Machine Learning in Imaging

BME 590L
Roarke Horstmeyer

Lecture 14: Beyond classification – object detection and segmentation

Class project details

- Can work alone or in a group (up to 4 people), required effort will scale with # of people
- Select a “base” dataset (online, or from a list I’ll make)
- Simulate parameters of a physical (imaging) system with base dataset
- Train deep neural net with simulated dataset
- Report results

Class project details

- Can work alone or in a group (up to 4 people), required effort will scale with # of people
- Select a "base" dataset (online, or from a list I'll make)
- Simulate parameters of a physical (imaging) system with base dataset
- Train deep neural net with simulated dataset
- Report results

What you'll need to submit:
1) The project's source code

2) A short research-style paper (3 pages minimum, 5 pages maximum) that includes an introduction, results, a discussion section, references and at least 2 figures

3) A completed web template containing the main results from the research paper

4) An 8-minute presentation that each student will deliver to the class

Example project topics:

Can we design a new lens/transducer/antenna shape to improve classification of X?

What is the tradeoff between image resolution and classification accuracy for X?

Can we determine an optimal set of colors to improve fluorophore distinguishability?

If we capture 2 images that are overlapped on one sensor, what is the best way to pre-blur them to then be able to tell them apart? Or to be able to classify them together?

If we just had a few sensors, how should be arrange them e.g. a mask to be able to predict the position of X?

Is there some optimal shift-variant blur that we can to use for a particular task?

Or, given a shift-variant PSF image, can we establish a good deconvolution using locally connected layers?

What is the optimal way to layout filters on a sensor to capture a color image for classification? Or an HDR image?

HDR image generation with filters over pixels – what is optimal design?

What if we could make a sensor with different sized pixels – how should they be laid out to achieve the best X?

Class project – what are the first steps?

1. Think about it!
2. Discuss with your friends/others in the class (feel free to use Slack!)
3. Schedule a short 15 meeting with me:
    - Friday 3/1, 3:30pm – 6:00pm
    - Next Monday 3/4, 1:00pm – 3:30pm
    - Next Wednesday 3/6, 10:00am – 1:00pm
4. Start to write-up a proposal
    - General aim: 1 paragraph with specification of physical layer
    - Discussion: (a) data source(s), (b) expected simulations, (c) expected CNN, (d) quantitative analysis of physical layer (comparison, plot, etc).

- Project proposal due date: **Thursday March 7, 2019**
- Revised project proposal due date: **Tuesday March 19, 2019**

Example project topics:

Can we design a new lens/transducer/antenna shape to improve classification of X?

What is the tradeoff between image resolution and accuracy for X (classification, segmentation, etc.)? What if we had access to n low-resolution cameras – how might we position them to get the best performance?

Can we determine an optimal set of colors to improve fluorophore distinguishability?

If we capture 2 images that are overlapped on one sensor, what is the best way to pre-blur them to then be able to tell them apart? Or to be able to classify them together?

If we just had a few sensors, how should be arrange them e.g. a mask to be able to predict the position of X?

Is there some optimal shift-variant blur that we can to use for a particular task?

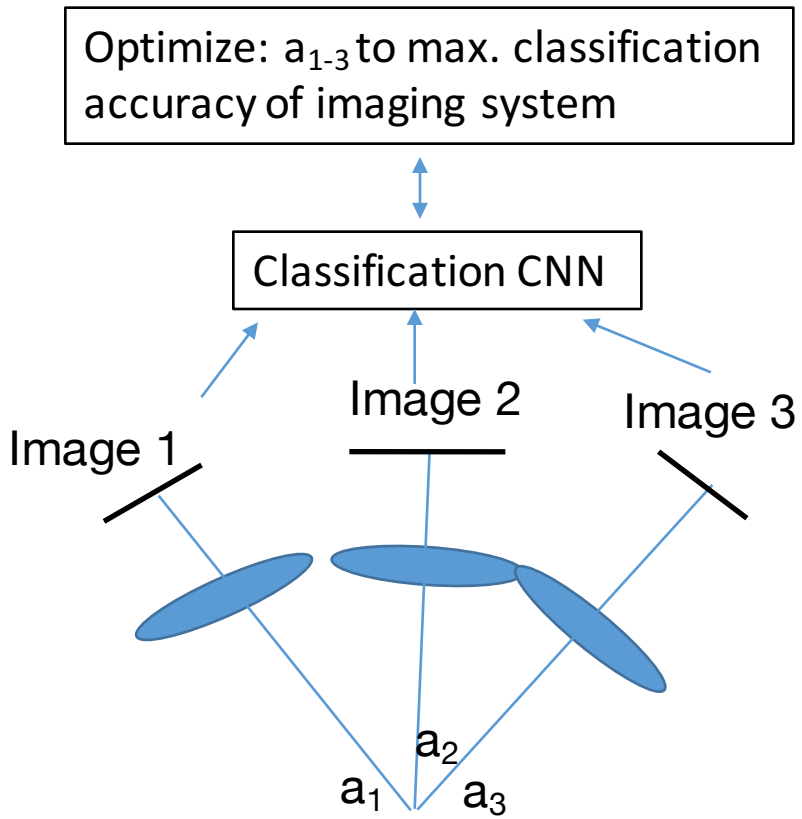
Or, given a shift-variant PSF image, can we establish a good deconvolution using locally connected layers?

What is the optimal way to layout filters on a sensor to capture a color image for classification? Or an HDR image?

HDR image generation with filters over pixels – what is optimal design?

What if we could make a sensor with different sized pixels – how should they be laid out to achieve the best X?

What is the tradeoff between image resolution and accuracy for image segmentation? What if we had access to n low-resolution cameras – how might we position them to get the best performance?



I propose to simulate the classification performance of a new type of microscope, which will have 3 different lenses and sensors. Each lens and sensor will capture an image of a flat object from a unique angular perspective, and the image classification will be performed with all of the data. The physical parameter that I will optimize is the angle of tilt of each lens with respect to the object to maximize classification accuracy.

Dataset: 12,500 images of 4 types of blood cell
https://www.kaggle.com/paultimothymooney/blood-cells

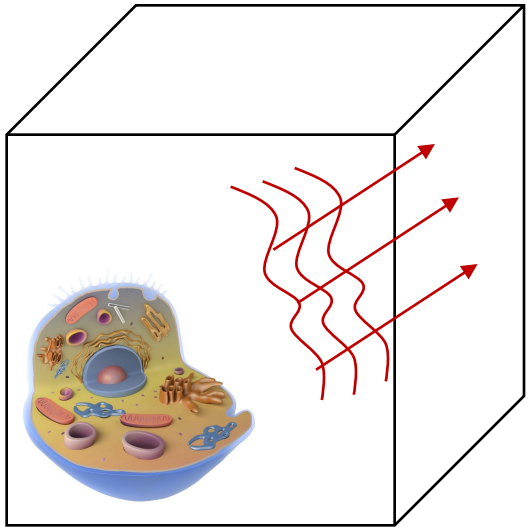
Simulation: Treat each image as a thin 2D object and is coherently illuminated. Assume each camera captures a unique component of the object spectrum, which will vary as a function of a_{1-3}. Start by neglecting size and shape of each camera.

CNN: Digital layer: Alexnet. Physical layer: simulate object spectrum, sample object spectrum re-centered by angle a_{1-3} (which are weight variables), form images and classify them together

Quantitative analysis: I will plot classification performance as a function of the number of allowed cameras for a fixed CNN architecture, and will also compare the classification performance to the case of a single image

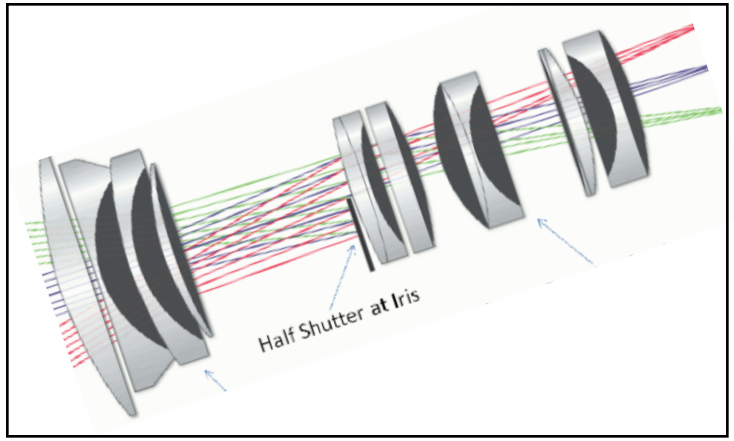
ML+Imaging pipeline + plan

Real World



Continuous complex fields

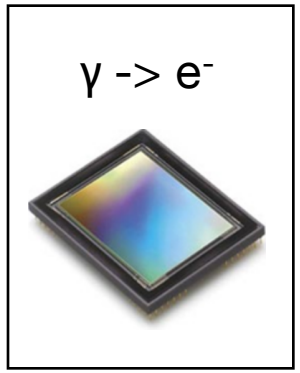
Measurement device



Black box transformations

- Convolution
- Fourier Transform

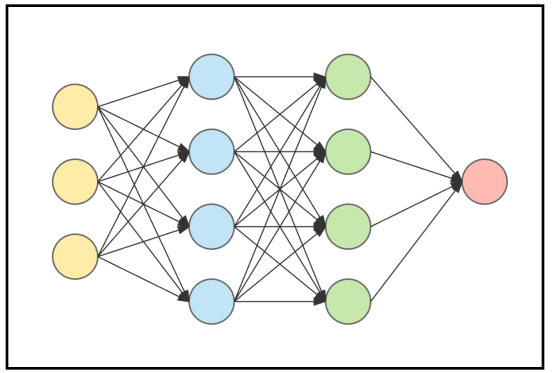
Digitization



Sampling Theorem

Discrete math & Linear algebra

Machine Learning



Optimization

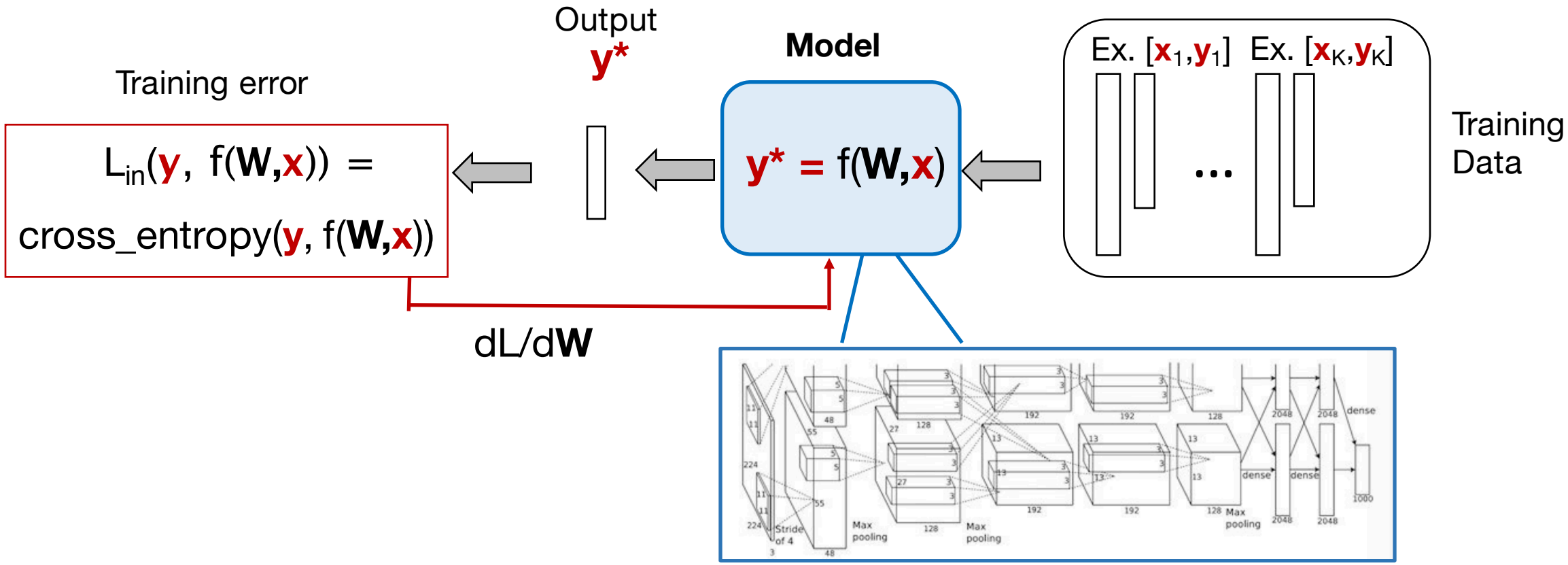
Linear classification

Logistic classifier

Neural networks

Convolutional NN's

March → April



Dimensional analysis for classification:

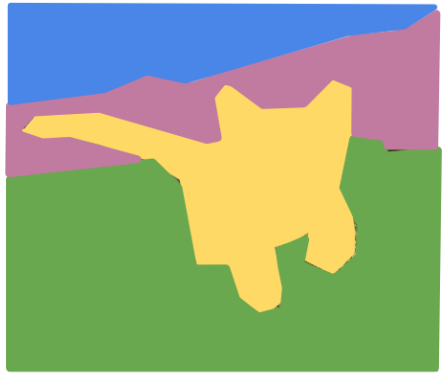
Input **x**: ~R^{1000}
Output **y***: ~R^2 – R^{10}

This class – let's make **y*** bigger!

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

Other Computer Vision Tasks

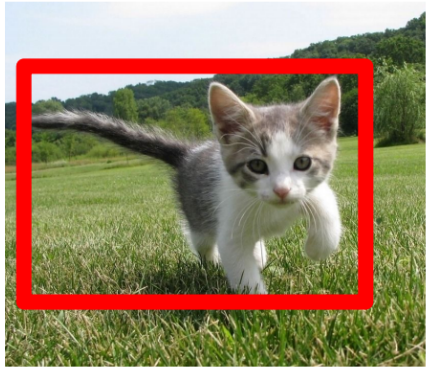
Semantic Segmentation



GRASS, CAT, TREE, SKY

No objects, just pixels

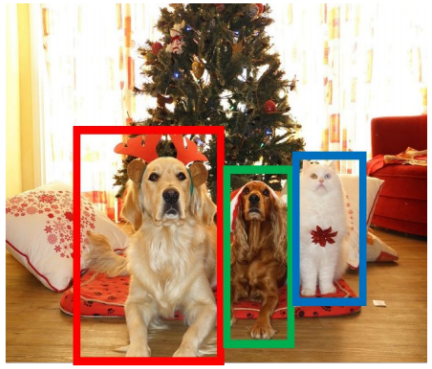
Classification + Localization



CAT

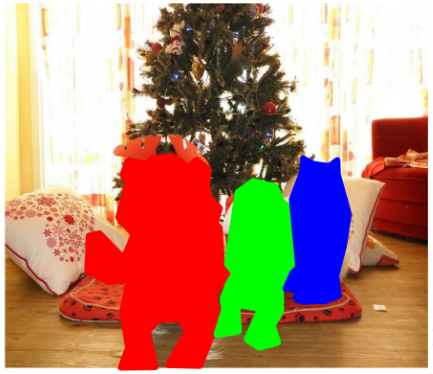
Single Object

Object Detection



DOG, DOG, CAT

Instance Segmentation



DOG, DOG, CAT

Multiple Object

Super-resolution

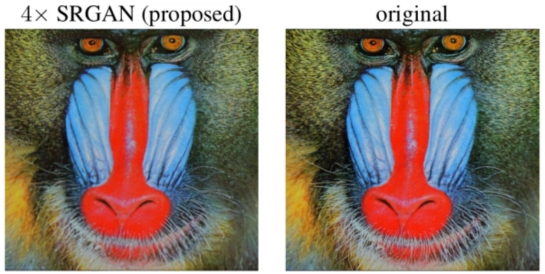
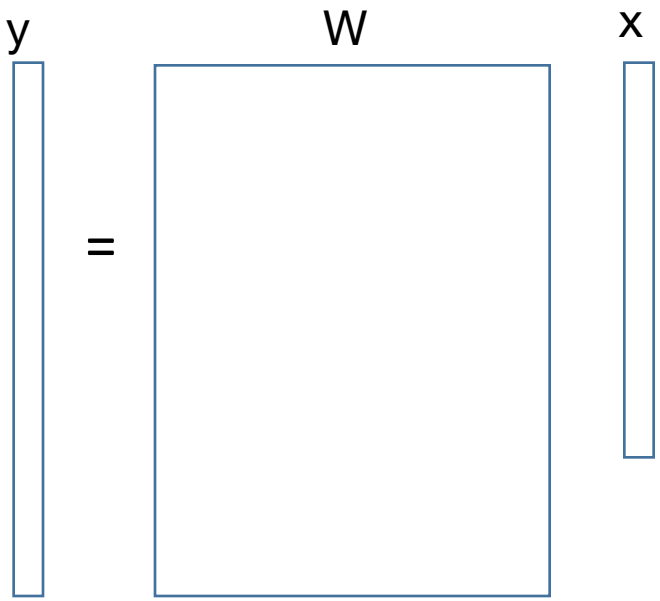
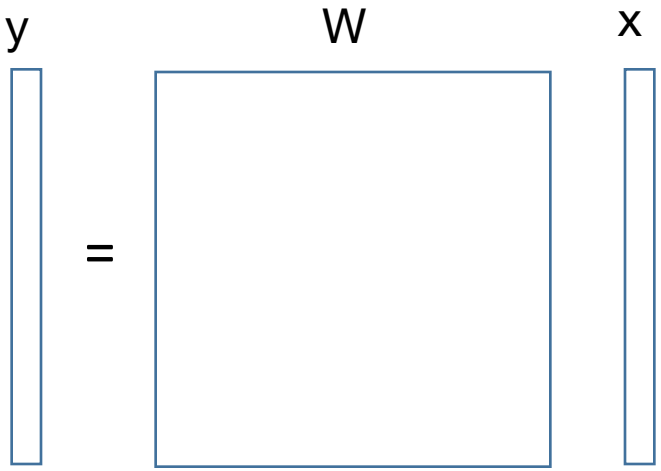


Figure 1: Super-resolved image (left) is almost indistinguishable from original (right). [4× upscaling]

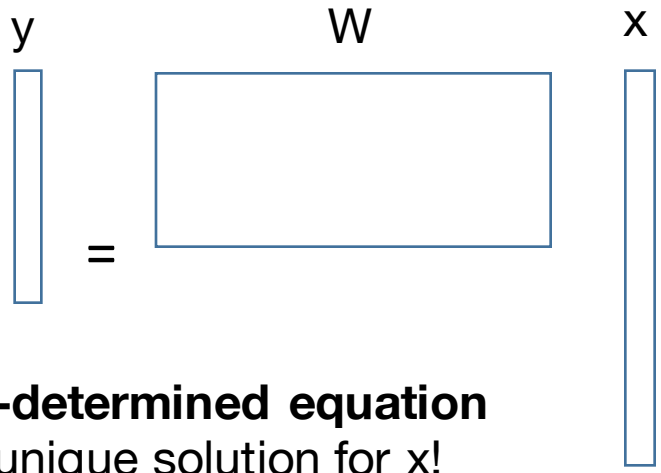
Over-determined, under-determined and balanced equations



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



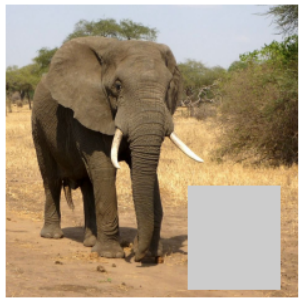
Balanced equation
- Invertible if A is nice
- Hard to invert

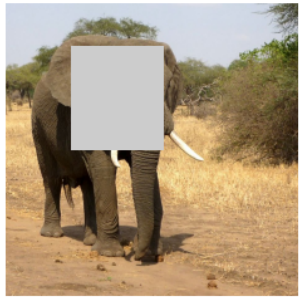


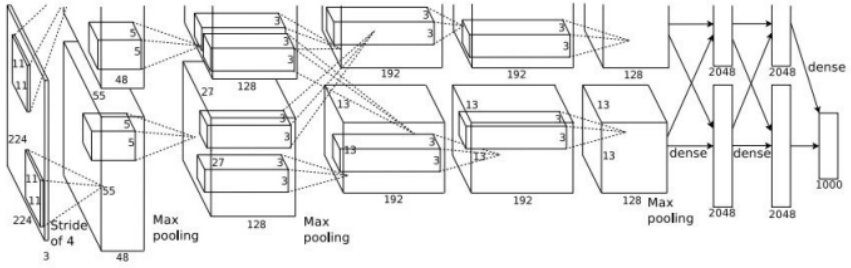
Under-determined equation
- No unique solution for x!
- Hard to invert
- Not a good place to be

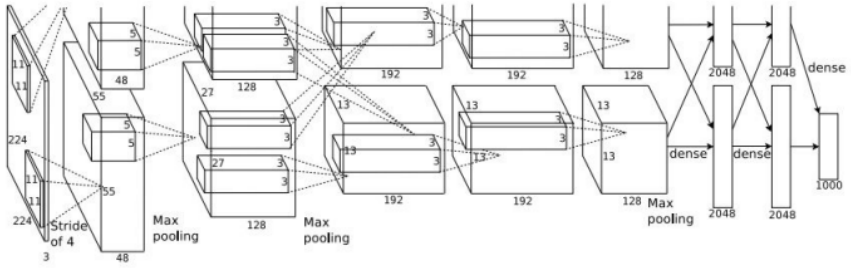
Approach #1: Sliding window + occlusion map (last lecture)

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

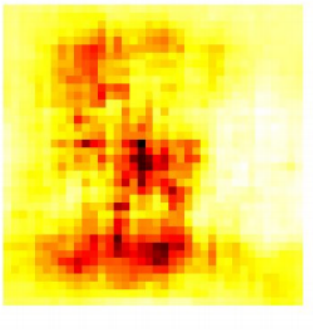




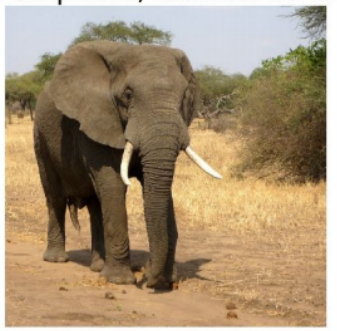


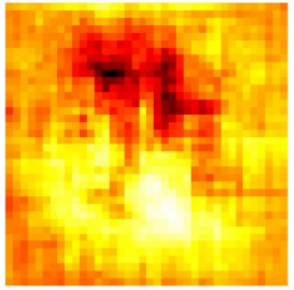


schooner

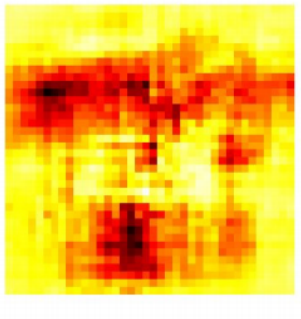


African elephant, Loxodonta africana





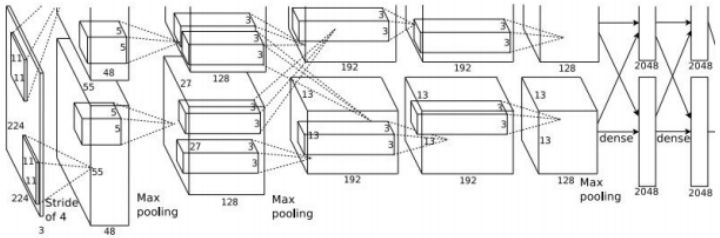
go-kart



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

Classification + Localization

This image is CC0 public domain



Vector: 4096

Fully Connected: 4096 to 1000

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

Softmax Loss

Correct label: Cat

Fully Connected: 4096 to 4

Box Coordinates (x, y, w, h)

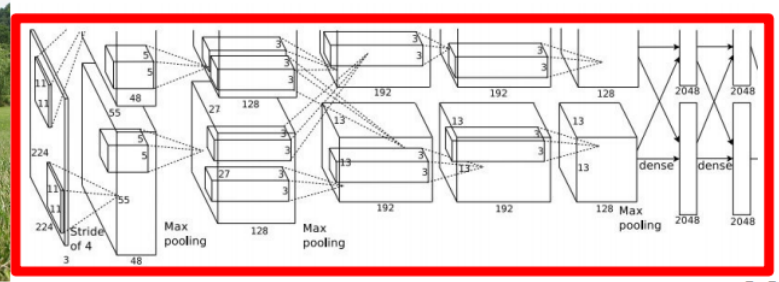
L2 Loss

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

Treat localization as a regression problem!

Classification + Localization

This image is CC0 public domain



Often pretrained on ImageNet
(Transfer learning)

Vector: 4096

Fully Connected: 4096 to 1000

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

Softmax Loss

Correct label: Cat

Fully Connected: 4096 to 4

Box Coordinates (x, y, w, h)

L2 Loss

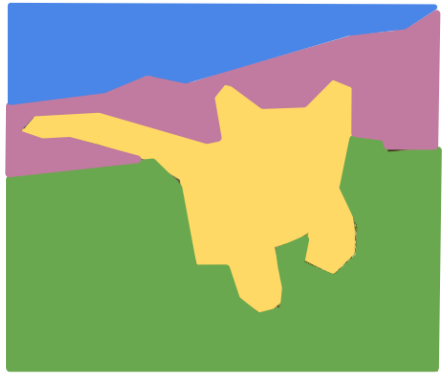
Correct box: (x', y', w', h')

Loss

Treat localization as a regression problem!

Other Computer Vision Tasks

Semantic Segmentation

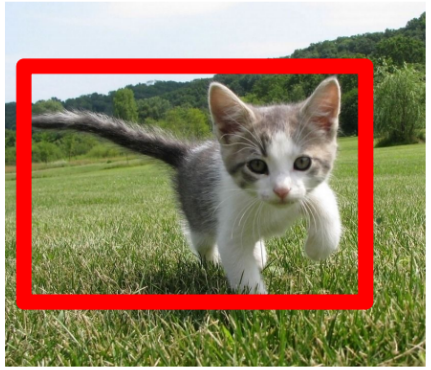


GRASS, CAT, TREE, SKY

No objects, just pixels

Balanced equation

Classification + Localization

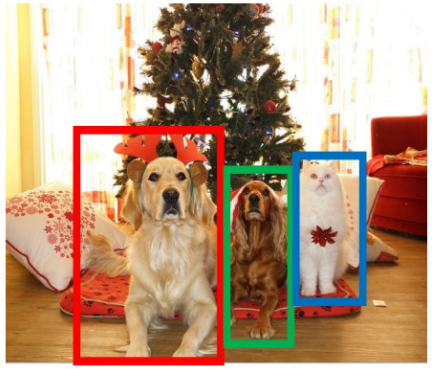


CAT

Single Object

Over-determined

Object Detection

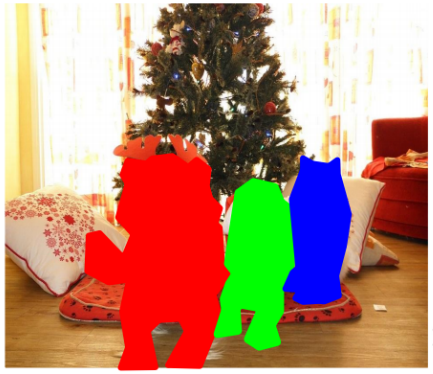


DOG, DOG, CAT

Multiple Object

Over-determined

Instance Segmentation



DOG, DOG, CAT

Over-determined

Super-resolution

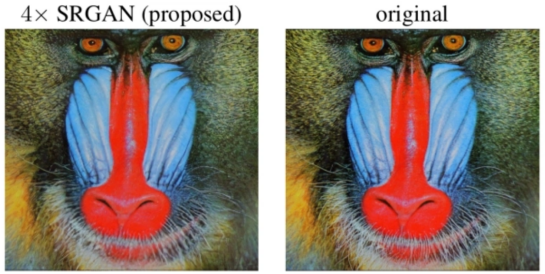
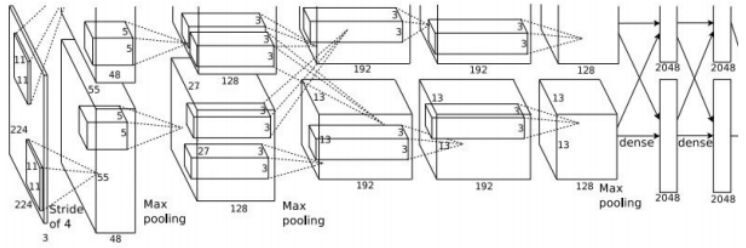


Figure 1: Super-resolved image (left) is almost indistinguishable from original (right). [4× upscaling]

Under-determined

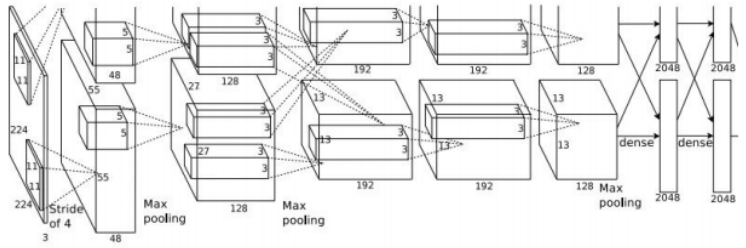
Object Detection as Regression?

Each image needs a different number of outputs!



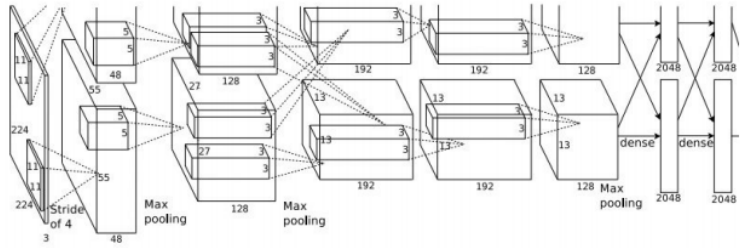
CAT: (x, y, w, h)

4 numbers



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

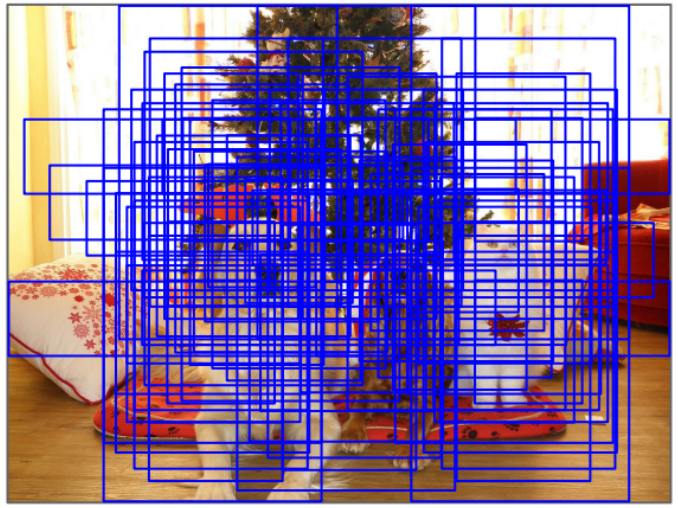
16 numbers



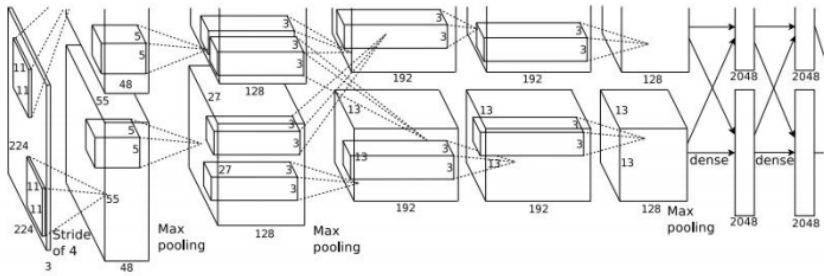
DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
….

Many numbers!

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

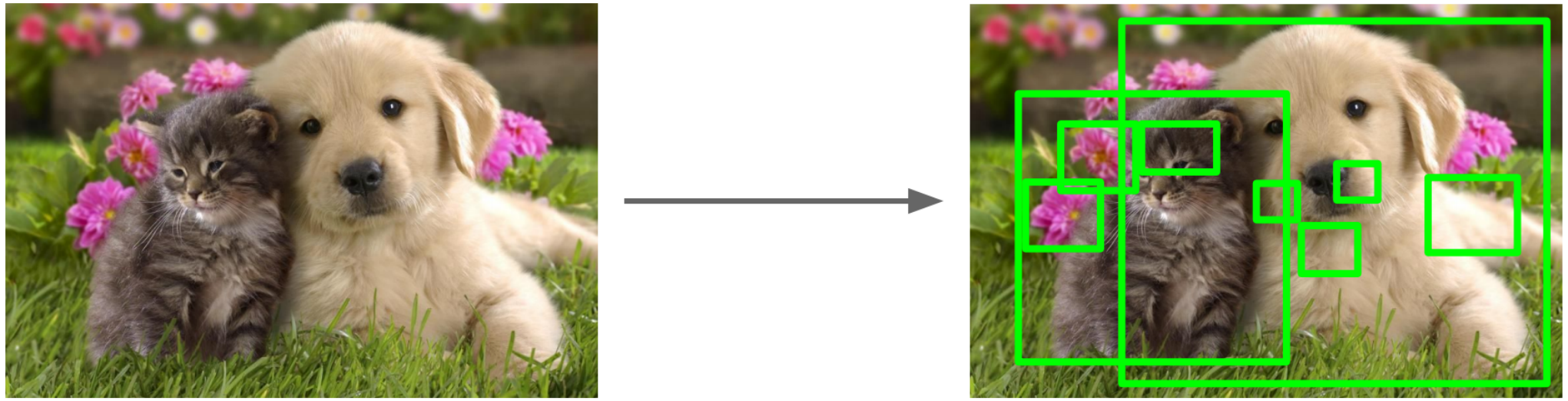
Problem: Need to apply CNN to huge number of locations, scales, and aspect ratios, very computationally expensive!

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

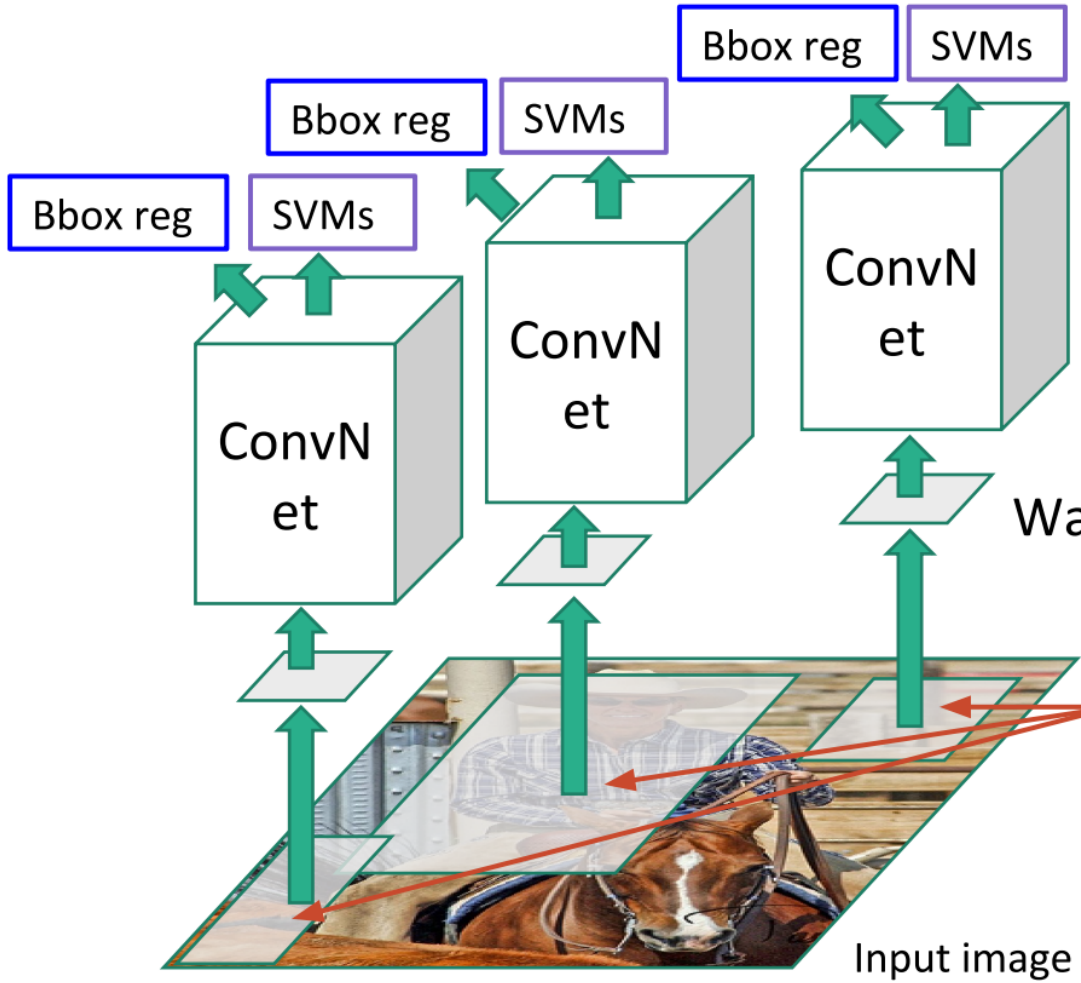
- Find “blobby” image regions that are likely to contain objects
- Relatively fast to run; e.g. Selective Search gives 2000 region proposals in a few seconds on CPU



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

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

R-CNN



Linear Regression for bounding box offsets

Classify regions with SVMs

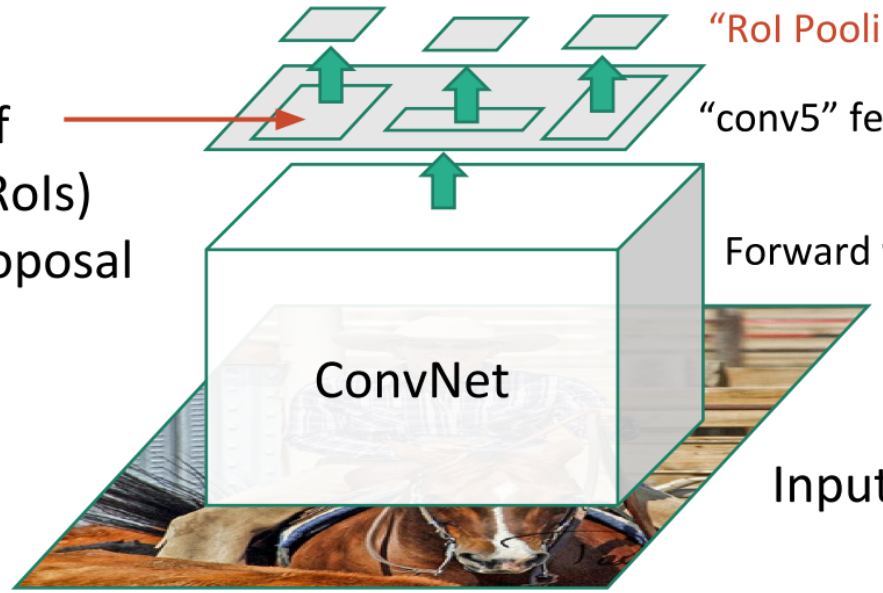
Forward each region through ConvNet

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.

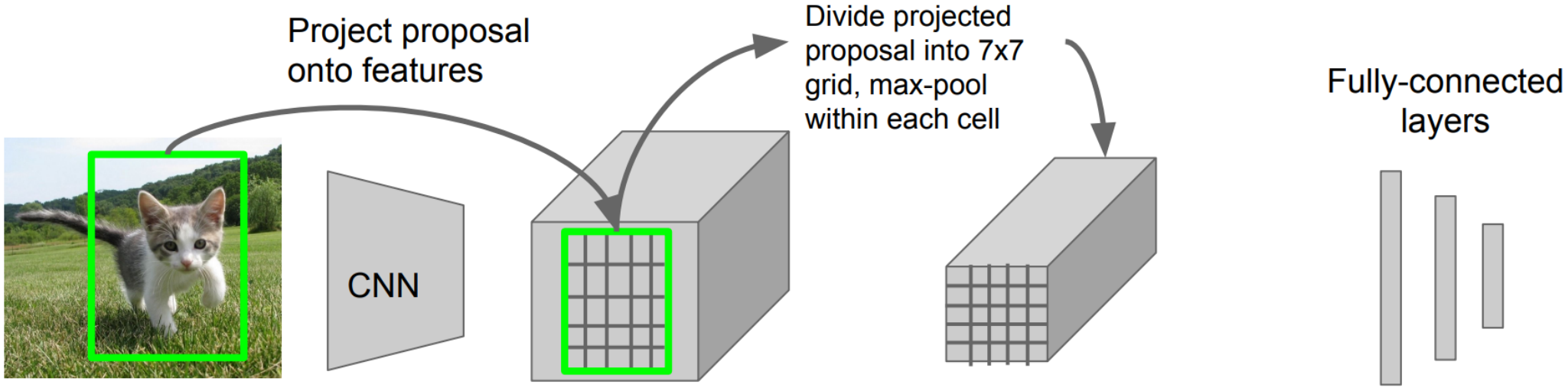
Fast R-CNN



(Main features first, instead of repeating 2000X)

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

Fast R-CNN: RoI Pooling



Hi-res input image: 3 x 640 x 480 with region proposal

Hi-res conv features: 512 x 20 x 15;

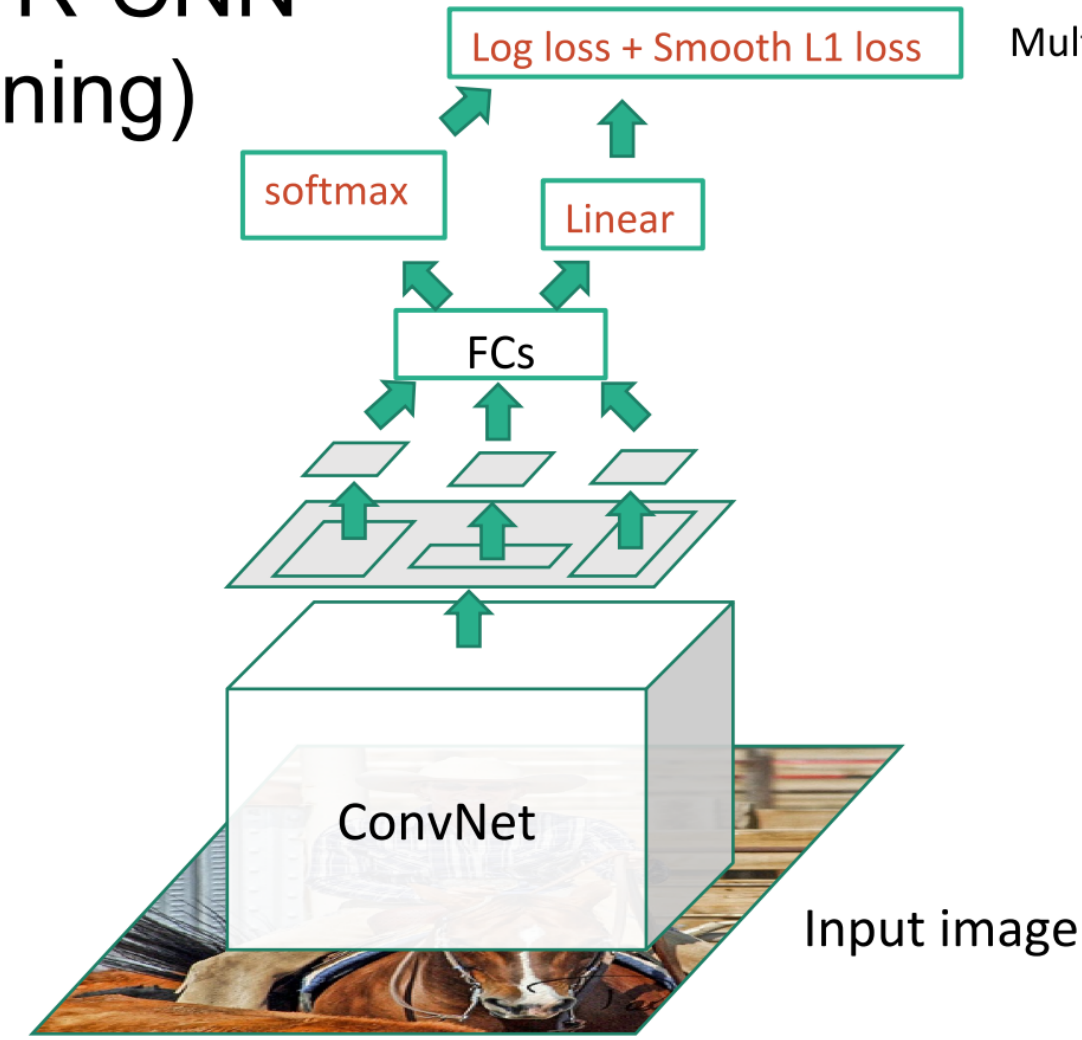
Projected region proposal is e.g. 512 x 18 x 8 (varies per proposal)

RoI conv features: 512 x 7 x 7 for region proposal

Fully-connected layers expect low-res conv features: 512 x 7 x 7

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

Fast R-CNN (Training)

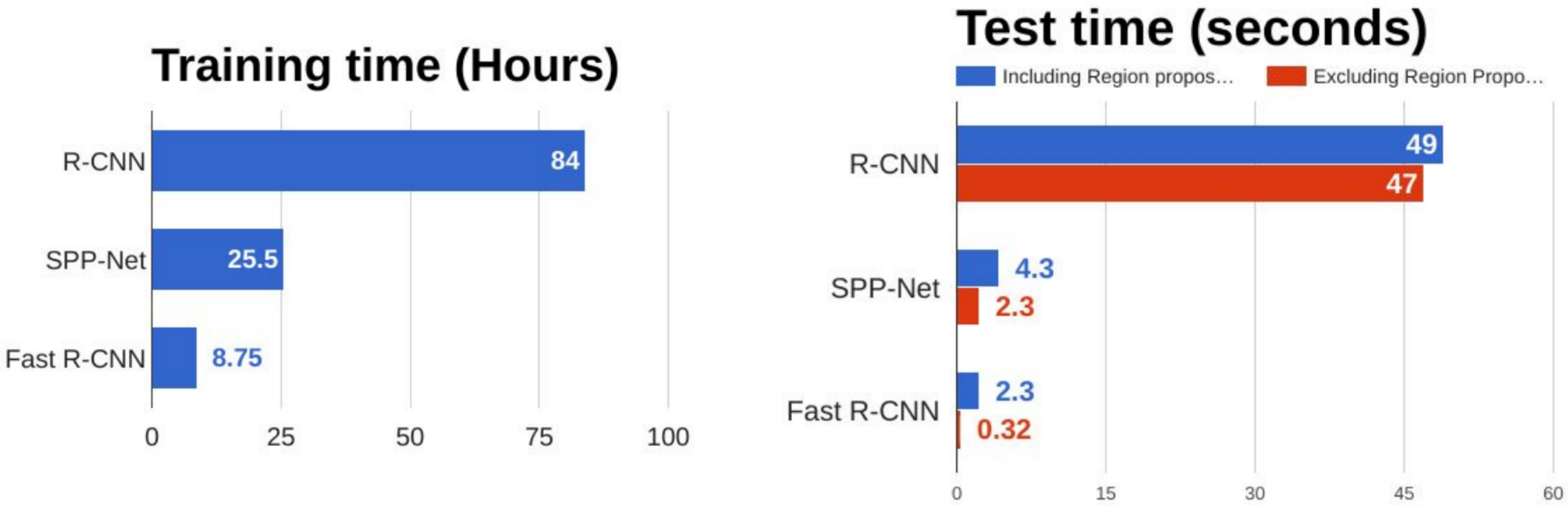


Multi-task loss

Input image

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

R-CNN vs SPP 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

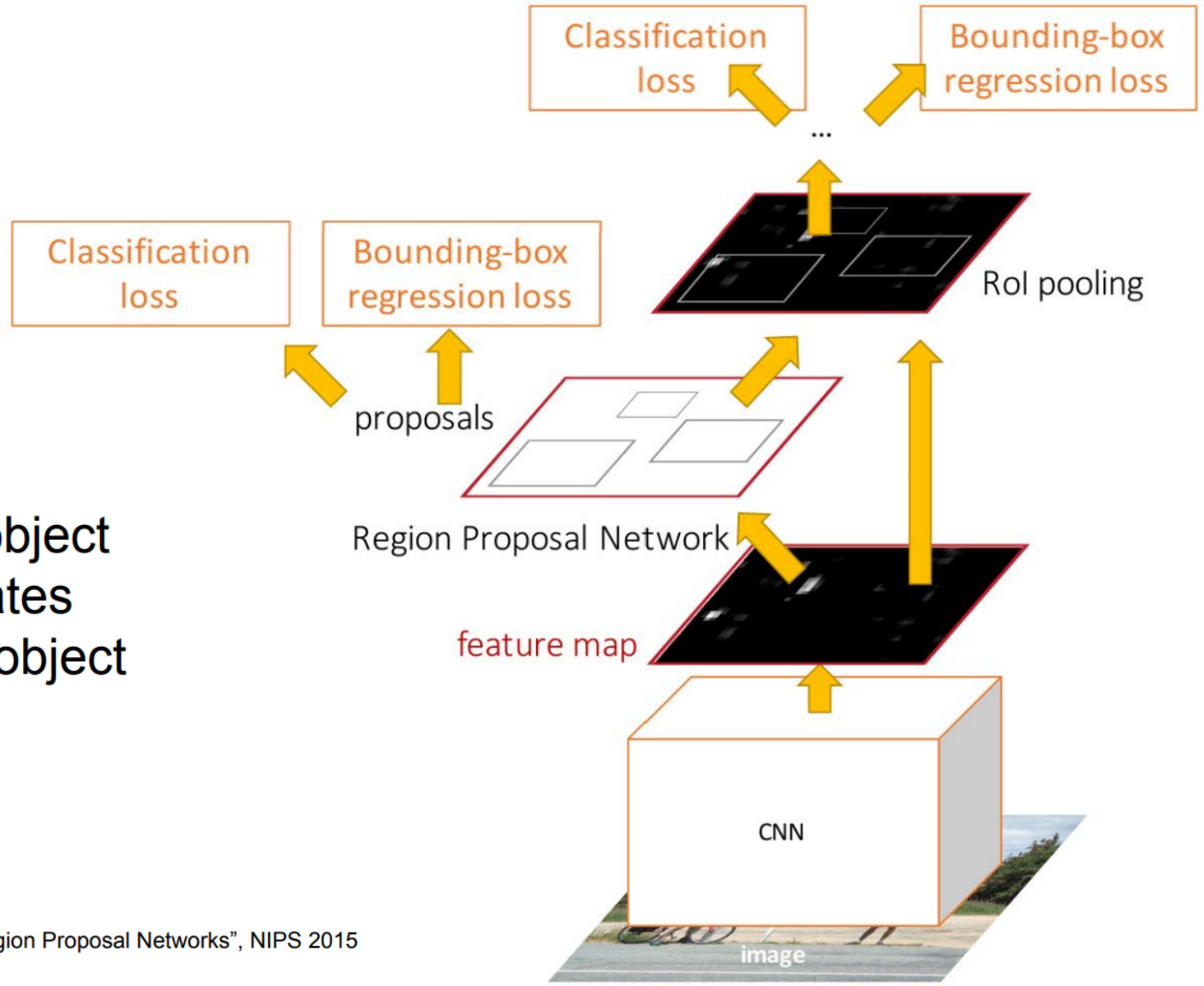
Faster R-CNN:

Make CNN do proposals!

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

Jointly train with 4 losses:

1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates

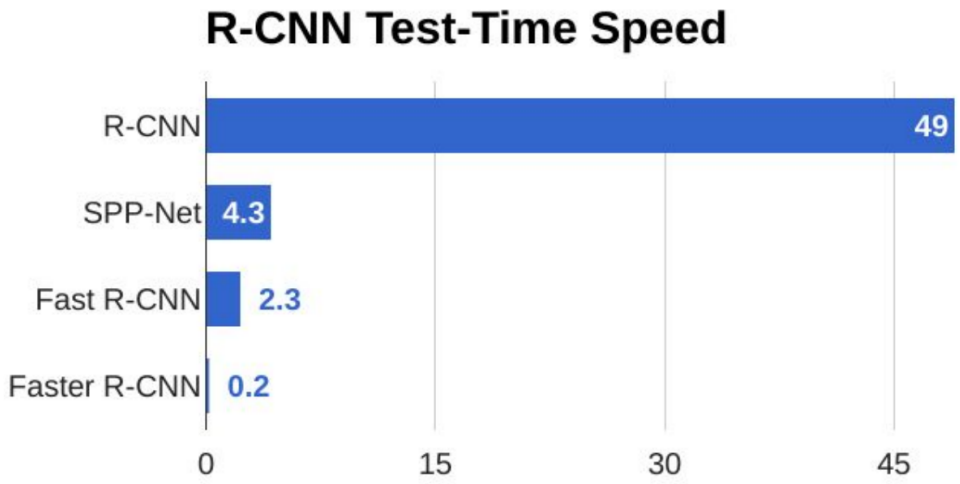


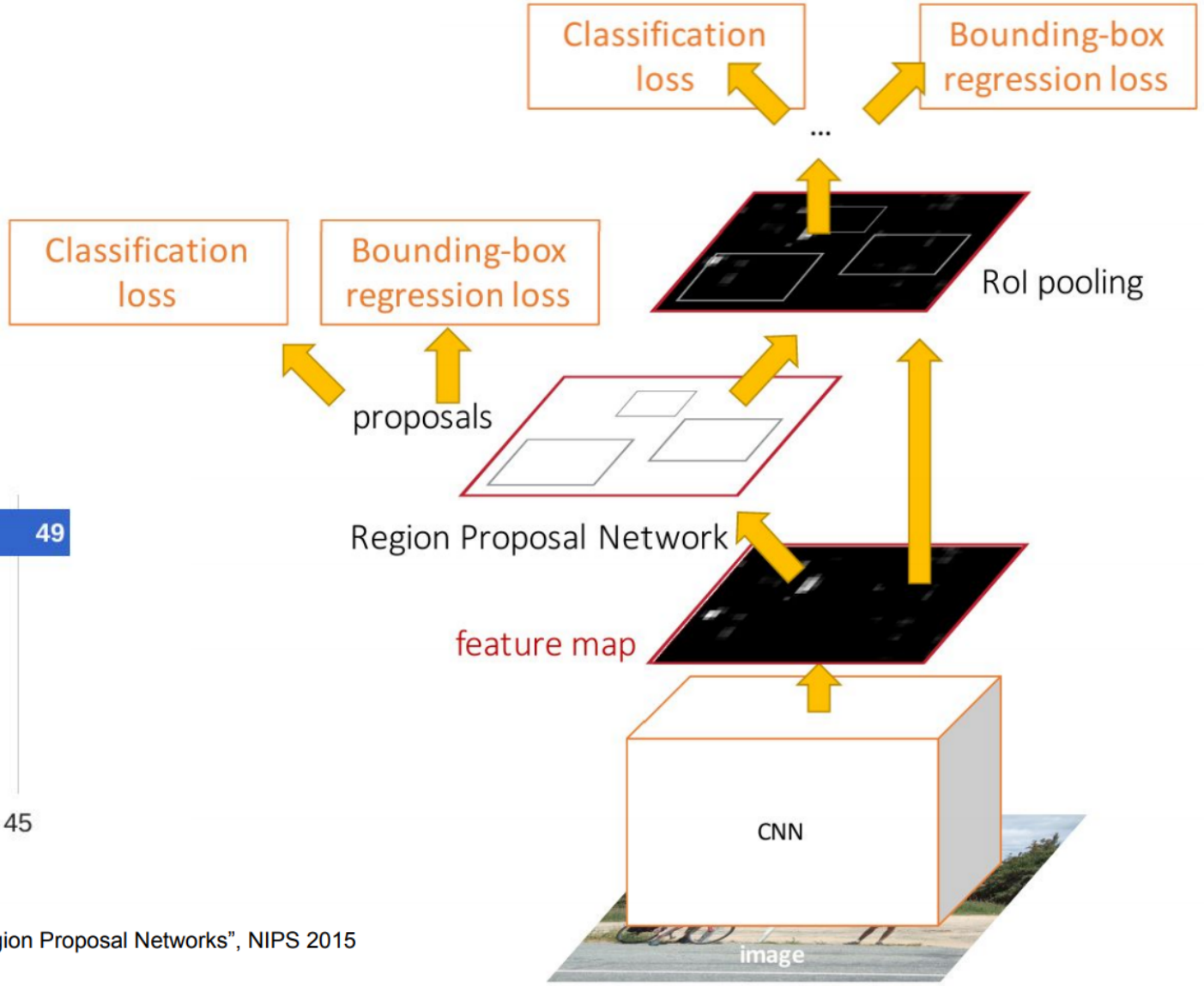
Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015
Figure copyright 2015, Ross Girshick; reproduced with permission

Faster R-CNN:

Make CNN do proposals!

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



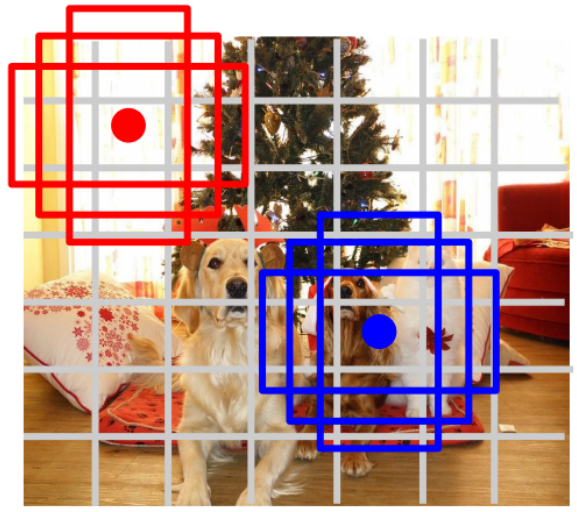


Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

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

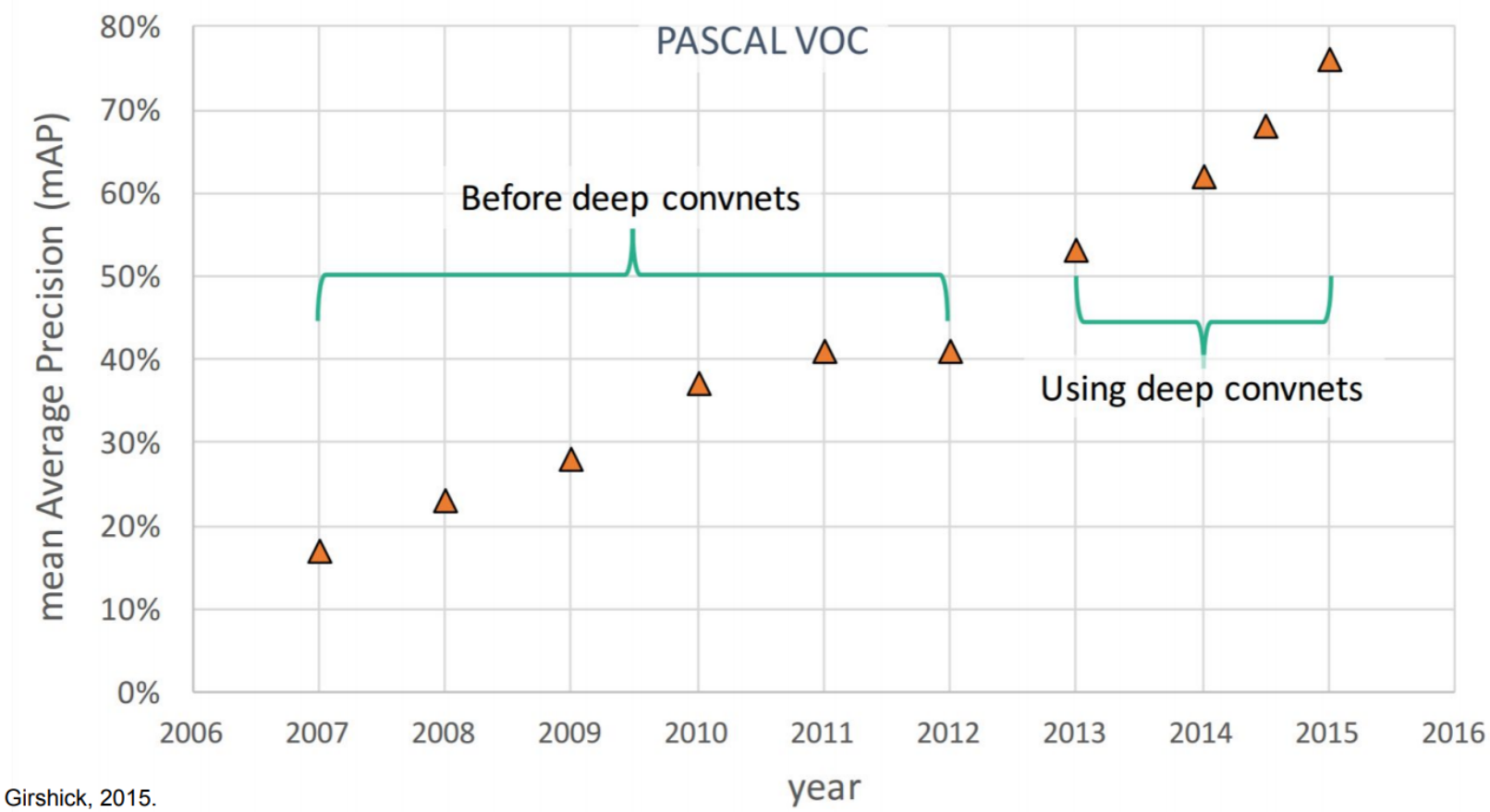
Within each grid cell:

- 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 x 7 x (5 * B + C)

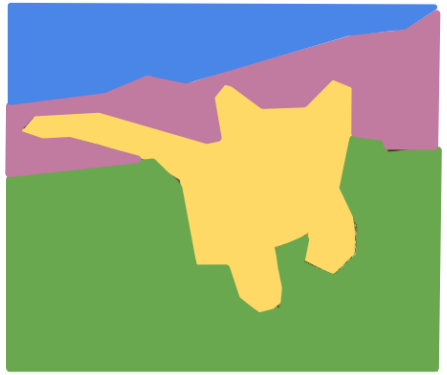
Redmon et al, "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016
Liu et al, "SSD: Single-Shot MultiBox Detector", ECCV 2016

Object Detection: Impact of Deep Learning



Other Computer Vision Tasks

Semantic Segmentation

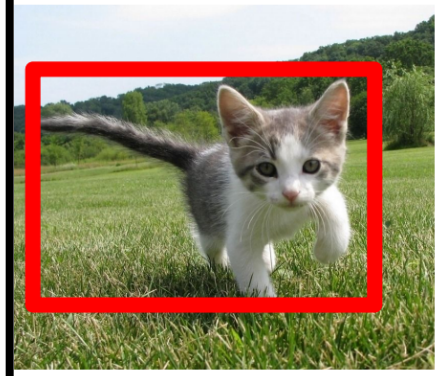


GRASS, CAT, TREE, SKY

No objects, just pixels

Balanced equation

Classification + Localization

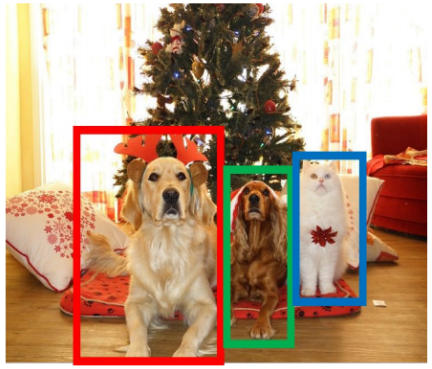


CAT

Single Object

Over-determined

Object Detection

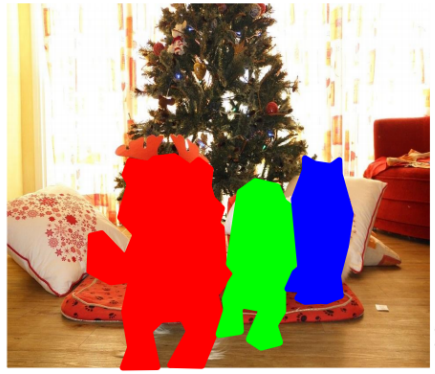


DOG, DOG, CAT

Multiple Object

Over-determined

Instance Segmentation



DOG, DOG, CAT

Over-determined

Super-resolution

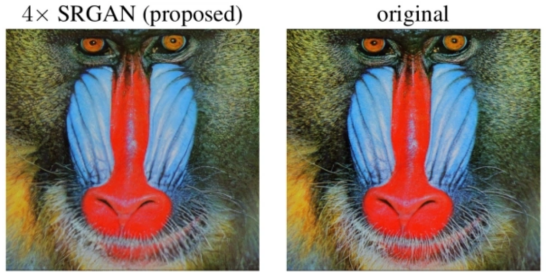


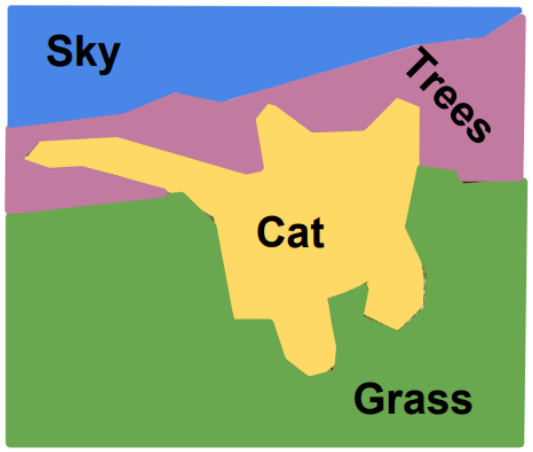
Figure 1: Super-resolved image (left) is almost indistinguishable from original (right). [4× upscaling]

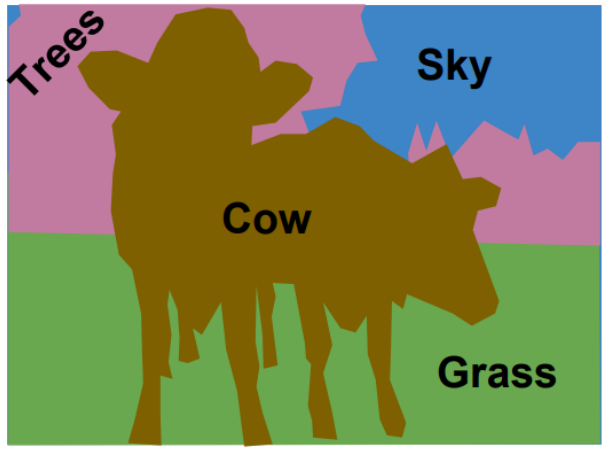
Under-determined

Semantic Segmentation

Label each pixel in the image with a category label

Don’t differentiate instances, only care about pixels

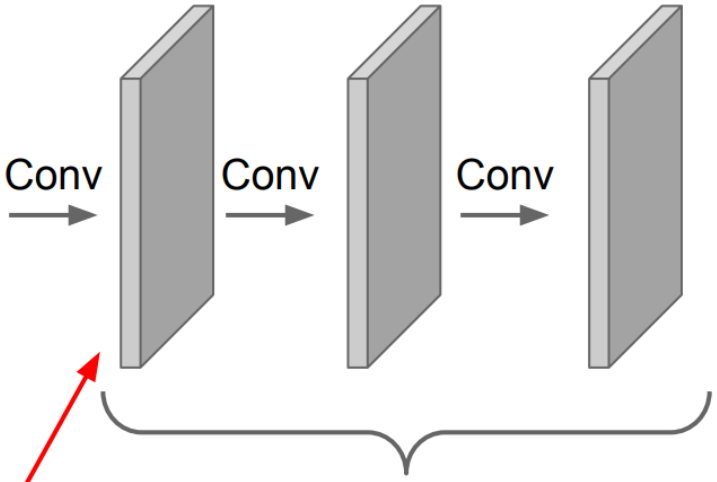




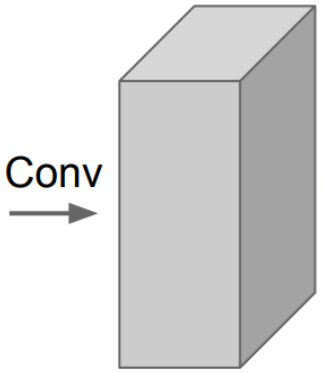
Semantic Segmentation Idea: Fully Convolutional ?

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

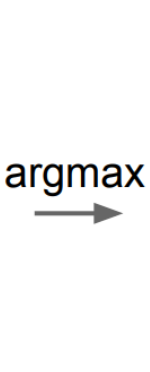
Input:
3 x H x W



Convolutions:
D x H x W



Scores:
C x H x W



Predictions:
H x W

Problem: convolutions at original image resolution will be very expensive ...

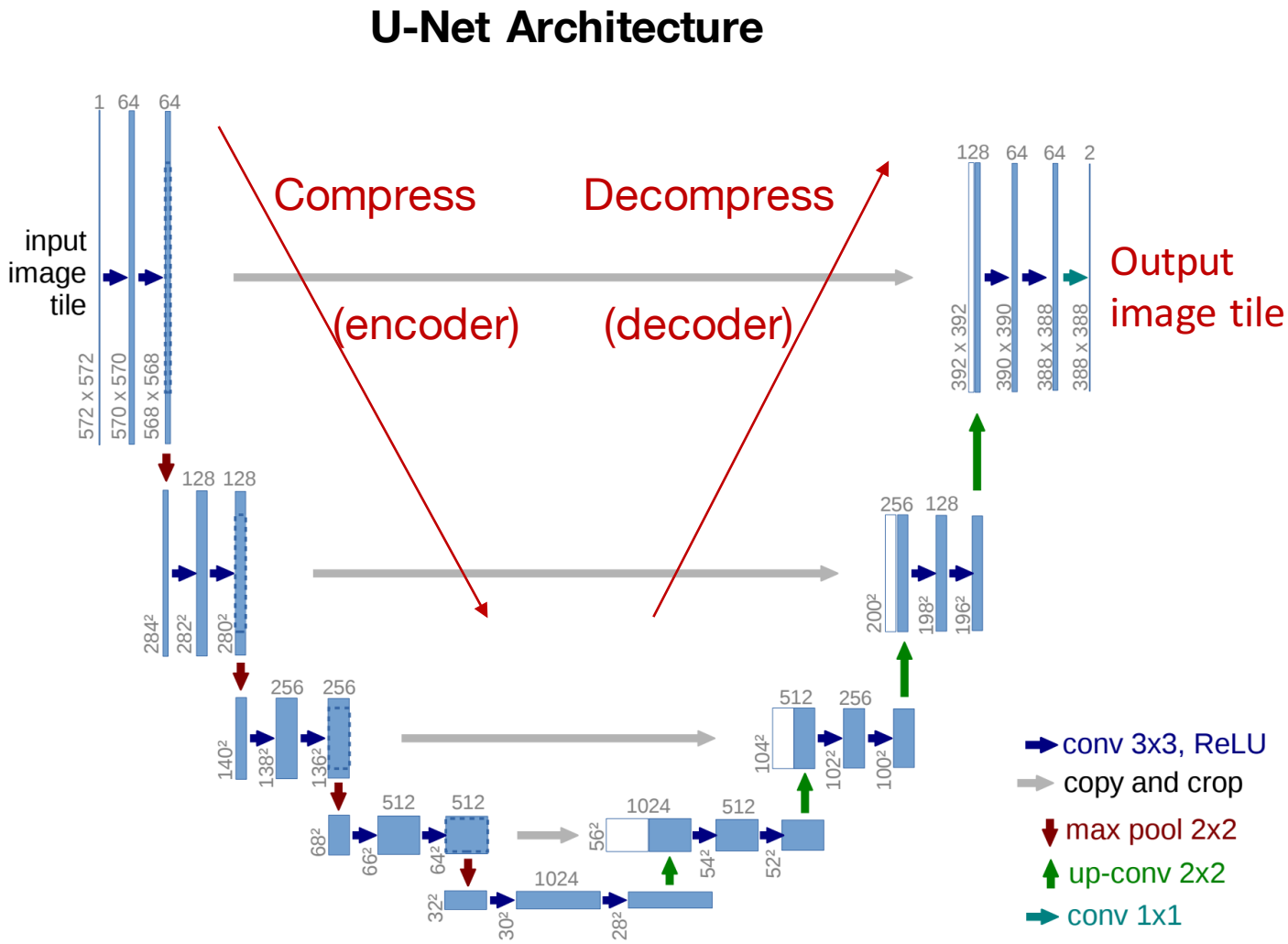
Instead, *compress x-y dimensions of* input image

- Compress spatial features into learned filters
- Then, decompress learned filters back into same spatial dimensions
- **Can be an autoencoder**
- Analogous to image compression
- A very powerful idea…

U-Net: Convolutional Networks for Biomedical Image Segmentation

Olaf Ronneberger, Philipp Fischer, and Thomas Brox

Computer Science Department and BIOSS Centre for Biological Signalling Studies, University of Freiburg, Germany
ronneber@informatik.uni-freiburg.de,
WWW home page: http://lmb.informatik.uni-freiburg.de/



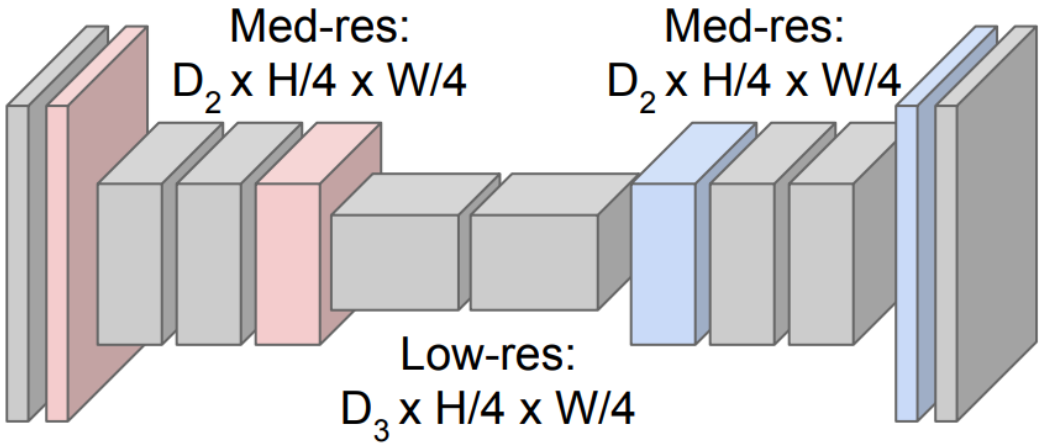
Semantic Segmentation Idea: Fully Convolutional

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