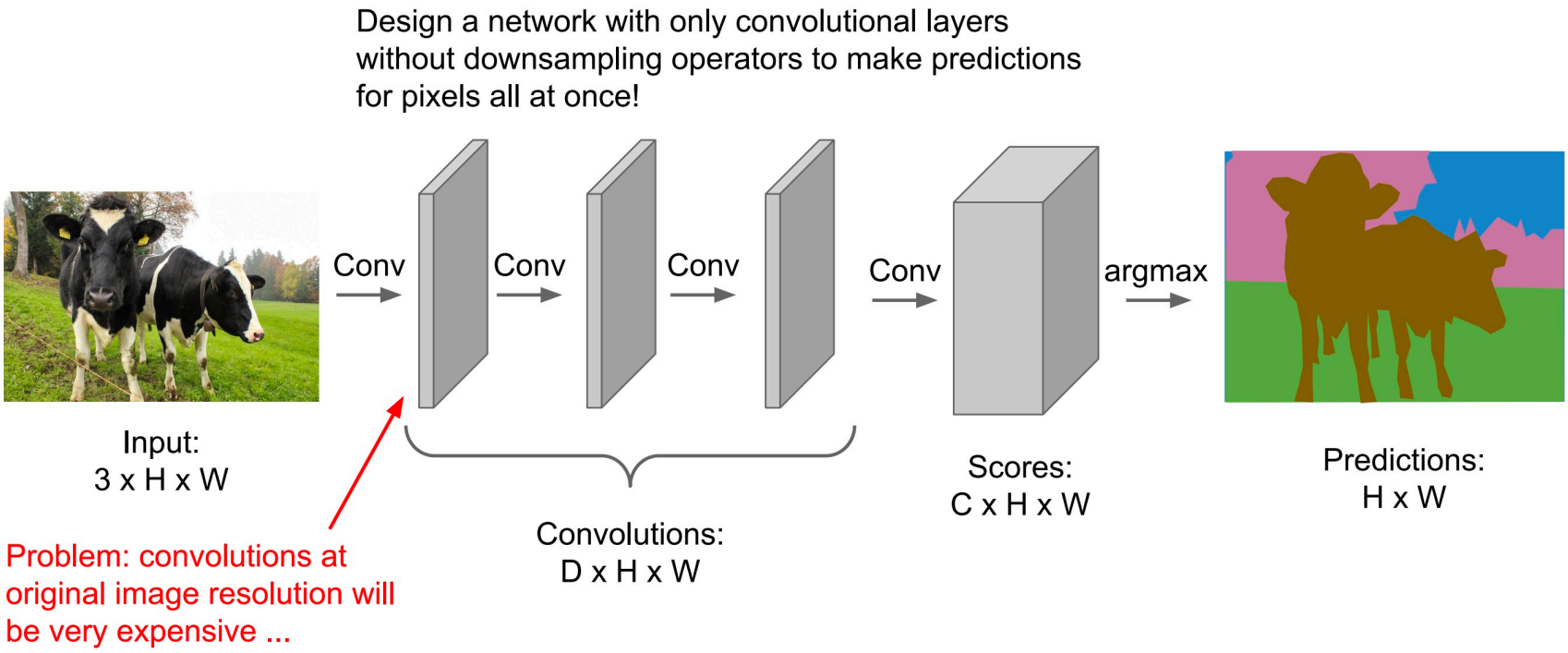
An intuitive idea: encode the entire image with conv net, and do semantic segmentation on top.

Problem: classification architectures often reduce feature spatial sizes to go deeper, but semantic segmentation requires the output size to be the same as input size.

Semantic Segmentation Idea: Fully Convolutional

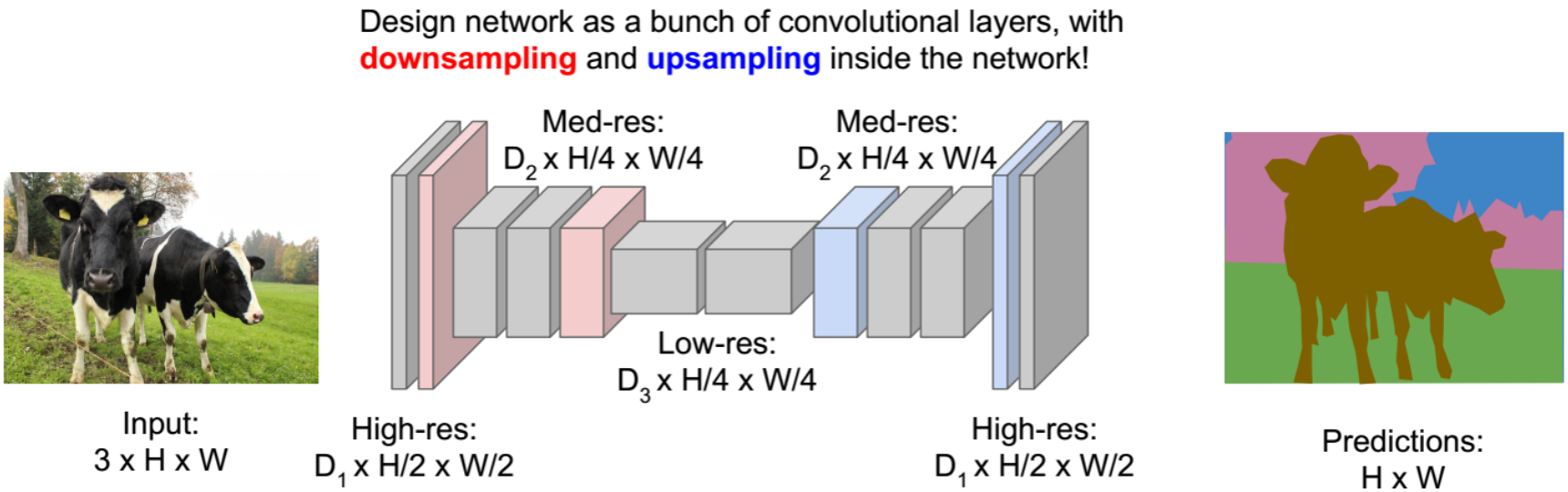


Semantic Segmentation Idea: Fully Convolutional



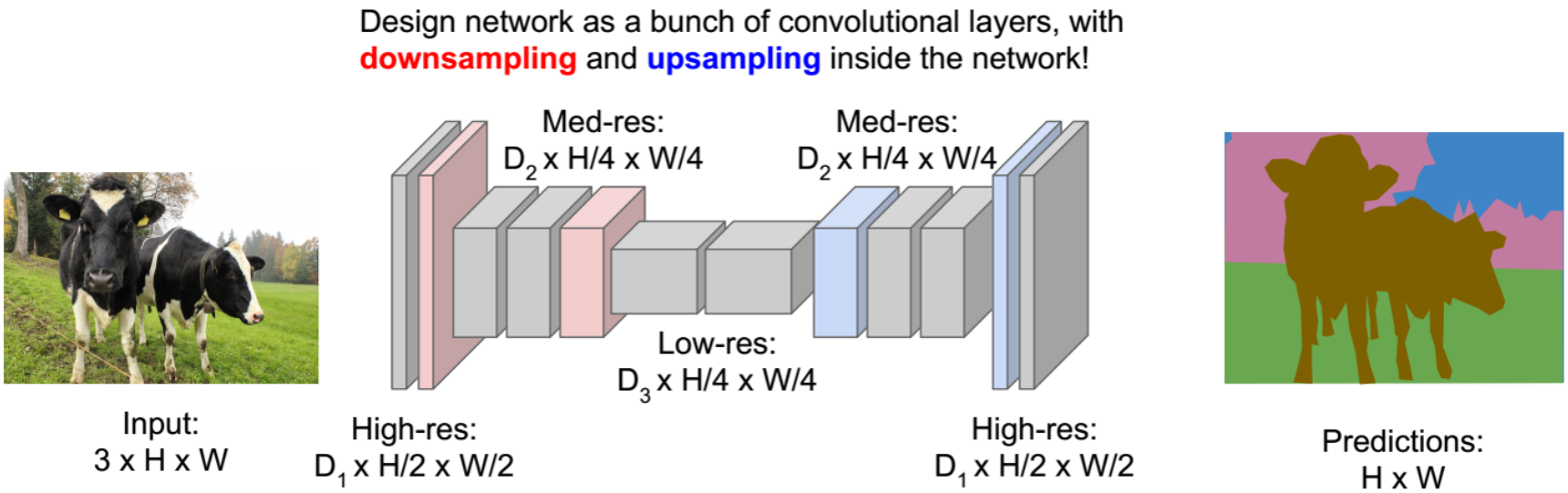
Fully Convolutional Network (FCN，全卷积网络)

- Encoder-decoder framework: reduce size of feature maps, then increase to original size



Fully Convolutional Network (FCN，全卷积网络)

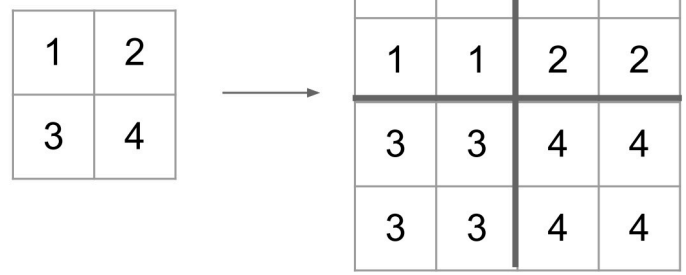
- Encoder-decoder framework: reduce size of feature maps, then increase to original size



- But how to realize "upsampling" (上采样)?

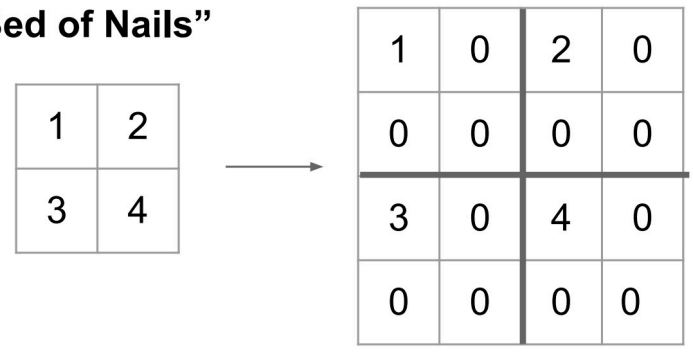
In-Network Upsampling: "Unpooling" (上池化)

Nearest Neighbor



Input: 2 x 2 Output: 4 x 4

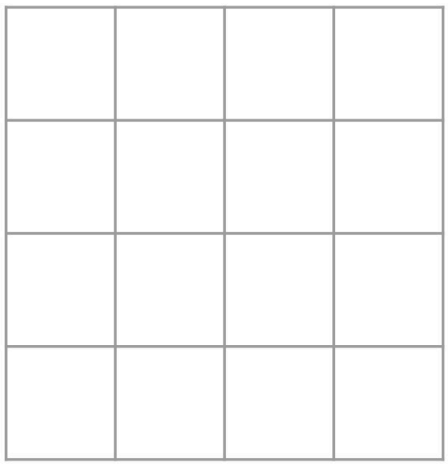
"Bed of Nails"



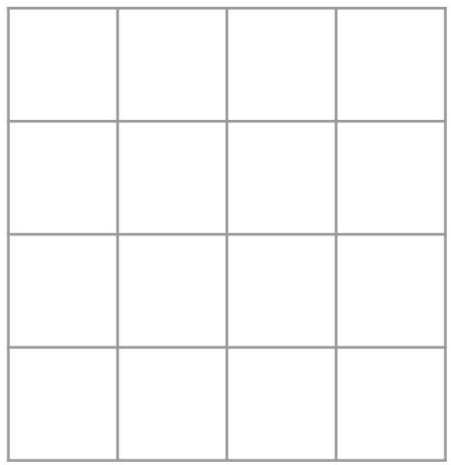
Input: 2 x 2 Output: 4 x 4

Transposed convolution (反卷积): learnable upsampling

Recall: Normal 3 x 3 convolution, stride 1 pad 1



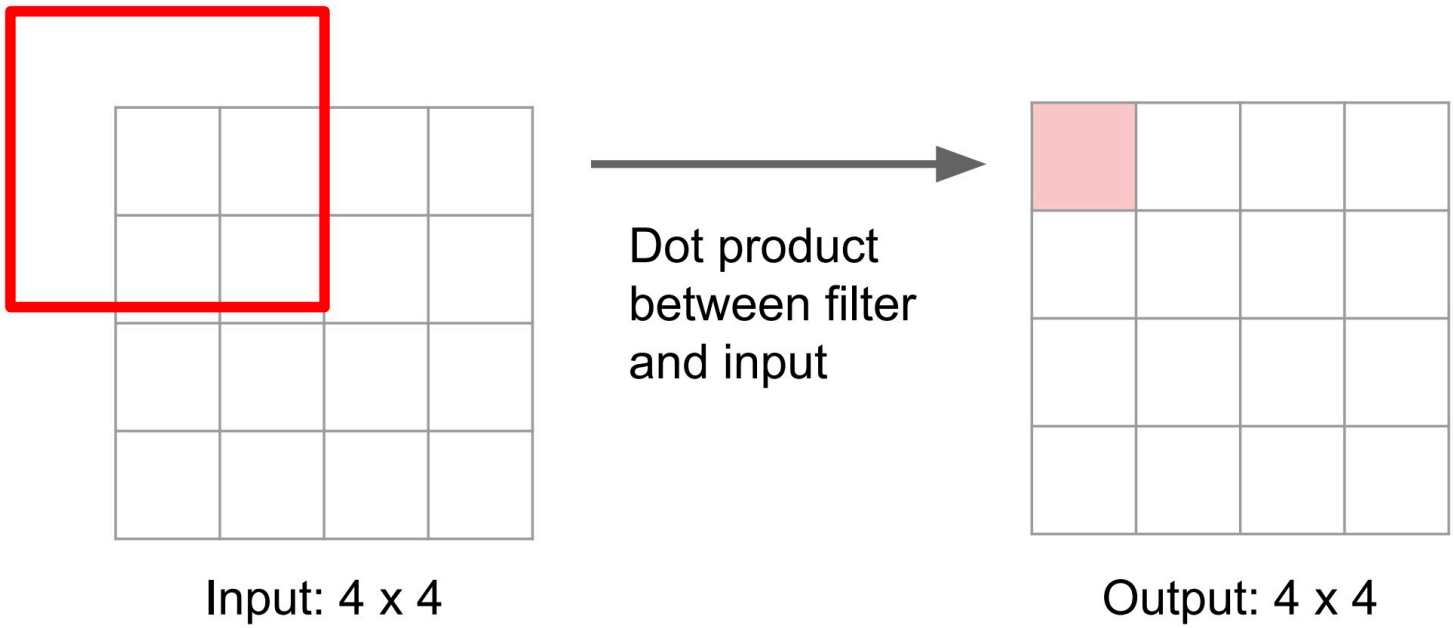
Input: 4 x 4



Output: 4 x 4

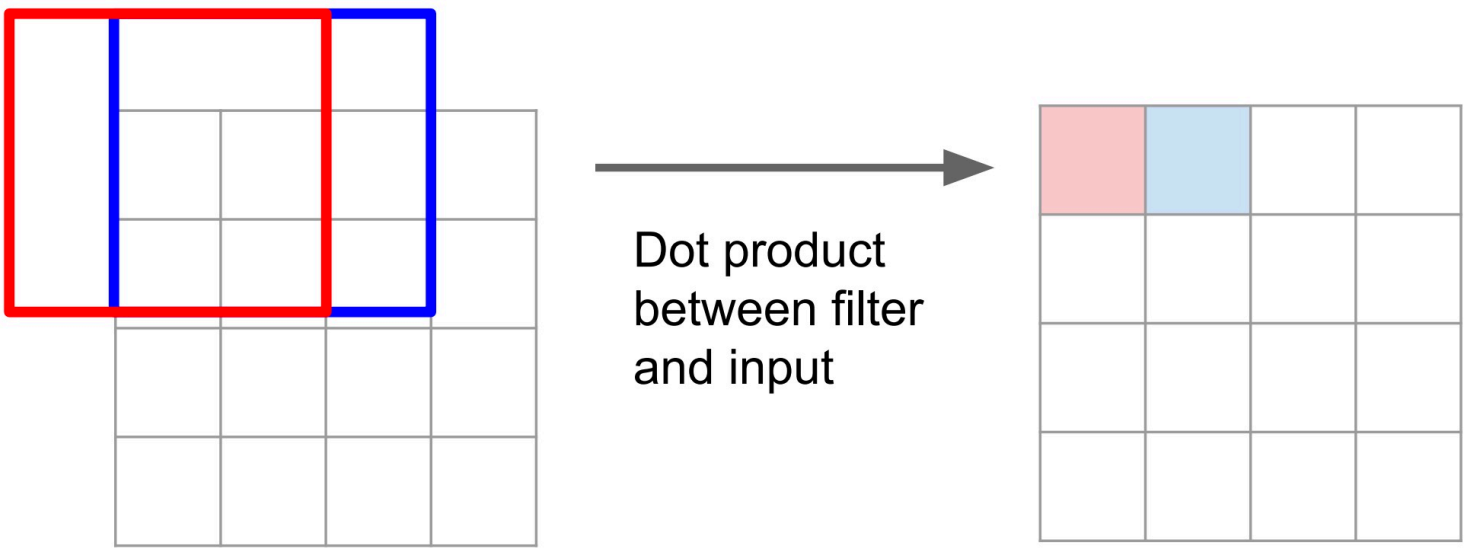
Transposed convolution (反卷积): learnable upsampling

Recall: Normal 3 x 3 convolution, stride 1 pad 1



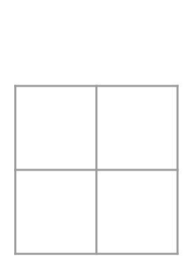
Transposed convolution (反卷积): learnable upsampling

Recall: Normal 3 x 3 convolution, stride 1 pad 1

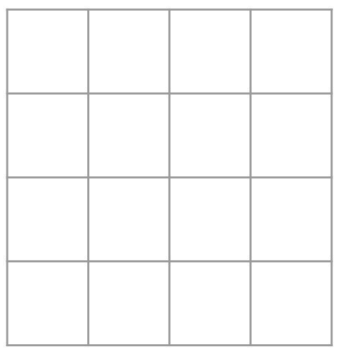


Learnable Upsampling: Transposed Convolution

3 x 3 **transposed** convolution, stride 2 pad 1

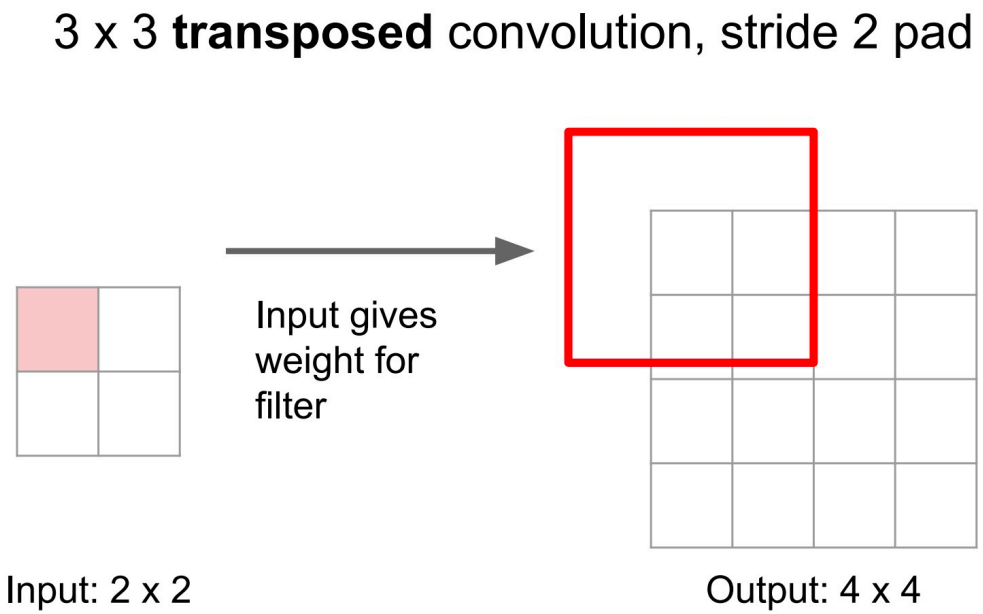


Input: 2 x 2



Output: 4 x 4

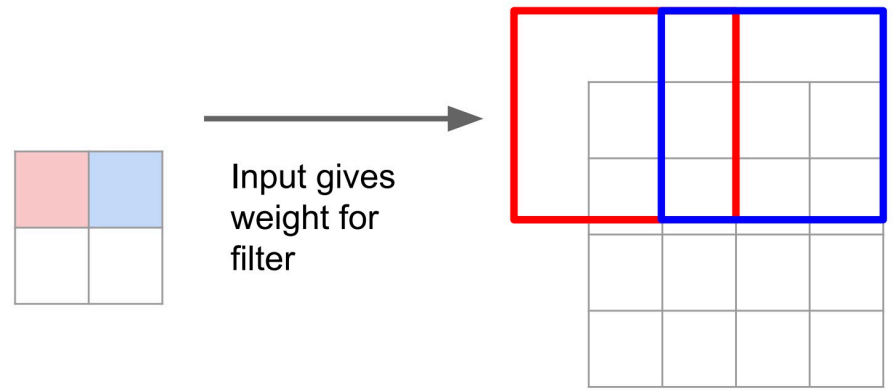
Learnable Upsampling: Transposed Convolution



Transposed convolution (反卷积): learnable upsampling

Learnable Upsampling: Transposed Convolution

3 x 3 **transposed** convolution, stride 2 pad 1

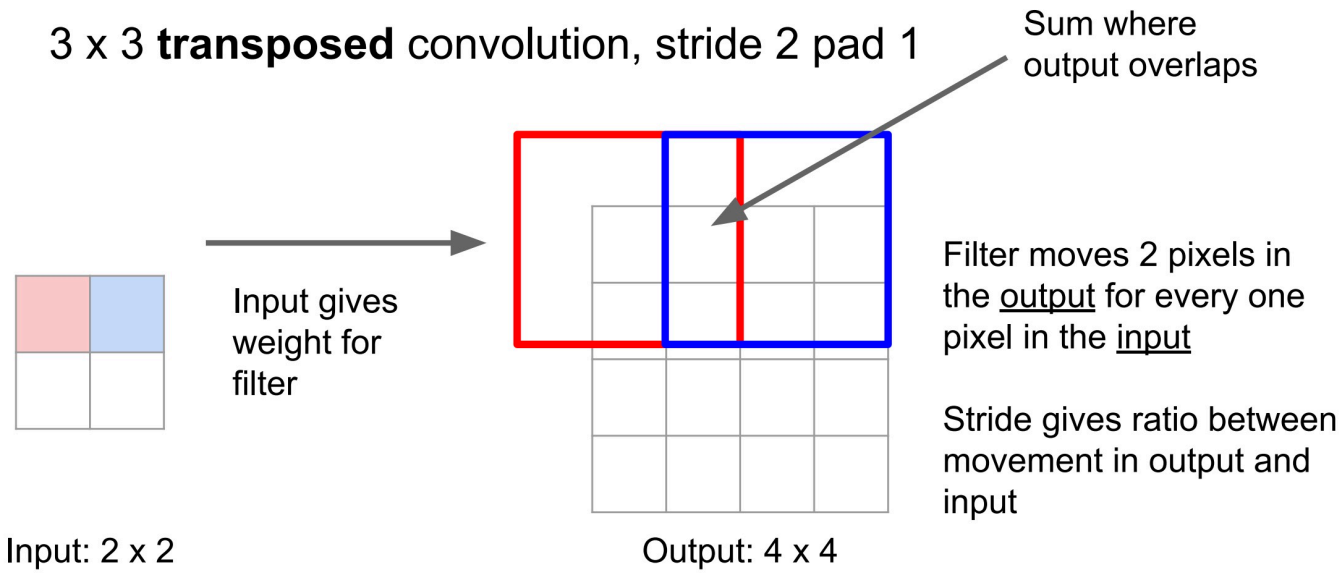


Filter moves 2 pixels in the output for every one pixel in the input

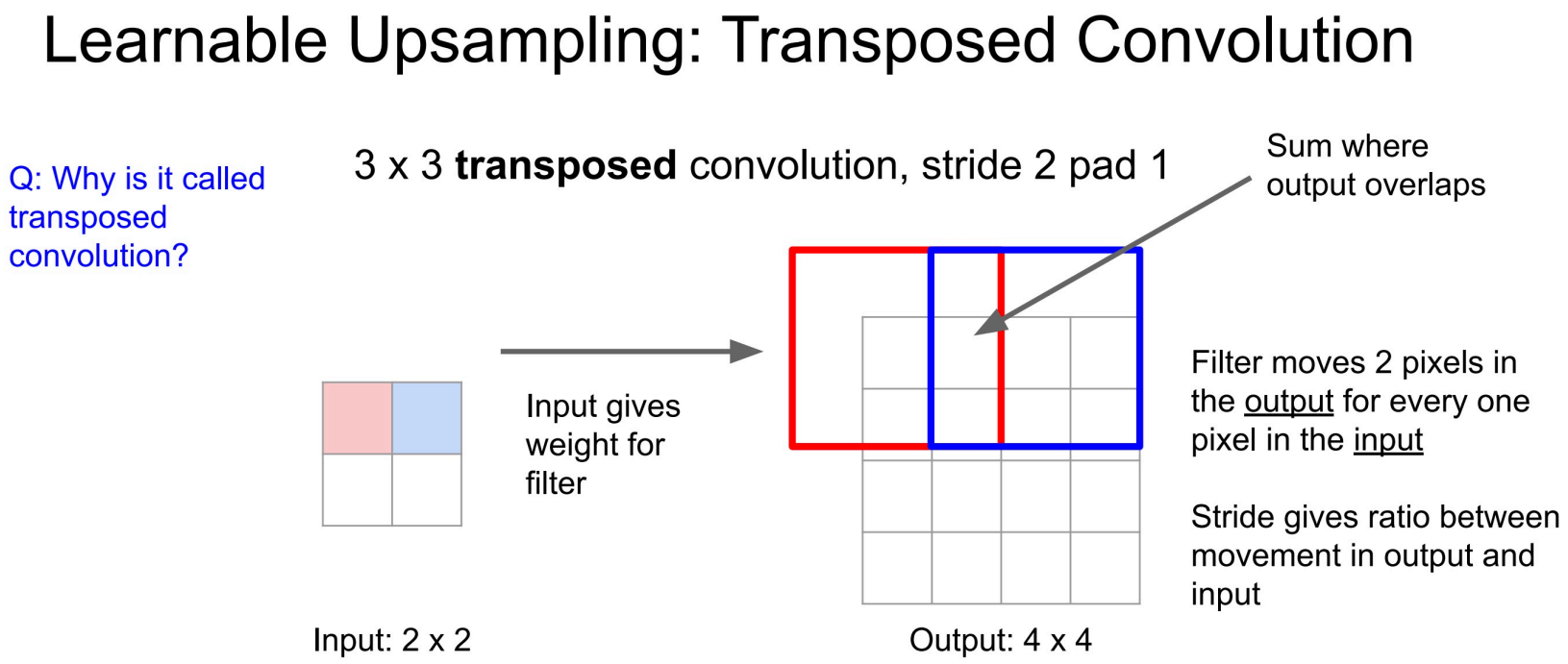
Stride gives ratio between movement in output and input

Transposed convolution (反卷积): learnable upsampling

Learnable Upsampling: Transposed Convolution

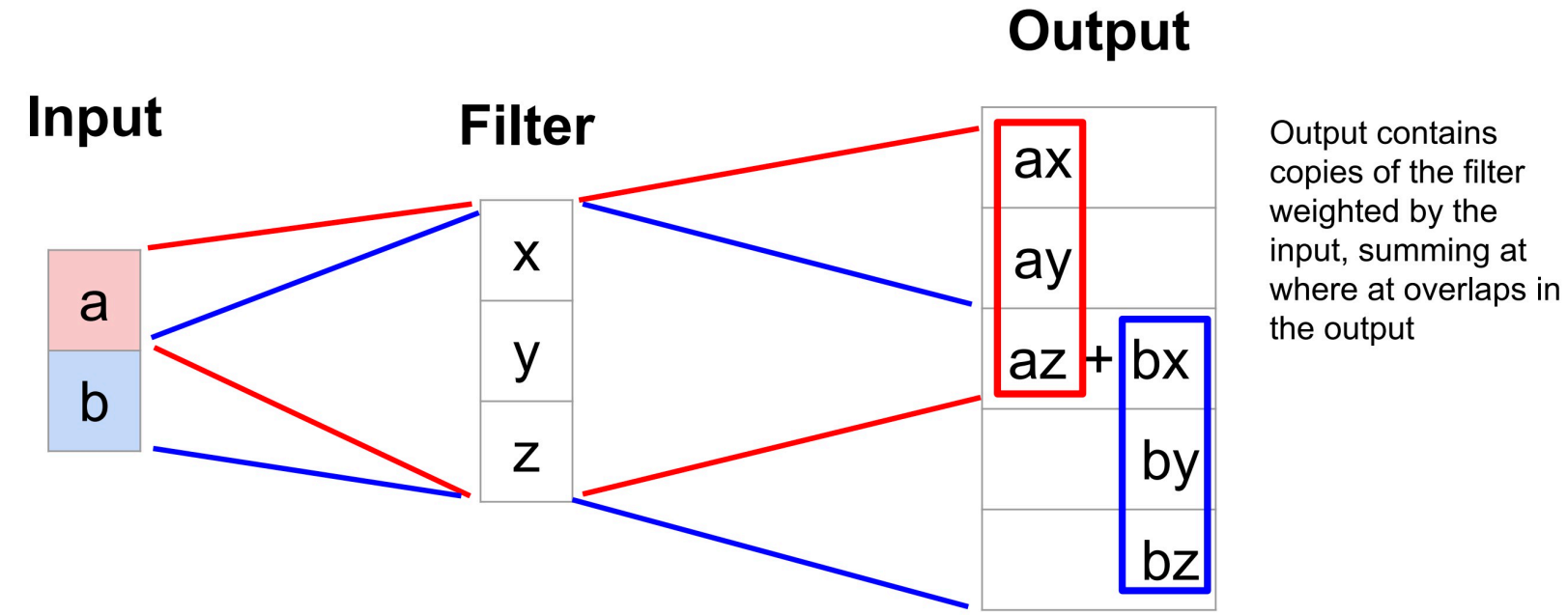


Transposed convolution (反卷积): learnable upsampling



Padding: unlike in the regular convolution where padding is applied to input, it is applied to output in the transposed convolution.

Transposed convolution (反卷积): learnable upsampling



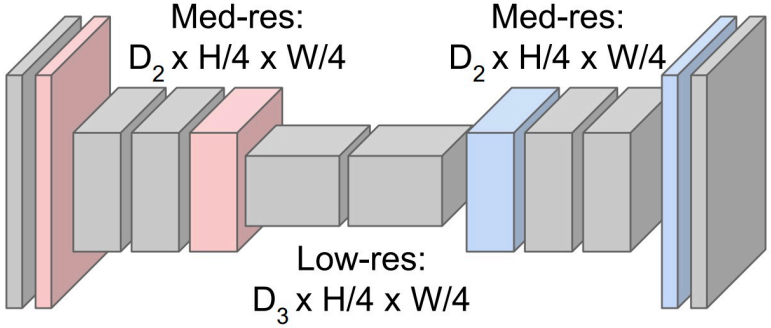
Fully Convolutional Network (FCN，全卷积网络)

Downsampling:
Pooling, strided convolution

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Upsampling:
Unpooling or strided transposed convolution

Input:
3 x H x W



High-res:
D_1 x H/2 x W/2

High-res:
D_1 x H/2 x W/2

Predictions:
H x W

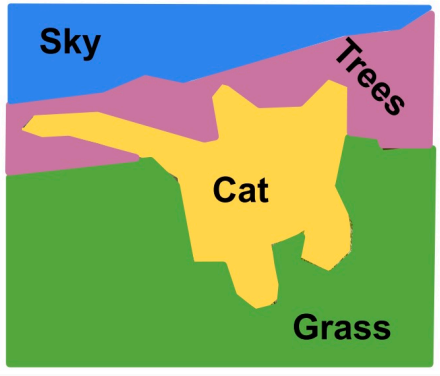
Lecture 8.2 Object Detection 目标检测

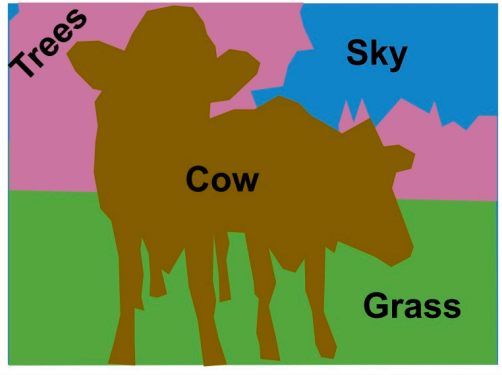
Semantic Segmentation (语义分割)

Semantic Segmentation

Label each pixel in the image with a category label

Don't differentiate instances, only care about pixels





Object Detection

Classification

CAT

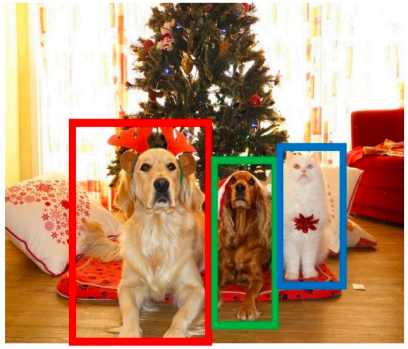
No spatial extent

Semantic Segmentation

GRASS, CAT, TREE, SKY

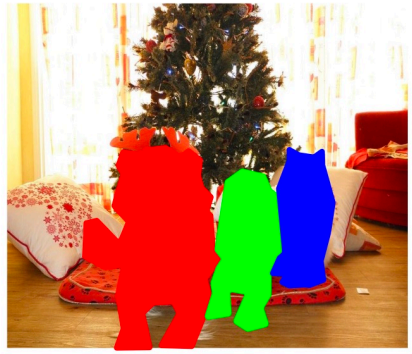
No objects, just pixels

Object Detection



DOG, DOG, CAT

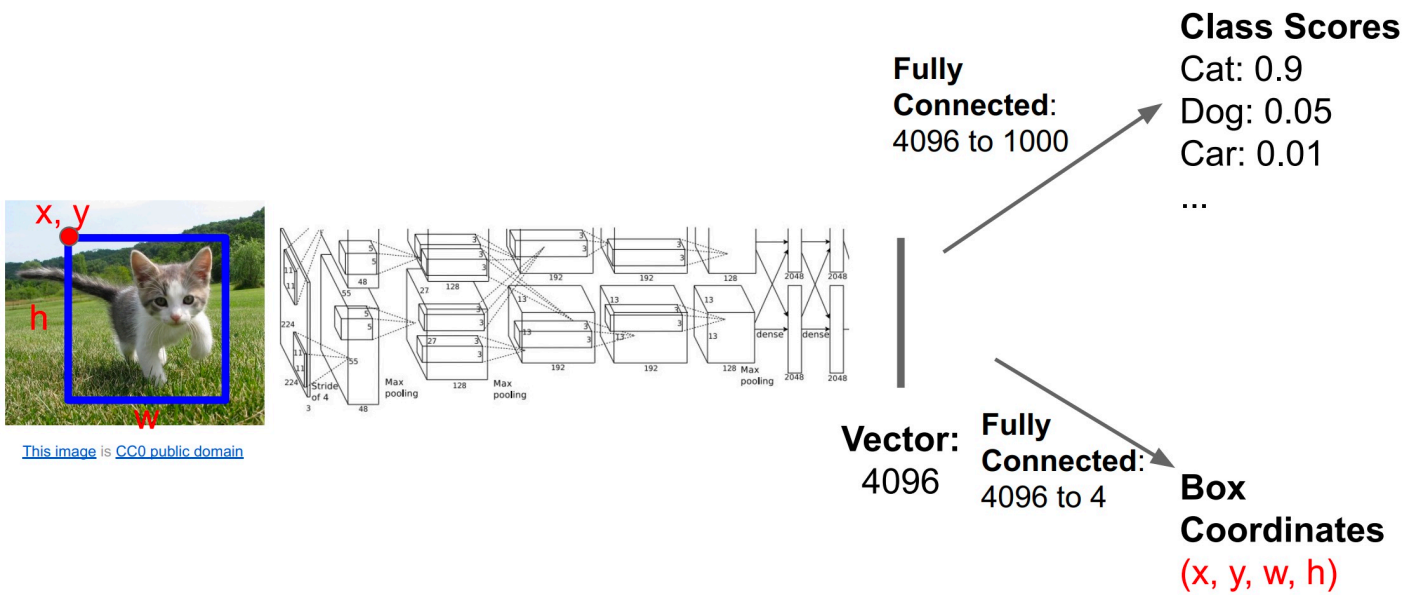
Instance Segmentation



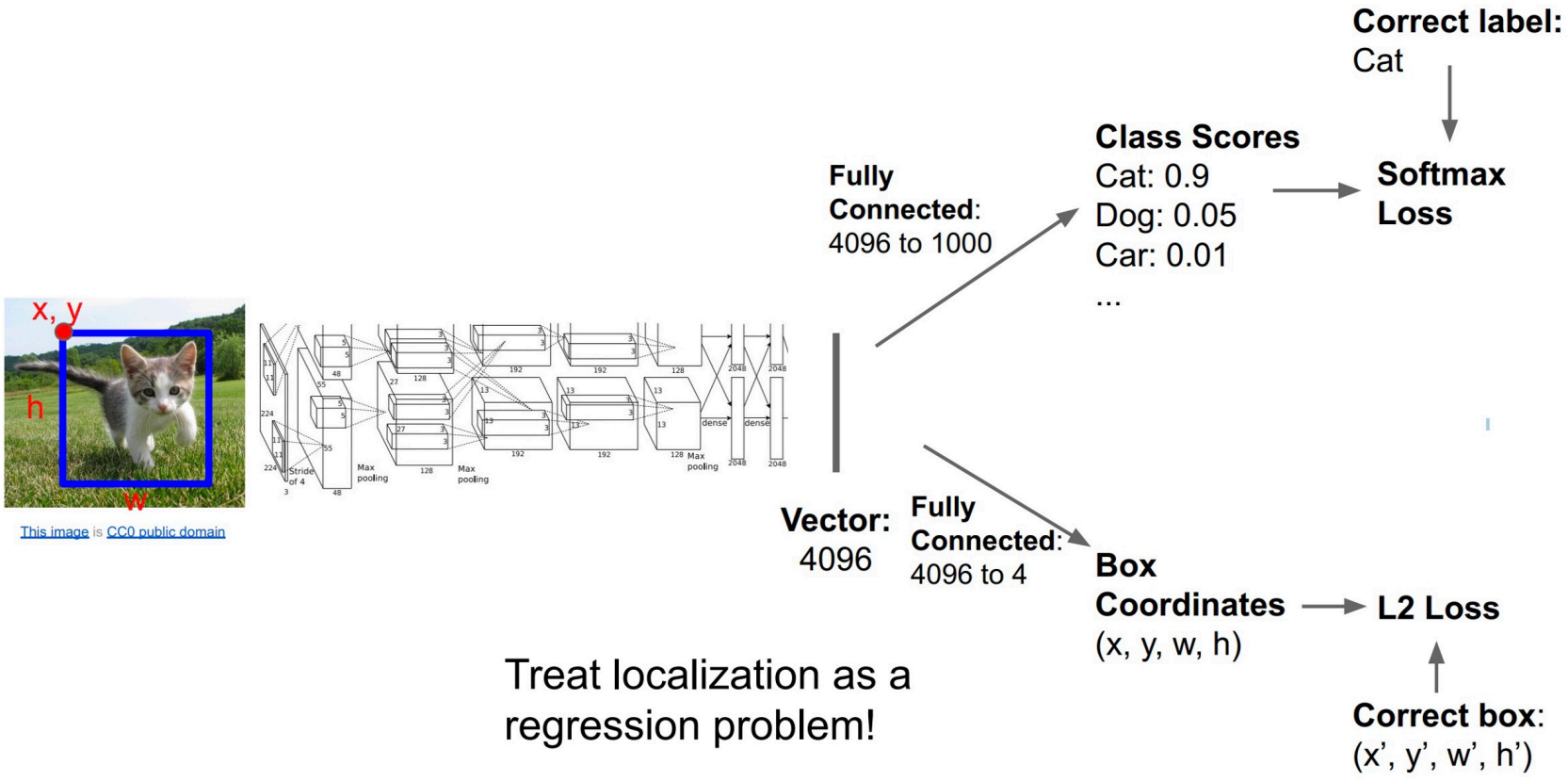
DOG, DOG, CAT

Multiple Object

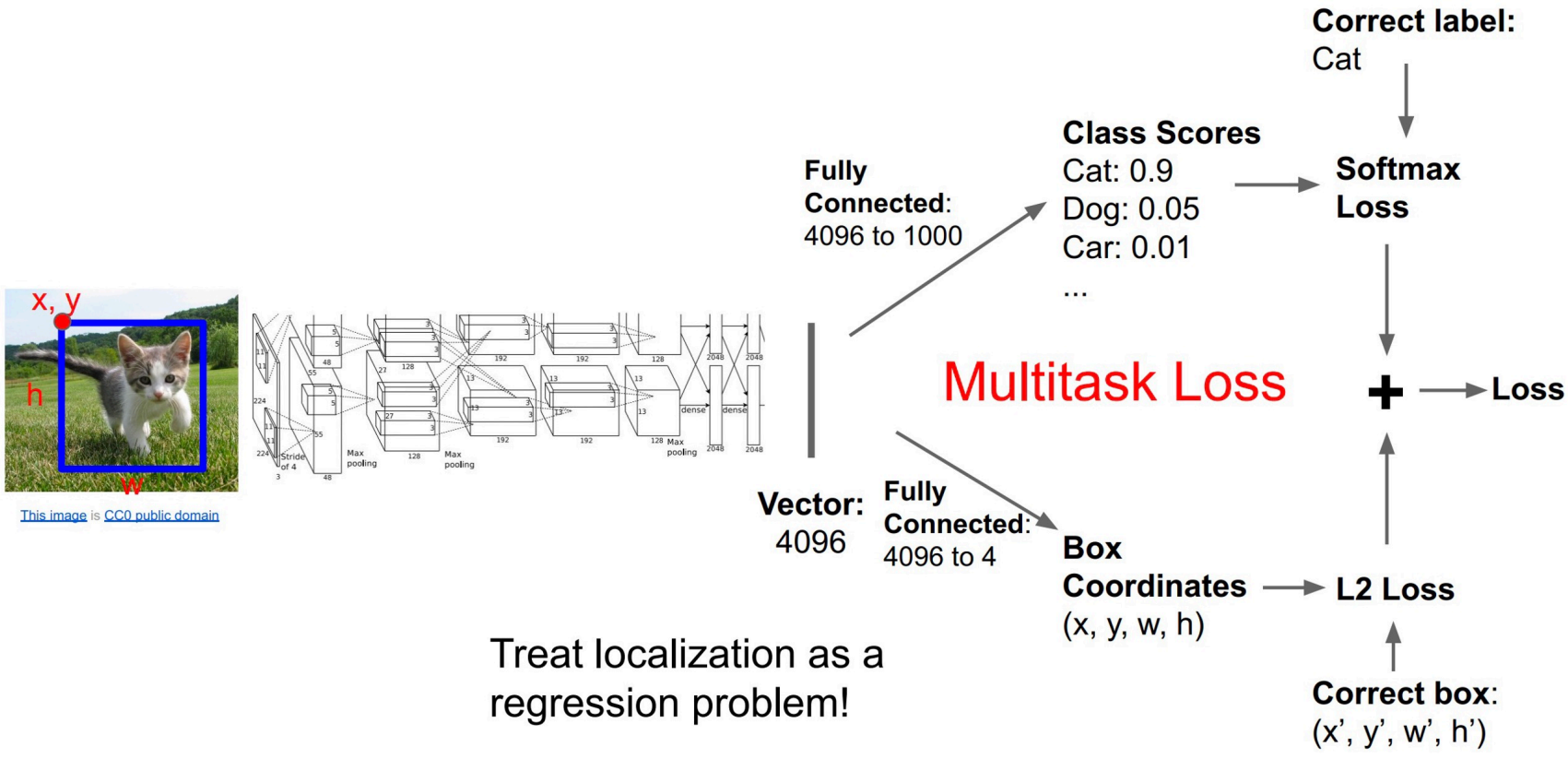
Object Detection: Single Object (Classification + Localization)



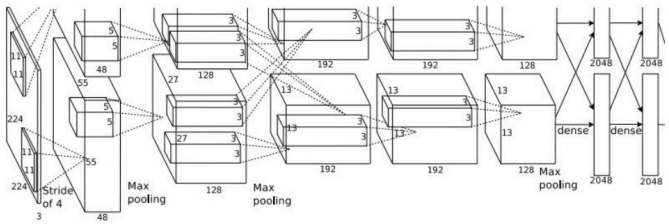
Object Detection: Single Object (Classification + Localization)



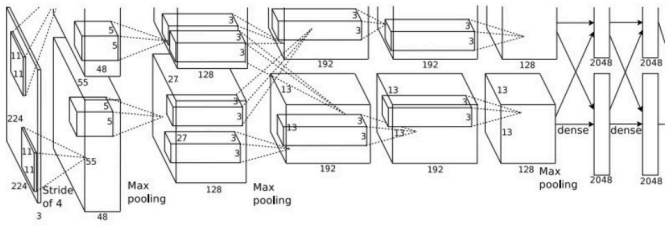
Object Detection: Single Object (Classification + Localization)



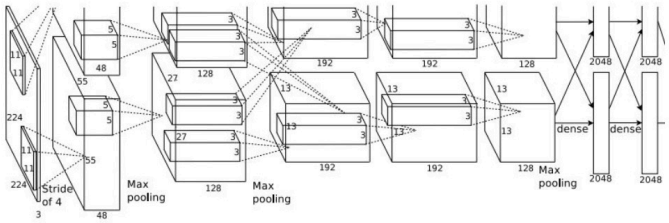
Object Detection: Multiple Objects



CAT: (x, y, w, h)



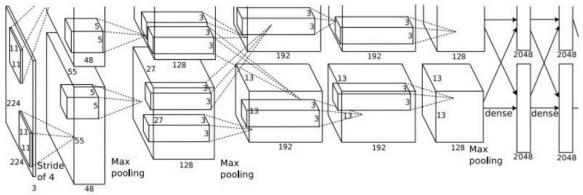
DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)



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

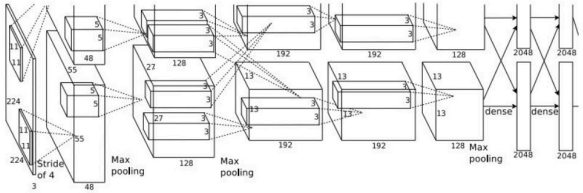
Object Detection: Multiple Objects

Each image needs a different number of outputs!



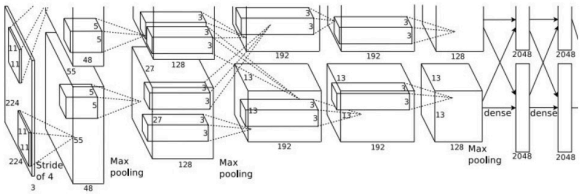
CAT: (x, y, w, h)

4 numbers



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

12 numbers

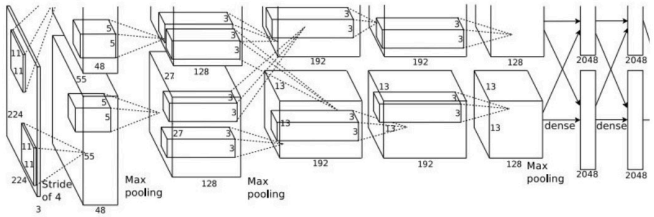


DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
....

Many numbers!

Object Detection: Multiple Objects

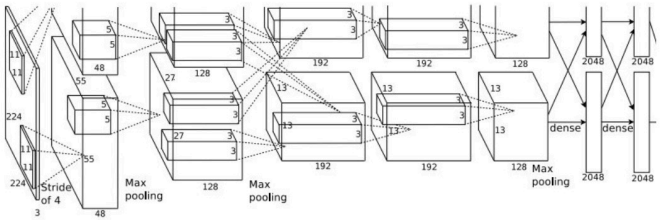
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? NO
Background? YES

Object Detection: Multiple Objects

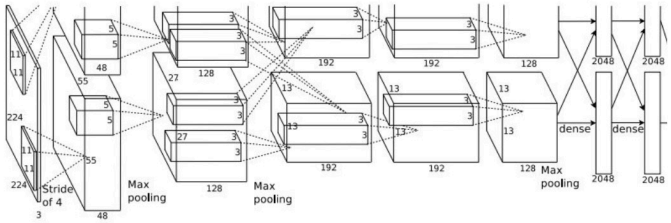
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES
Cat? NO
Background? NO

Object Detection: Multiple Objects

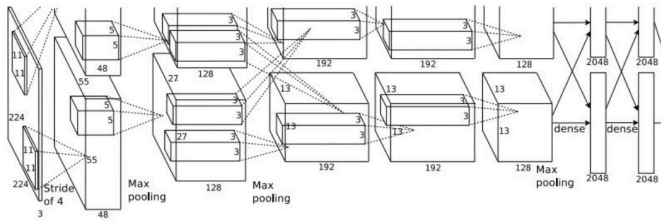
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES
Cat? NO
Background? NO

Object Detection: Multiple Objects

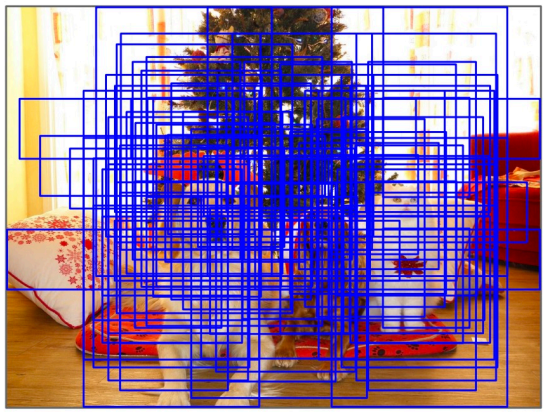
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



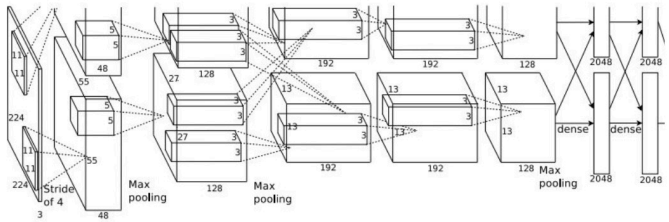
Dog? NO
Cat? YES
Background? NO

Q: What's the problem with this approach?

Object Detection: Multiple Objects



Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

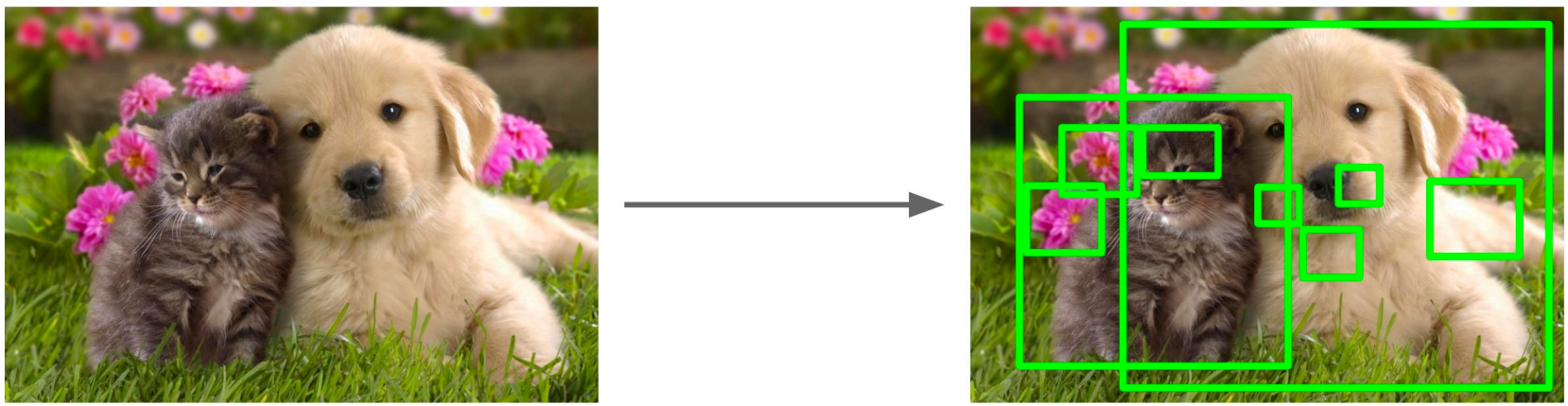


Dog? NO
Cat? YES
Background? NO

Problem: Need to apply CNN to huge number of locations, scales, and aspect ratios, very computationally expensive!

Region Proposals: Selective Search 候选区域: 选择搜索

- Find “blobby” image regions that are likely to contain objects
- Relatively fast to run; e.g. Selective Search gives 2000 region proposals in a few seconds on CPU

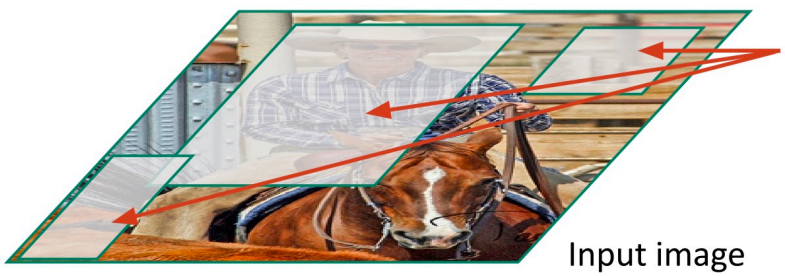


R-CNN

Input image

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

R-CNN

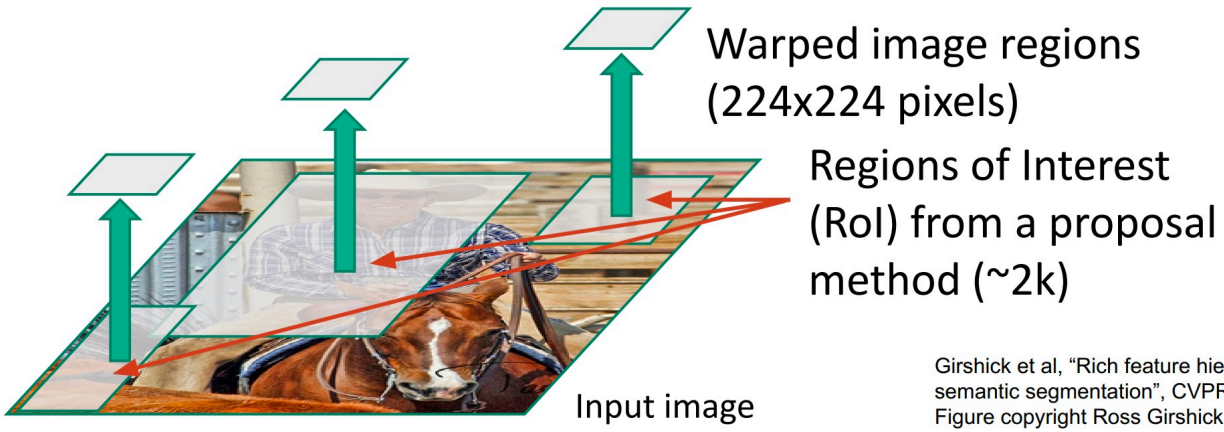


Input image

Regions of Interest (RoI) from a proposal method (~2k)

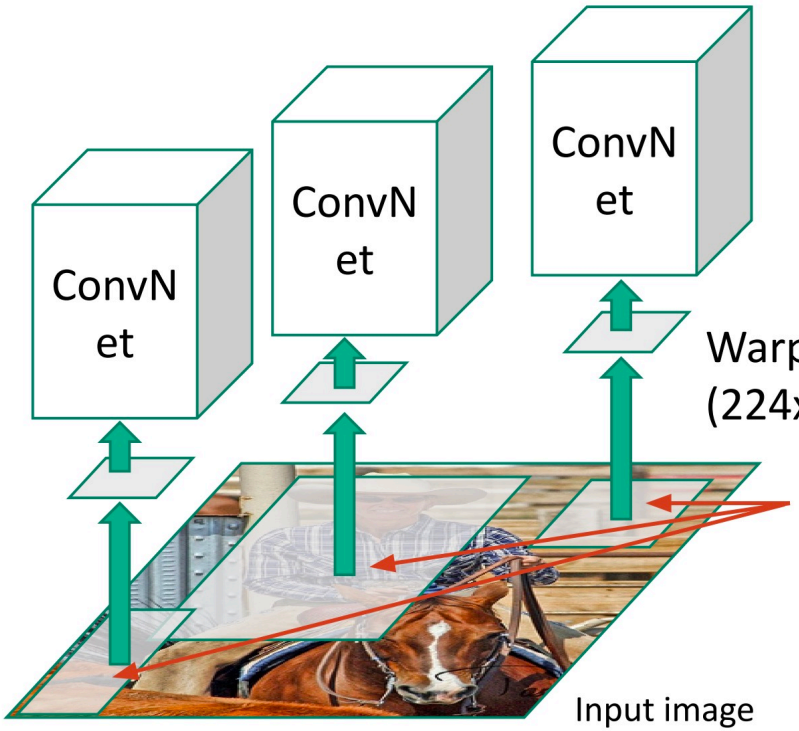
Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

R-CNN



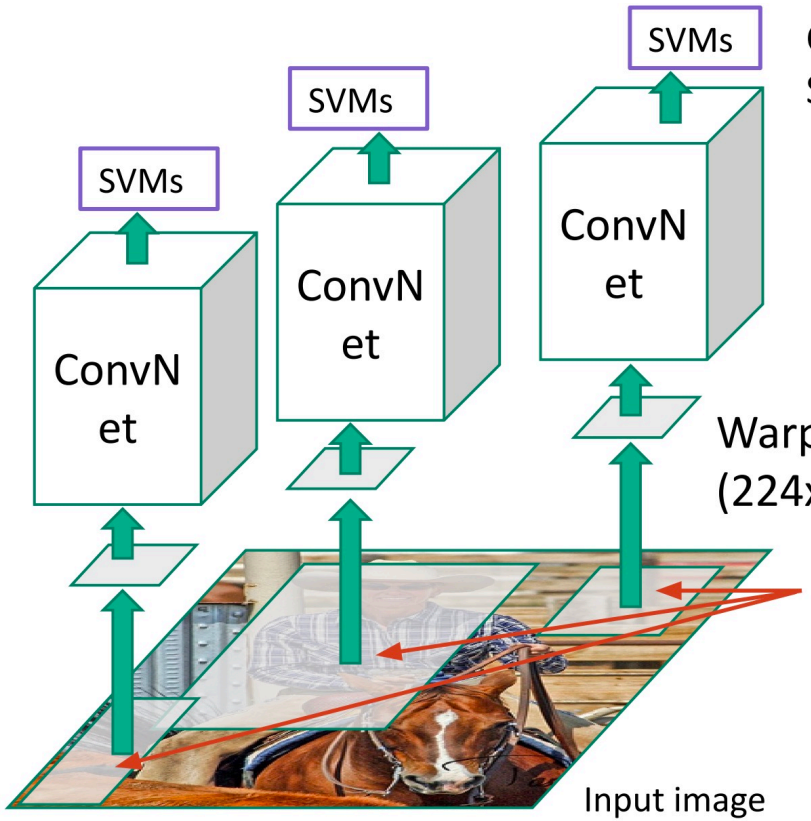
Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

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.

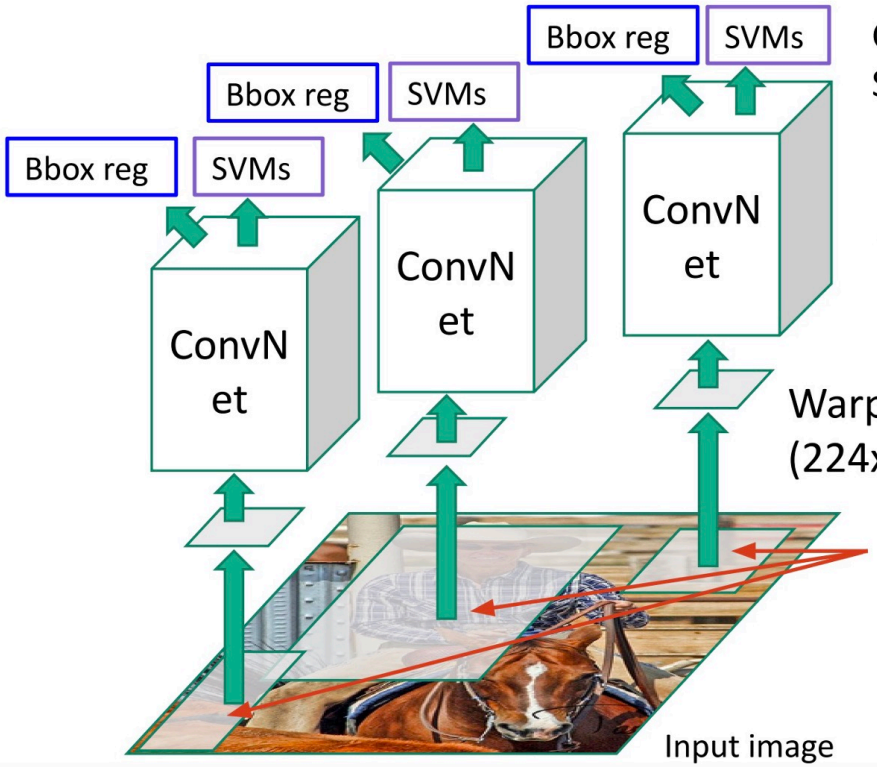
R-CNN



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)



Classify regions with SVMs

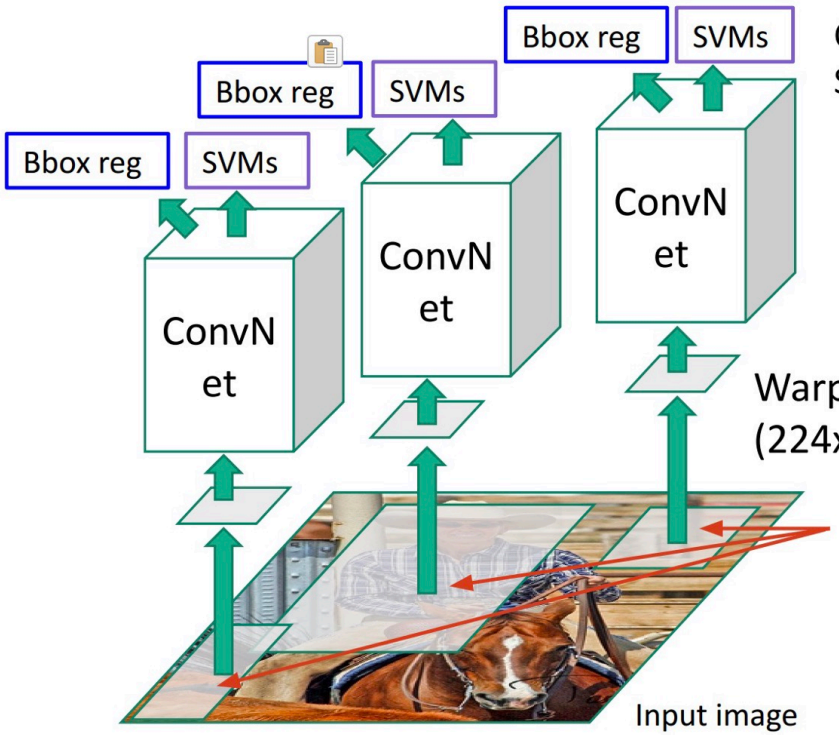
Forward each region through ConvNet (ImageNet-pretranied)

Warped image regions (224x224 pixels)

Regions of Interest (RoI) from a proposal method (~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

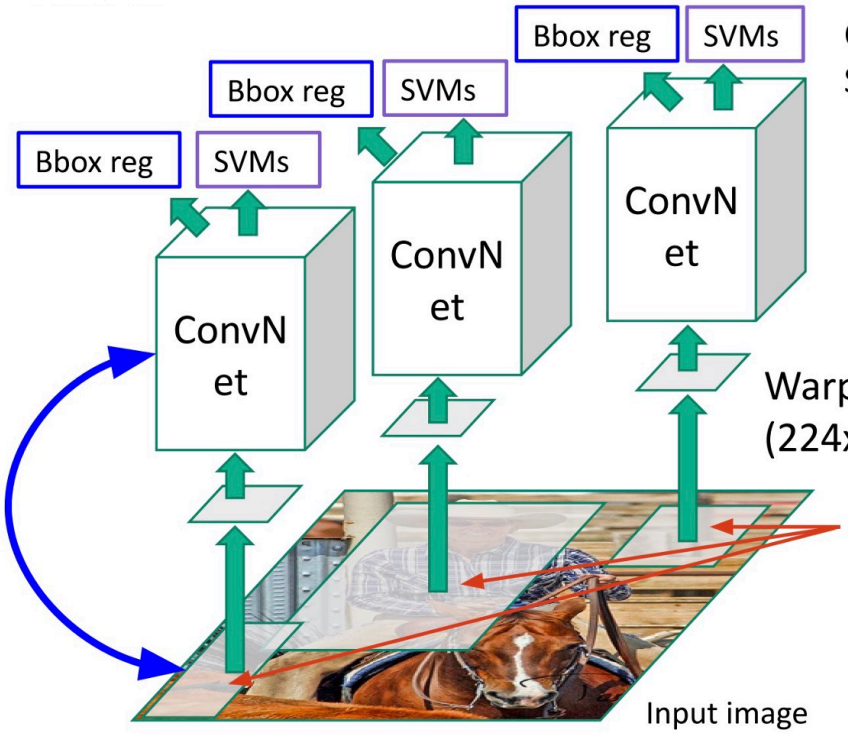


Problem: Very slow! Need to do ~2k independent forward passes for each image!

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

"Slow" R-CNN

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



Classify regions with SVMs

Forward each region through ConvNet

Warped image regions (224x224 pixels)

Regions of Interest (RoI) from a proposal method (~2k)

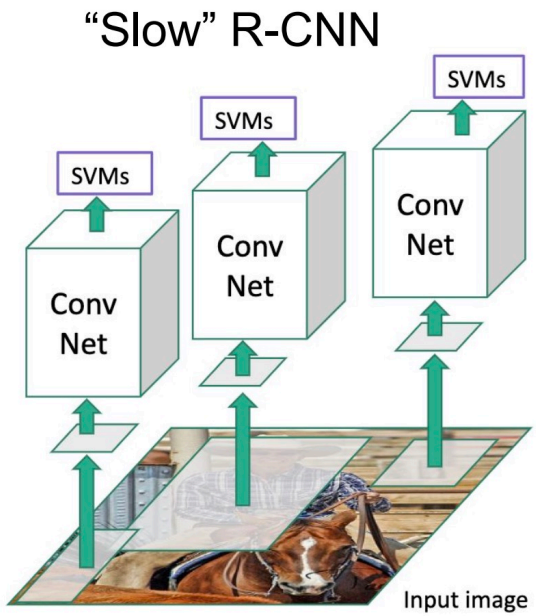
Problem: Very slow! Need to do ~2k independent forward passes for each image!

Idea: Pass the image through convnet before cropping! Crop the conv feature instead!

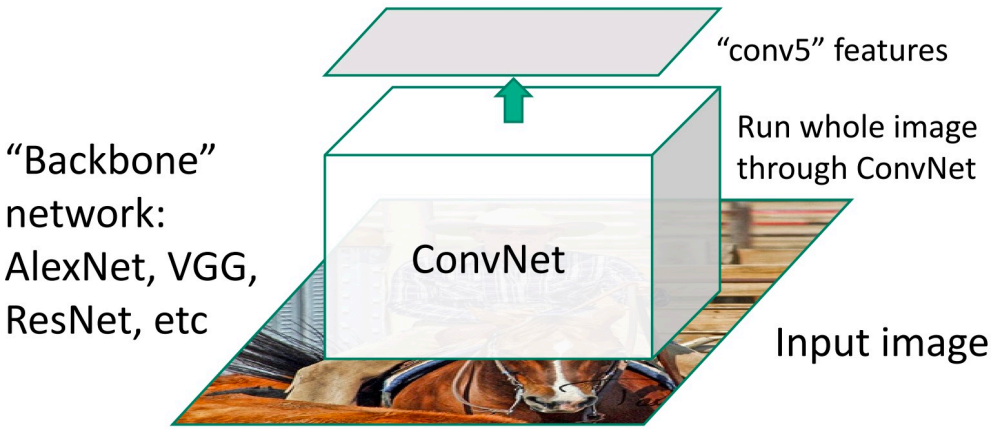
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

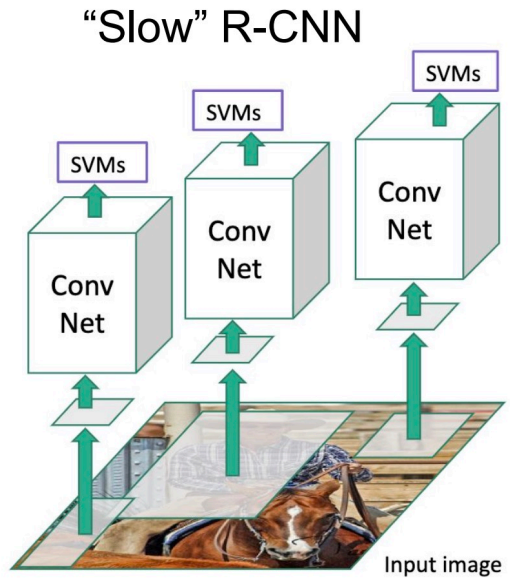
Fast R-CNN

Input image

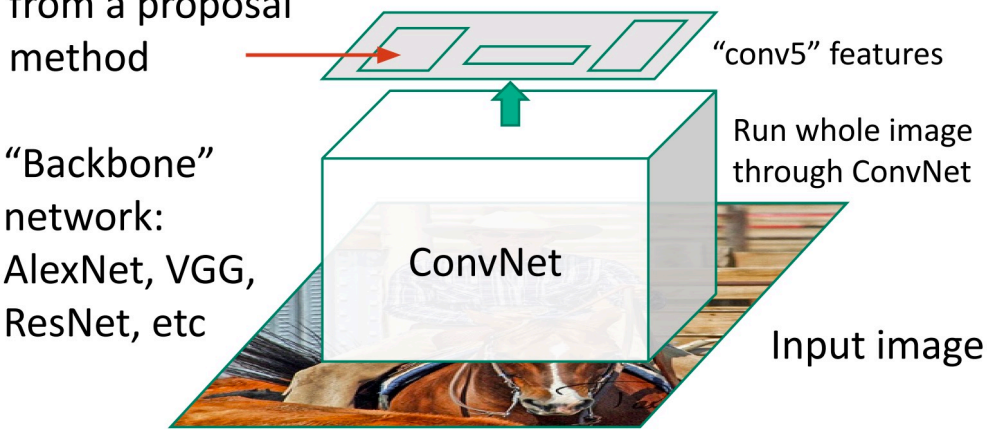


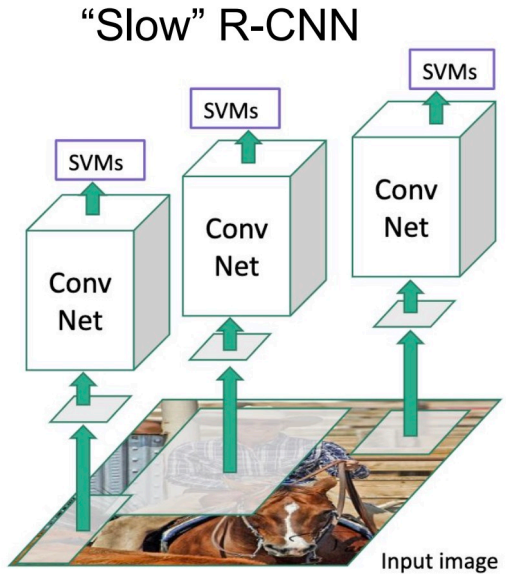
Fast R-CNN


"Backbone" network: AlexNet, VGG, ResNet, etc
ConvNet
"conv5" features
Run whole image through ConvNet
Input image

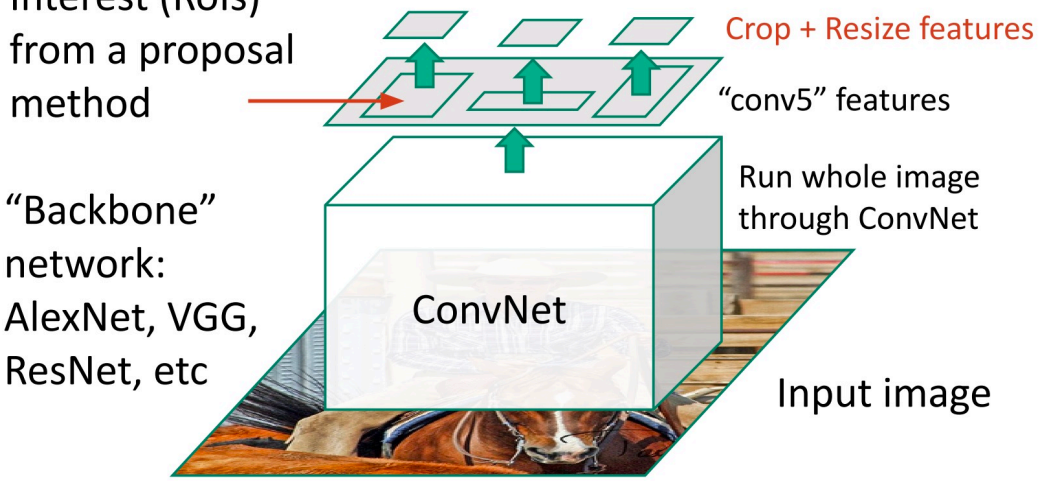

"Slow" R-CNN
SVMs
SVMs
SVMs
Conv Net
Conv Net
Conv Net
Input image

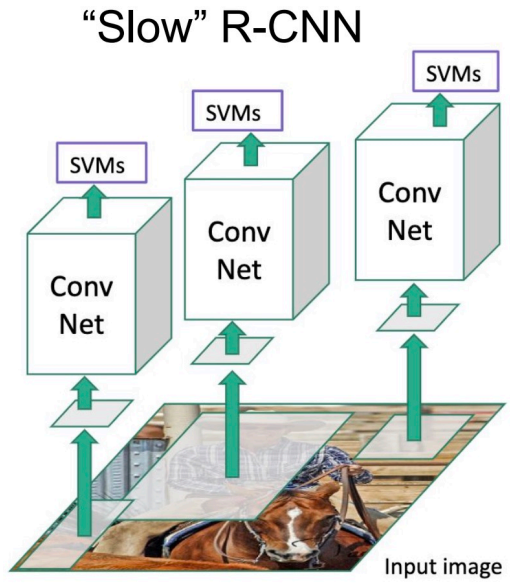
Fast R-CNN


Regions of Interest (RoIs) from a proposal method
"conv5" features
Run whole image through ConvNet
"Backbone" network: AlexNet, VGG, ResNet, etc
ConvNet
Input image

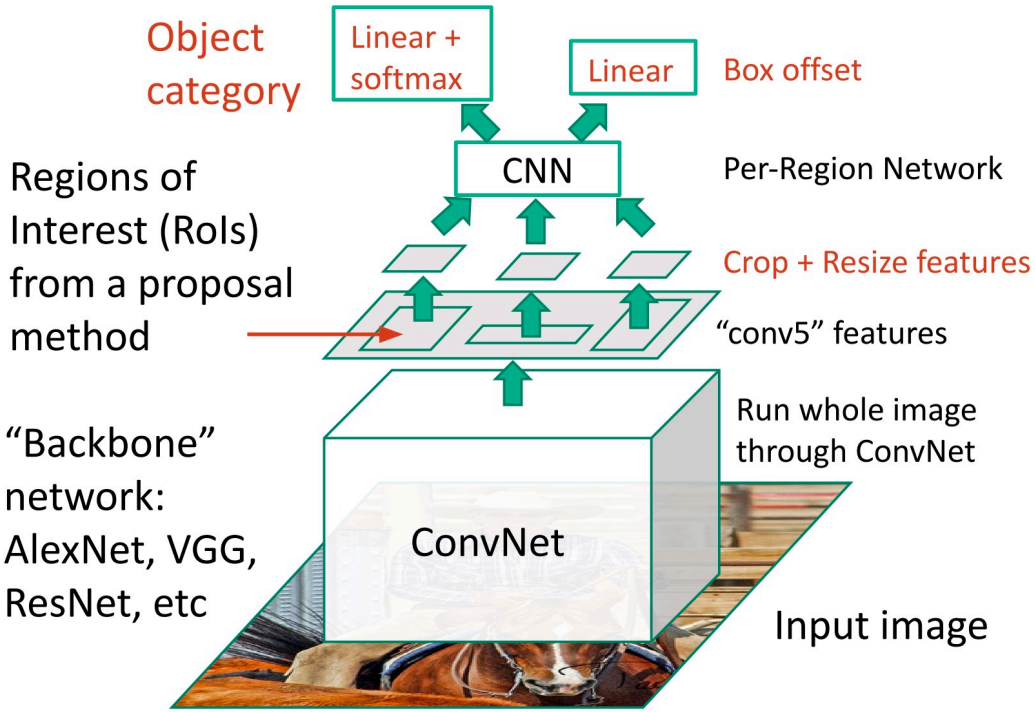

"Slow" R-CNN
SVMs
SVMs
SVMs
Conv Net
Conv Net
Conv Net
Input image

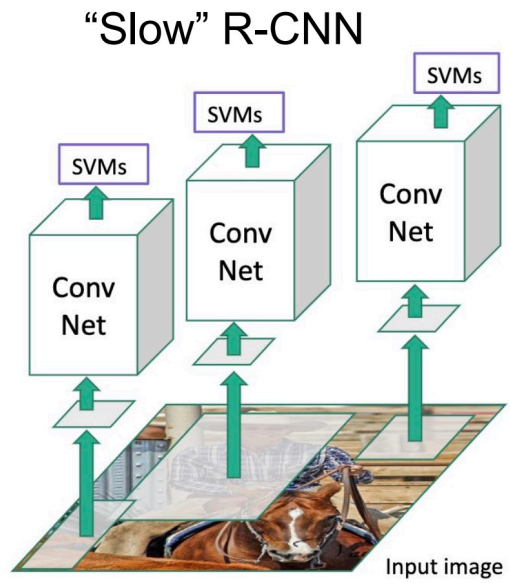
Fast R-CNN


Regions of Interest (RoIs) from a proposal method
Crop + Resize features
"conv5" features
"Backbone" network: AlexNet, VGG, ResNet, etc
Run whole image through ConvNet
ConvNet
Input image


"Slow" R-CNN
SVMs
SVMs
SVMs
Conv Net
Conv Net
Conv Net
Input image

Fast R-CNN


Object category
Linear + softmax
Linear
Box offset
CNN
Per-Region Network
Regions of Interest (RoIs) from a proposal method
Crop + Resize features
"conv5" features
Run whole image through ConvNet
"Backbone" network: AlexNet, VGG, ResNet, etc
ConvNet
Input image


"Slow" R-CNN
SVMs
SVMs
SVMs
Conv Net
Conv Net
Conv Net
Input image

Thank you!