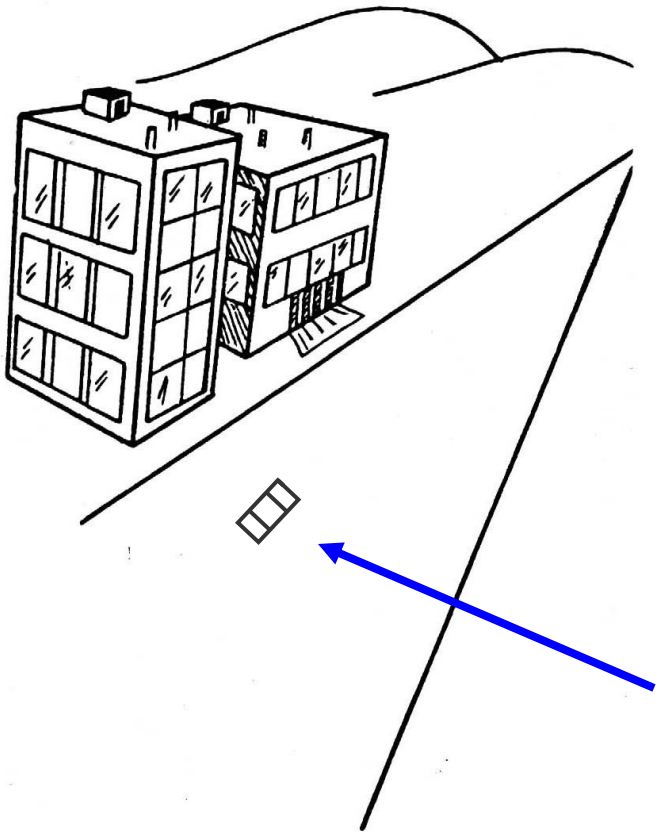
Image Segmentation

How many zebras?

From Sandlot Science

Why context is important?

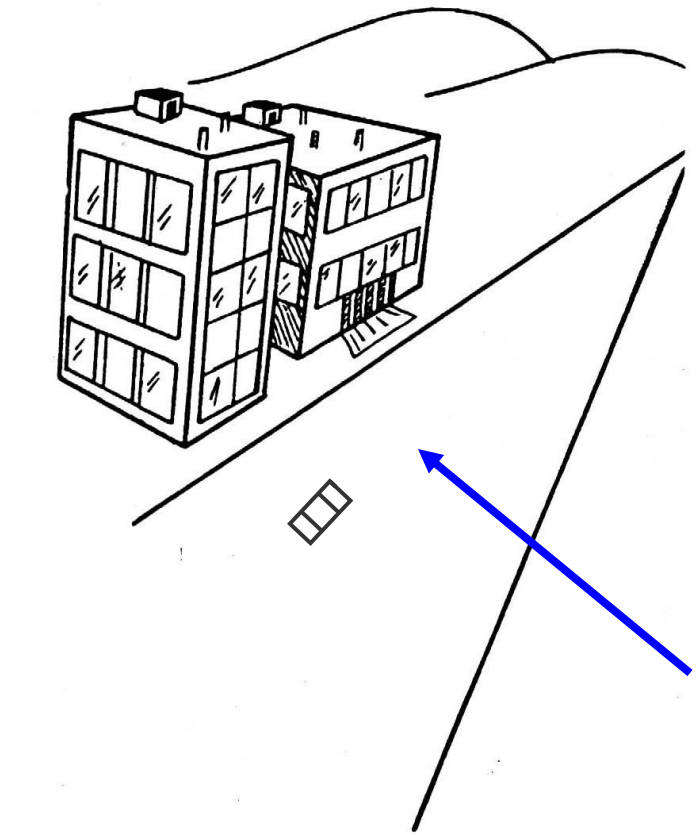


What is this?

slide by Takeo Kanade

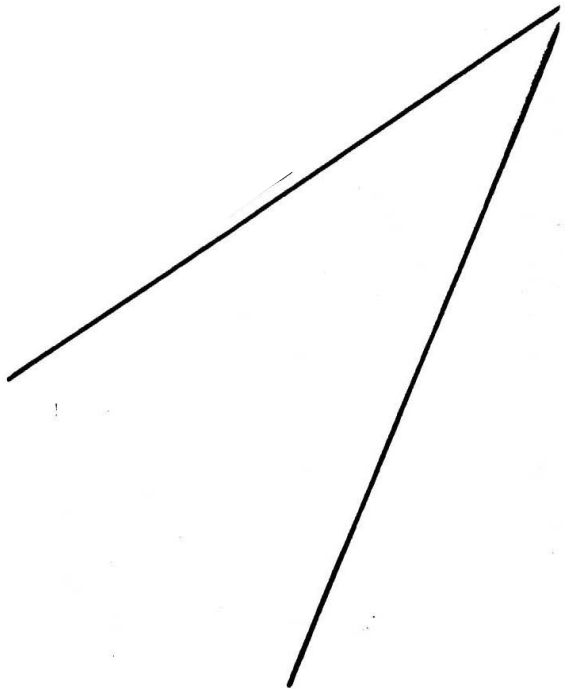
Why is this a car?

…because it's on the road!



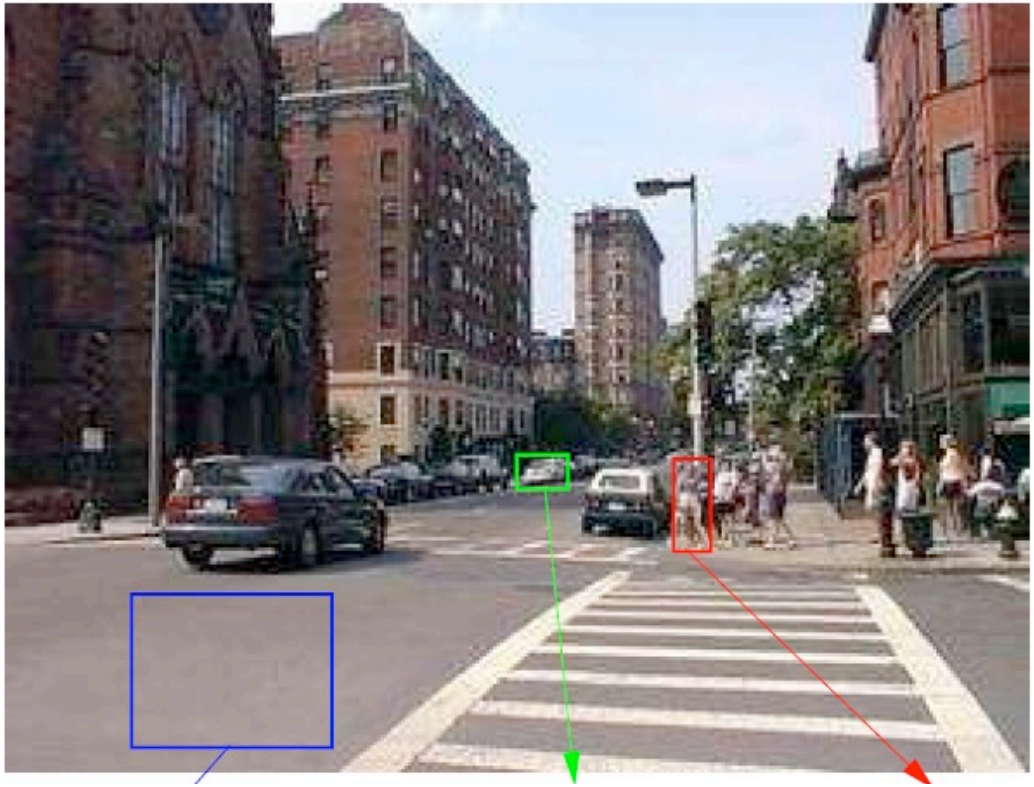
Why is this road?

Why is this a road?

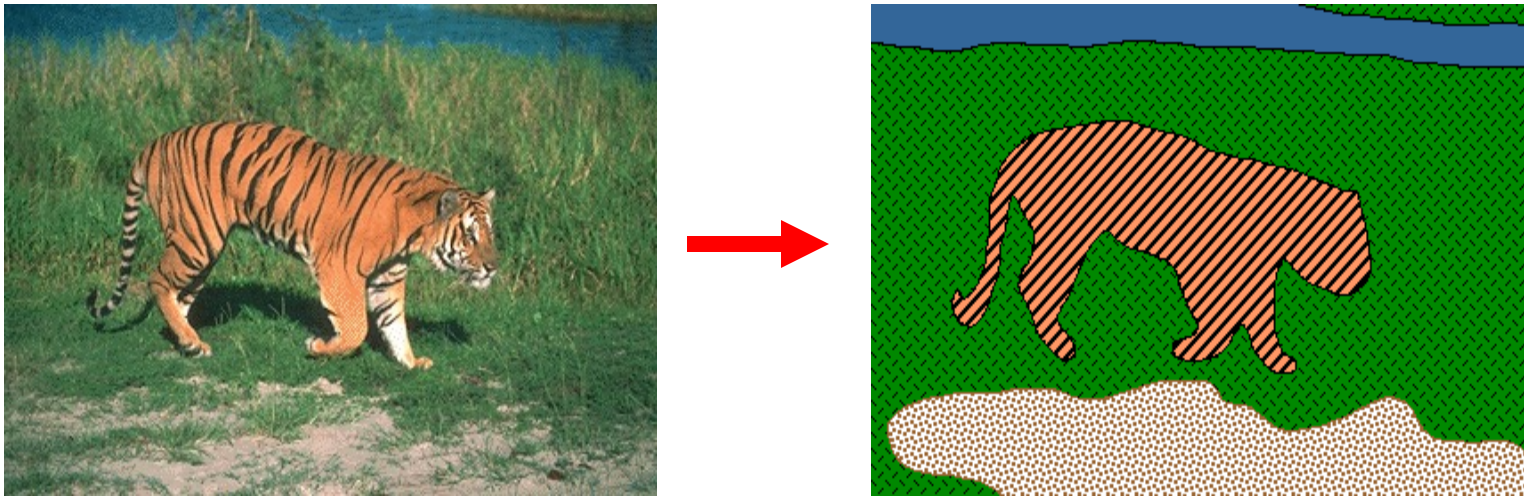


Context is very important!

Same problem in real scenes



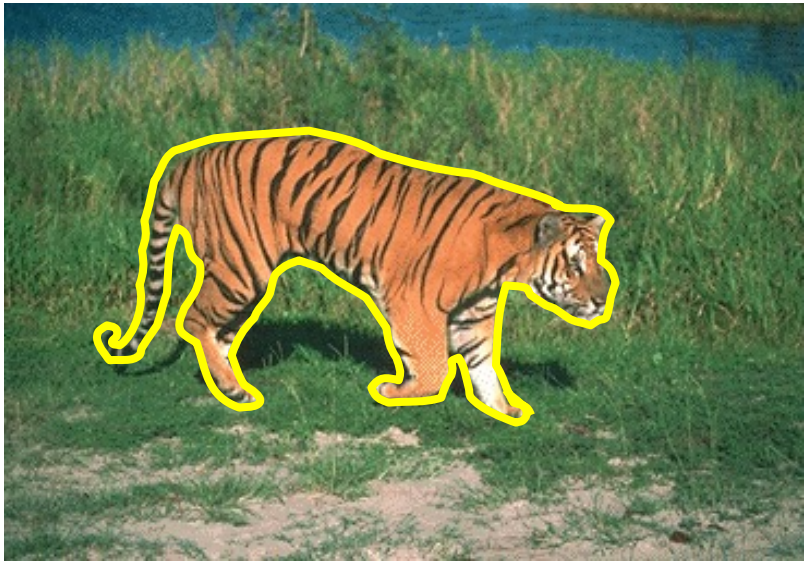
From images to objects



What defines an object?

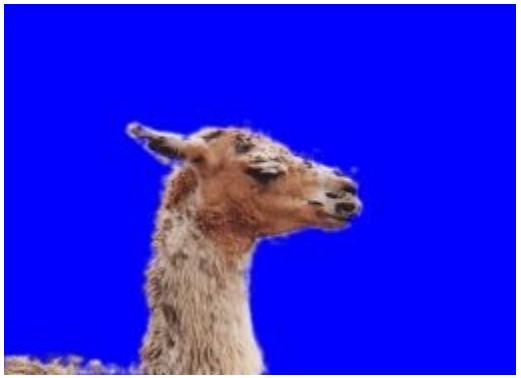
- Subjective problem, but has been well-studied

Extracting objects



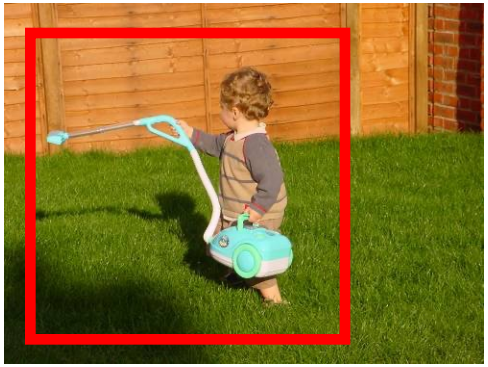
How could we do this automatically (or at least semi-automatically)?

Semi-automatic binary segmentation

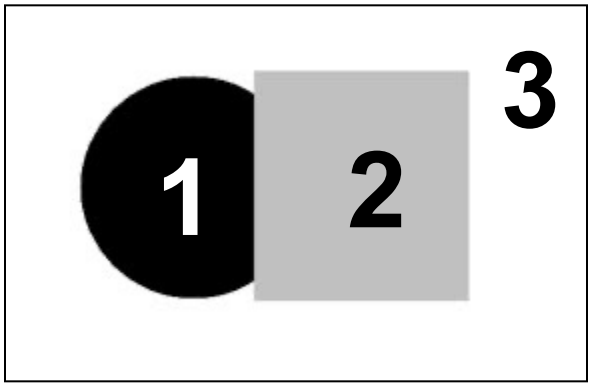


Simplifying the user interaction

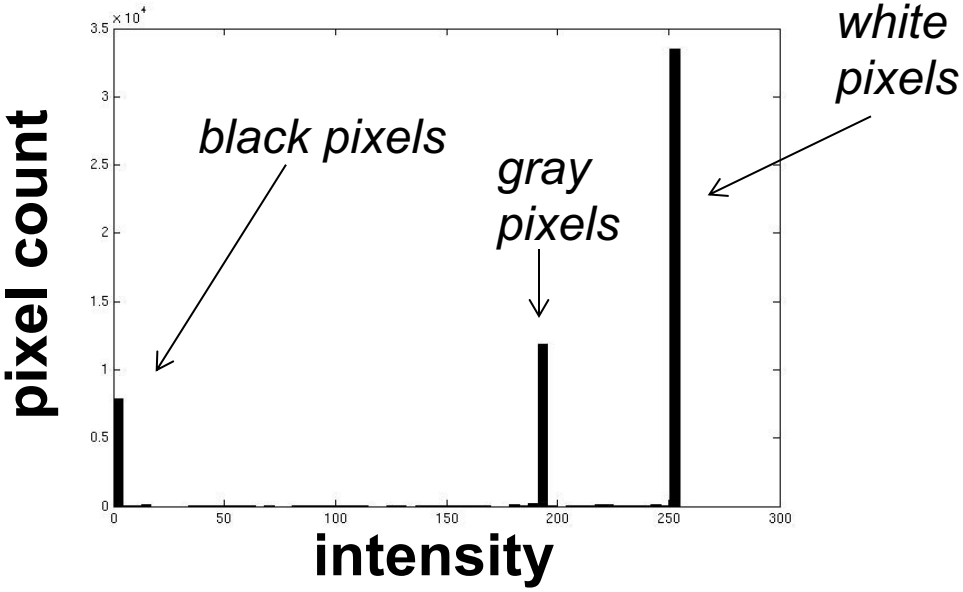
Grabcut [Rother et al., SIGGRAPH 2004]



Auto segmentation: toy example

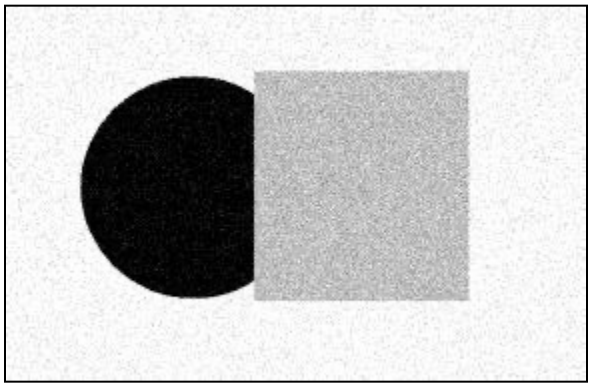


input image

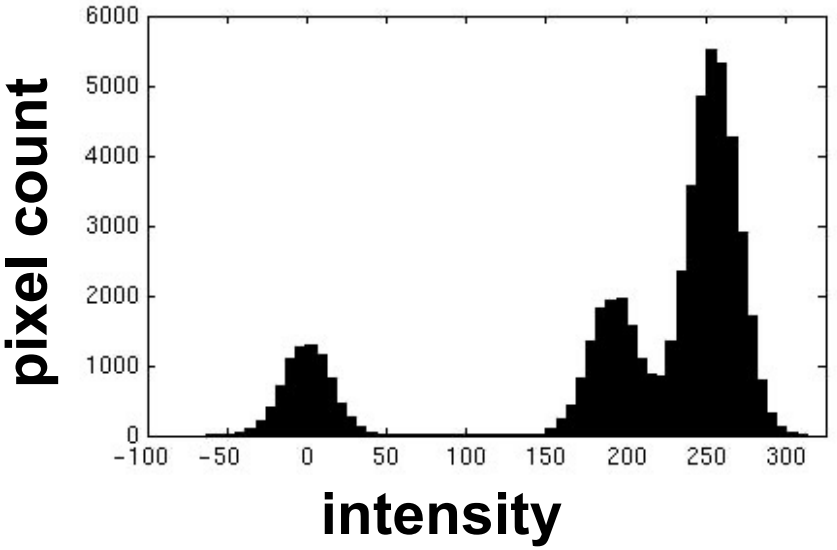


- These intensities define the three groups.
- We could label every pixel in the image according to which of these primary intensities it is.
  - i.e., *segment* the image based on the intensity feature.
- But … image isn't quite so simple …

Source: K. Grauman



input image

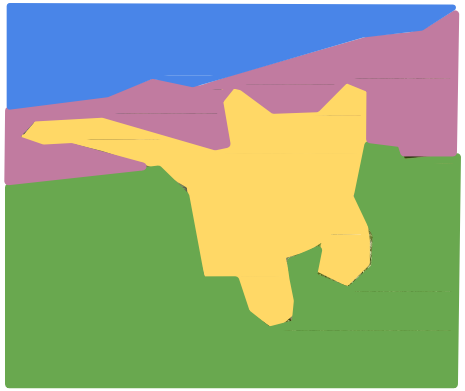


- Now how to determine the three main intensities that define our groups?
- We need to ***cluster.***

Source: K. Grauman

Deep Learning

Semantic Segmentation



GRASS, CAT, TREE, SKY

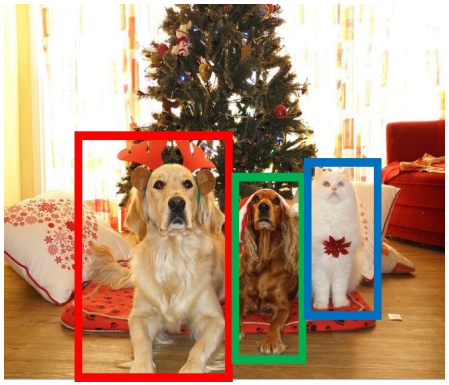
Pixel-level

Classification + Localization

CAT

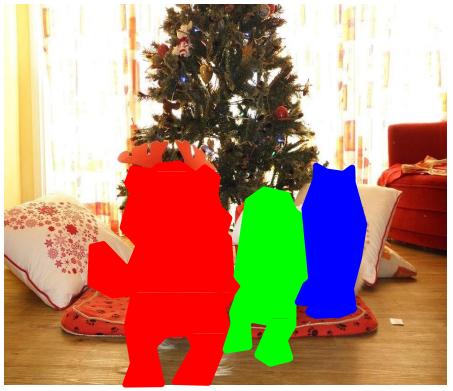
Single Object

Object Detection



DOG, DOG, CAT

Instance Segmentation



DOG, DOG, CAT

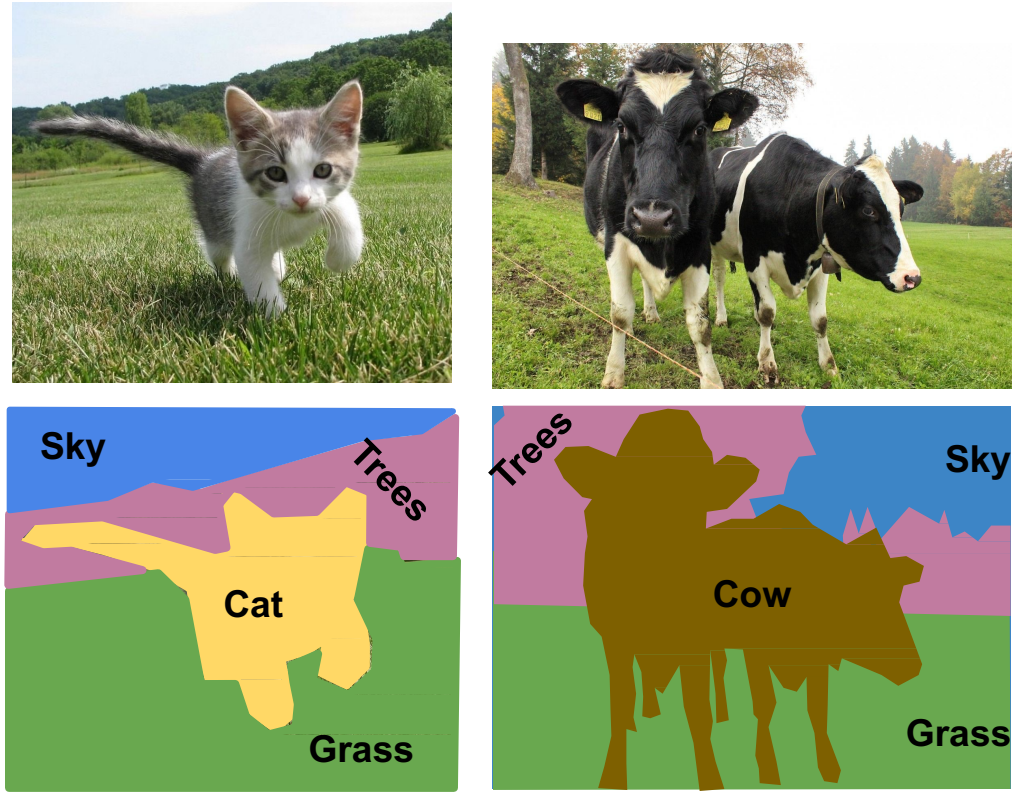
Multiple Object

Segmentation+Classification

Slide by: Justin Johnson

Semantic Segmentation

Label each pixel in the image with a category label

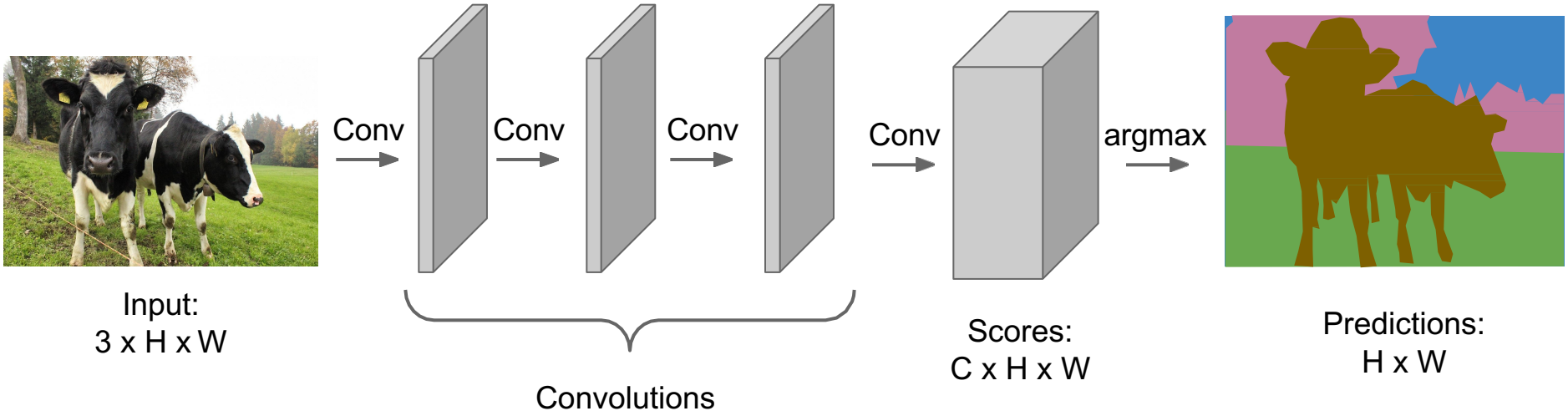


Don’t differentiate instances, only care about pixels

Slide by: Justin Johnson

Semantic Segmentation Idea: Fully Convolutional

Design a network as a bunch of convolutional layers to make predictions for pixels all at once!



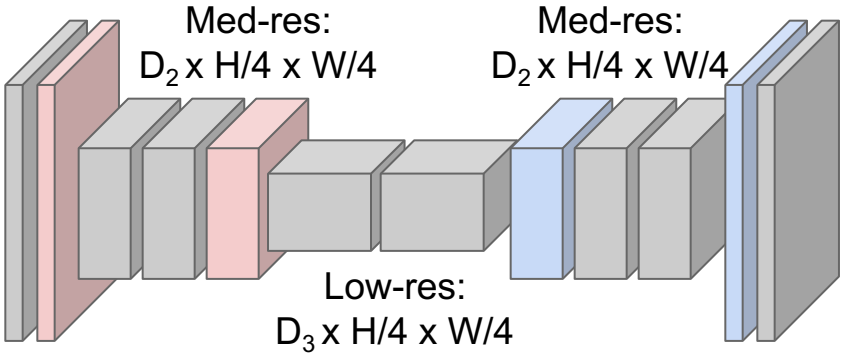
Each channel is a class
C channels->C classes

Slide by: Justin Johnson

Semantic Segmentation Idea: Fully Convolutional

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

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

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

Slide by: Justin Johnson

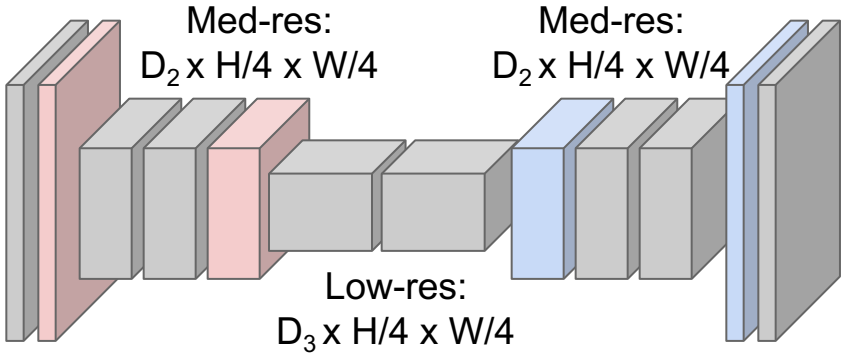
Semantic Segmentation Idea: Fully Convolutional

Downsampling: Pooling, strided convolution

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

Upsampling: ???

Input:
3 x H x W



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

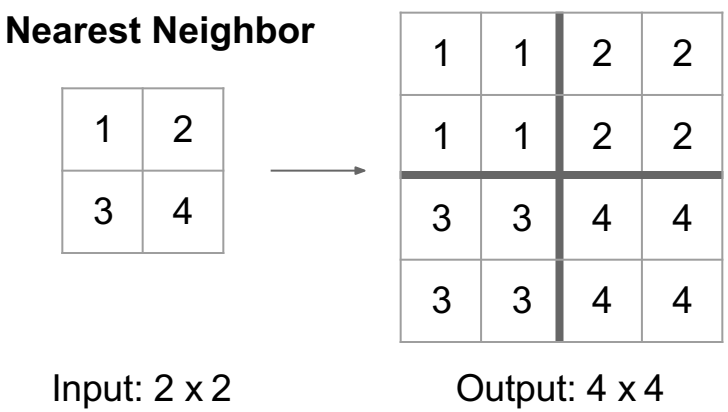
High-res:
D_1 x H/2 x W/2

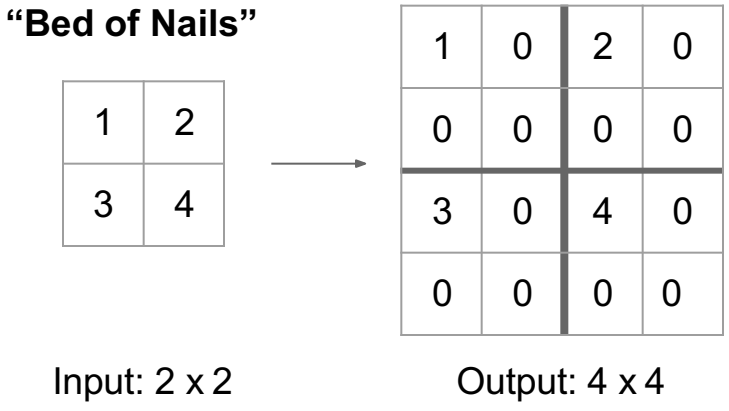
Predictions:
H x W

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

Slide by: Justin Johnson

In-Network upsampling: “Unpooling”





Slide by: Justin Johnson

In-Network upsampling: “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

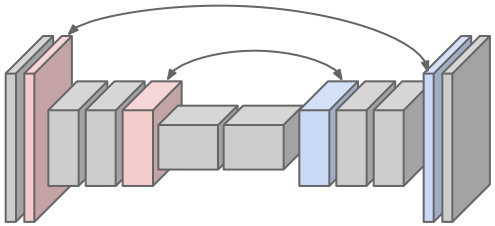
1	2
3	4

Input: 2 x 2

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

Output: 4 x 4

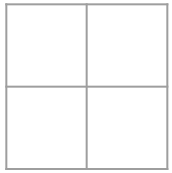
Corresponding pairs of downsampling and upsampling layers



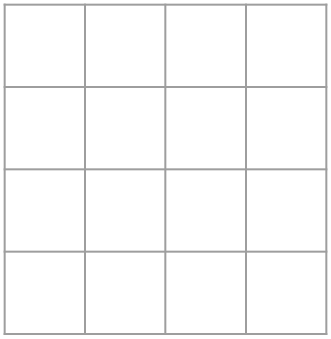
Slide by: Justin Johnson

Learnable Upsampling

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



Input: 2 x 2

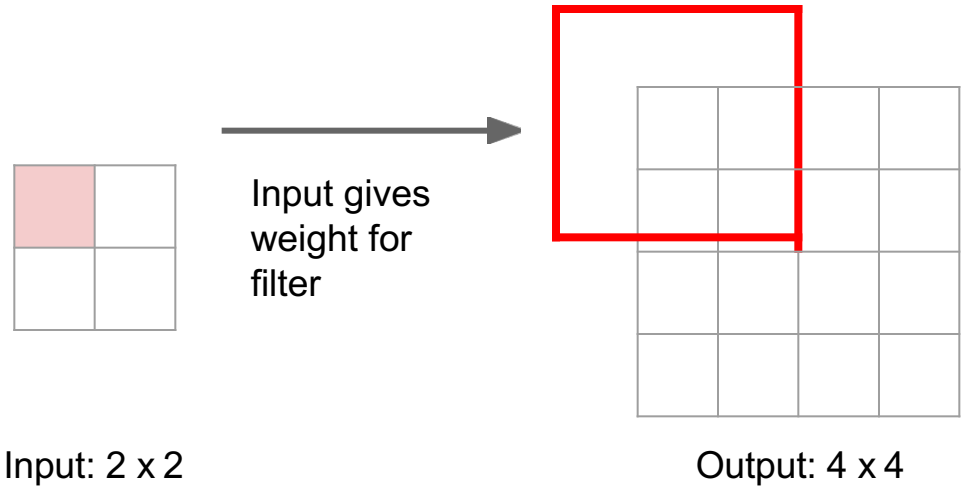


Output: 4 x 4

Slide by: Justin Johnson

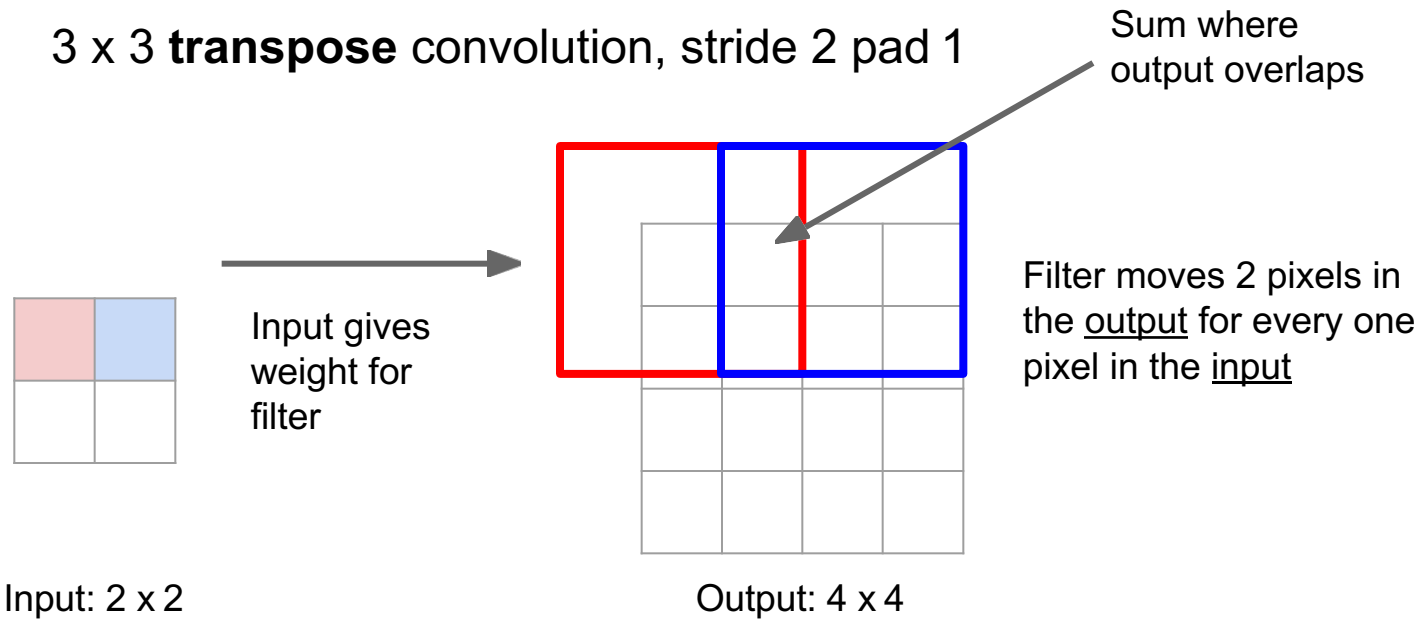
Learnable Upsampling

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



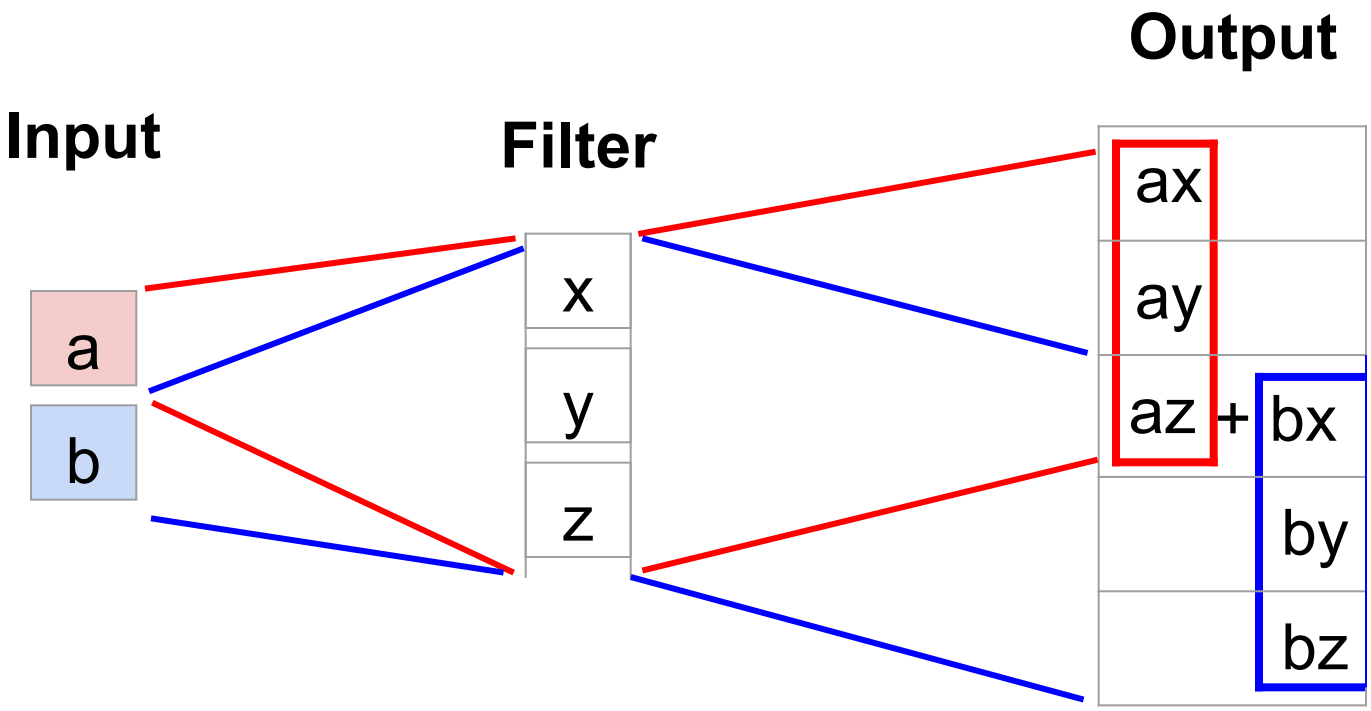
Slide by: Justin Johnson

Learnable Upsampling



Slide by: Justin Johnson

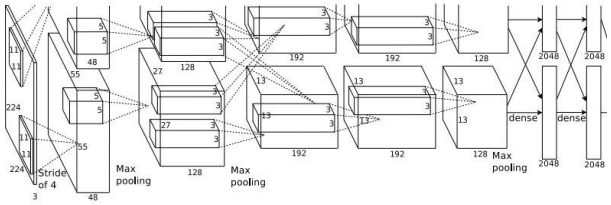
Transpose Convolution: 1D Example



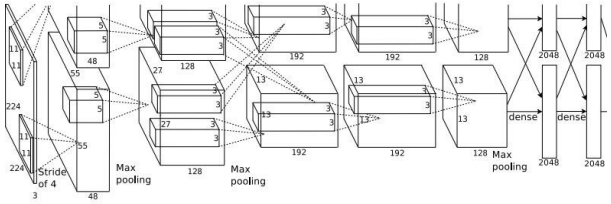
Output contains copies of the filter weighted by the input, summing at where at overlaps in the output

Adapted from Justin Johnson

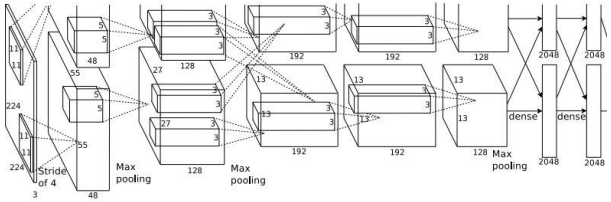
Object Detection as Regression?



CAT: (x, y, w, h)



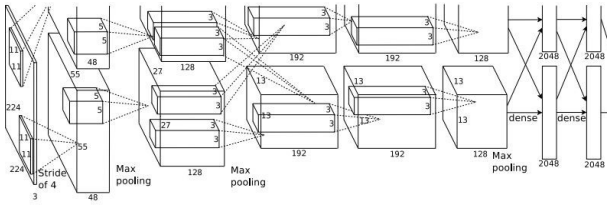
DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)



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

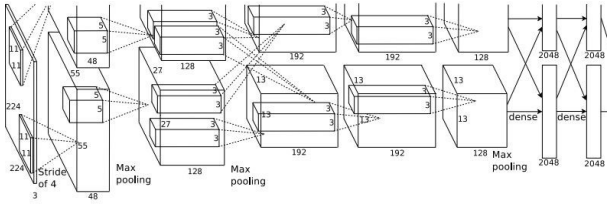
Slide by: Justin Johnson

Object Detection as Regression?



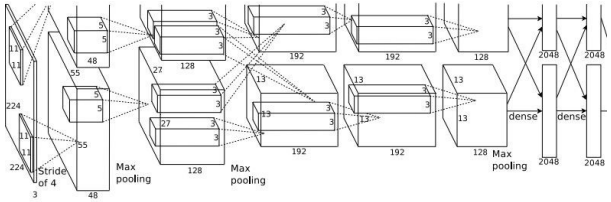
CAT: (x, y, w, h)

4 numbers



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

16 numbers



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

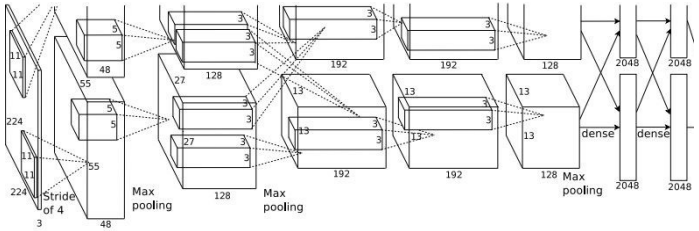
Many numbers!

Each image needs a different number of outputs!

Slide by: Justin Johnson

Object Detection as Classification: Sliding Window

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

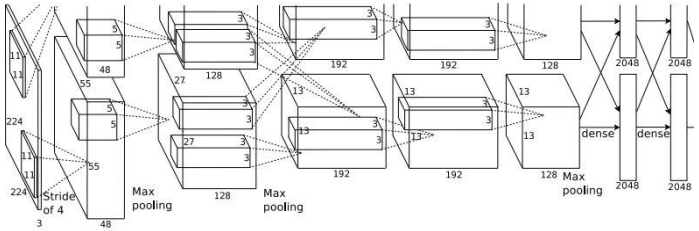


Dog? NO Cat? NO
Background? YES

Slide by: Justin Johnson

Object Detection as Classification: Sliding Window

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

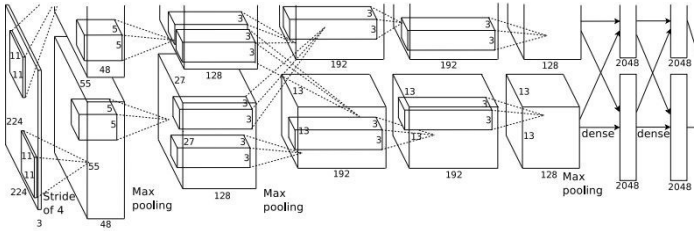


Dog? YES
Cat? NO
Background? NO

Slide by: Justin Johnson

Object Detection as Classification: Sliding Window

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

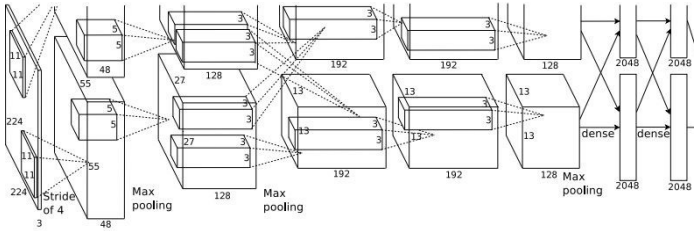


Dog? YES
Cat? NO
Background? NO

Slide by: Justin Johnson

Object Detection as Classification: Sliding Window

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

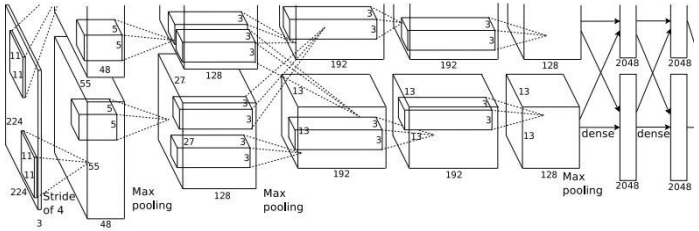


Dog? NO Cat? YES
Background? NO

Slide by: Justin Johnson

Object Detection as Classification: Sliding Window

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



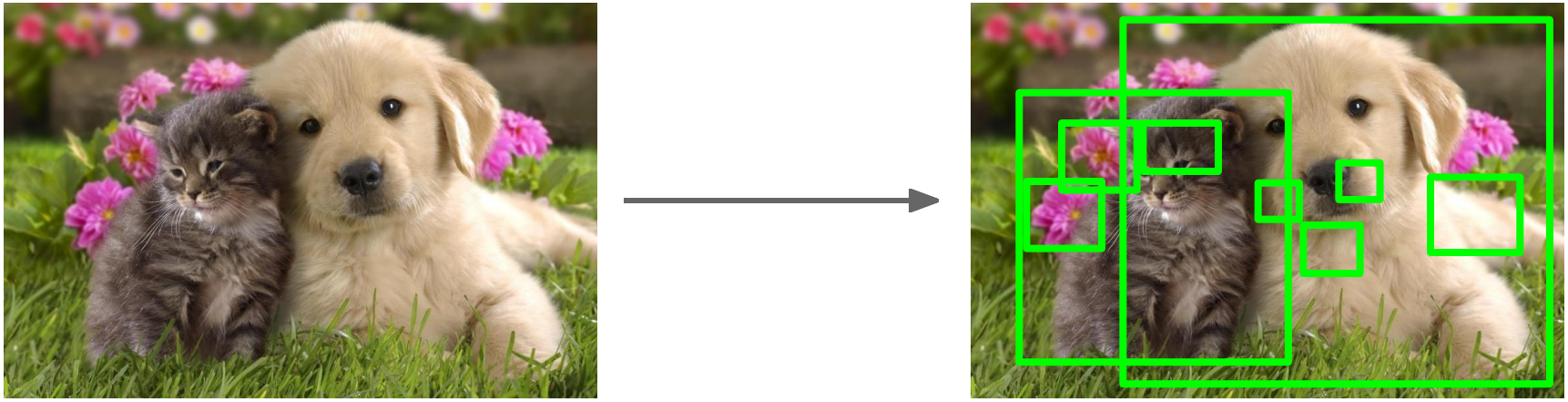
Dog? NO Cat? YES
Background? NO

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

Slide by: Justin Johnson

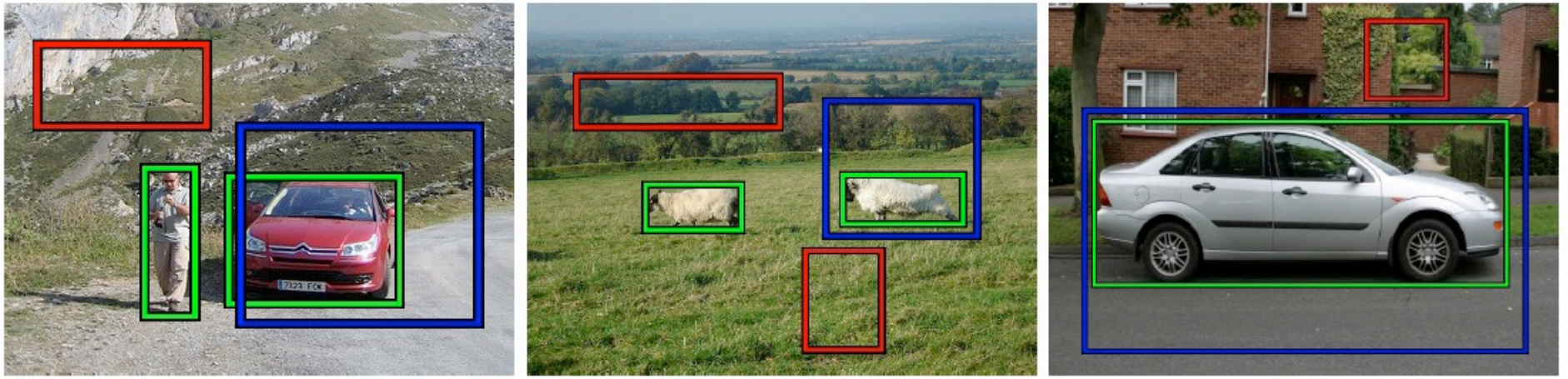
Region Proposals

- Find image regions that are likely to contain objects
- Relatively fast to run; e.g. Selective Search gives 1000 region proposals in a few seconds on CPU



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

Slide by: Justin Johnson



(a) (b) (c)

Fig. 1: **Desired behavior of an objectness measure.** *The desired objectness measure should score the blue windows, partially covering the objects, lower than the ground truth windows (green), and score even lower the red windows containing only stuff or small parts of objects.*

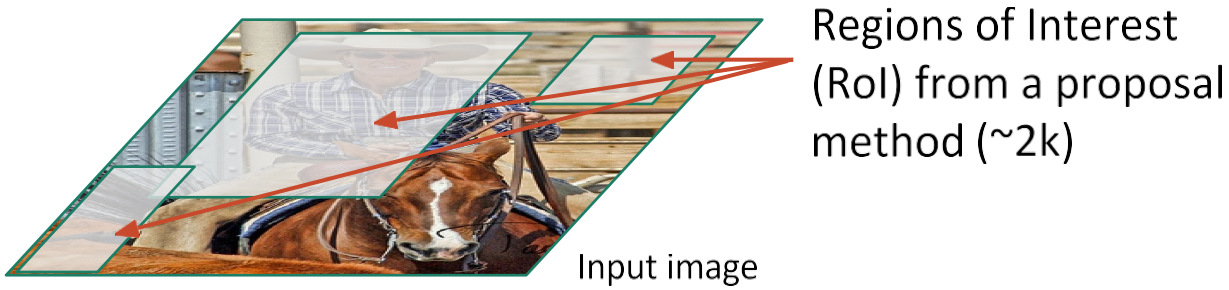
Alexe et al., CVPR 2010

R-CNN

Input image

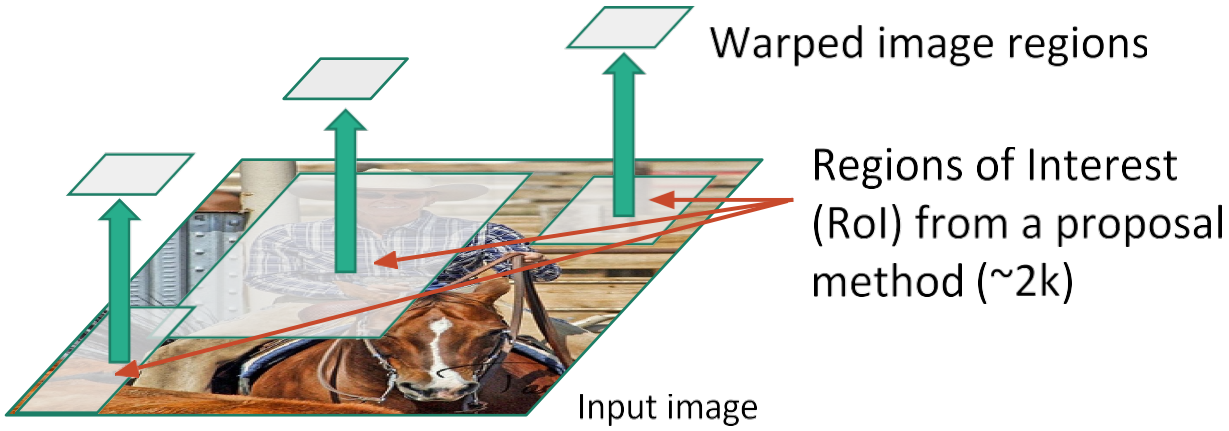
Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

R-CNN



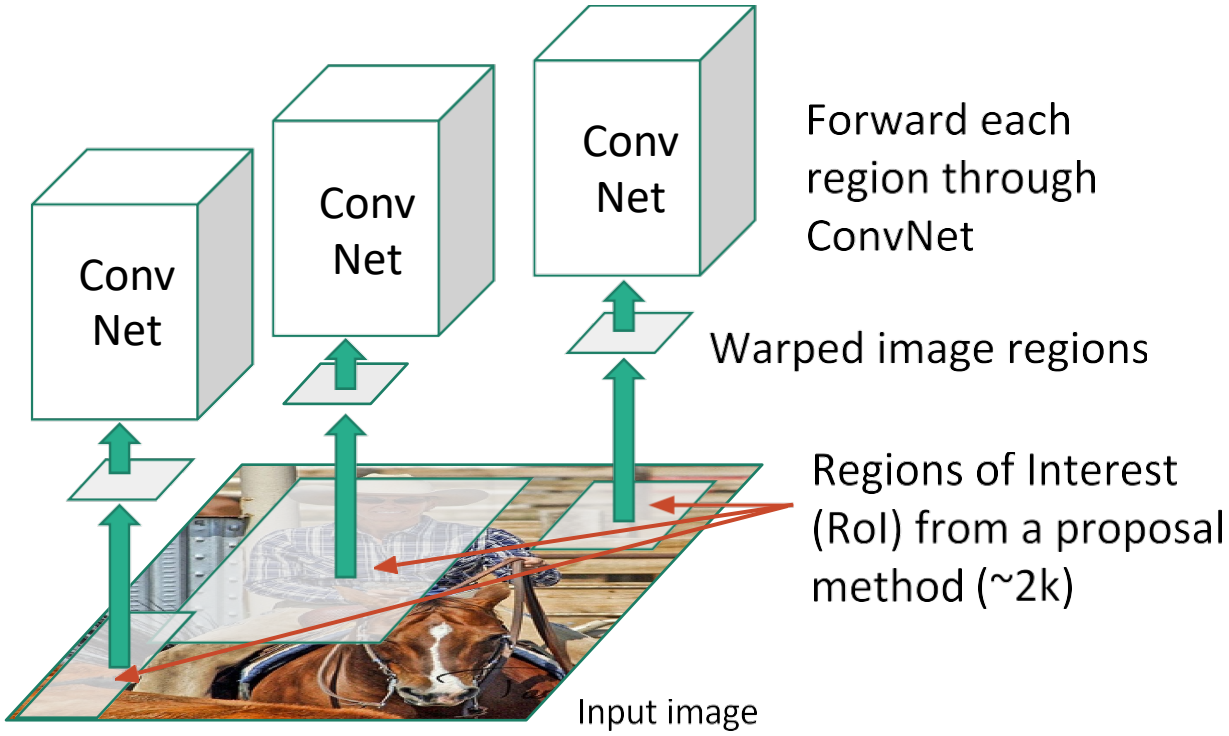
Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

R-CNN



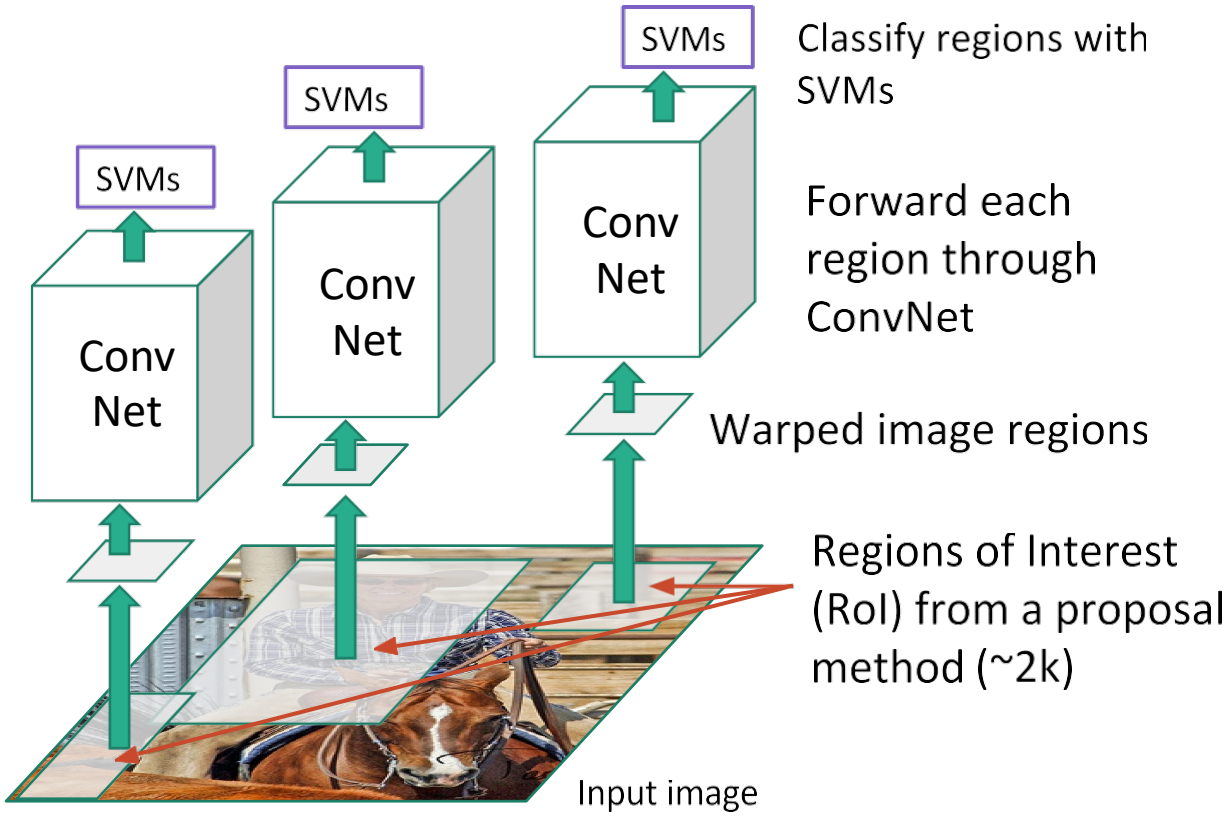
Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

R-CNN



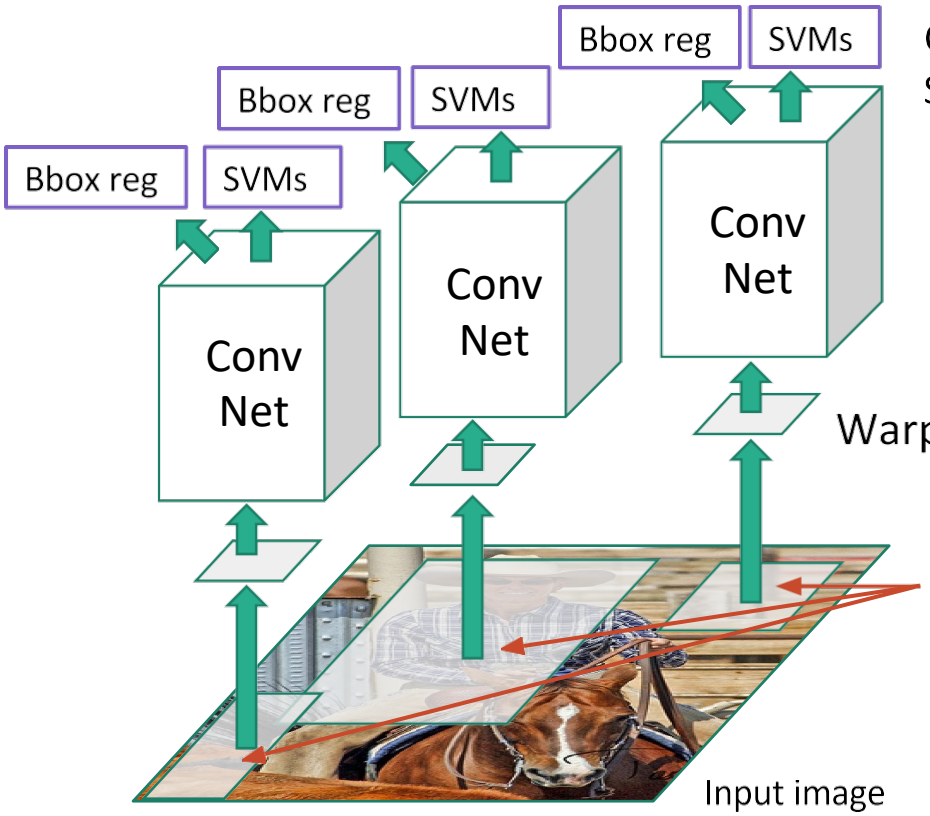
Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

R-CNN



Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

R-CNN



Linear Regression for bounding box offsets
Classify regions with SVMs

Forward each region through ConvNet

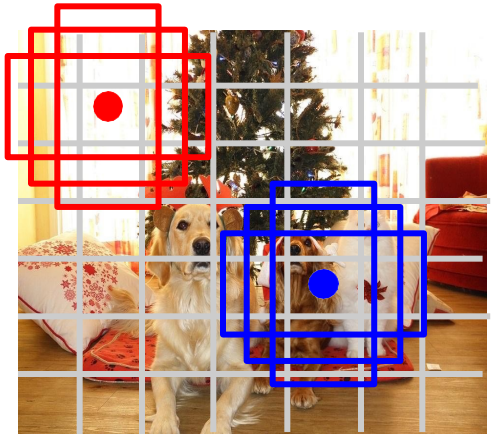
Warped image regions

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

Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

Detection without Proposals: YOLO

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)

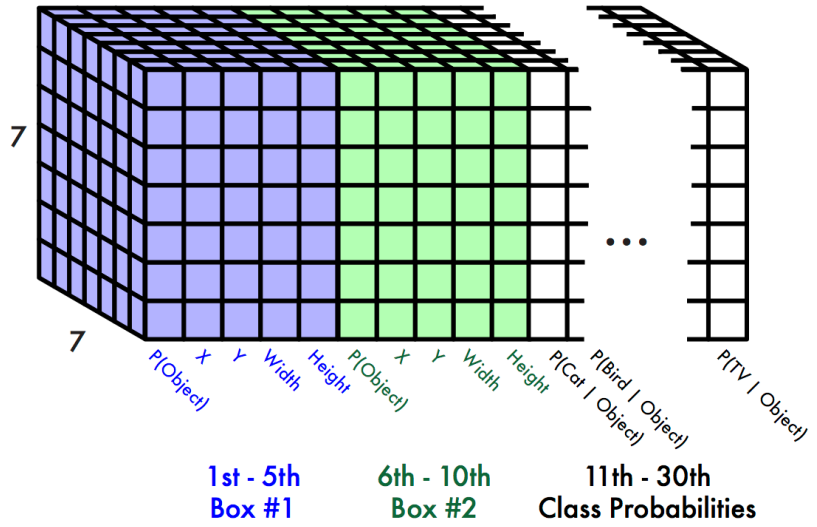
Output:
7 x 7 x (5 * B + C)

Redmon et al, "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016
Liu et al, "SSD: Single-Shot MultiBox Detector", ECCV 2016

Slide by: Justin Johnson

This parameterization fixes the output size

- Each cell predicts:
  - For each bounding box:
    - 4 coordinates (x, y, w, h)
    - 1 confidence value
  - Some number of class probabilities

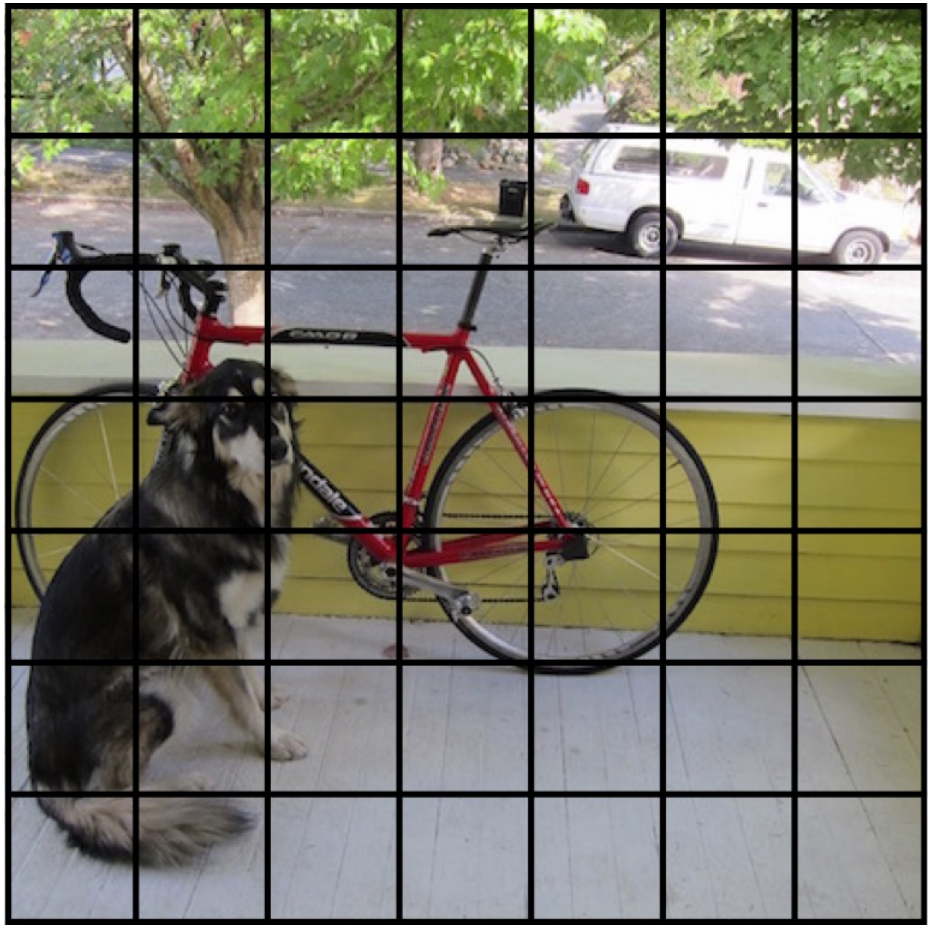


- For Pascal VOC:
  - 7x7 grid
  - 2 bounding boxes / cell
  - 20 classes

- 7 x 7 x (2 x 5 + 20) = 7 x 7 x 30 tensor = **1470 outputs**

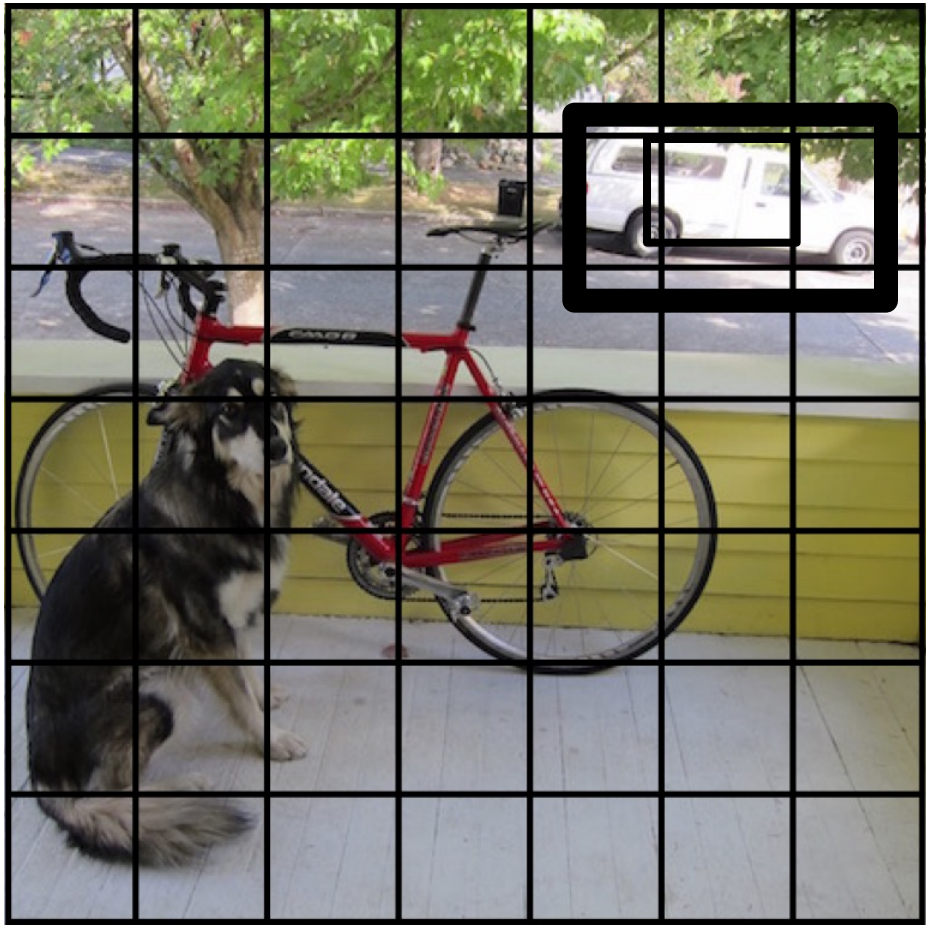
Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016

Split the image into a grid



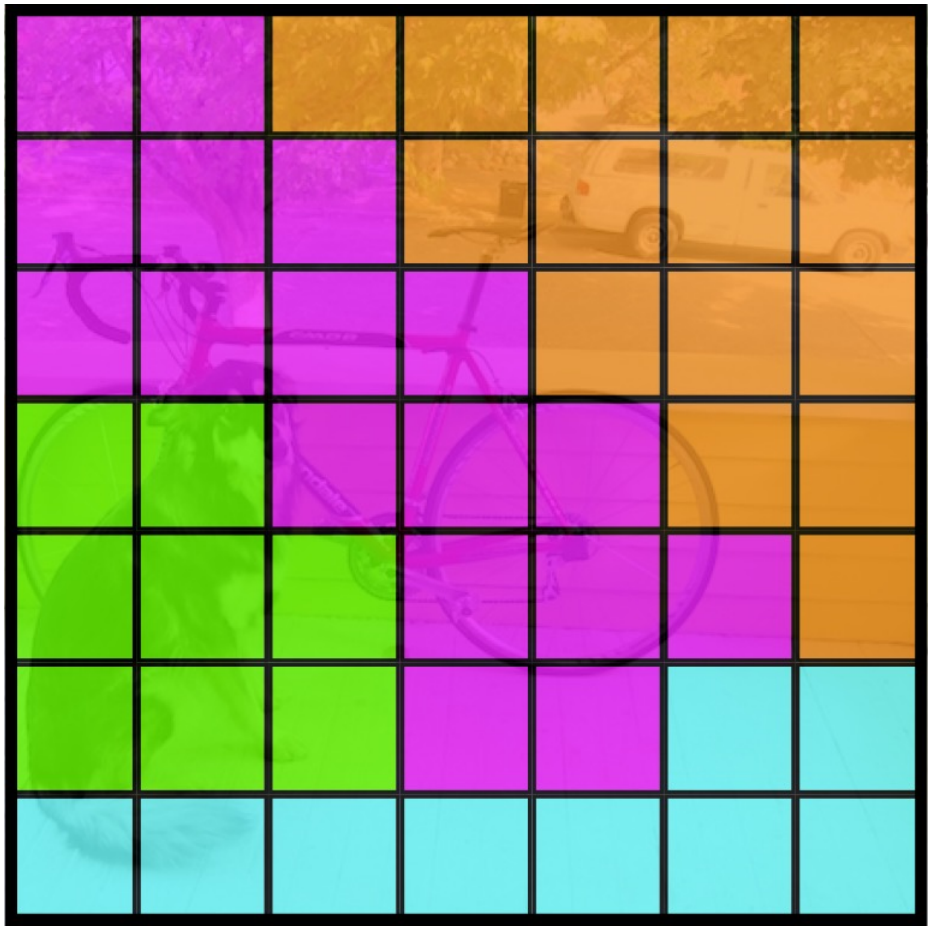
Redmon et al., “You Only Look Once: Unified, Real-Time Object Detection”, CVPR 2016

Each cell predicts boxes and confidences: P(Object)



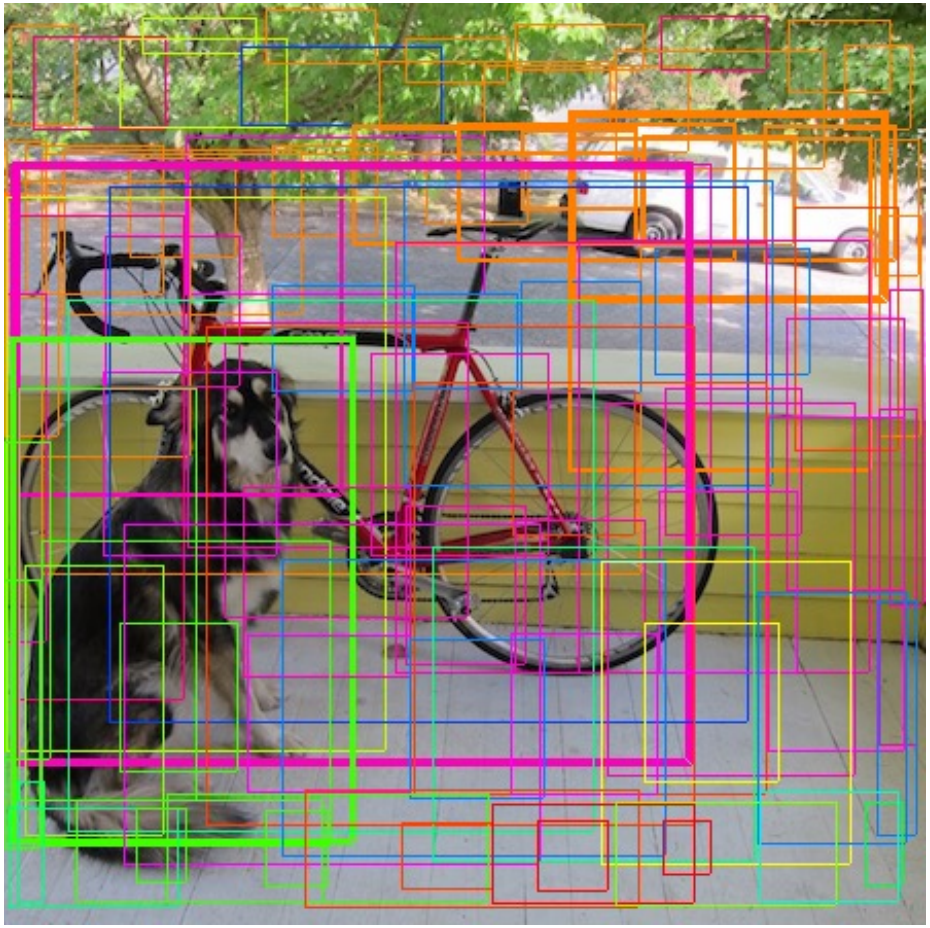
Redmon et al., “You Only Look Once: Unified, Real-Time Object Detection”, CVPR 2016

Each cell also predicts a probability P(Class | Object)



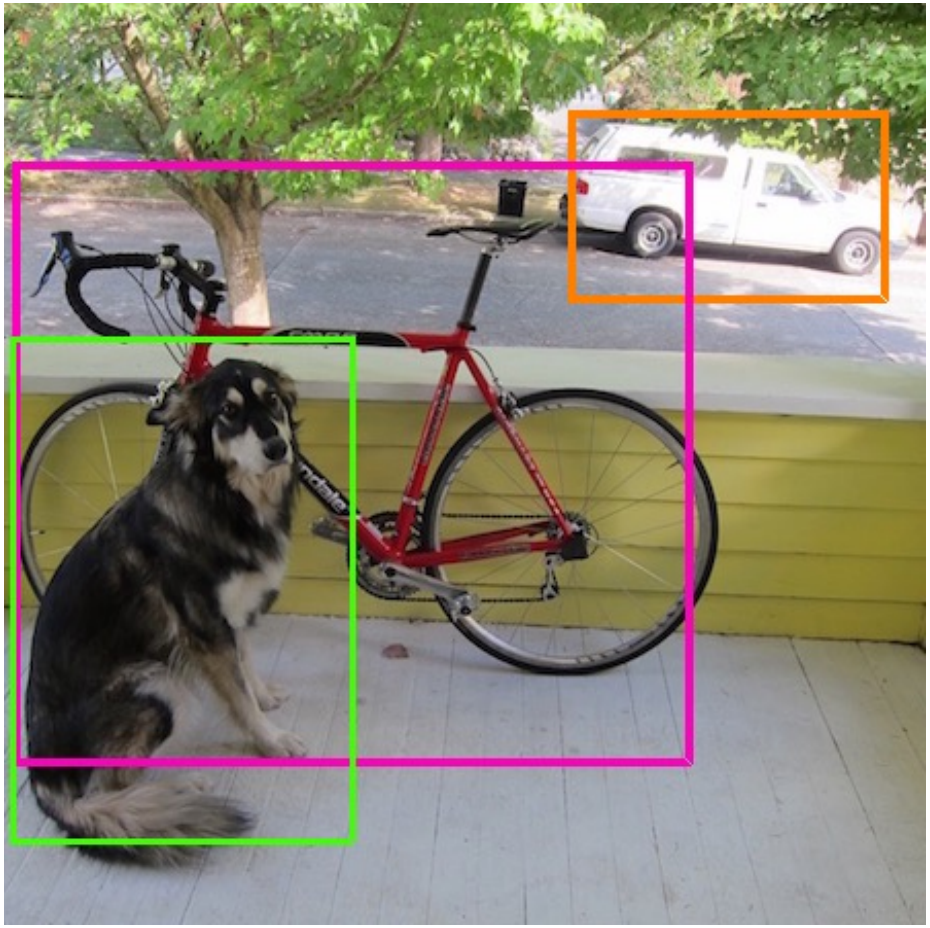
Redmon et al., “You Only Look Once: Unified, Real-Time Object Detection”, CVPR 2016

Combine the box and class predictions



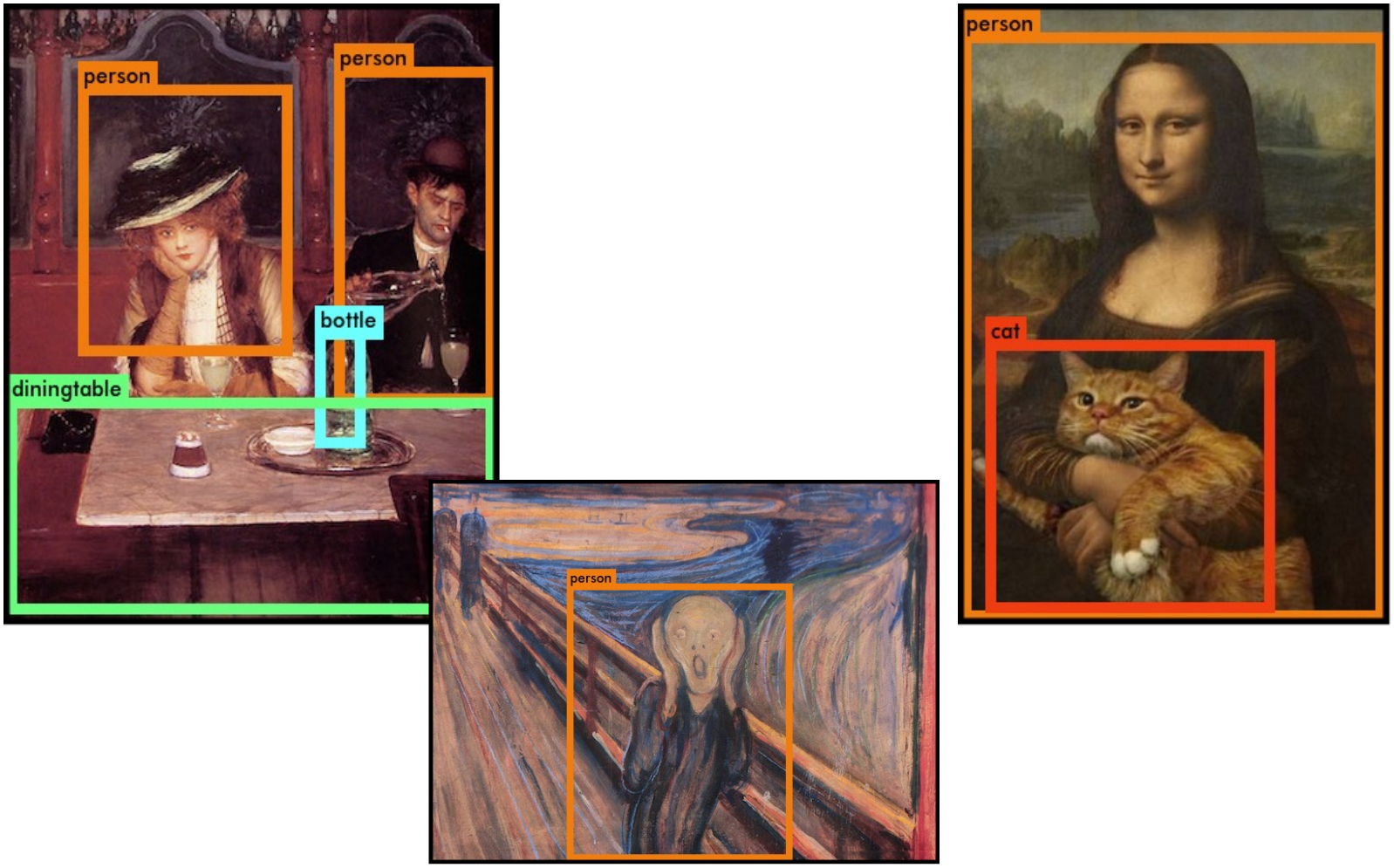
Redmon et al., “You Only Look Once: Unified, Real-Time Object Detection”, CVPR 2016

Finally do non-maximum suppression and threshold detections



Redmon et al., “You Only Look Once: Unified, Real-Time Object Detection”, CVPR 2016

It also generalizes well to new domains



Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016