

Lecture 14a: Beyond classification – object detection and segmentation

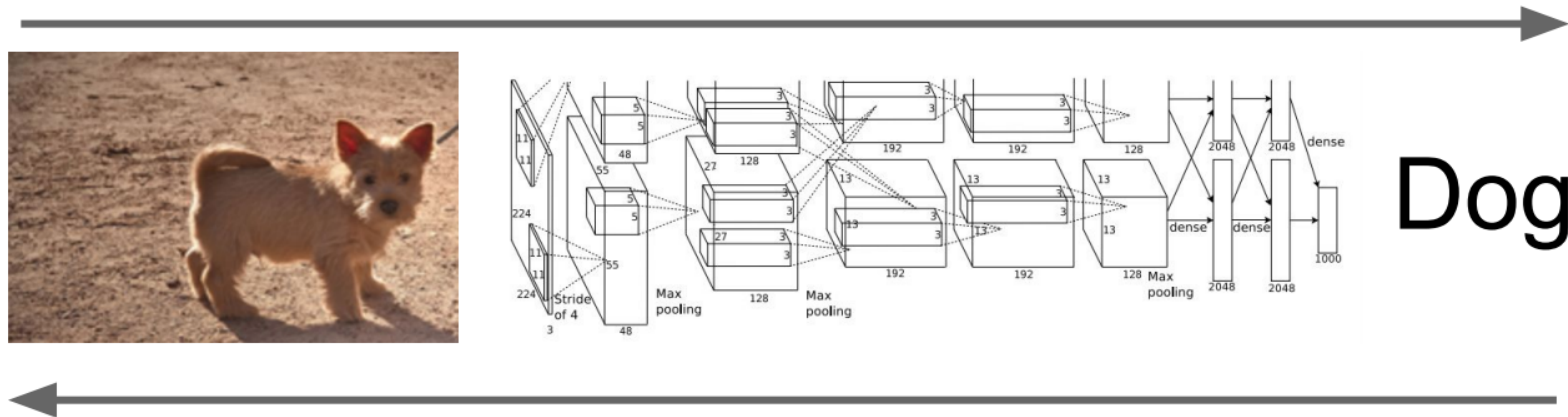
Machine Learning and Imaging

BME 548L

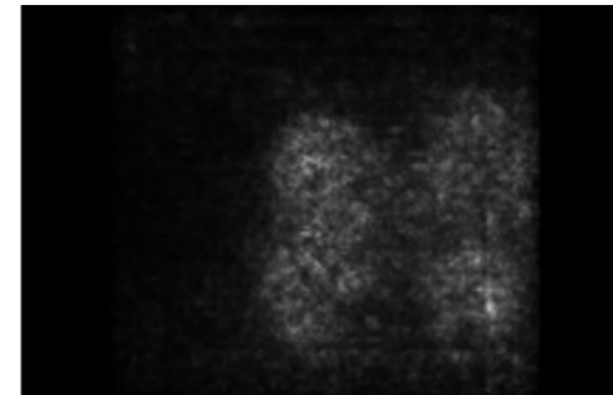
Roarke Horstmeyer

Which pixels matter: Saliency via Backprop

Forward pass: Compute probabilities

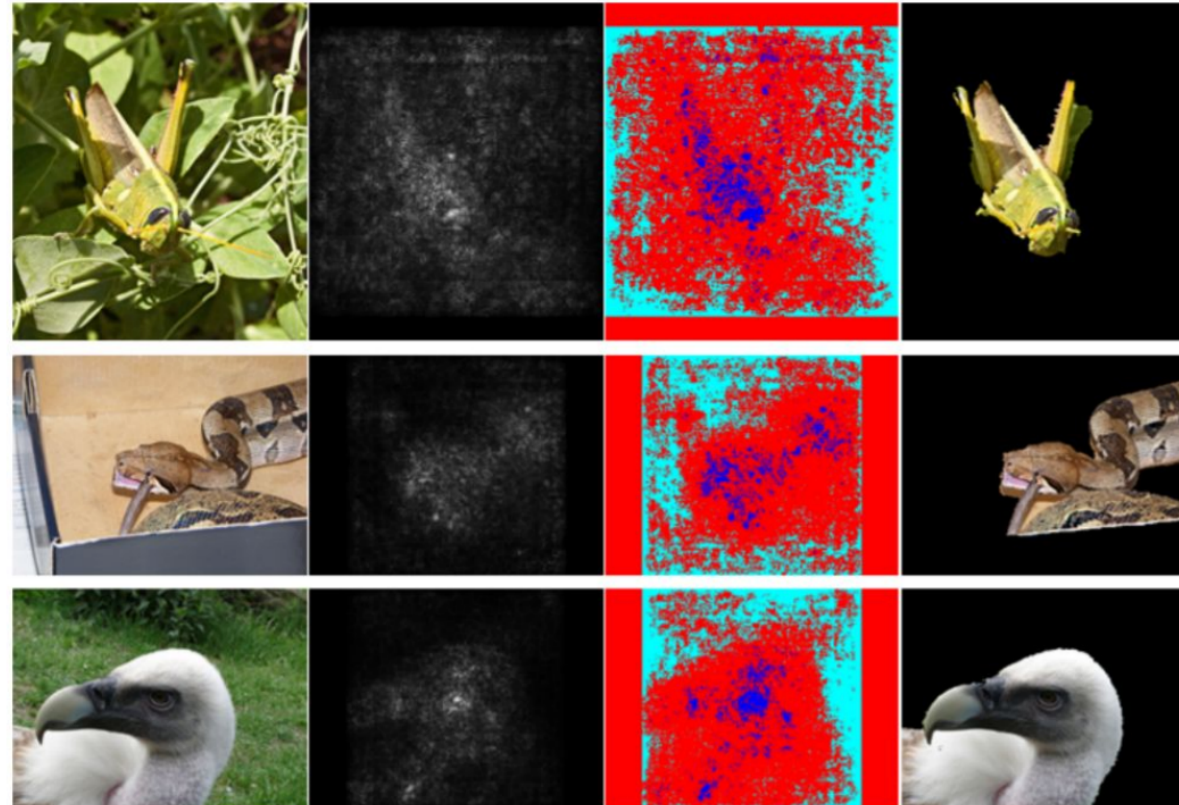


Compute gradient of (unnormalized) class score with respect to image pixels, take absolute value and max over RGB channels



Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.
Figures copyright Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, 2014; reproduced with permission.

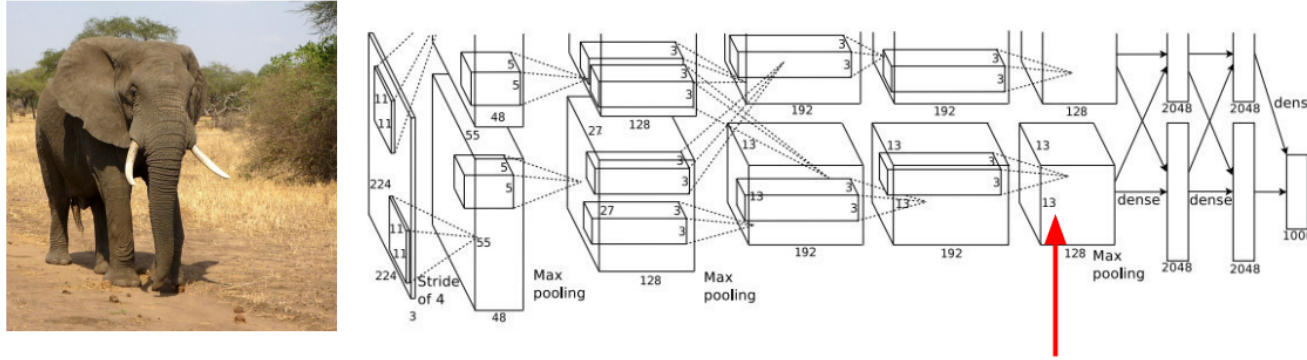
Saliency Maps: Segmentation without supervision



Use GrabCut on saliency map

Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.
Figures copyright Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, 2014; reproduced with permission.
Rother et al. "Grabcut: Interactive foreground extraction using iterated graph cuts". ACM TOG 2004

Intermediate Features via (guided) backprop



Pick a single intermediate neuron, e.g. one value in 128 x 13 x 13 conv5 feature map

Compute gradient of neuron value with respect to image pixels

Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014
Springenberg et al, "Striving for Simplicity: The All Convolutional Net", ICLR Workshop 2015

Intermediate features via (guided) backprop



Maximally activating patches
(Each row is a different neuron)



Guided Backprop

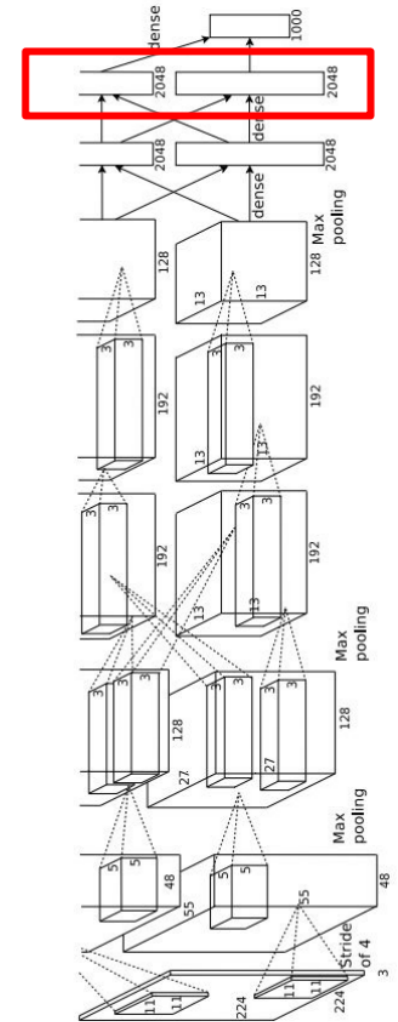
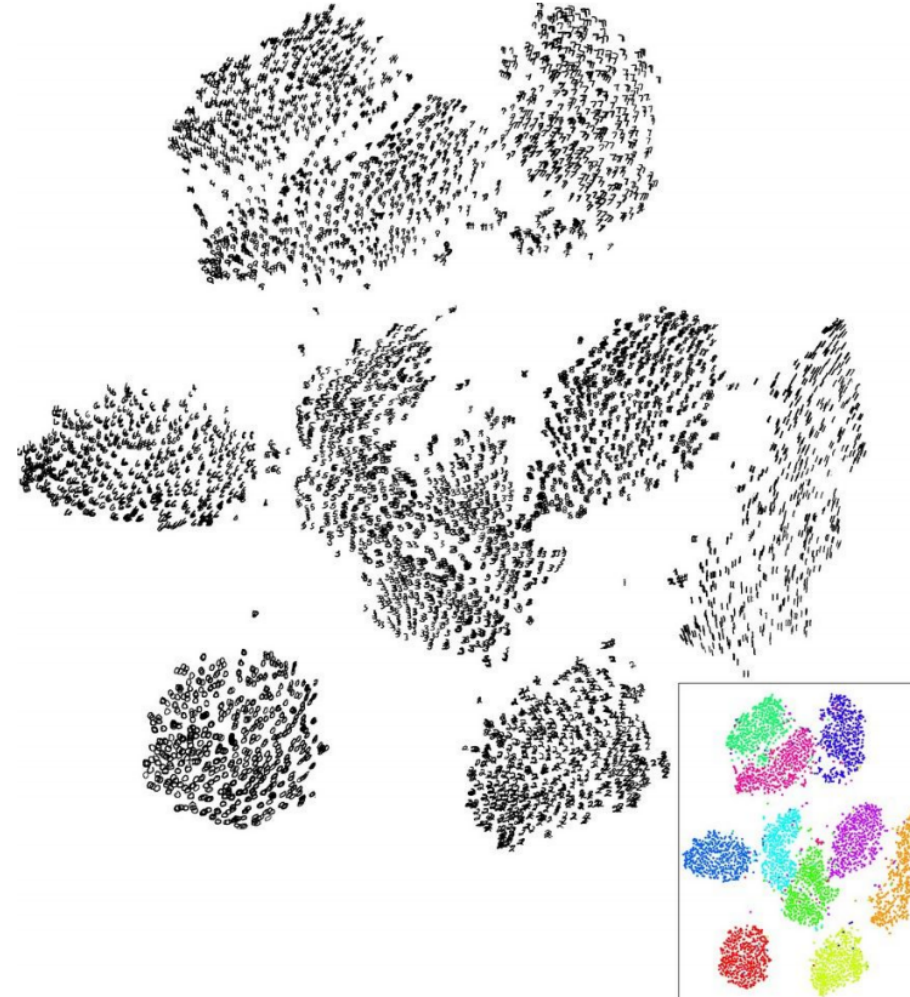
Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014
Springenberg et al, "Striving for Simplicity: The All Convolutional Net", ICLR Workshop 2015
Figure copyright Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, Martin Riedmiller, 2015; reproduced with permission.

Last Layer: Dimensionality Reduction

Visualize the “space” of FC7 feature vectors by reducing dimensionality of vectors from 4096 to 2 dimensions

Simple algorithm: Principal Component Analysis (PCA)

More complex: **t-SNE**

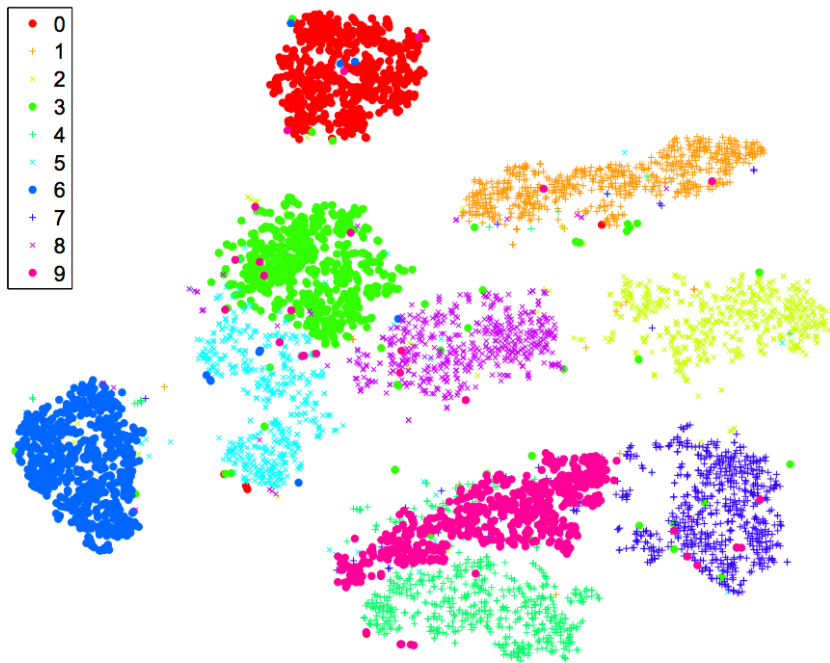


Van der Maaten and Hinton, “Visualizing Data using t-SNE”, JMLR 2008
 Figure copyright Laurens van der Maaten and Geoff Hinton, 2008. Reproduced with permission.

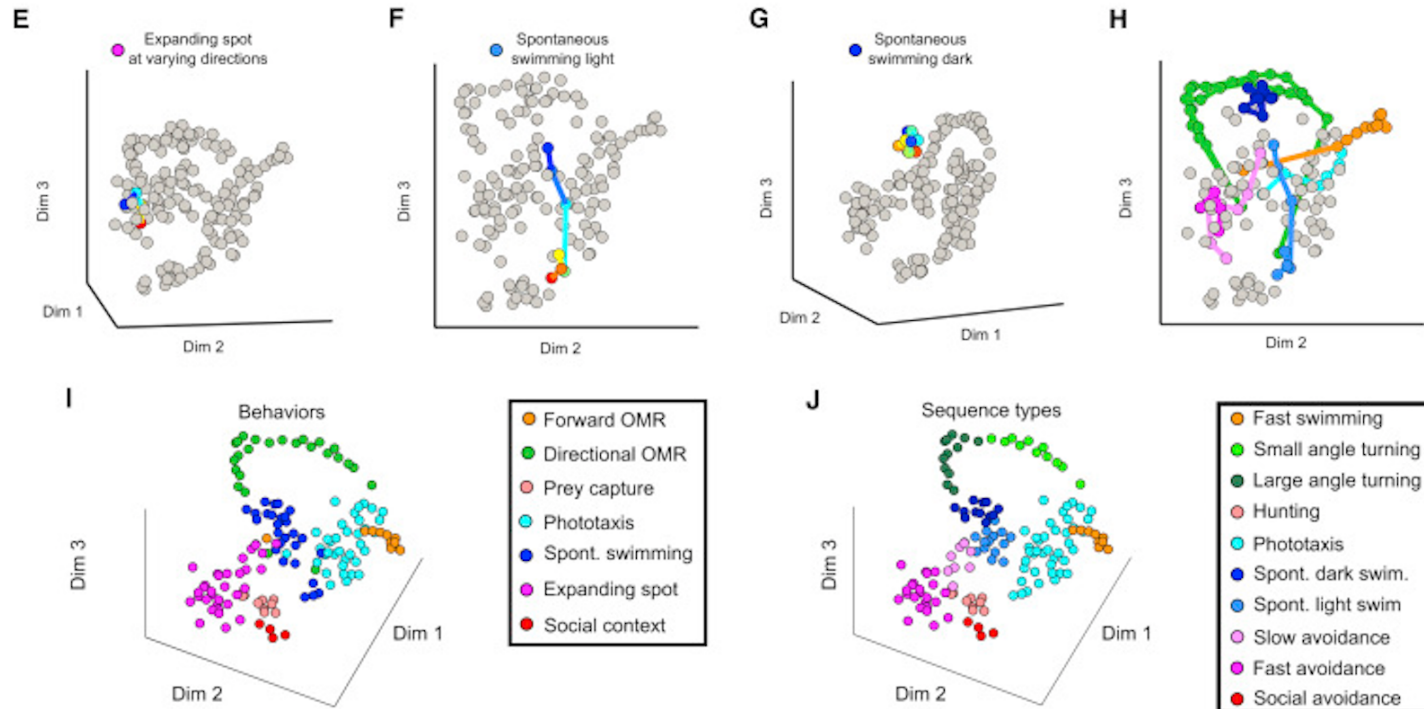
TSNE for data visualization

- Reduce data dimensions to enable visualization in 2D or 3D
 - $nD \rightarrow 2D$ or $3D$
 - Preserve local structure of data to highlight groups
 - Unsupervised – clusters unlabeled data

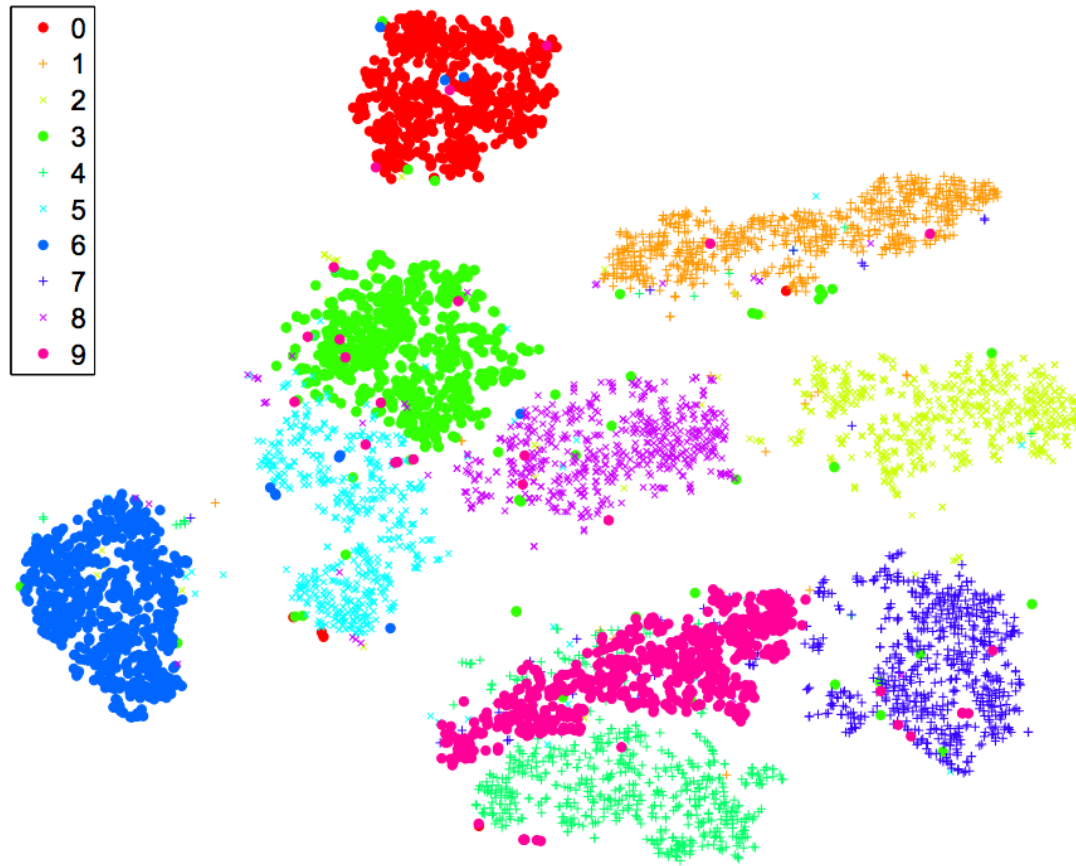
Applied to MNIST digits



Applied to movies of zebrafish behavior



Aside about clustering data – why do we need deep learning at all?

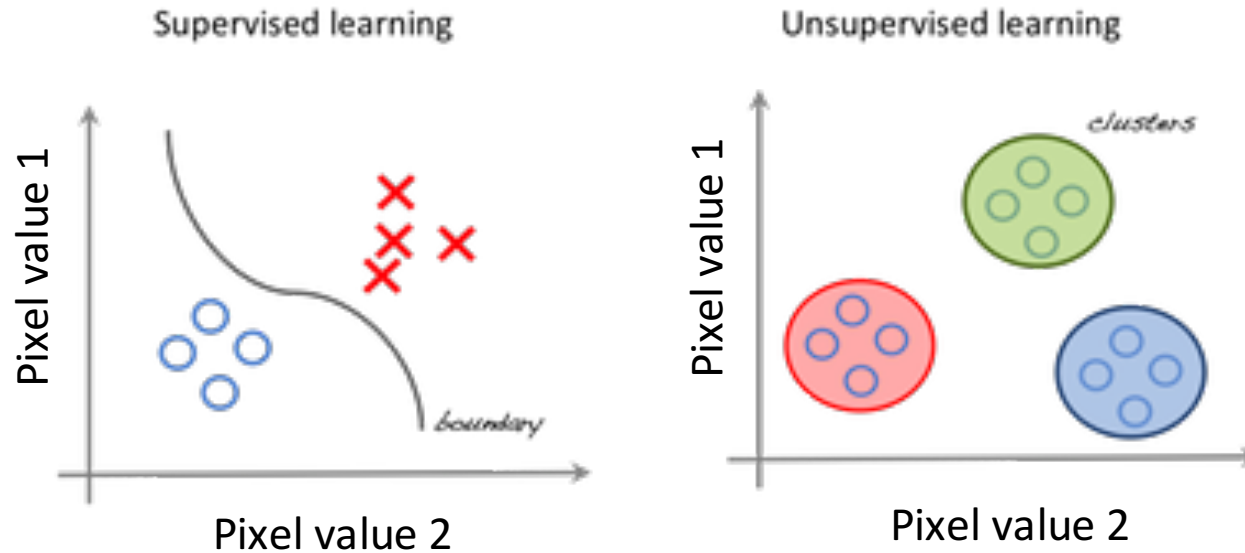


Isn't this good enough?

Unsupervised learning in a nutshell

Definition of Unsupervised Learning:

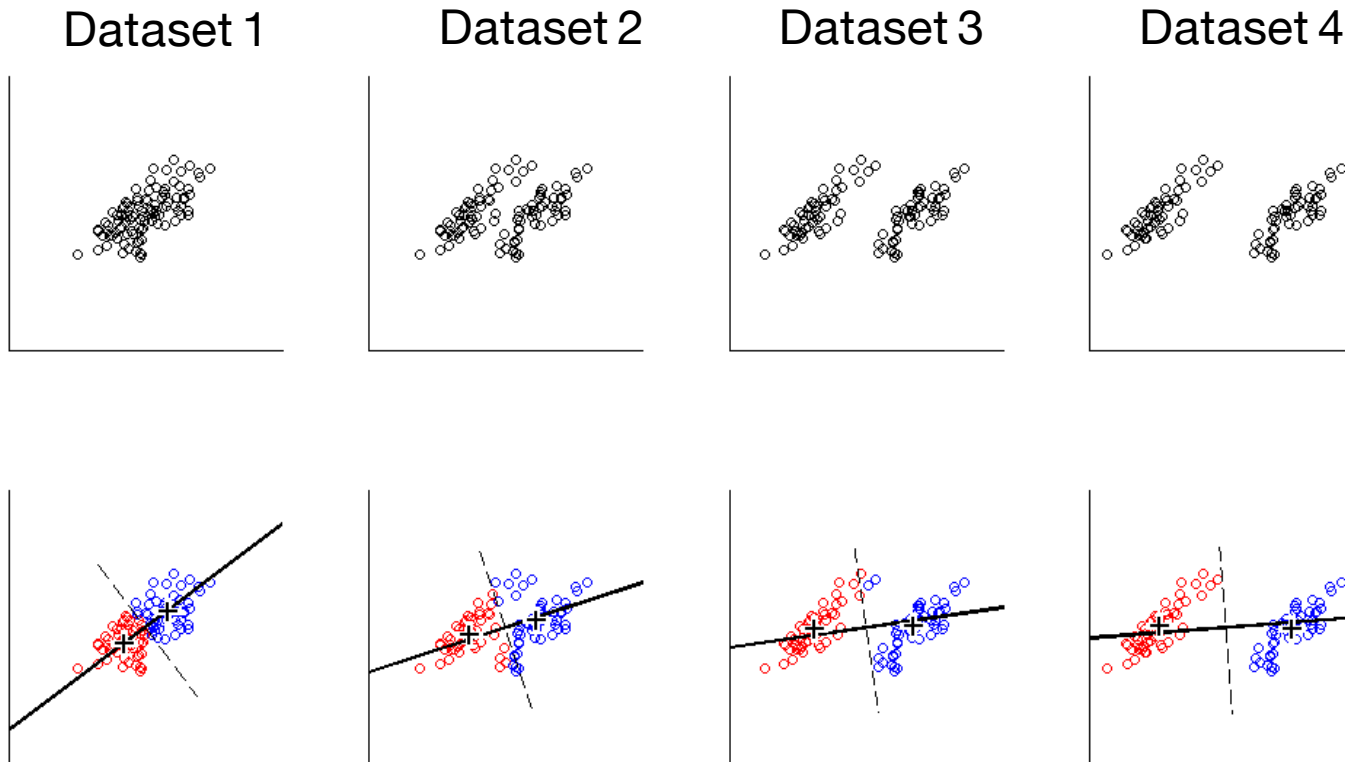
Learning useful structure *without* labeled classes, optimization criterion, feedback signal, or any other information beyond the raw data



Unsupervised learning in a nutshell

Mathematical tools for finding patterns in data:

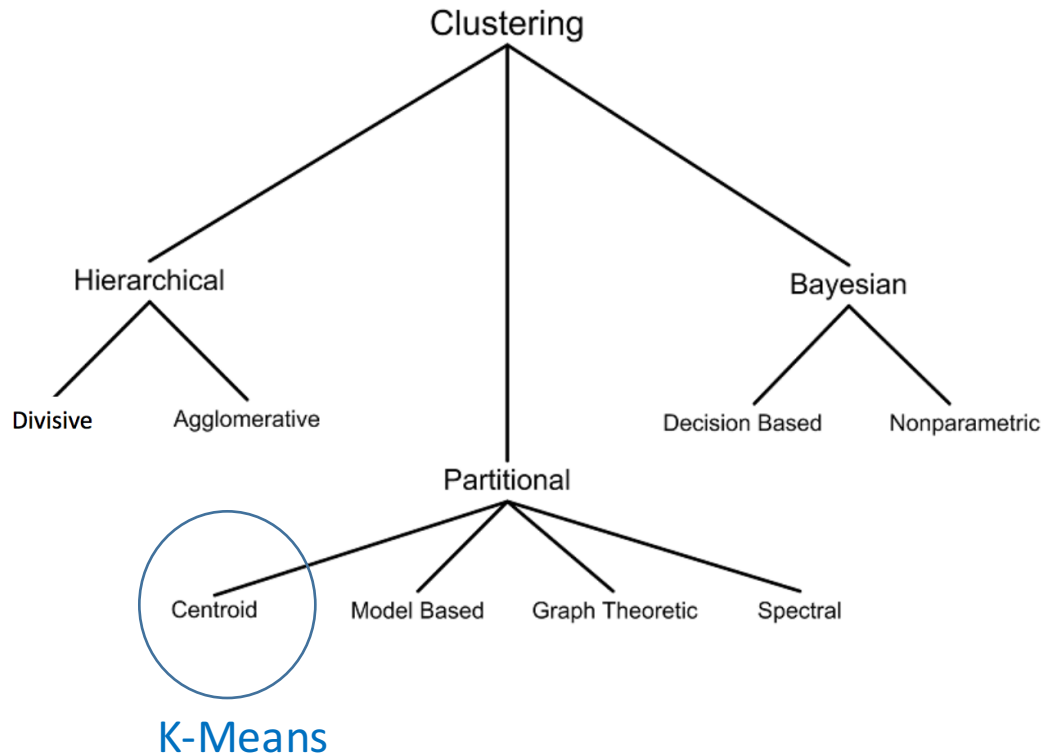
- Eigenvector decomposition
- Principal component analysis
- Singular value decomposition



<https://stats.stackexchange.com/questions/183236/wh-at-is-the-relation-between-k-means-clustering-and-pca>

Iterative methods for unsupervised learning - Clustering

Clustering techniques

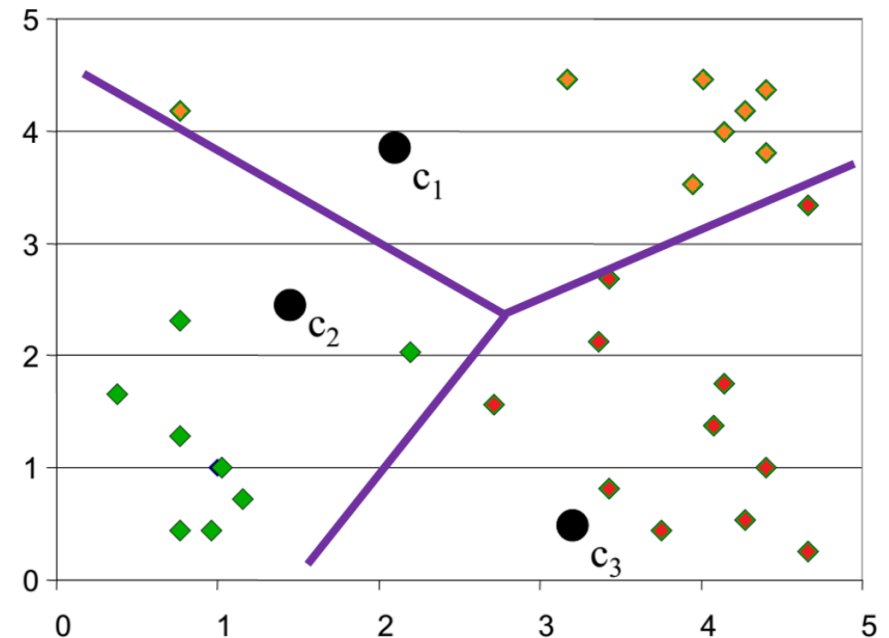


- **Hierarchical** algorithms find successive clusters using previously established clusters. These algorithms can be either **agglomerative** (“*bottom-up*”) or **divisive** (“*top-down*”):
 - 1 **Agglomerative algorithms** begin with each element as a separate cluster and merge them into successively larger clusters;
 - 2 **Divisive algorithms** begin with the whole set and proceed to divide it into successively smaller clusters.
- **Partitional** algorithms typically determine all clusters at once, but can also be used as divisive algorithms in the hierarchical clustering.
- **Bayesian** algorithms try to generate a *posteriori distribution* over the collection of all partitions of the data.

K-Means Clustering

- Given k , the k -means algorithm works as follows:
 - Choose k (random) data points (**seeds**) to be the initial **centroids**, cluster centers
 - Assign each data point to the closest **centroid**
 - Re-compute the **centroids** using the current cluster memberships
 - If a convergence criterion is not met, repeat steps 2 and 3

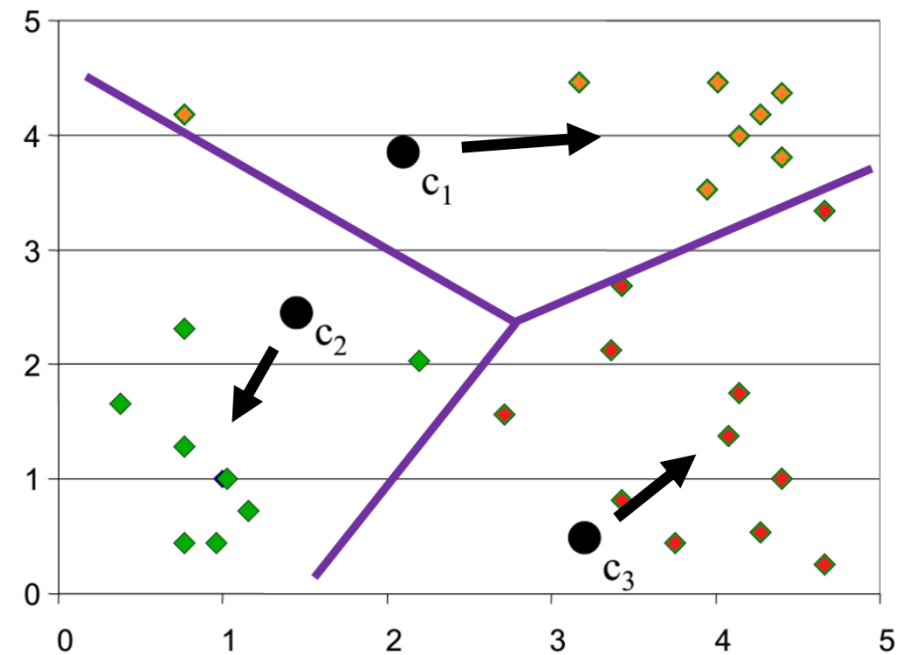
Determine cluster membership for each data point



K-Means Clustering

- Given k , the k -means algorithm works as follows:
 1. Choose k (random) data points (**seeds**) to be the initial **centroids**, cluster centers
 2. Assign each data point to the closest **centroid**
 3. Re-compute the **centroids** using the current cluster memberships
 4. If a convergence criterion is not met, repeat steps 2 and 3

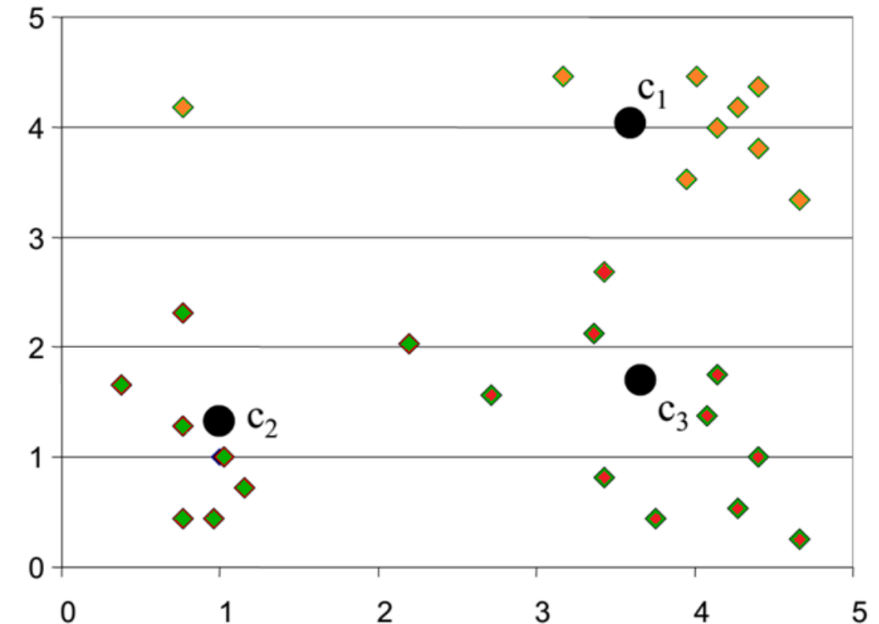
Compute and update new cluster center



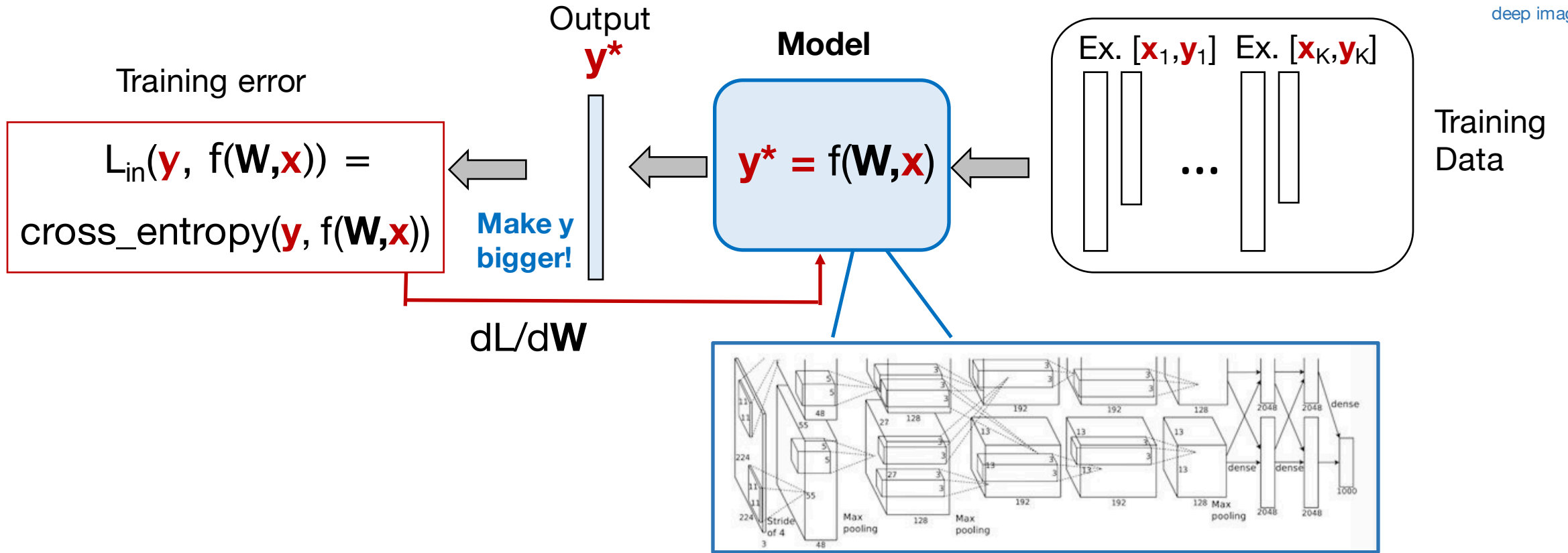
K-Means Clustering

- Given k , the k -means algorithm works as follows:
 1. Choose k (random) data points (**seeds**) to be the initial **centroids**, cluster centers
 2. Assign each data point to the closest **centroid**
 3. Re-compute the **centroids** using the current cluster memberships
 4. If a convergence criterion is not met, repeat steps 2 and 3

Result of first iteration



Next step: let's consider other automated tasks
besides image classification!



Dimensional analysis for classification:

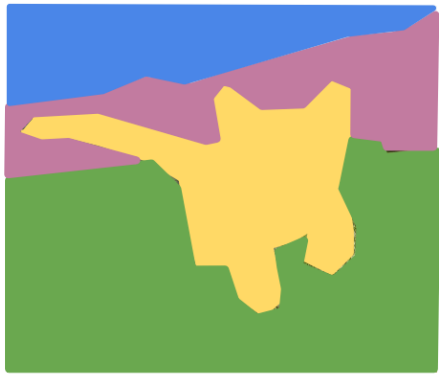
Input \mathbf{x} : $\sim R^{1000}$
 Output \mathbf{y}^* : $\sim R^2 - R^{10}$

This class – let's make \mathbf{y}^* bigger!

- Object detection
- Segmentation
- Creating 3D volumes
- Better resolution

Other Computer Vision Tasks

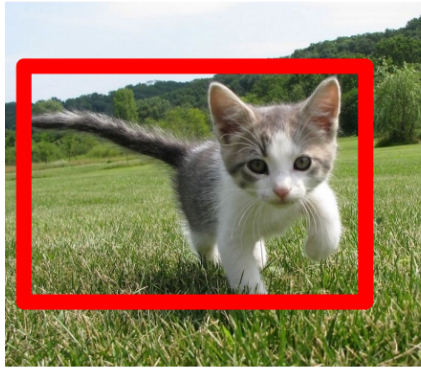
Semantic Segmentation



GRASS, CAT,
TREE, SKY

No objects, just pixels

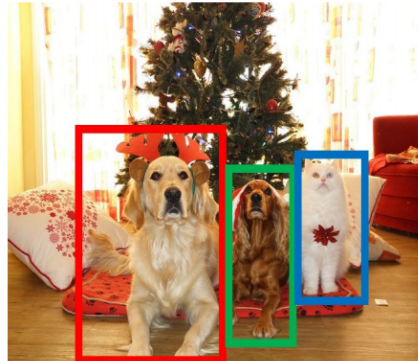
Classification + Localization



CAT

Single Object

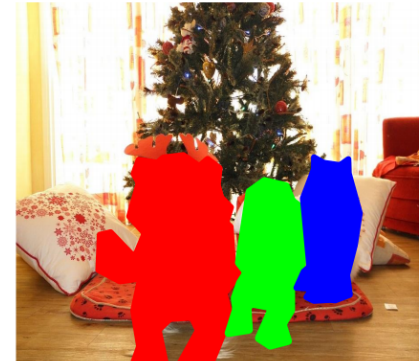
Object Detection



DOG, DOG, CAT

Multiple Object

Instance Segmentation



DOG, DOG, CAT

Super-resolution

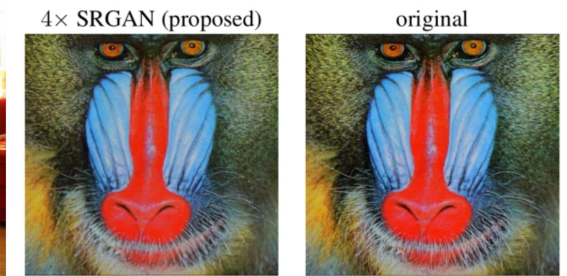
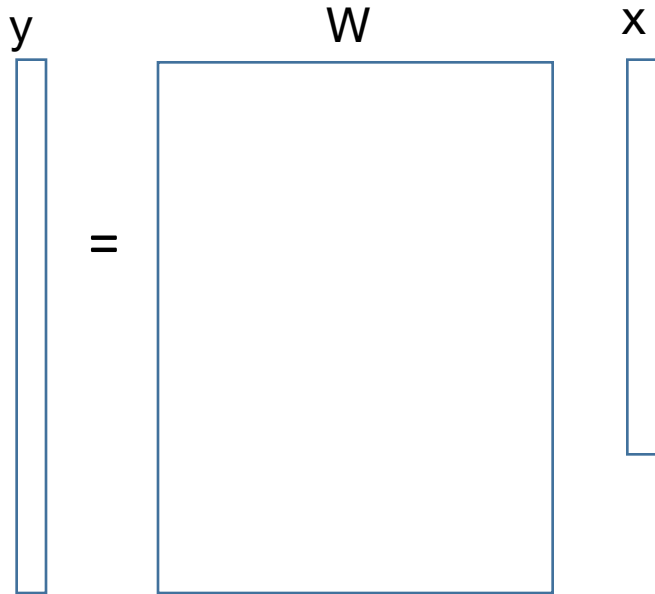


Figure 1: Super-resolved image (left) is almost indistinguishable from original (right). [4x upscaling]

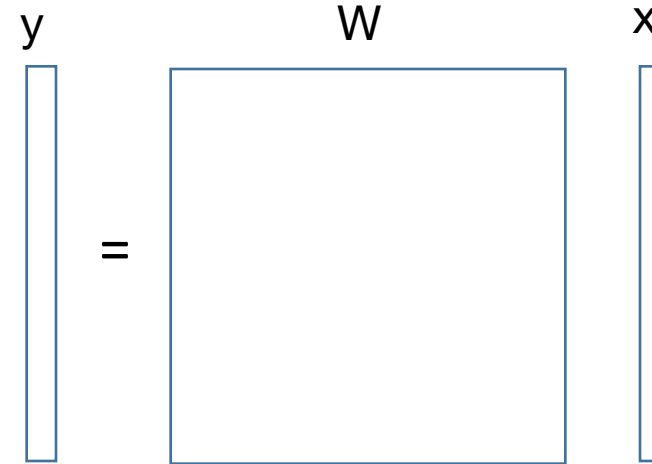
This image is CC0 public domain

Over-determined, under-determined and balanced inverse equations



Over-determined equation

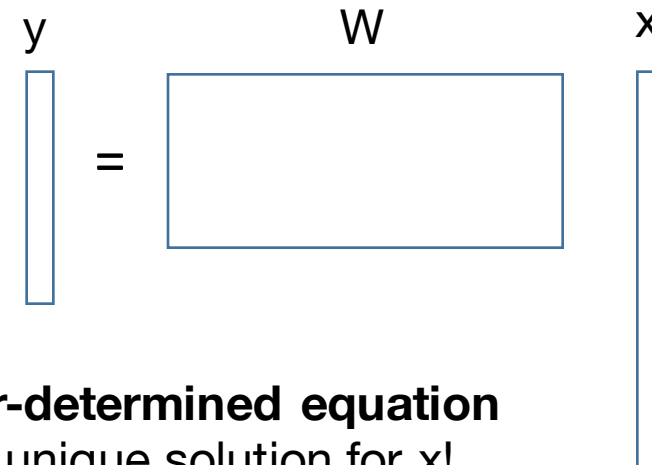
- Unique solution can exist
- If not, it's easy to get close
- Good place – more measurements than unknowns



Balanced equation

- Invertible if W is “nice”
- Hard to invert otherwise


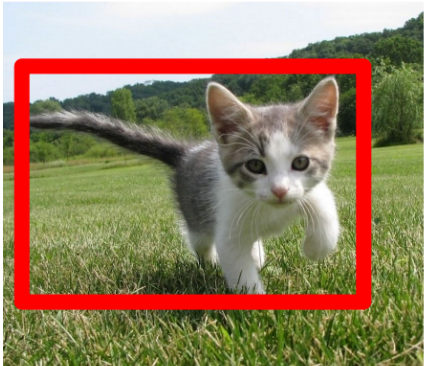
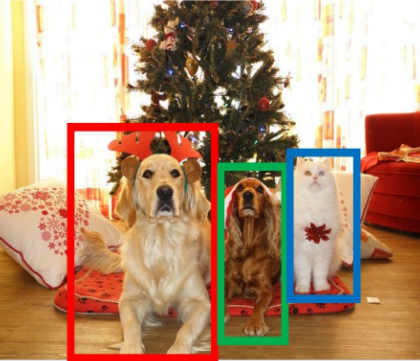

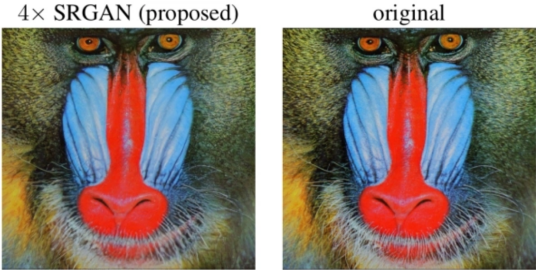
$$x = W^{-1} y$$



Under-determined equation

- No unique solution for x !
- Hard to invert
- Not a good place to be

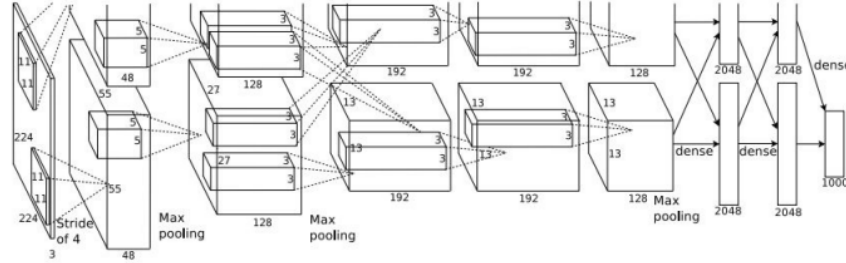
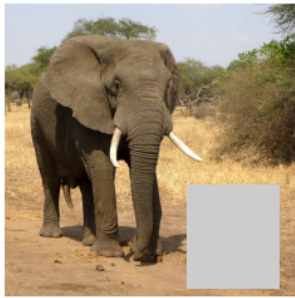
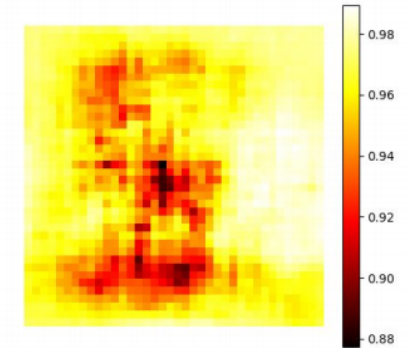
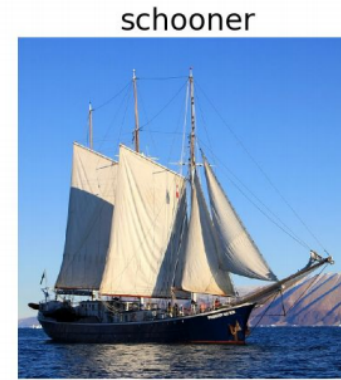
Other Computer Vision Tasks

Semantic Segmentation	Classification + Localization	Object Detection	Instance Segmentation	Super-resolution
 <p>GRASS, CAT, TREE, SKY</p> <p>No objects, just pixels</p> <p>Balanced equation</p>	 <p>CAT</p> <p>Single Object</p> <p>Over-determined</p>	 <p>DOG, DOG, CAT</p> <p>Multiple Object</p> <p>Over-determined</p>	 <p>DOG, DOG, CAT</p> <p>Multiple Object</p> <p>Over-determined</p>	 <p>4x SRGAN (proposed) original</p> <p>Figure 1: Super-resolved image (left) is almost indistinguishable from original (right). [4x upscaling]</p> <p>Under-determined</p>

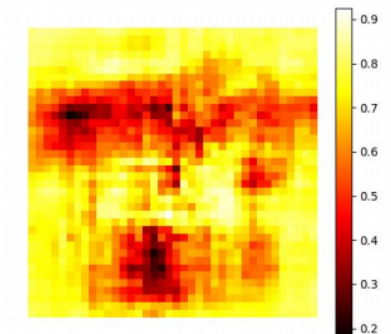
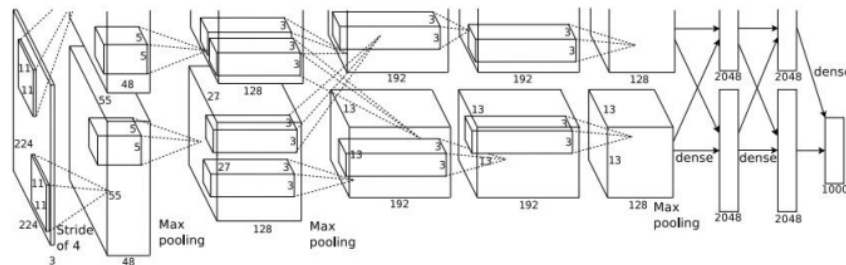
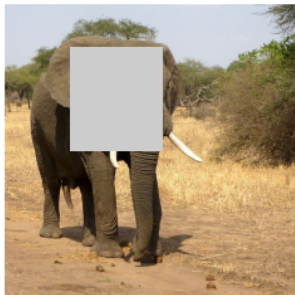
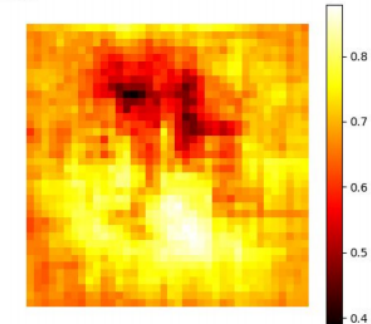
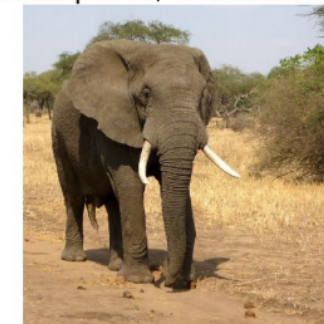
This image is CC0 public domain

Approach #1: Sliding window + occlusion map (last lecture)

Problem: Inefficient – not sharing information between different sliding window positions (even w/ lots of overlap)



African elephant, *Loxodonta africana*

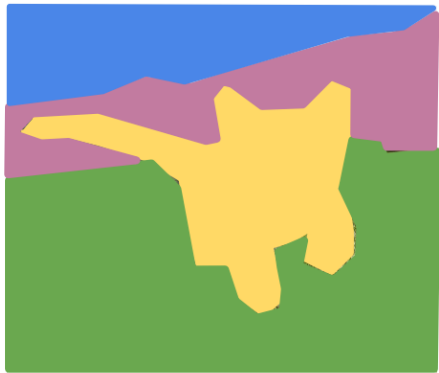


Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014

[Boat image](#) is CC0 public domain
[Elephant image](#) is CC0 public domain
[Go-Karts image](#) is CC0 public domain

Other Computer Vision Tasks

Semantic Segmentation

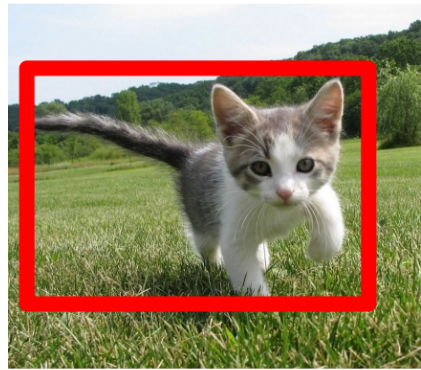


GRASS, CAT,
TREE, SKY

No objects, just pixels

Balanced equation

Classification + Localization

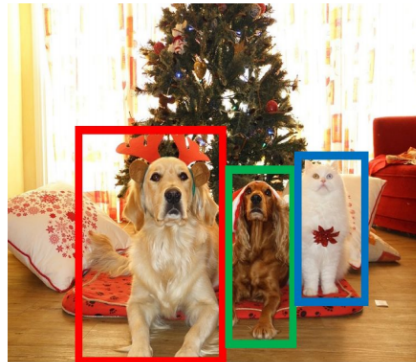


CAT

Single Object

Over-determined

Object Detection



DOG, DOG, CAT

Multiple Object

Over-determined

Instance Segmentation



DOG, DOG, CAT

Over-determined

Super-resolution

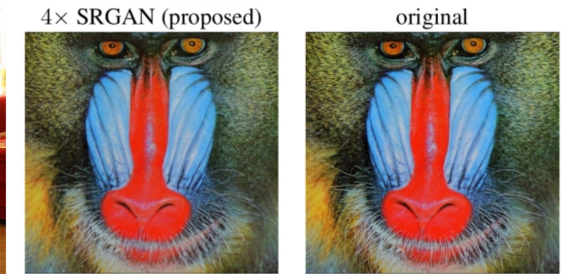


Figure 1: Super-resolved image (left) is almost indistinguishable from original (right). [4x upscaling]

Under-determined

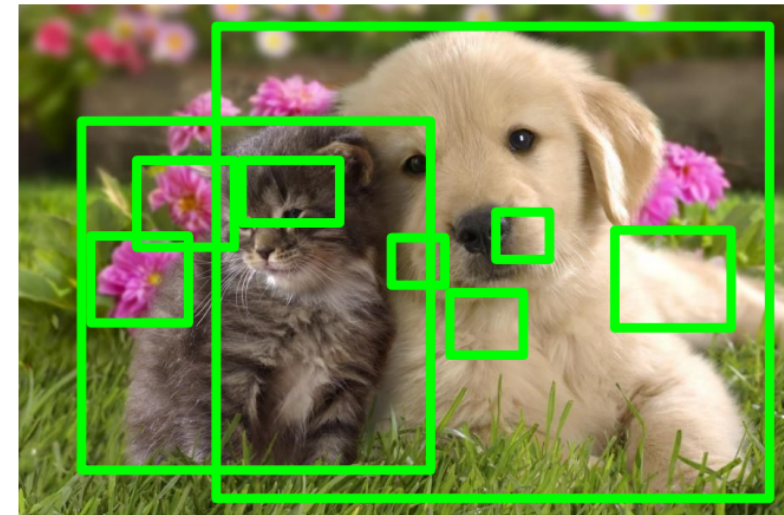
Solution: First apply a fixed ROI scheme to pull out “blobs” of interest



(Image source: van de Sande et al. ICCV'11)

Region Proposals / Selective Search

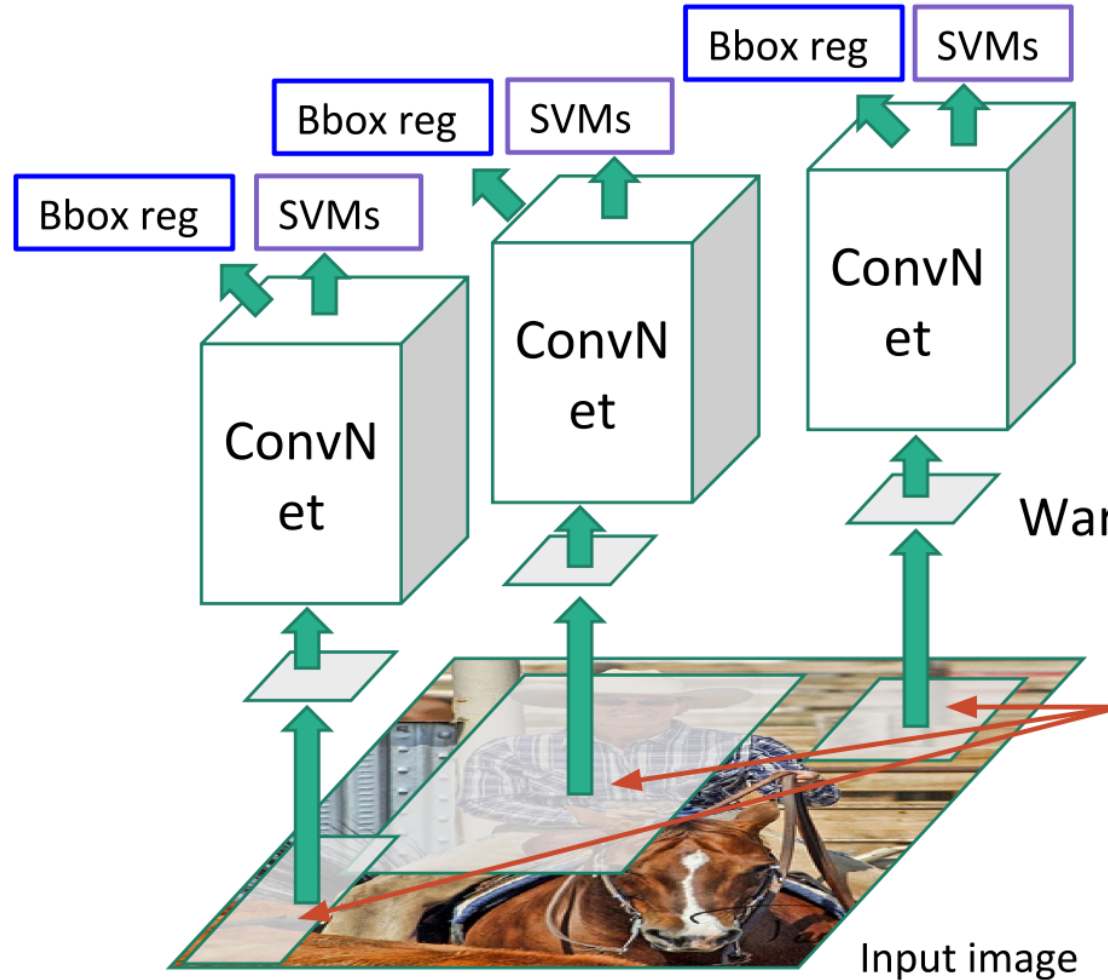
- Find “blobby” image regions that are likely to contain objects
- Relatively fast to run; e.g. Selective Search gives 2000 region proposals in a few seconds on CPU



Note: Training dataset has marked boxes, so don't necessarily need to do selective search for training, just evaluation/testing

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



Linear Regression for bounding box offsets

Classify regions with SVMs

Forward each region through ConvNet

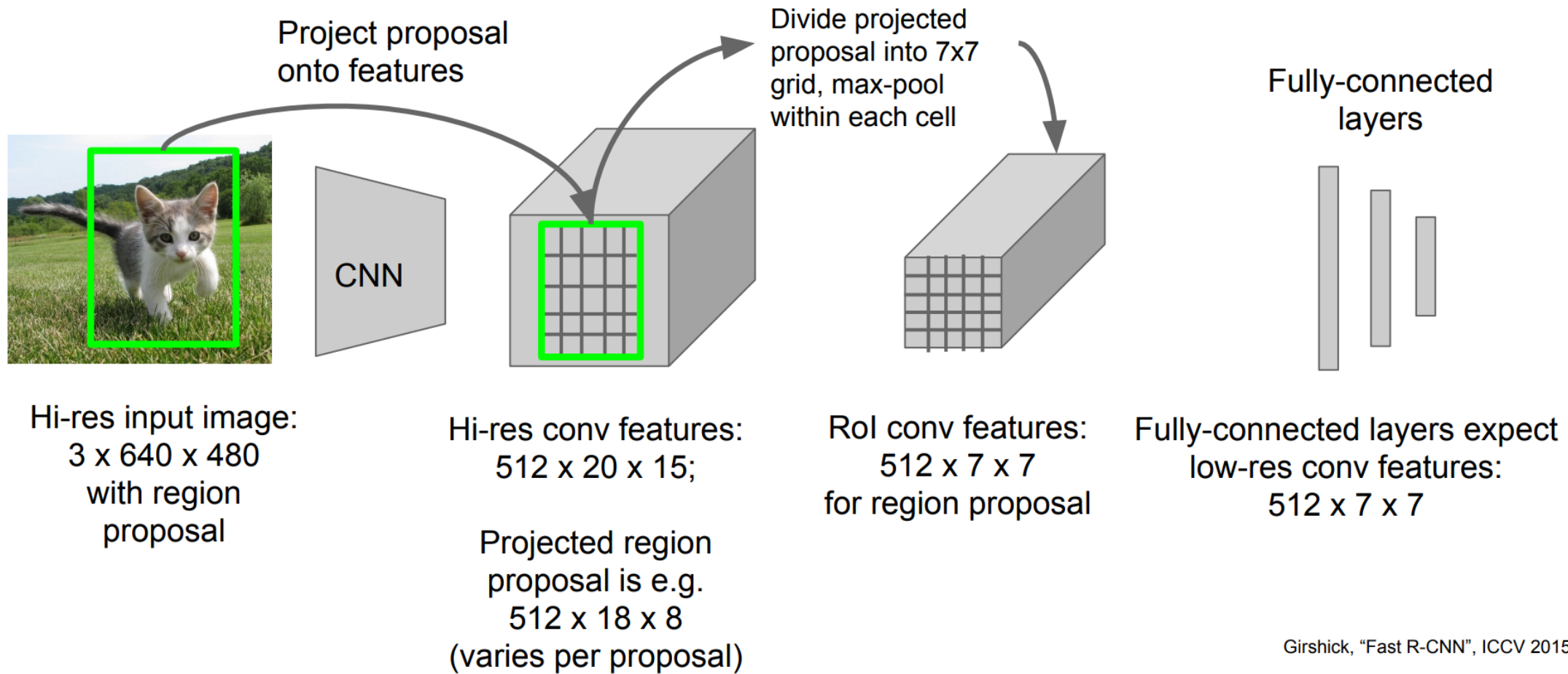
Warped image regions

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

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

Stanford CS231n - <http://cs231n.stanford.edu>

Fast R-CNN: RoI Pooling

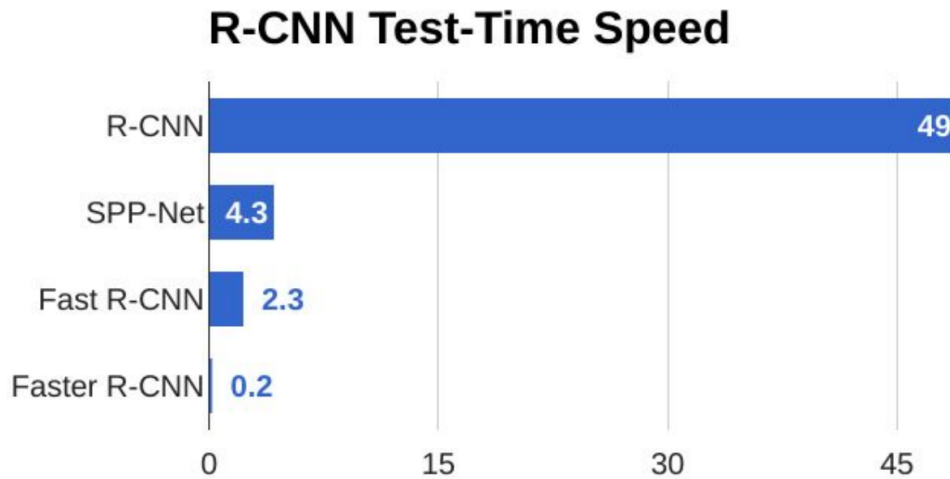


Girshick, "Fast R-CNN", ICCV 2015.

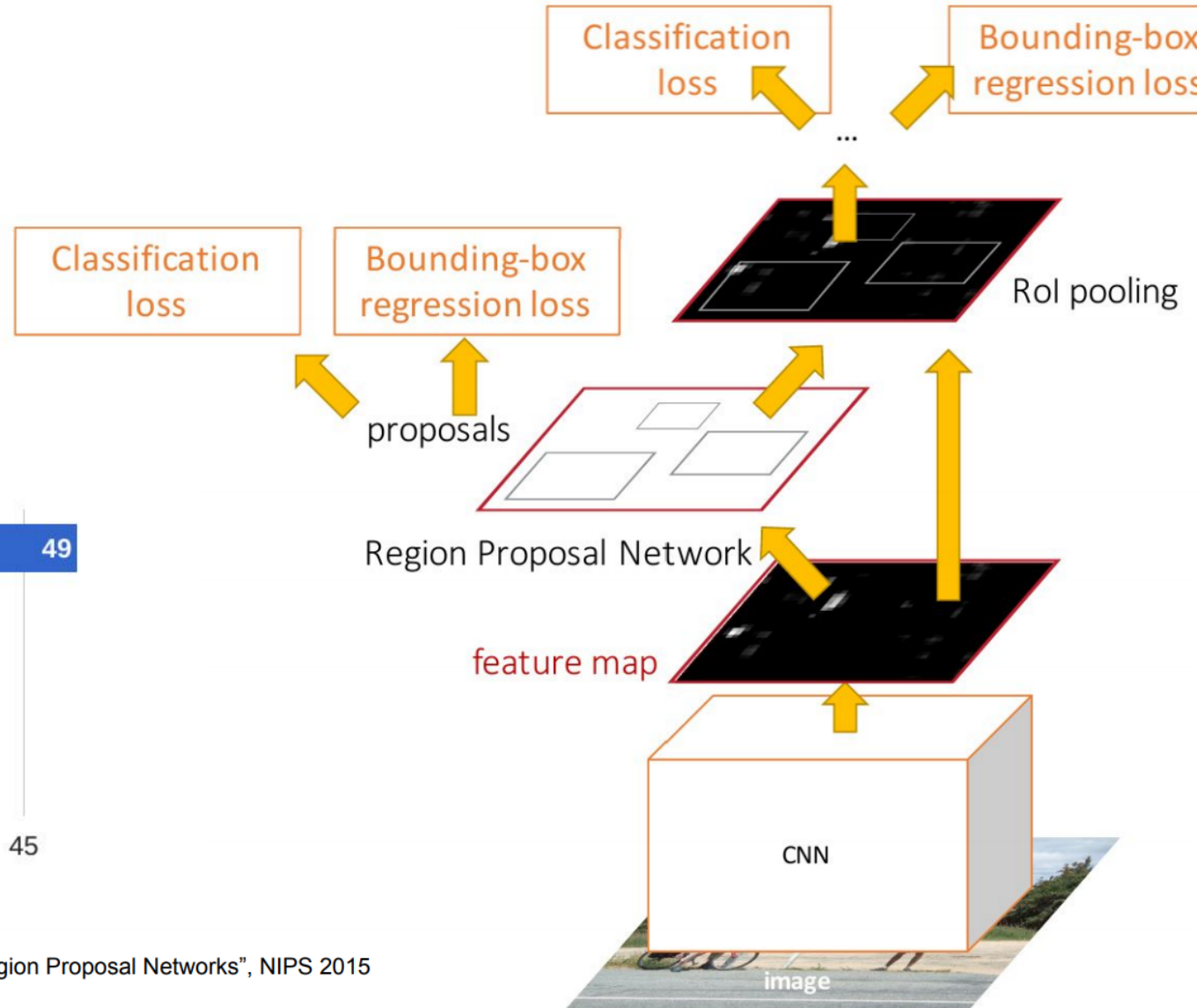
Faster R-CNN:

Make CNN do proposals!

Insert **Region Proposal Network (RPN)** to predict proposals from features



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

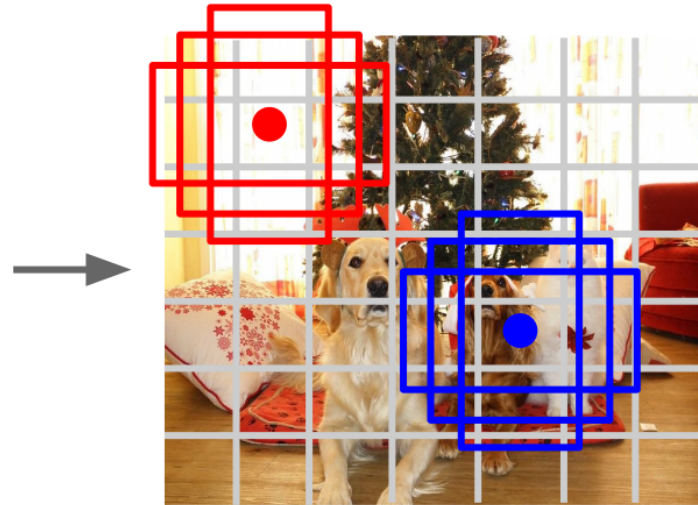


Detection without Proposals: YOLO / SSD

Go from input image to tensor of scores with one big convolutional network!



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:
 $7 \times 7 \times (5 * B + C)$

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

Object Detection: Impact of Deep Learning

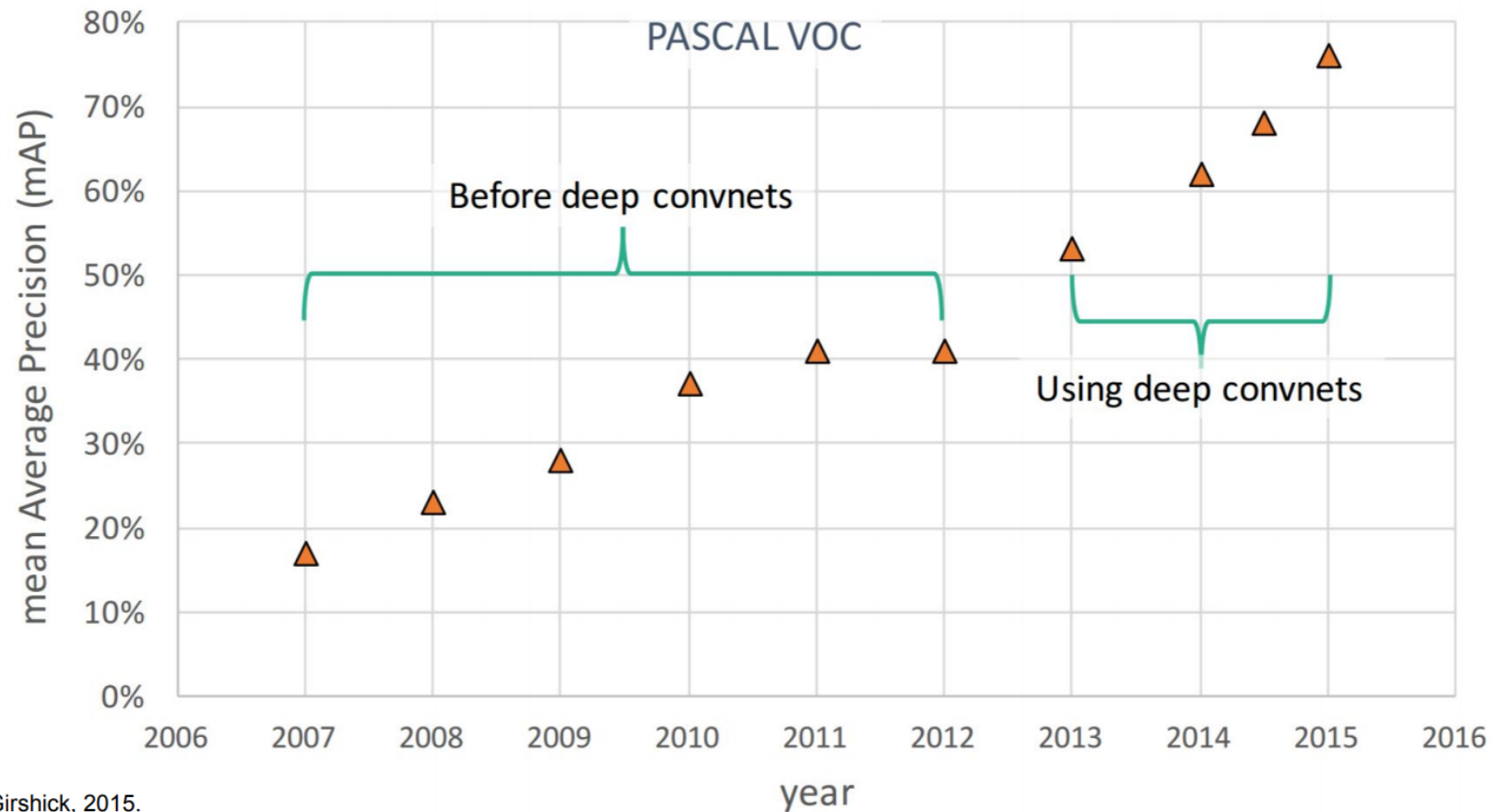


Figure copyright Ross Girshick, 2015.
Reproduced with permission.