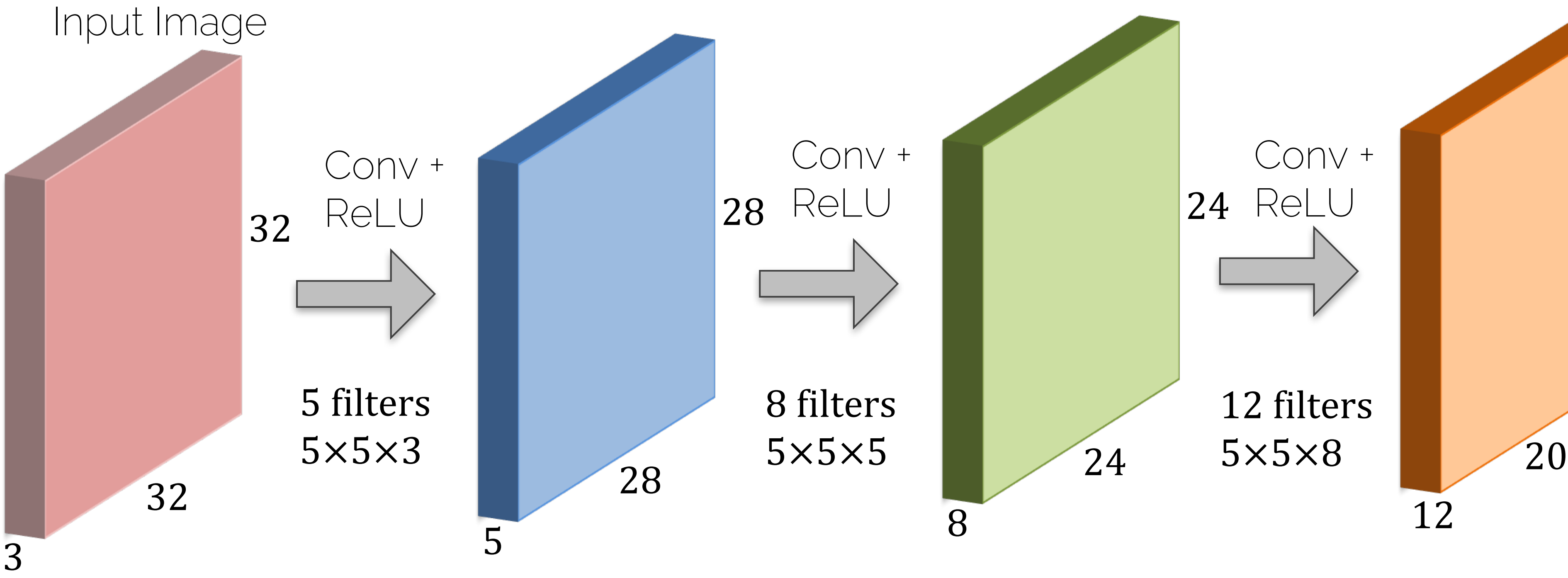
Lecture 8 Recap

Using CNNs in Computer Vision

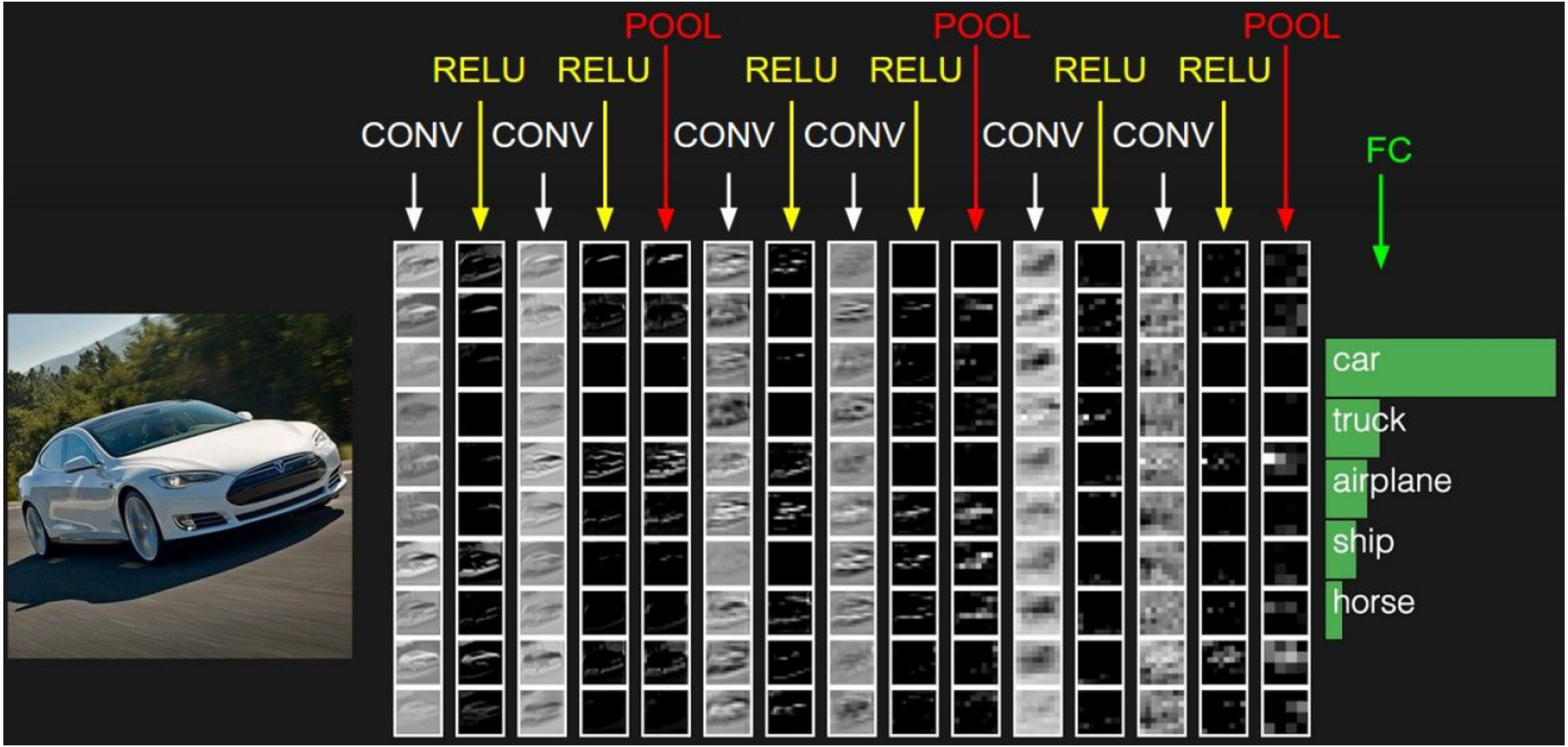
- We have CNNs (Convs, Pooling, FCs, Losses)
- We can employ them for classification
- We can employ them for regression

- Somewhat oversimplified: the "rest" is smart architectures and application of these tools
  - > of course it's more complicated ☺

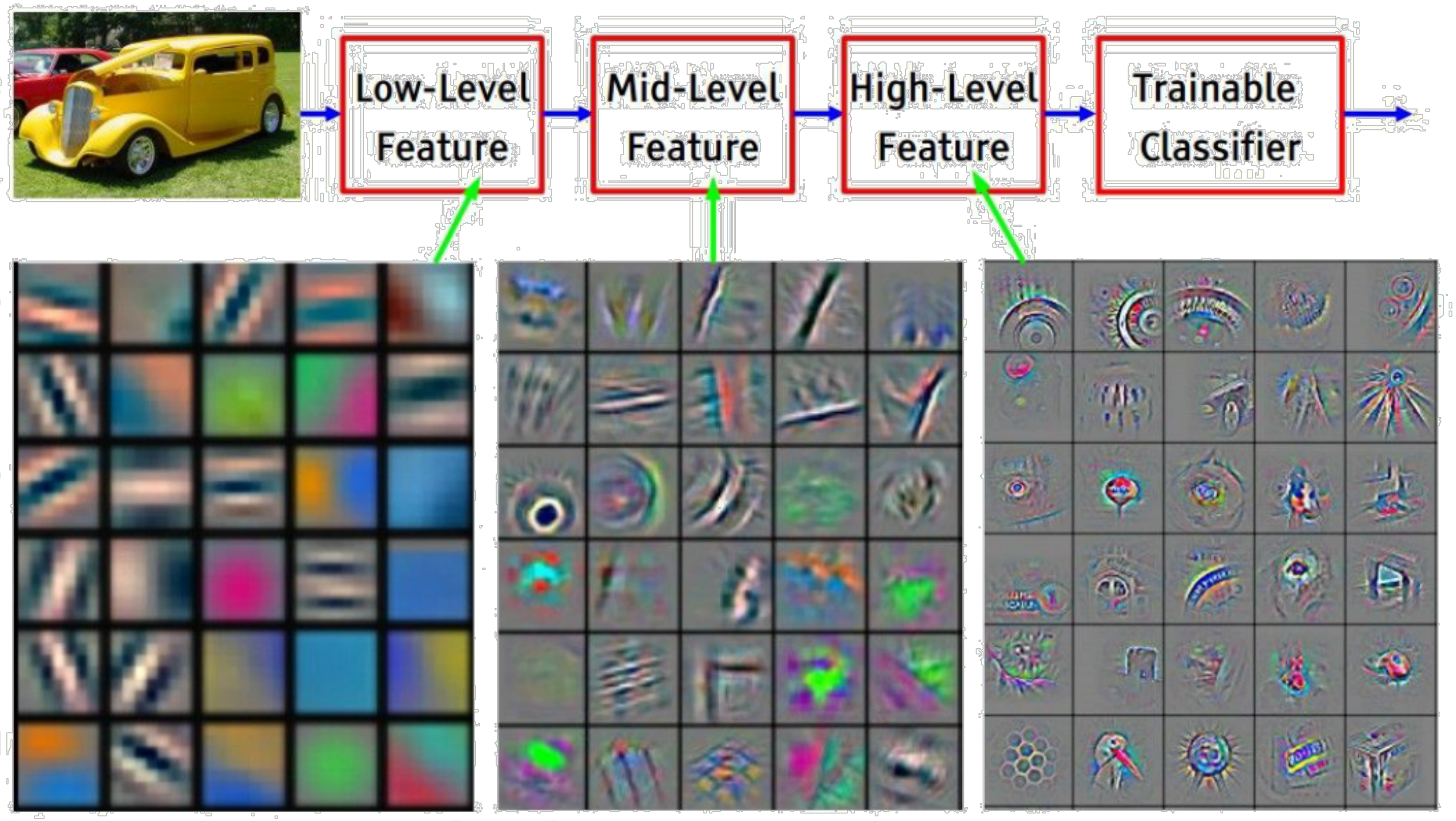
Convolution Layers: Dimensions



Convolutional Neural Network



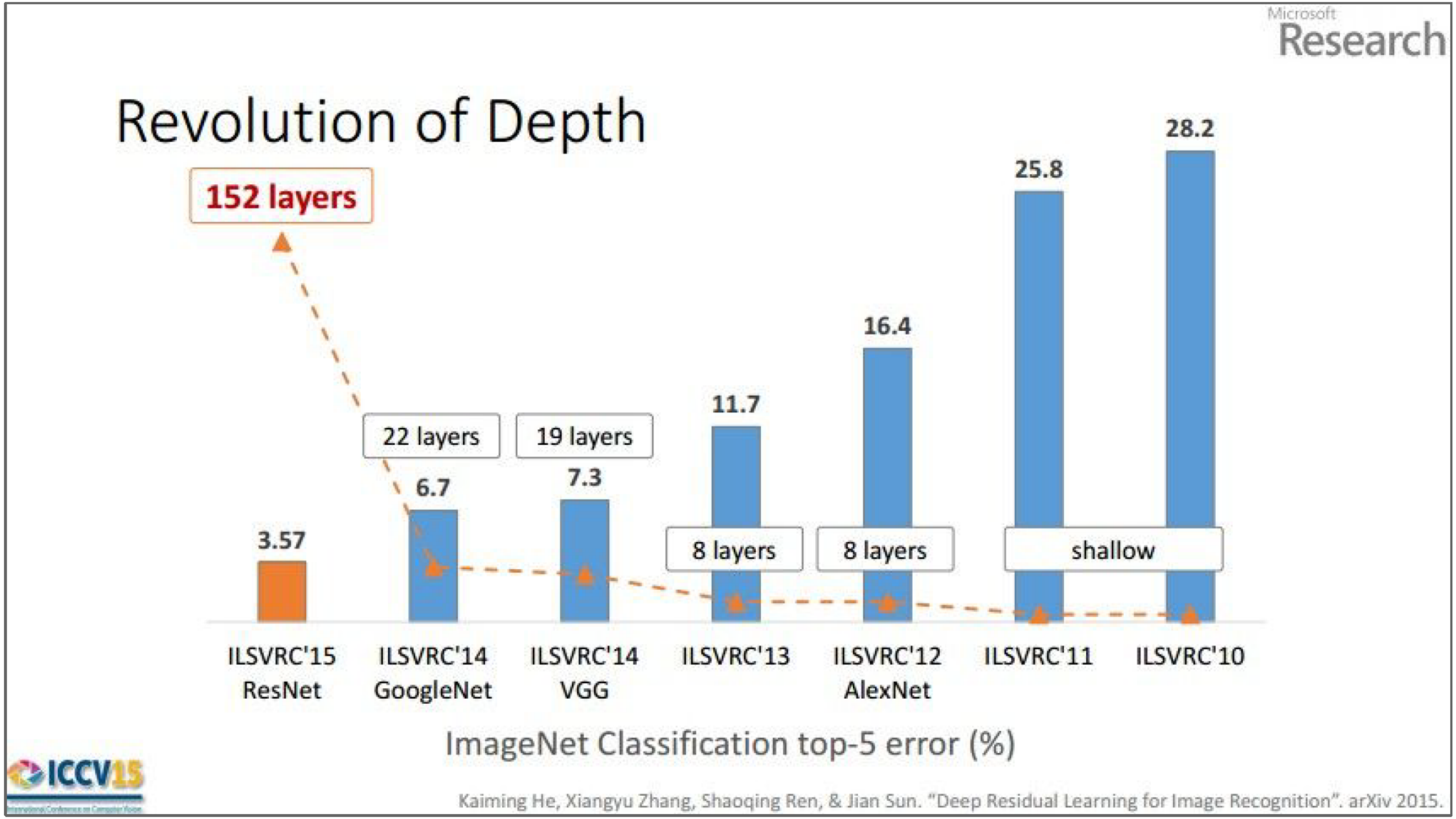
Convolutional Neural Network



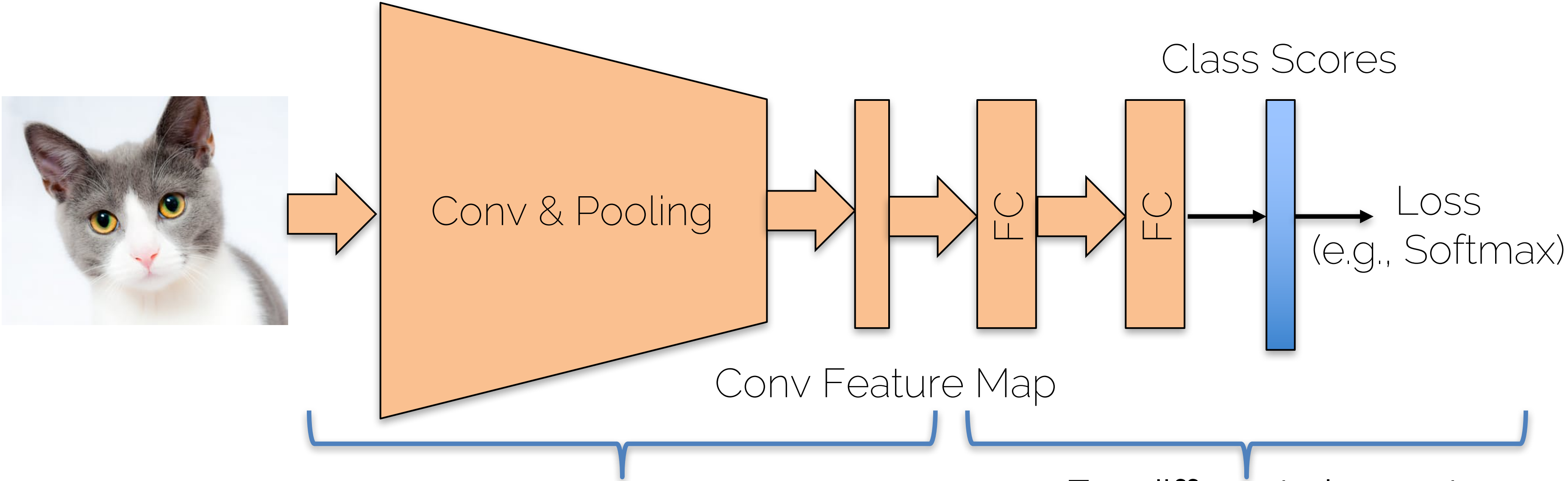
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Slide by LeCun

CNN Architectures



How to Train in Practice?

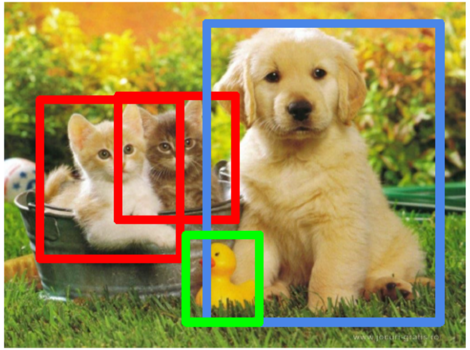


E.g. AlexNet, VGG, GoogLeNet

Use Pre-Trained Network (e.g., download model)
-> keep ConvLayers fixed

For different class set, only train FCs
-> new class scores
-> less training data
-> faster training

Using CNNs in Computer Vision

Classification	Classification + Localization	Object Detection	Instance Segmentation
			
CAT	CAT	CAT, DOG, DUCK	CAT, DOG, DUCK

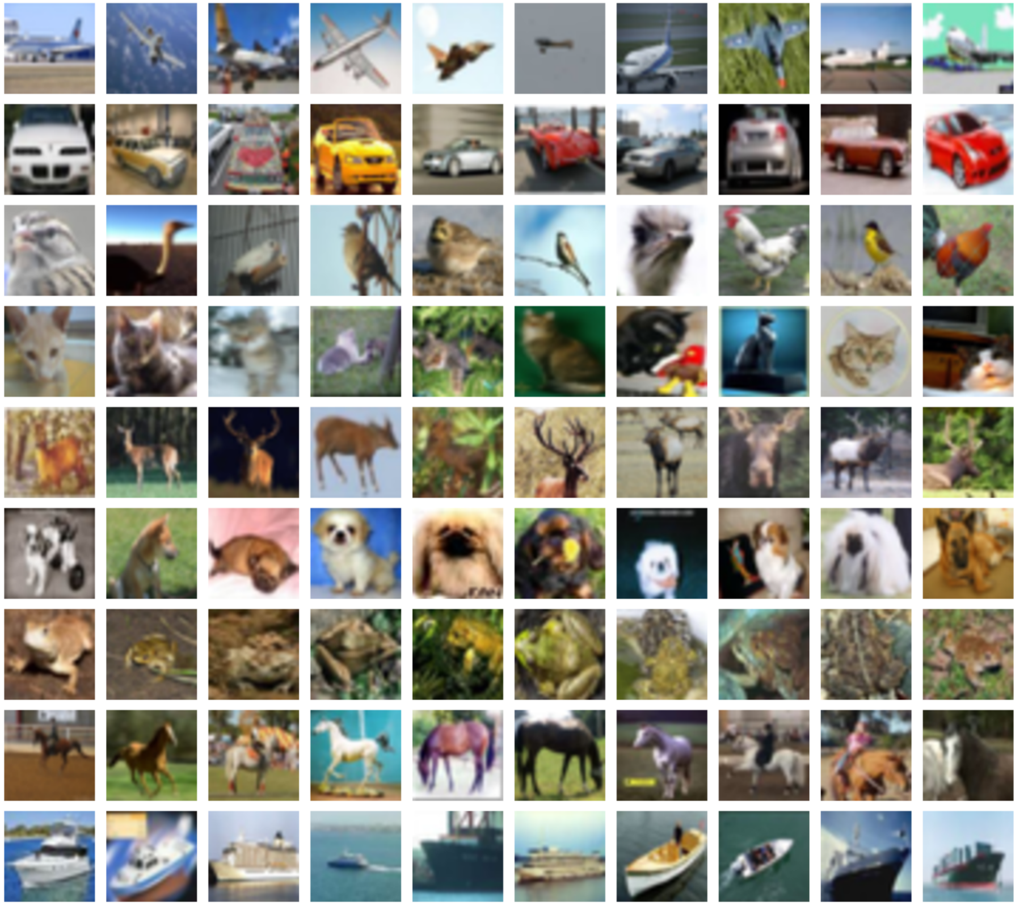
Single object

Multiple objects

Credit: Li/Karpathy/Johnson

Classification on CIFAR

airplane
automobile
bird
cat
deer
dog
frog
horse
ship



60k 32 x 32 RGB images
6k images per class
50k training and 10k test

[Krizhevsky 09]

Using CNNs in Computer Vision

Classification

Classification + Localization

Object Detection

Instance Segmentation

CIFAR 10 +
“raw” CNN ☺

Credit: Li/Karpathy/Johnson

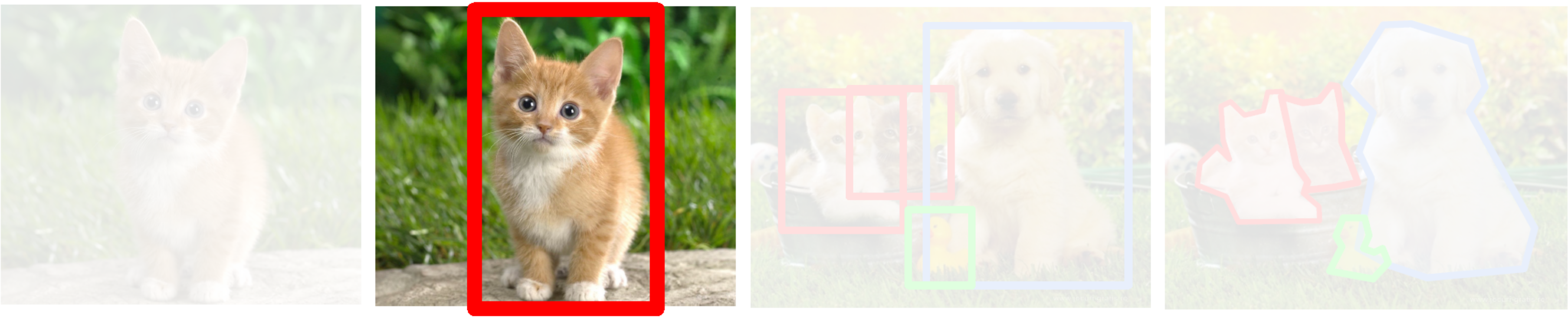
Using CNNs in Computer Vision

Classification

Classification + Localization

Object Detection

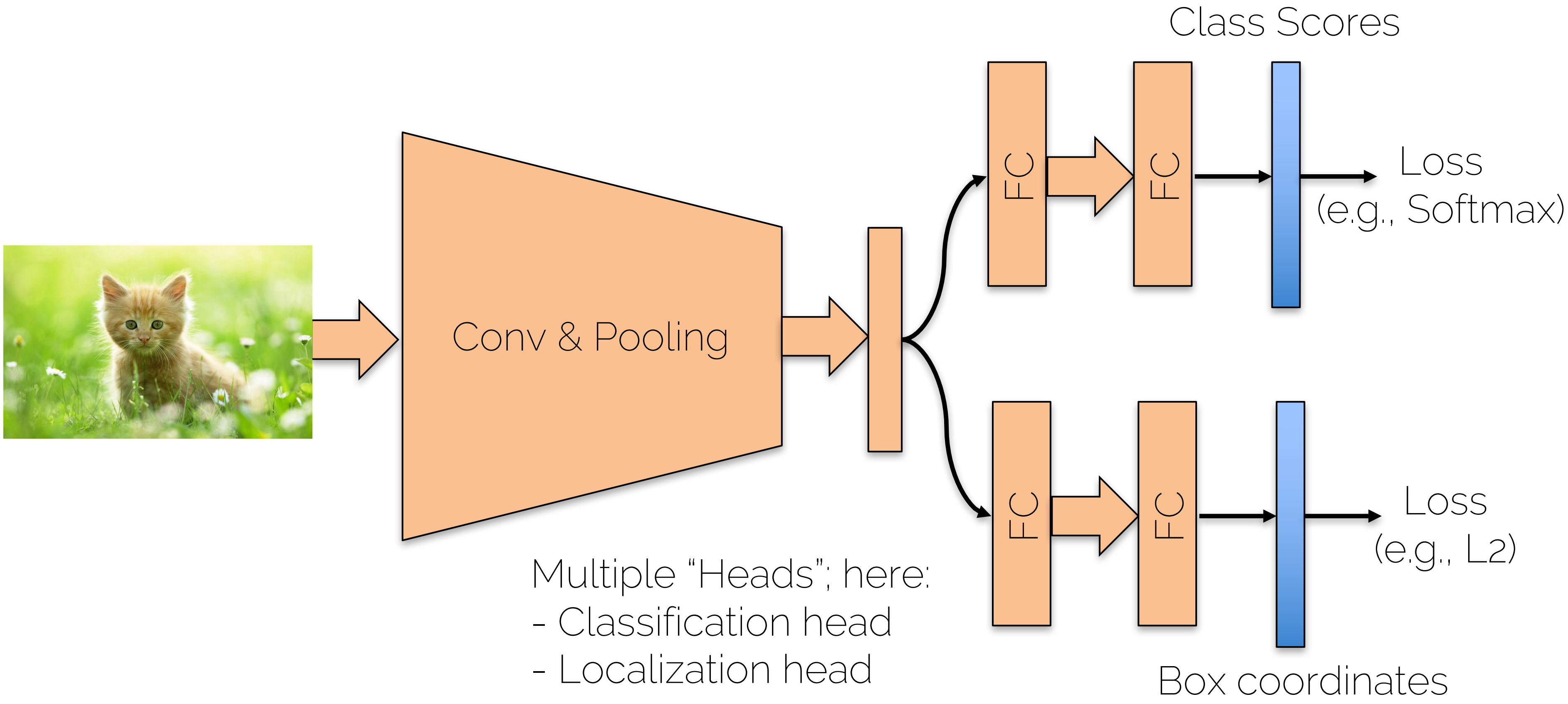
Instance Segmentation



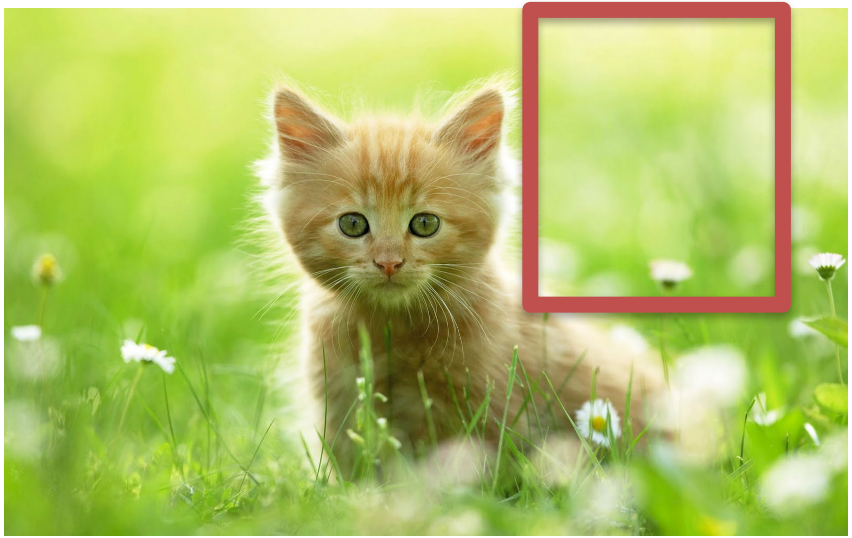
CIFAR 10 +
"raw" CNN ☺

Credit: Li/Karpathy/Johnson

Classification + Localization: Regression



Classification + Localization: Sliding Window

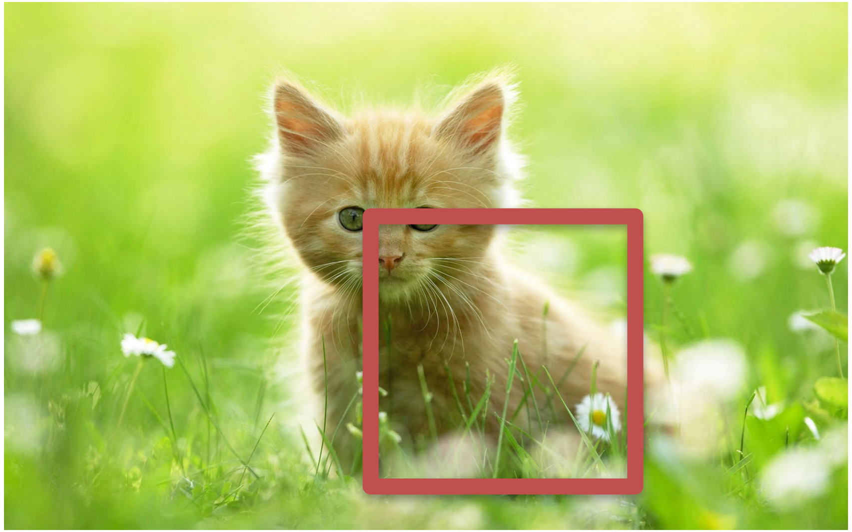


Class score (cat):
Box location 0 -> score 0.02

Classification + Localization: Sliding Window

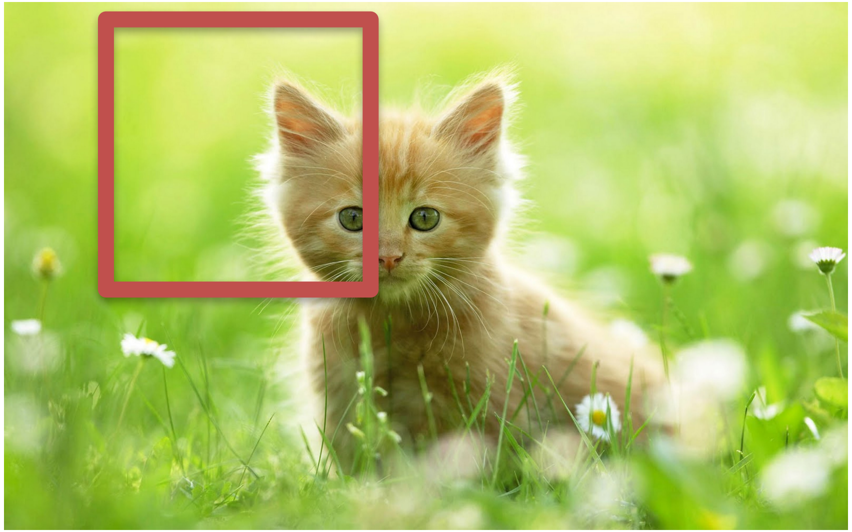
Class score (cat):
Box location 0 -> score 0.02
Box location 1 -> score 0.2

Classification + Localization: Sliding Window



Class score (cat):
Box location 0 -> score 0.02
Box location 1 -> score 0.2
Box location 2 -> score 0.42

Classification + Localization: Sliding Window



Class score (cat):

Box location 0	-> score 0.02
Box location 1	-> score 0.2
Box location 2	-> score 0.42
Box location 3	-> score 0.31

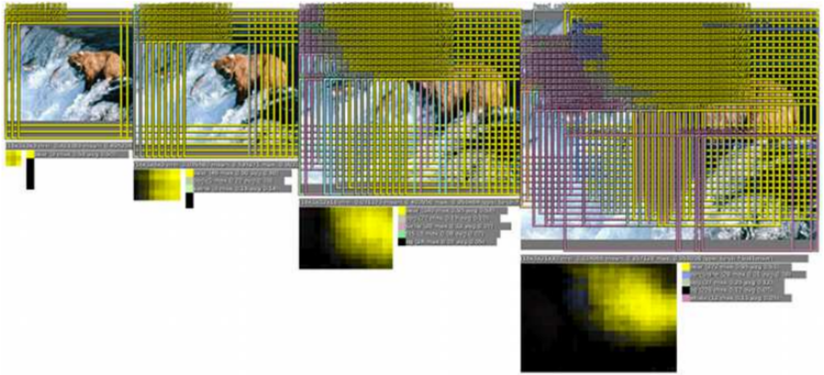
Classification + Localization: Sliding Window

Class score (cat):

Box location 0	-> score 0.02
Box location 1	-> score 0.2
Box location 2	-> score 0.42
Box location 3	-> score 0.31
Box location 4	-> score 0.8

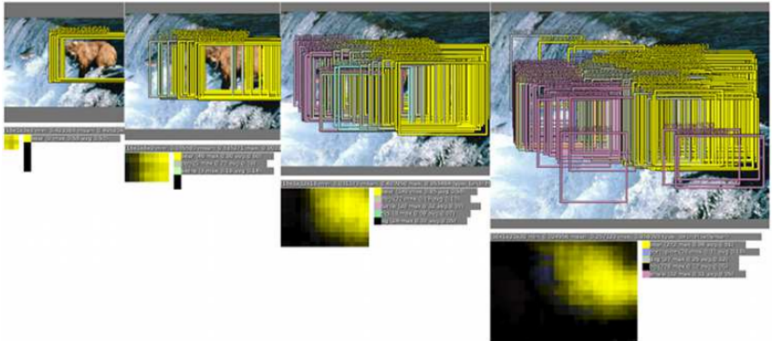
Take winning box location as result

Sliding Window: Overfeat



1) Window positions + score maps

3) Final bounding box prediction

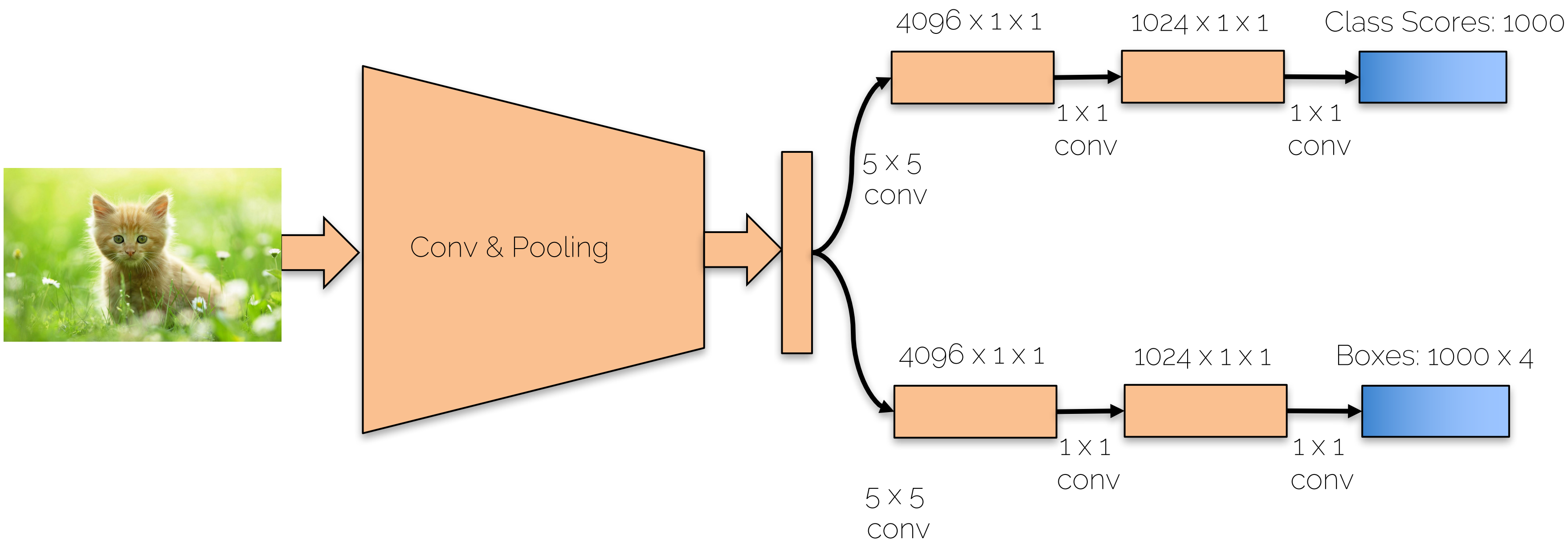


2) Box regression

[Sarmenet et al.: Overfeat, 14]

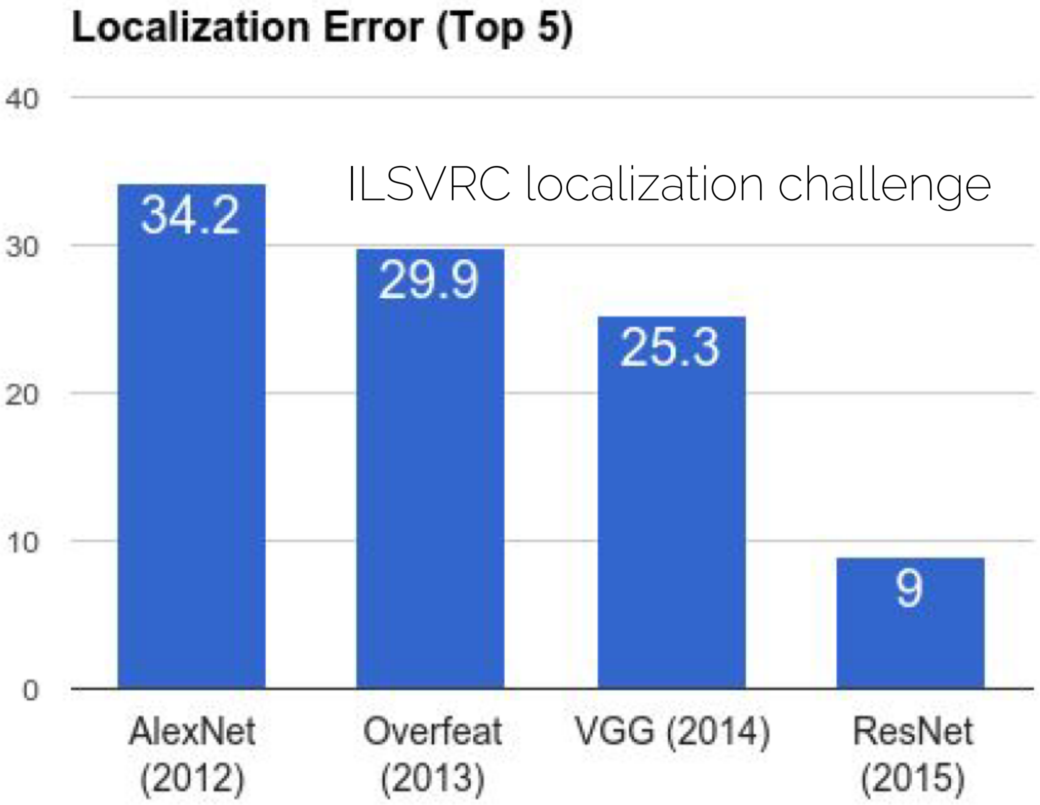
Sliding Window: Overfeat

Efficient sliding by converting FCs into convs



[Sarmenet et al.: Overfeat, 14]

ImageNet Classification + Localization



Overfeat: Multiscale conv regression with box merging

VGG: Mostly the same, but better network (also fewer scales and location, gain by better features)

ResNet: Crazy network, and different localization method (region proposals, RPN)

Credit: Li/Karpathy/Johnson

Important Datasets to Know

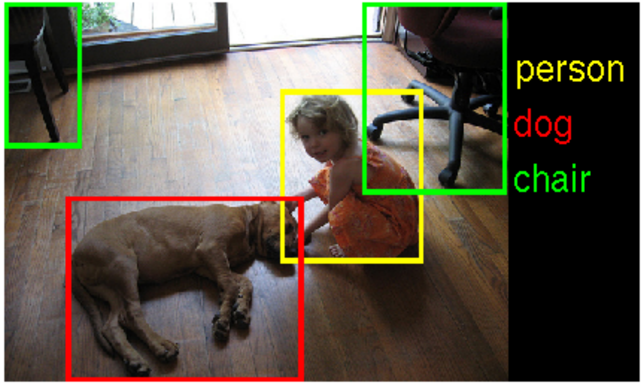
CIFAR-10: single object, centered, Krizhevsky et al.

MNIST: handwritten digits, LeCun et al.

Pascal VOC, 20 classes, 10k images, Everingham et al.

ImageNet: 10 mio images, Deng et al.

MS-COCO, 300k images, Lin et al. 15



Using CNNs in Computer Vision

Classification

CIFAR 10 +
"raw" CNN ☺

Classification + Localization

Regression and/or
sliding window

Object Detection

Instance Segmentation

Credit: Li/Karpathy/Johnson

Using CNNs in Computer Vision

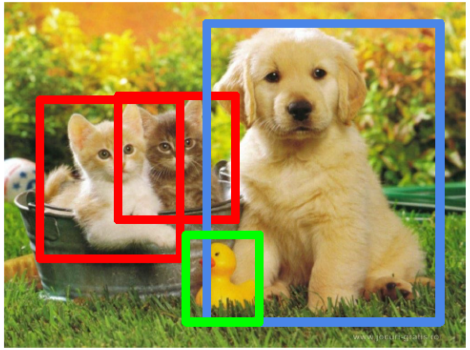
Classification

CIFAR 10 +
"raw" CNN ☺

Classification + Localization

Regression and/or
sliding window

Object Detection



Instance Segmentation

Multiple objects!
(but we don't know how many)

Credit: Li/Karpathy/Johnson

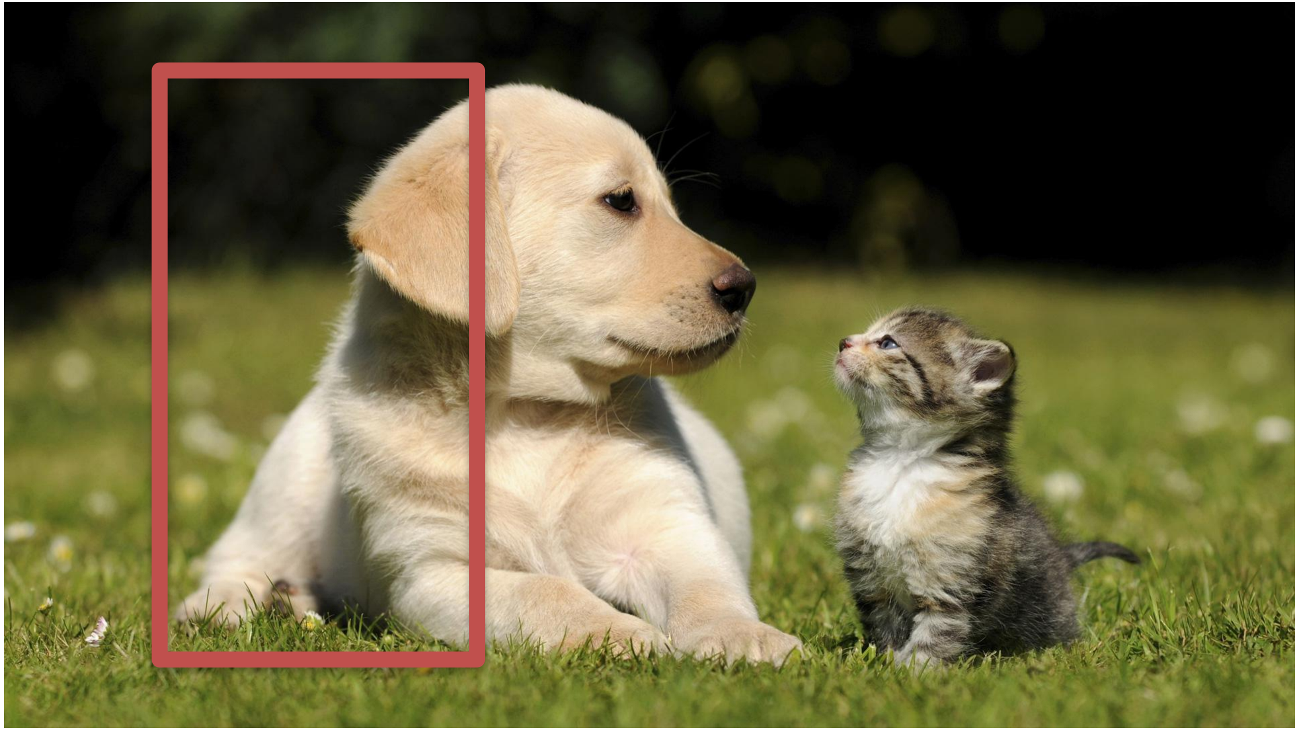
Object Detection as Classification

2 classes

Dog: no

Cat: no

Object Detection as Classification

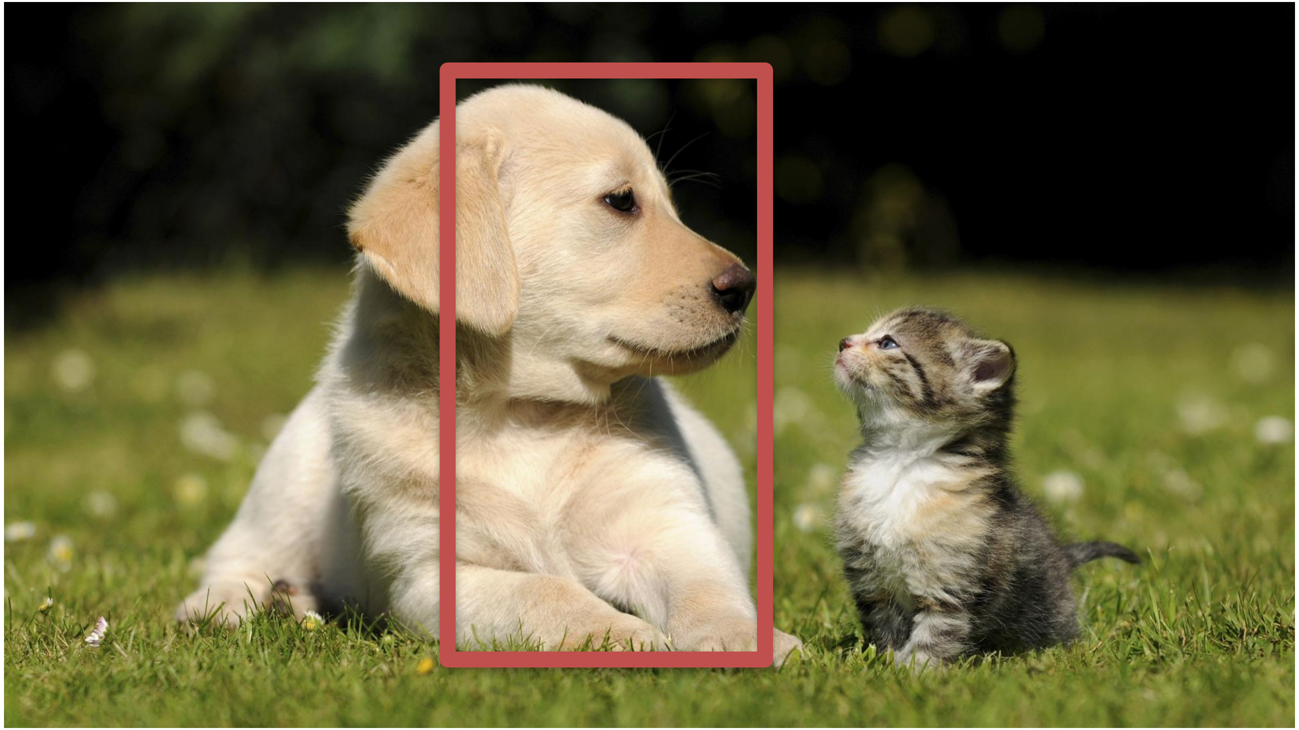


2 classes

Dog: maybe

Cat: no

Object Detection as Classification

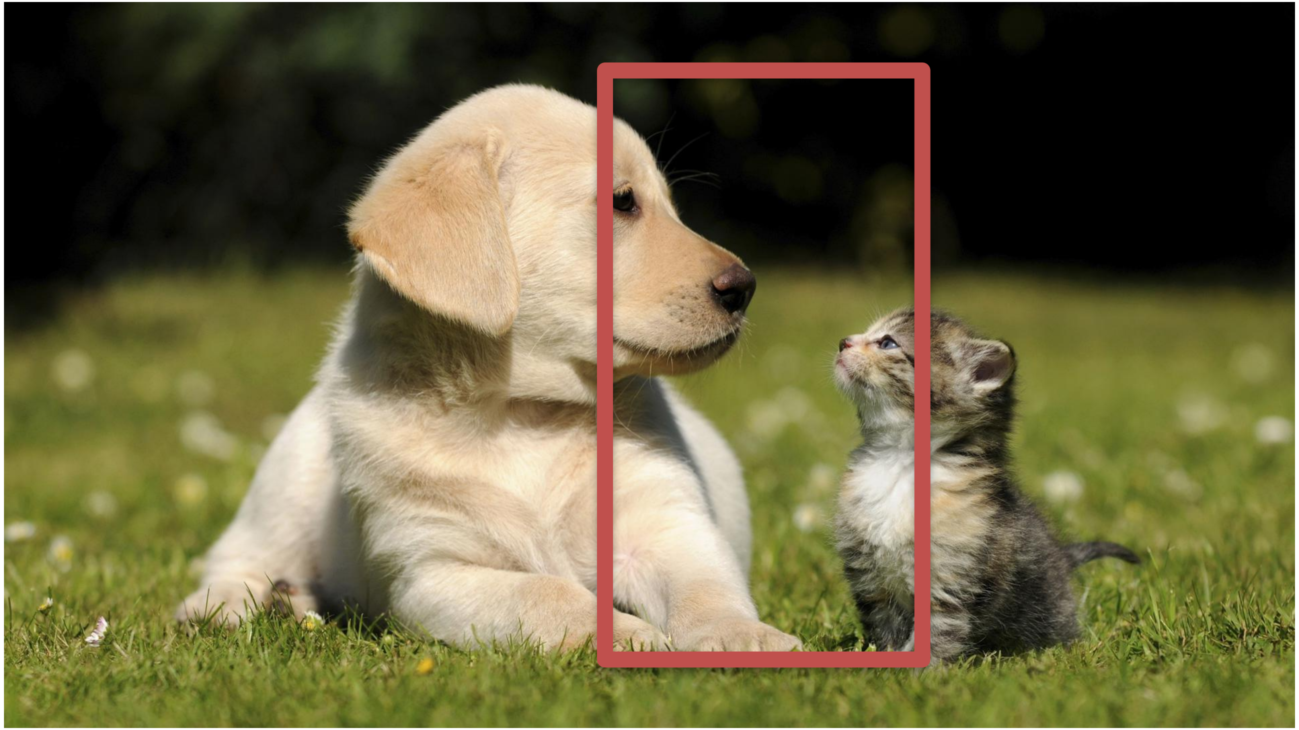


2 classes

Dog: yes

Cat: no

Object Detection as Classification

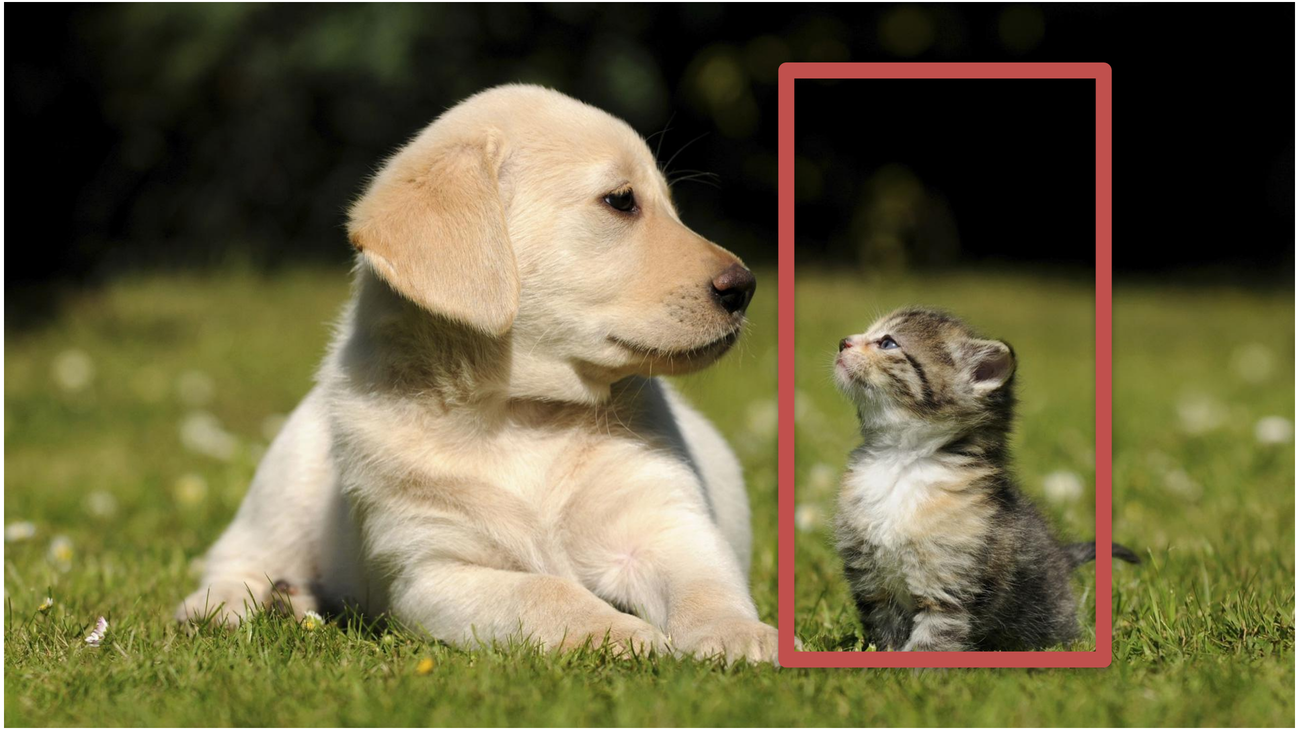


2 classes

Dog: maybe

Cat: maybe

Object Detection as Classification

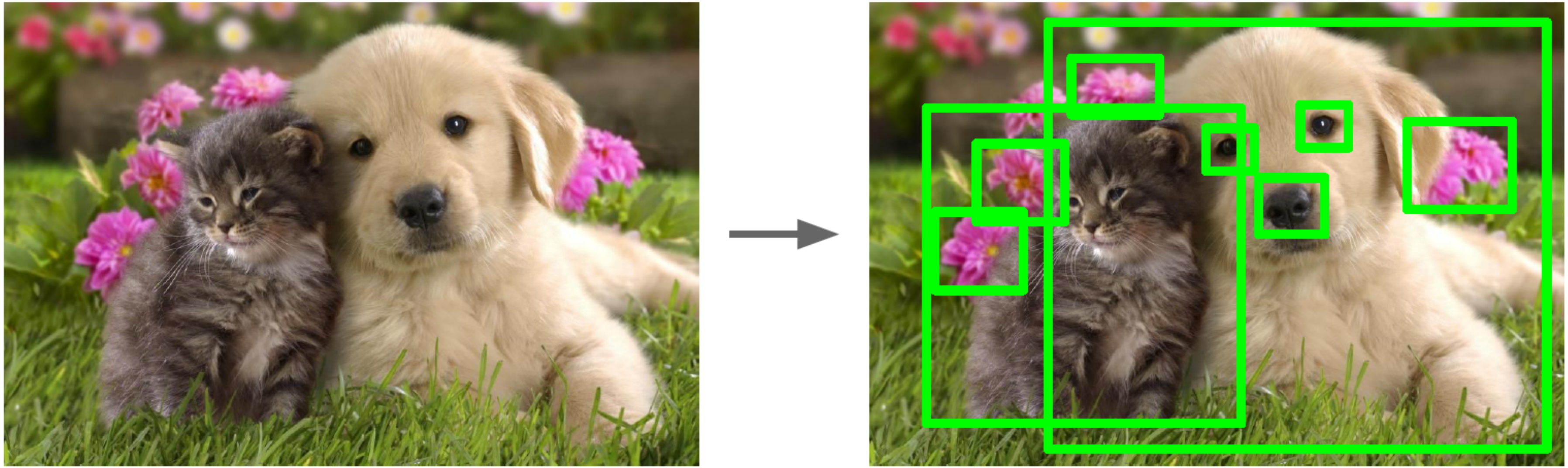


2 classes

Dog: no

Cat: yes

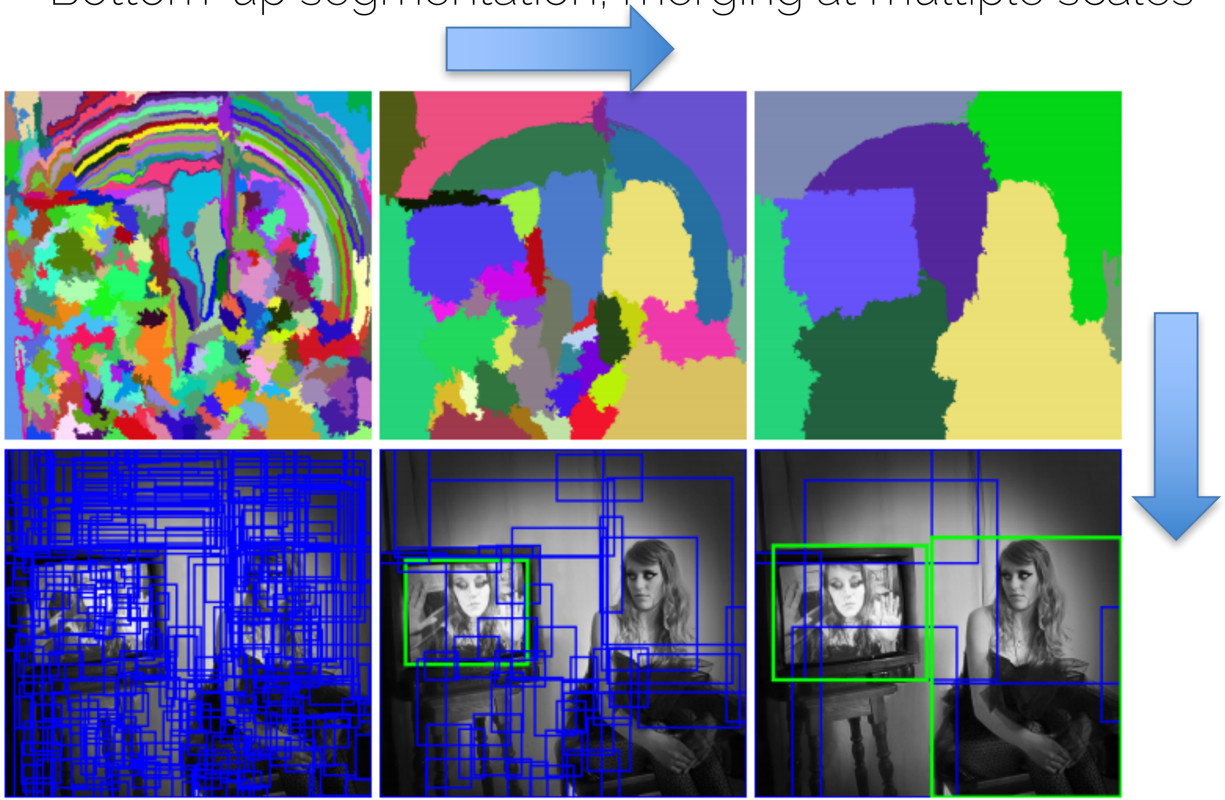
Region Proposals



Credit: Li/Karpathy/Johnson

Region Proposals: Selective Search

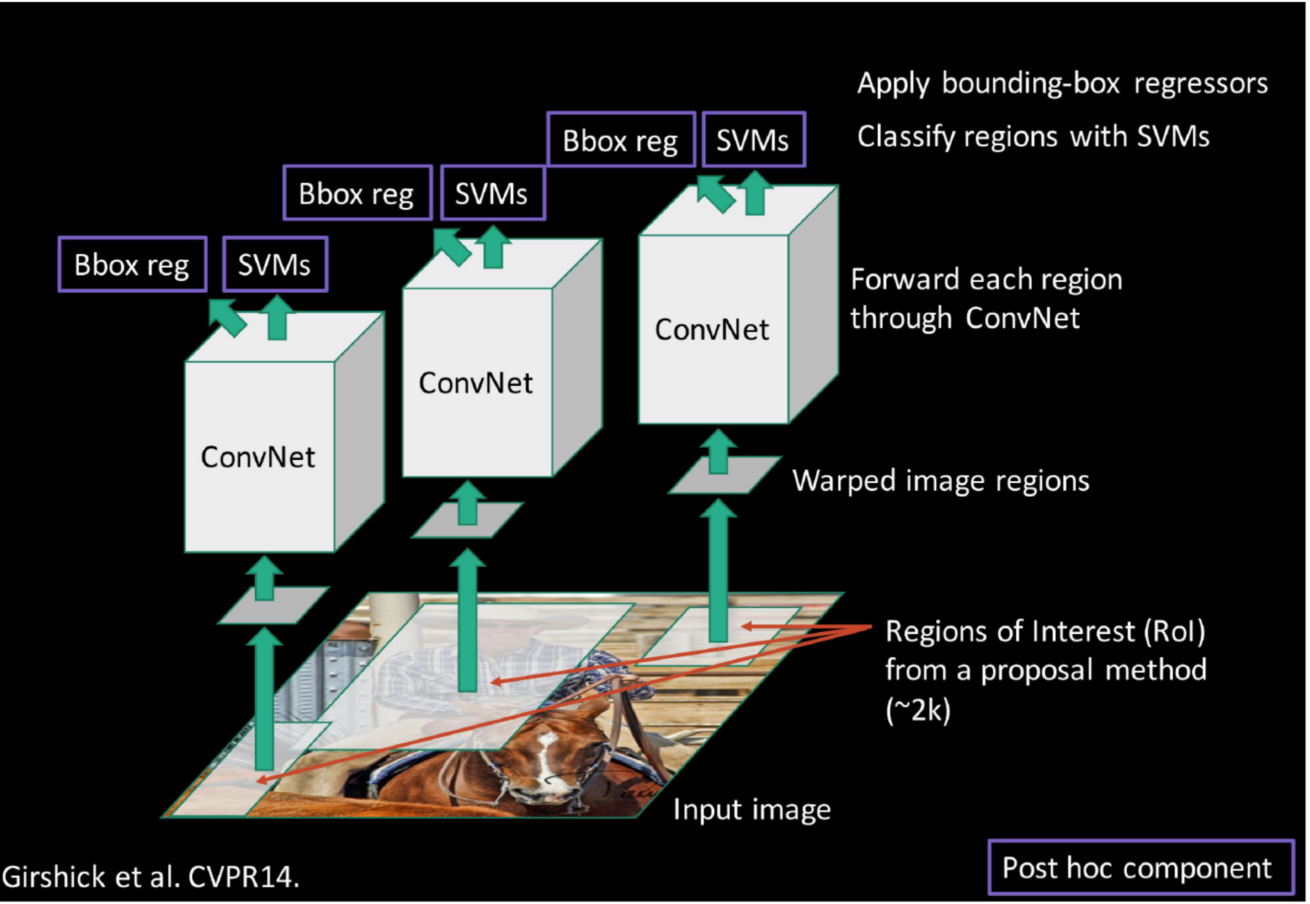
Bottom-up segmentation, merging at multiple scales



Convert regions to boxes

[Uijlings et al. 13, Selective Search for Object Recognition]

Putting it Together: R-CNN

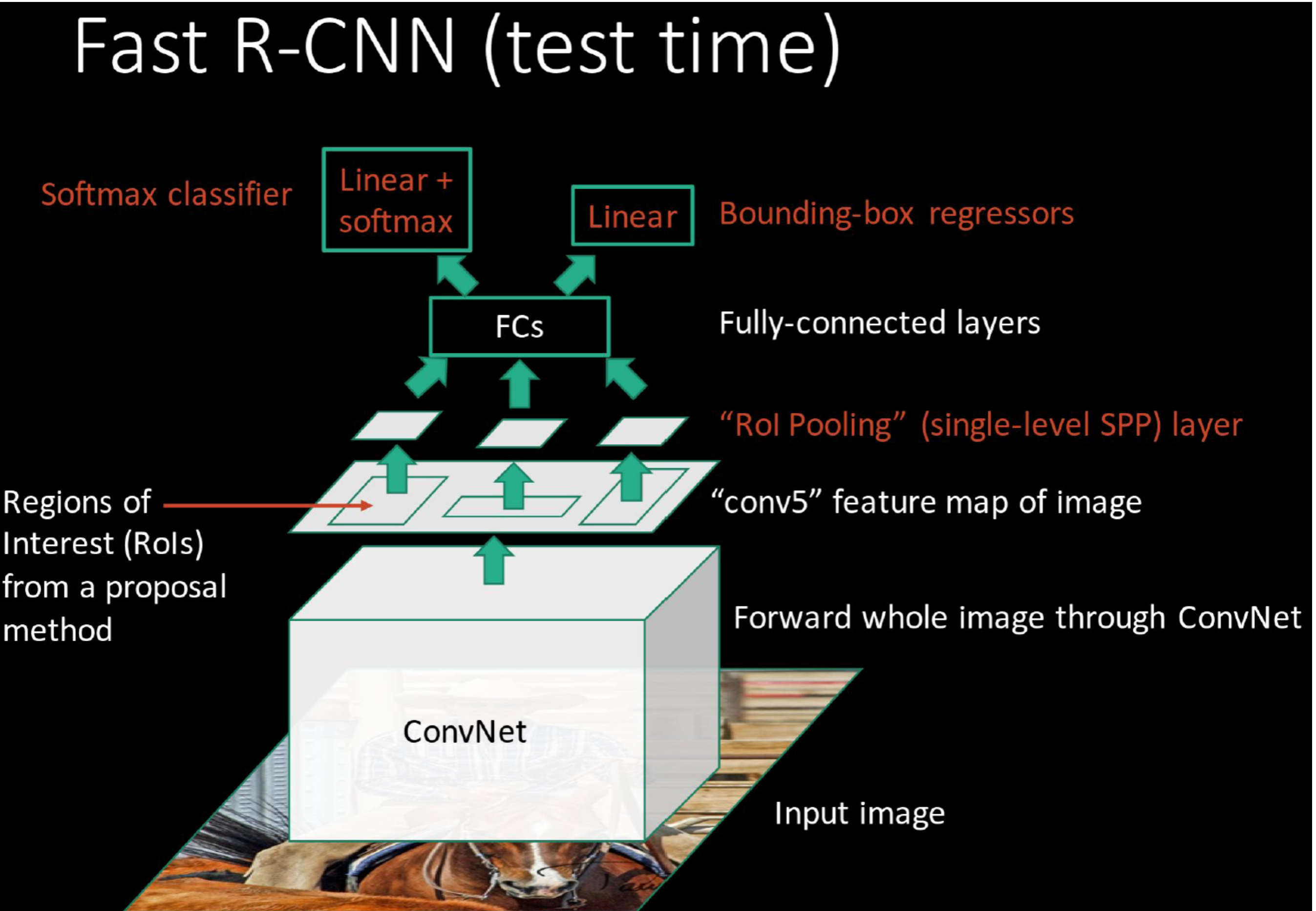


1) Run region proposal (e.g., selective search)

2) Warp (i.e., re-scale, re-size) to a fixed image size

3) This fixed output is fit into a CNN with class + regression head, which corrects for slightly off proposals

Fast R-CNN (testing)

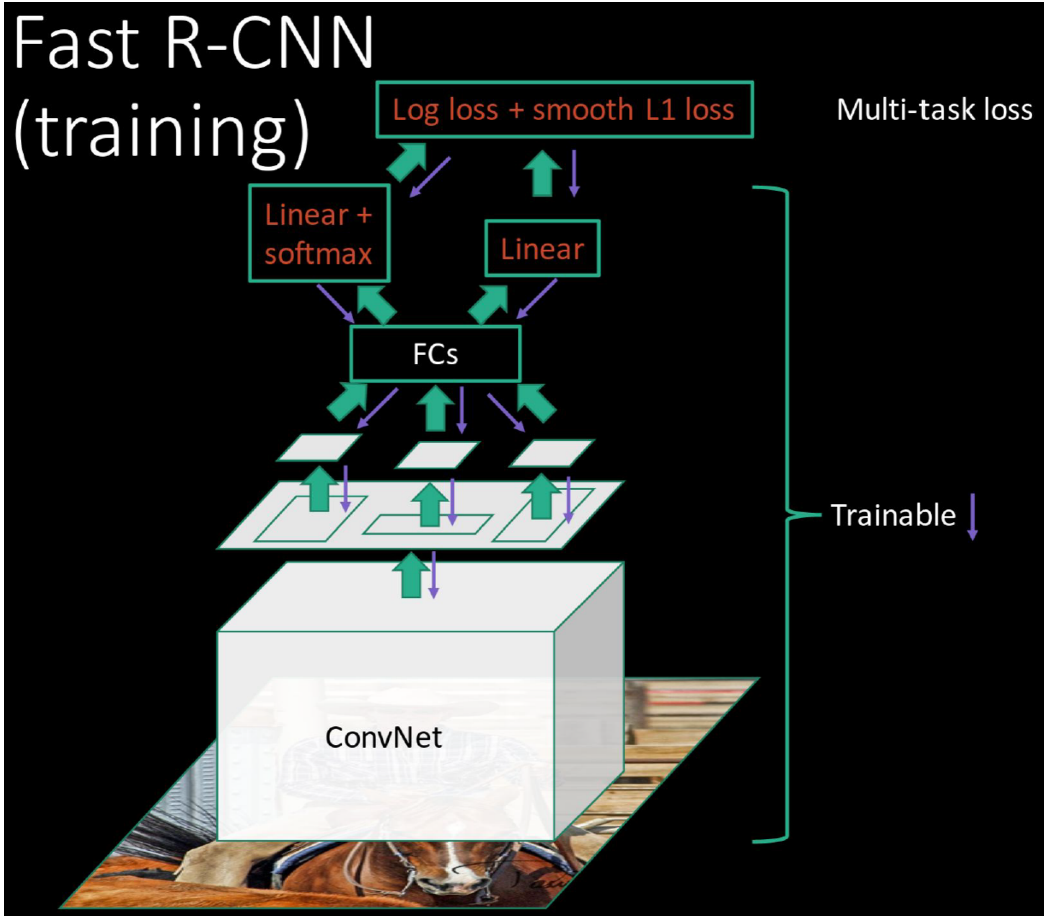


Solves test-time issue due to independent CNN forward passes

-> now one pass that shares computation of conv layers between proposals with in an image

[Girshick 15, Fast R-CNN]

Fast R-CNN (training)

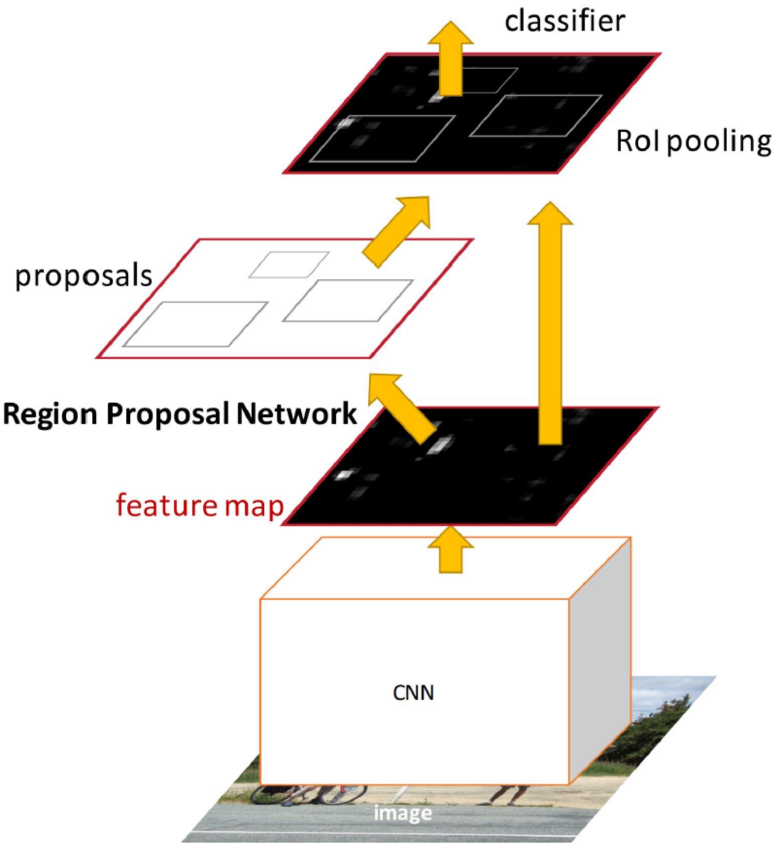


Solves training time issue: 1) CNN not updated with SVM losses. 2) Complex training pipeline

-> Just train whole thing end-to-end

[Girshick 15, Fast R-CNN]

Faster R-CNN



Solution: make the CNN also do region proposals!

Insert a Region Proposal Network (RPN) after last conv layer

RPN produces region proposals (one shot) -> no need for external proposals

After RPN, region of interest pooling, and use similar classifier and bbox regressor like Fast R-CNN

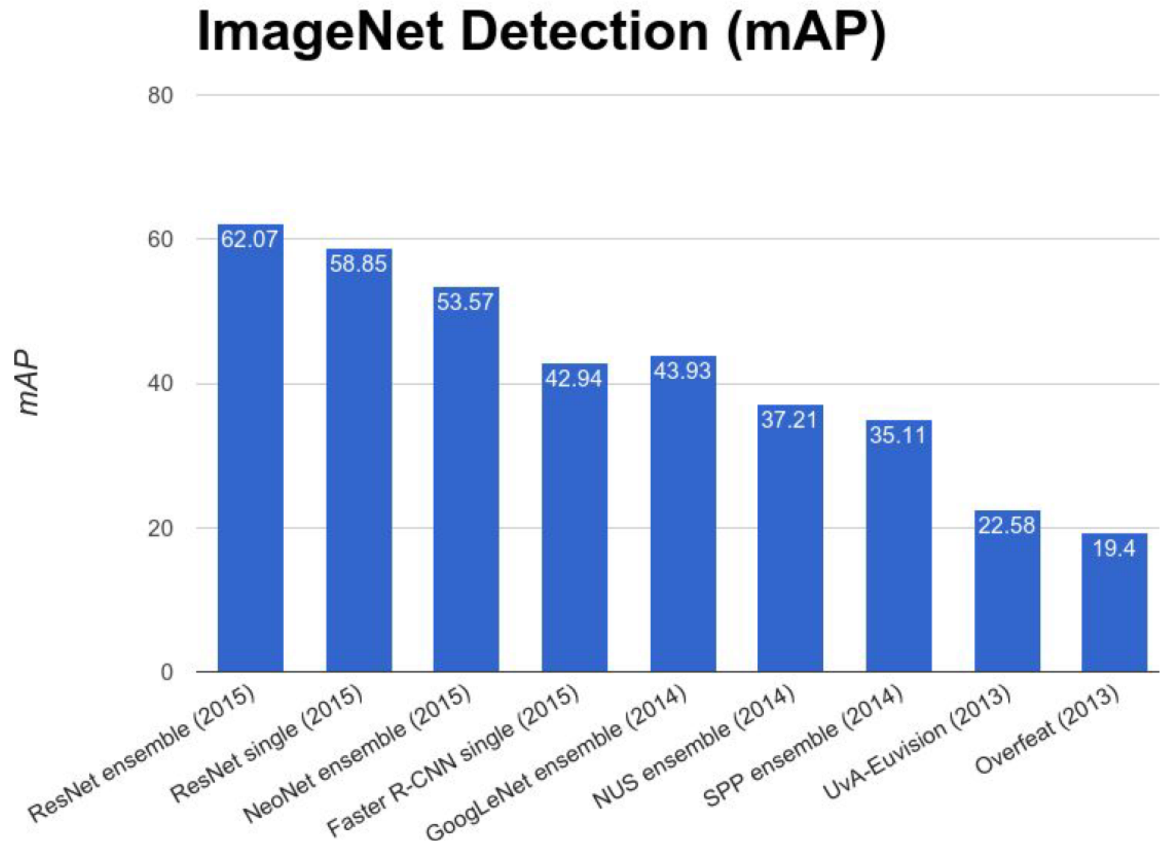
[Girshick 15, Faster R-CNN]

Faster R-CNN

	R-CNN	**Fast R-CNN**	**Faster R-CNN**
Test time per image (with proposals)	50 seconds	2 seconds	**0.2 seconds**
(Speedup)	1x	25x	**250x**
mAP (VOC 2007)	66.0	**66.9**	**66.9**

Credit: Li/Karpathy/Johnson

ImageNet Detection 2013 - 2015



Credit: Li/Karpathy/Johnson

Image Segmentation and Instance Segmentation

Using CNNs in Computer Vision

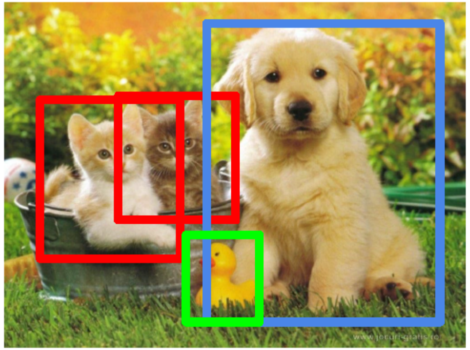
Classification

CIFAR 10 +
"raw" CNN ☺

Classification + Localization

Regression and/or
sliding window

Object Detection



Selective Search, (D)RP
(Fast(er)) R-CNN

Instance Segmentation

Credit: Li/Karpathy/Johnson

Using CNNs in Computer Vision

Classification

CIFAR 10 +
"raw" CNN ☺

Classification + Localization

Regression and/or
sliding window

Object Detection

Selective Search, RP
(Fast(er)) R-CNN

Instance Segmentation

Credit: Li/Karpathy/Johnson

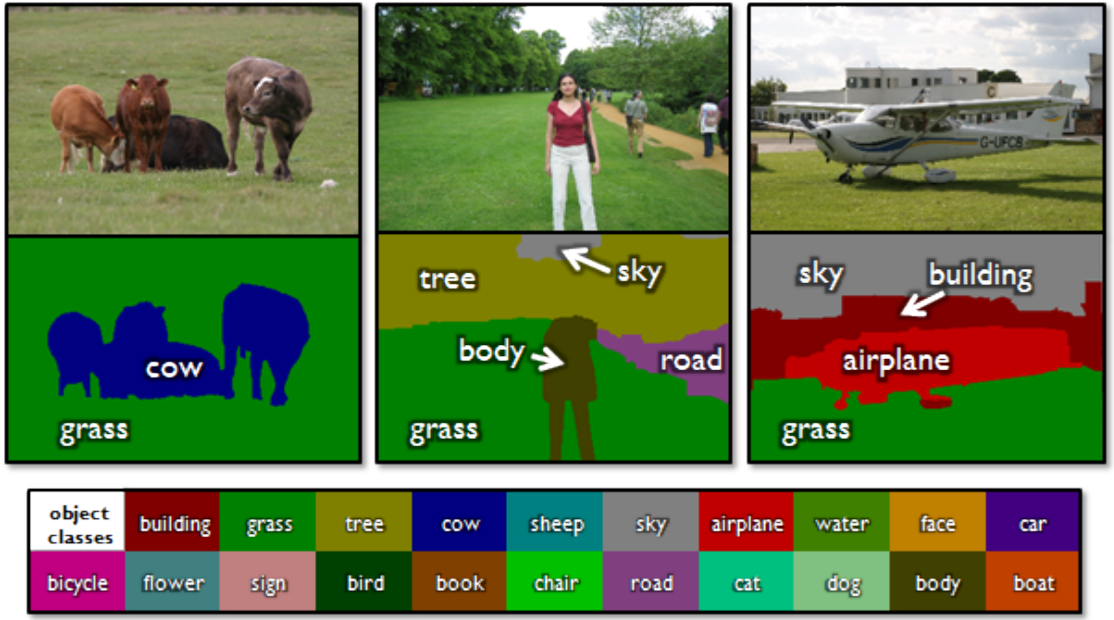
Semantic Segmentation

Predict class label for every pixel (i.e., dense pixel labeling)

No differentiation between instances

I.e., all objects of the same class receive same class label

Traditional computer vision task



[Shotton et al. 07] TextonBoost

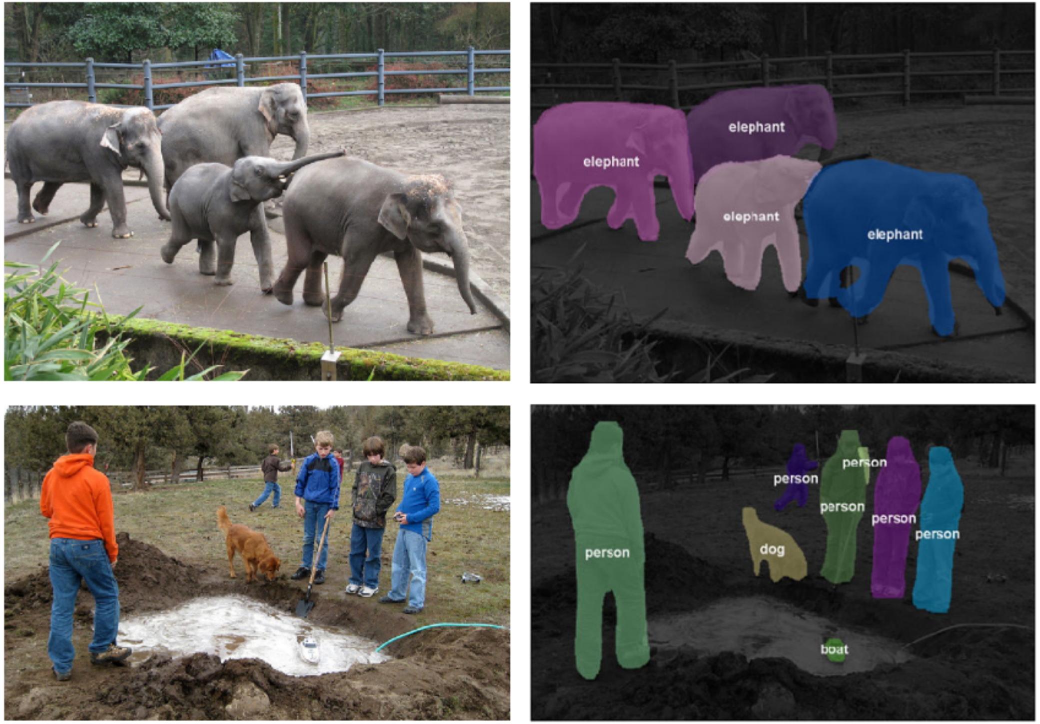
Instance Segmentation

Detect instances, classify category, label pixels of each instance;

Distinguish between instances within a category;
e.g., elephant1, elephant2, etc.

Simultaneous detection and segmentation (SDS)

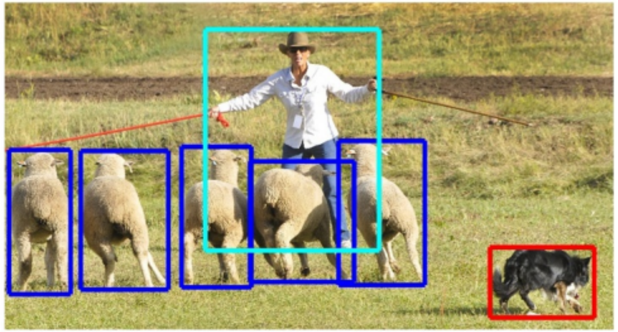
MS COCO is core dataset
-> lots of work around it



[Dai et al. 15] Instance-aware Semantic Segmentation

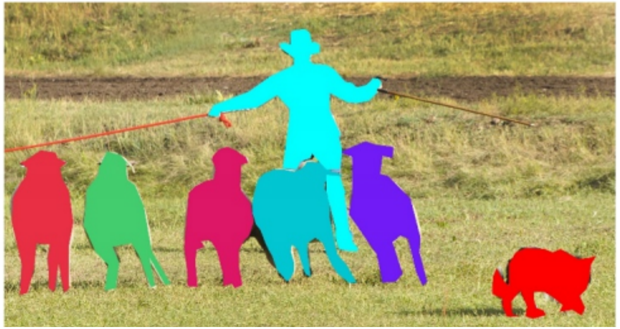
Semantic vs Instance Segmentation

(a) Image classification



(b) Object localization

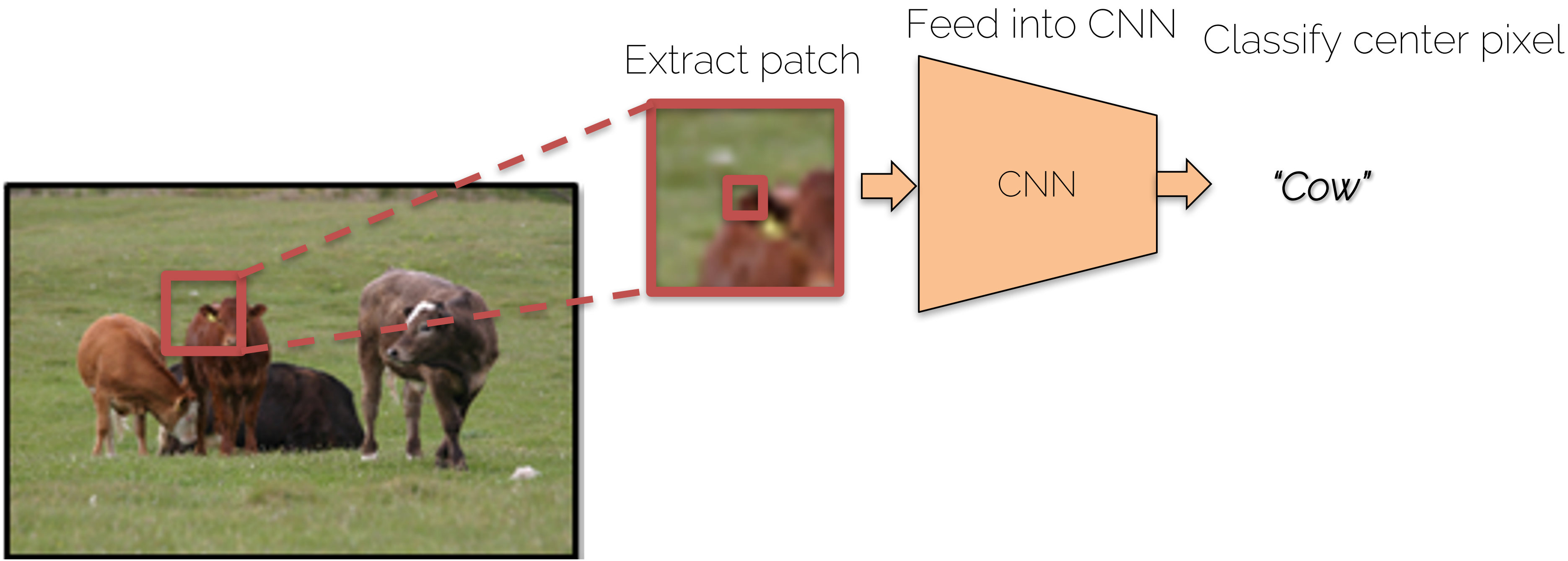
(c) Semantic segmentation



(d) Instance segmentation

[Lin et al. 15] Microsoft COCO: Common Objects in Context

Semantic Segmentation (Patch-based)

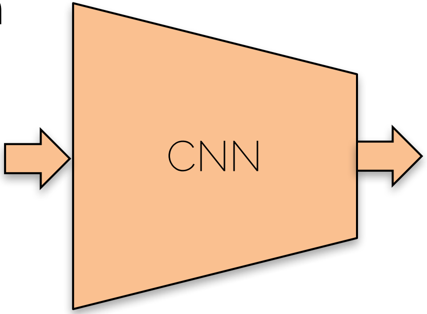


Semantic Segmentation (Patch-based)

Extract patch

Feed into CNN

Classify center pixel



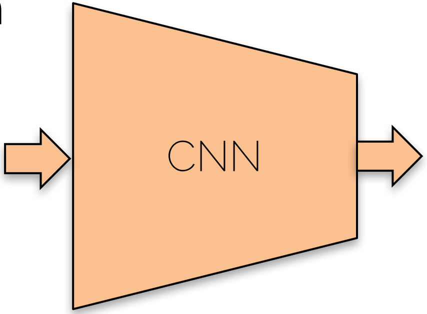
Run CNN for every pixel!

Semantic Segmentation (Patch-based)

Extract patch

Feed into CNN

Classify center pixel

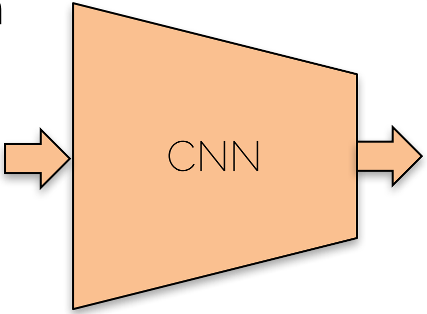


Run CNN for every pixel!

Semantic Segmentation (Patch-based)

Extract patch

Feed into CNN



Classify center pixel

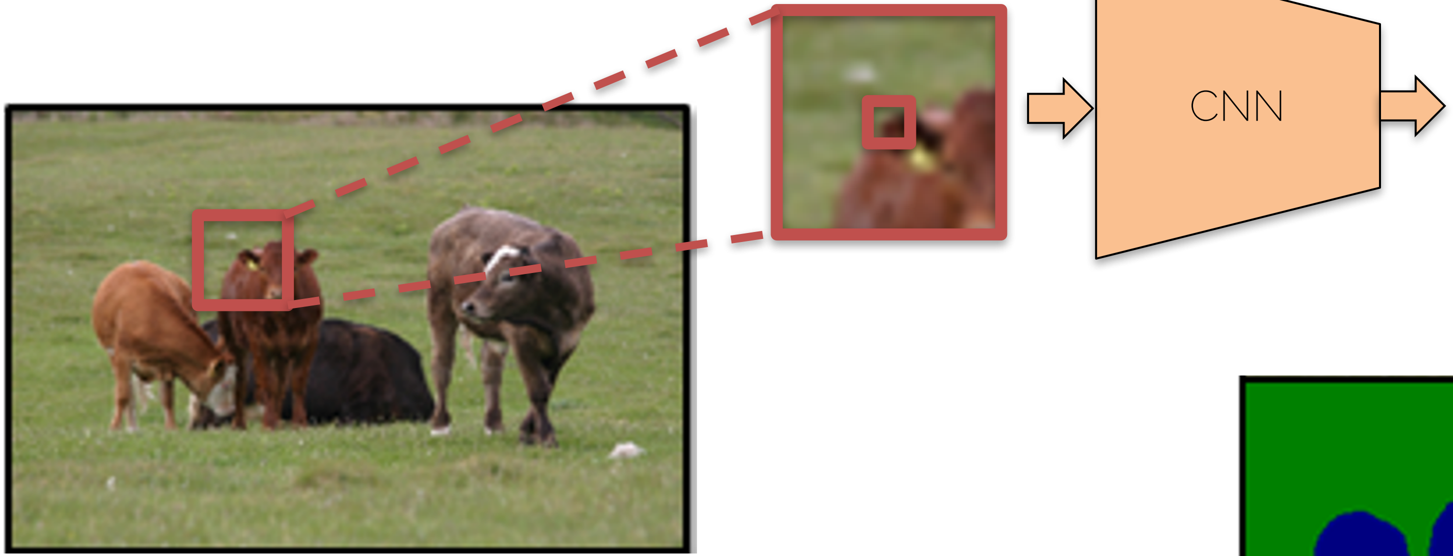
Run CNN for every pixel!

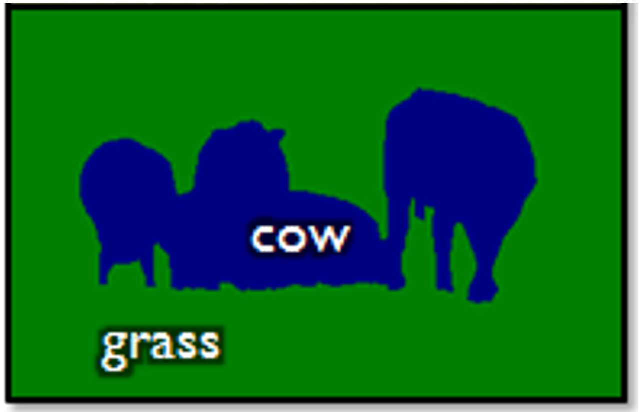
Semantic Segmentation (Patch-based)

Extract patch

Feed into CNN

Classify center pixel





Run CNN for every pixel!

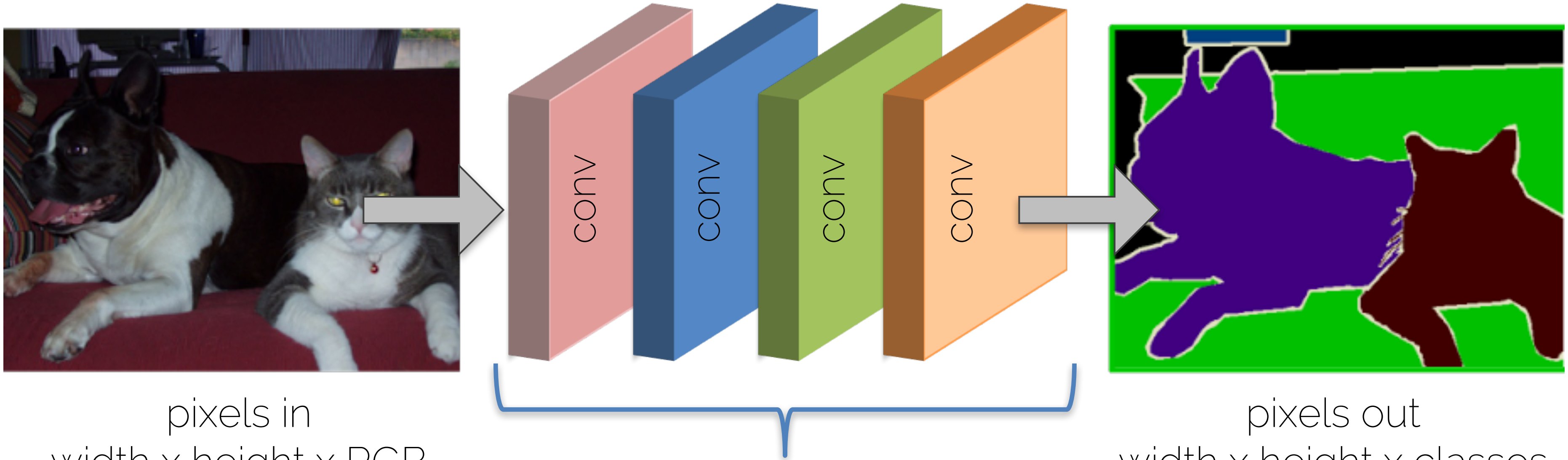
Possibly run a CRF at the end

Semantic Segmentation (Patch-based)

- Extract patch from image for every pixel
- Run every patch independently through a CNN

- Easy architecture: just classify -- use VGG/ResNet
- Easy to train: just use pixel center label for patch
- Expensive at test time

Semantic Segmentation



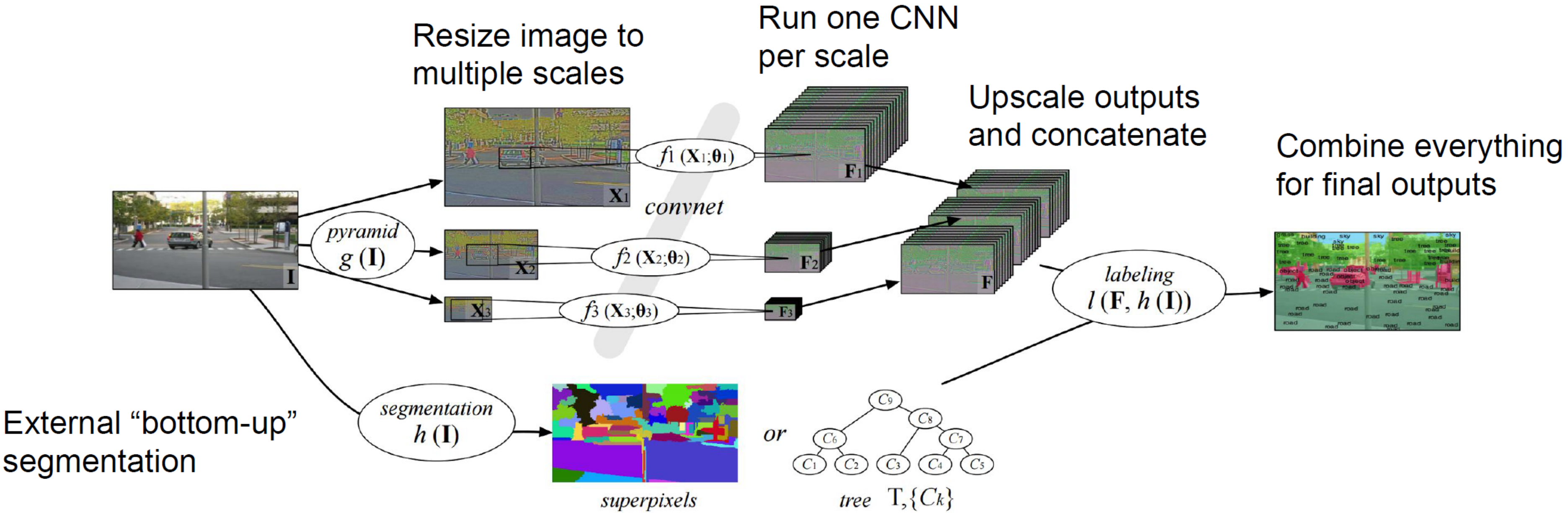
pixels in
width x height x RGB

Just convs & activations

pixels out
width x height x classes

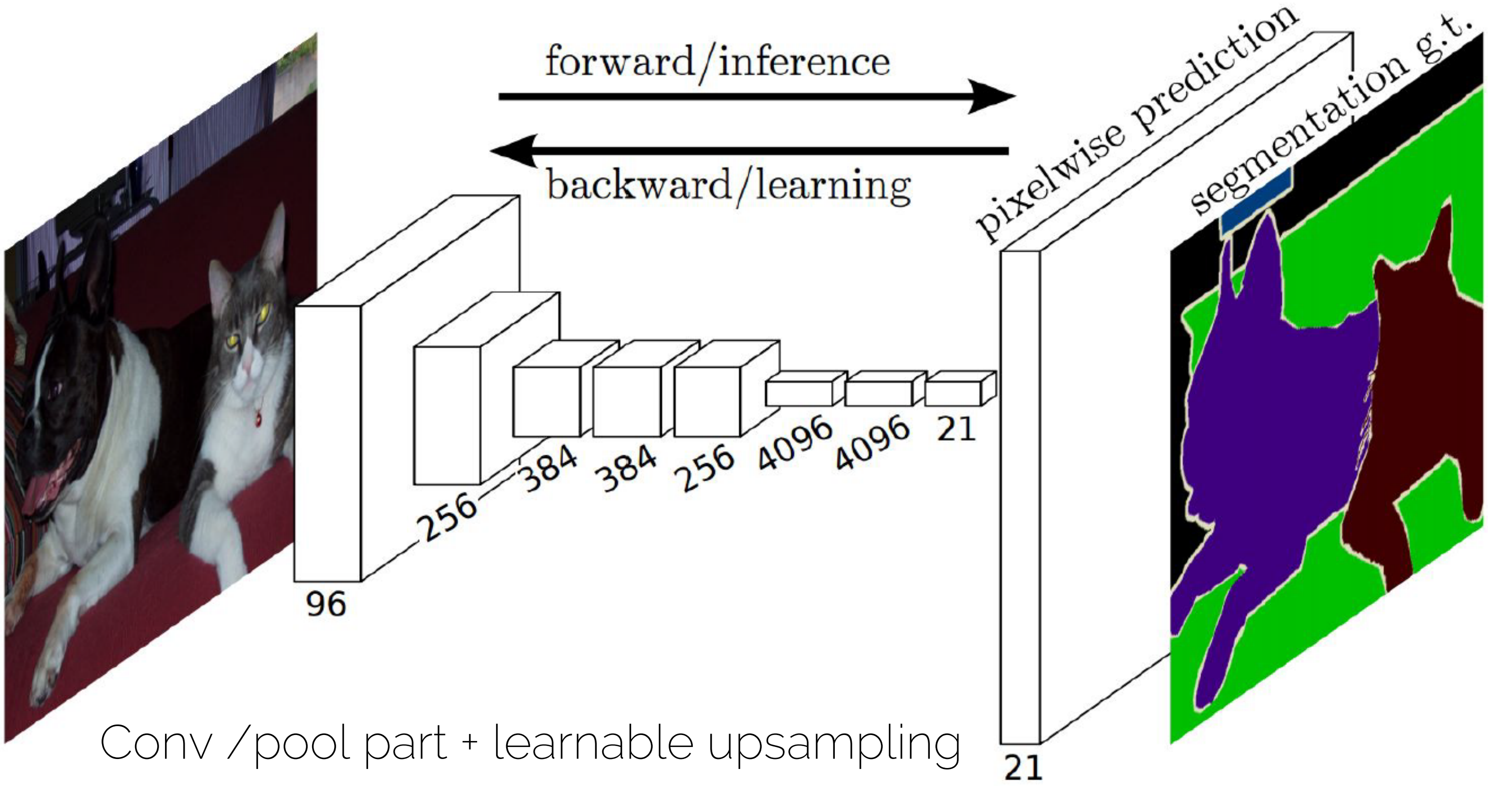
Fully Convolutional Network

Semantic Segmentation (Multi-Scale)



[Farabet et al. 13] Learning Hierarchical Features of Scene Labeling (Slide by Li/Karpathy/Johnson)

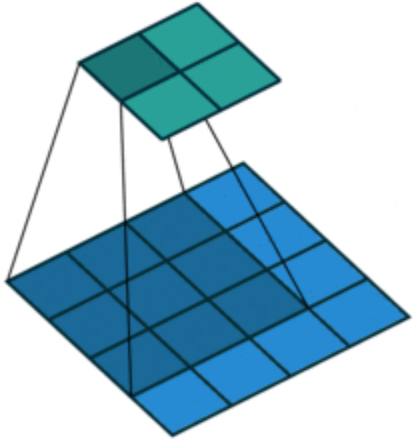
Semantic Segmentation (FCN)



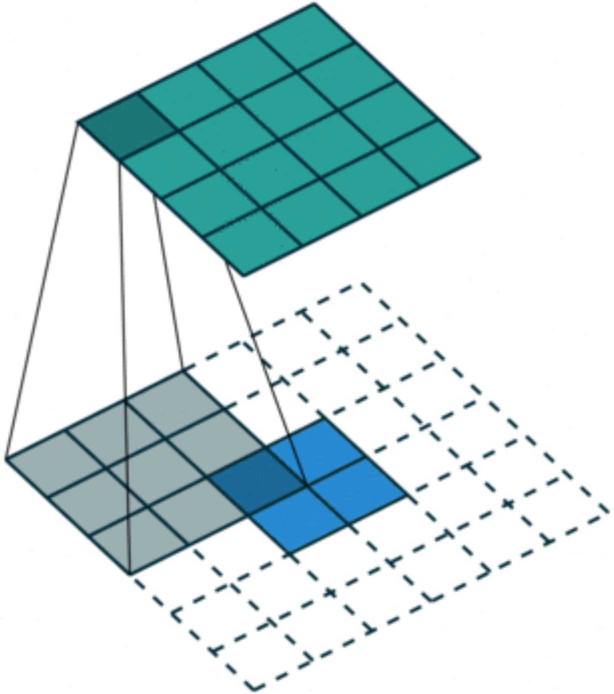
Conv /pool part + learnable upsampling

[Long et al. 15] Fully Convolutional Networks for Semantic Segmetnation (FCN)

Learnable Upsampling: Deconvolution



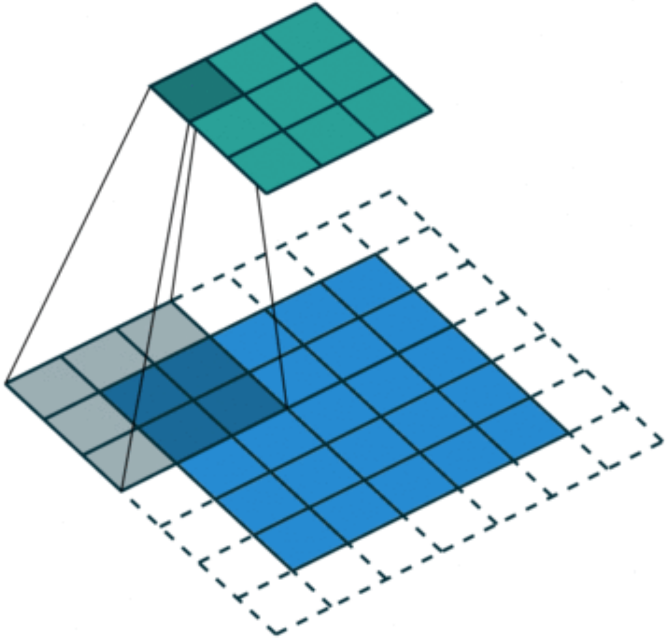
Convolution
no padding, no stride



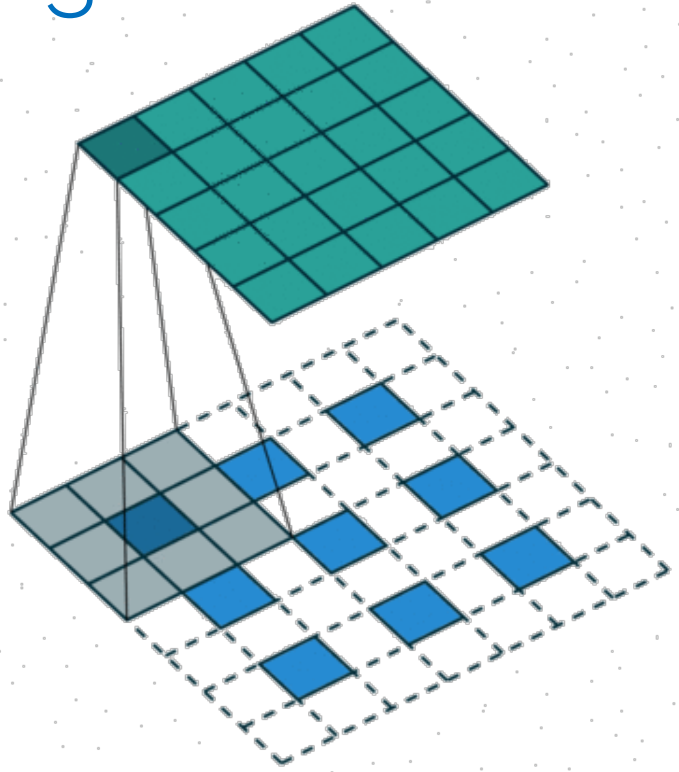
Transposed convolution
no padding, no stride

https://github.com/vdumoulin/conv_arithmetic

Learnable Upsampling: Deconvolution



Convolution
padding, stride



Transposed convolution
padding, stride

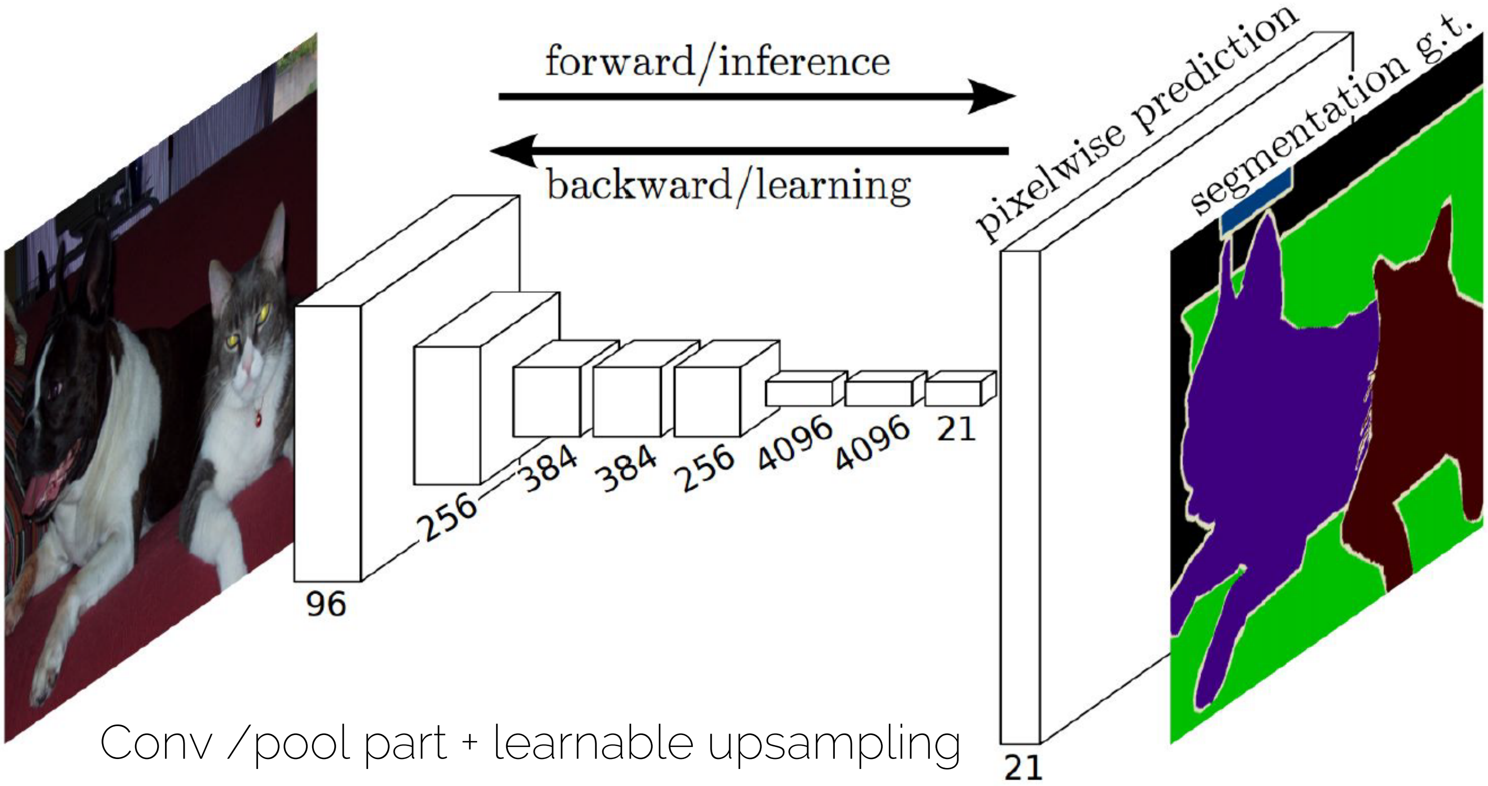
https://github.com/vdumoulin/conv_arithmetic

Learnable Upsampling: Deconvolution

- "Deconvolution" is not a great name, but widely used

- Also named:
  - Upconvolution
  - Convolution transpose
  - Backward strided convolution
  - ½ strided convolution

Semantic Segmentation (FCN)



Conv /pool part + learnable upsampling

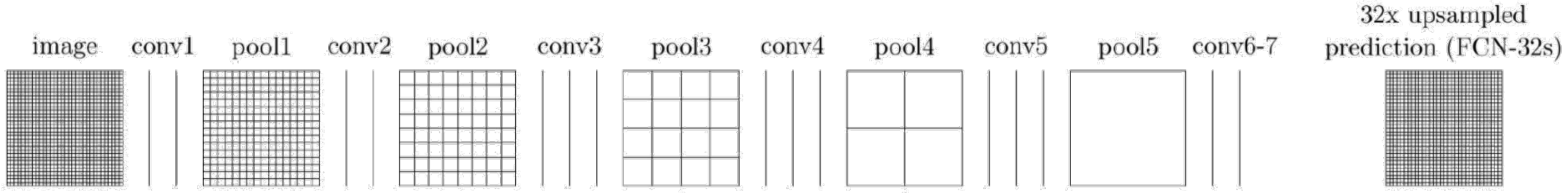
[Long et al. 15] Fully Convolutional Networks for Semantic Segmetnation (FCN)

Semantic Segmentation (FCN)

- Run "fully convolutional" network (FCN)
- Take all pixels at once as input
- Bottle neck + learnable upsampling
- Predict class for every pixel simultaneously

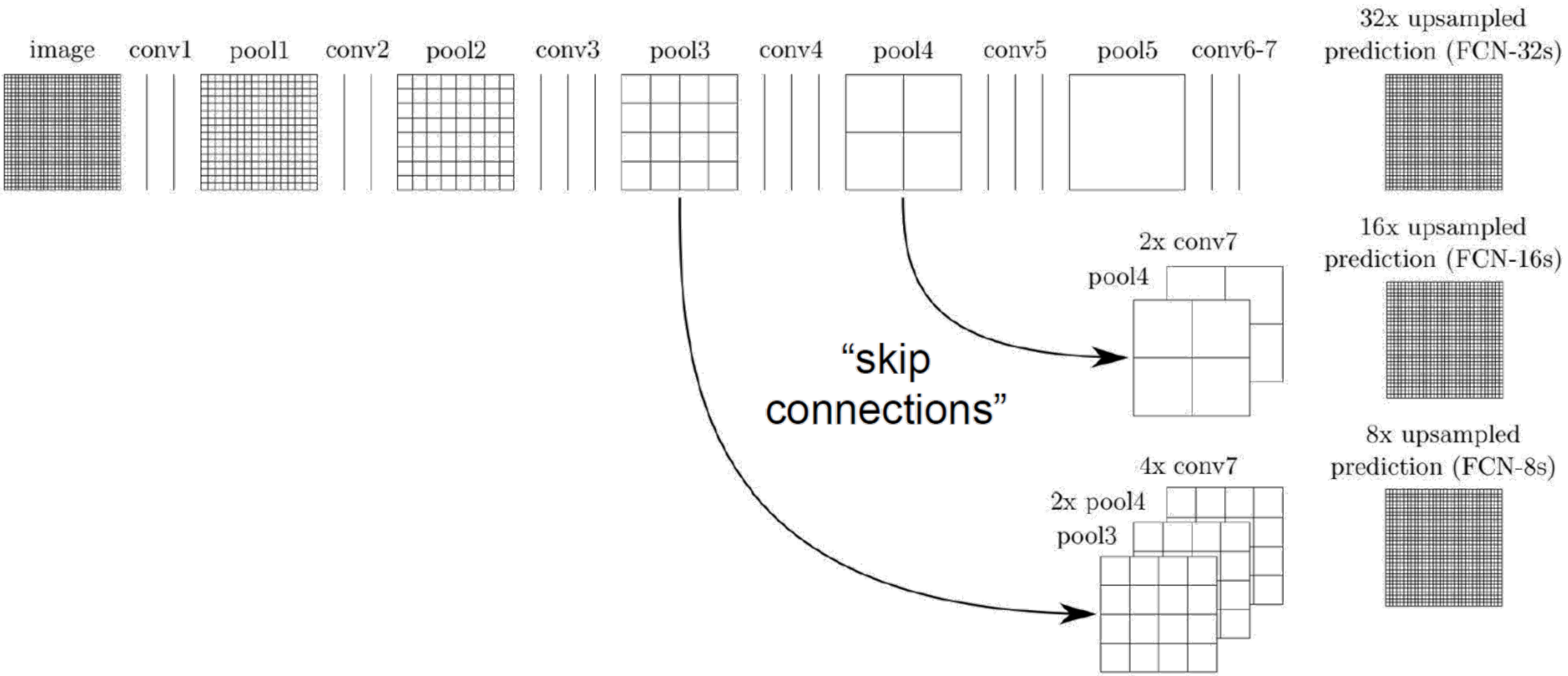
[Long et al. 15] Fully Convolutional Networks for Semantic Segmetnation (FCN)

Semantic Segmentation (FCN)



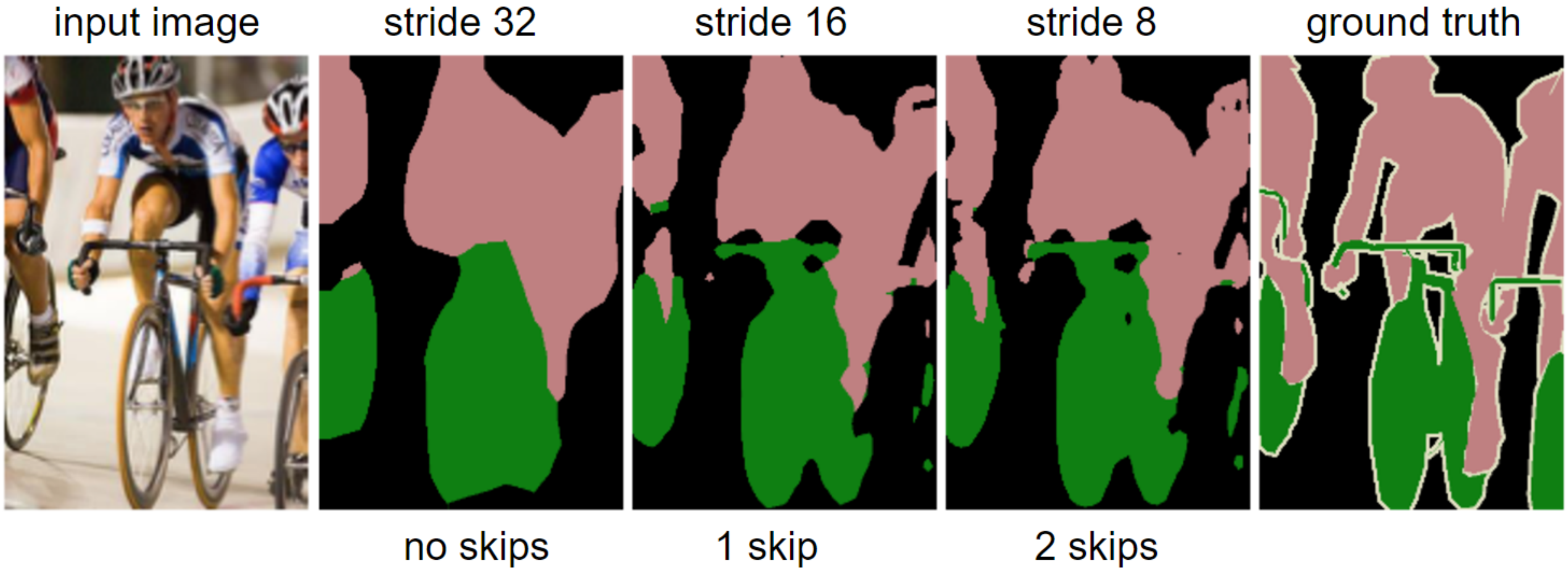
[Long et al. 15] Fully Convolutional Networks for Semantic Segmetnation (FCN)

Semantic Segmentation (FCN)



[Long et al. 15] Fully Convolutional Networks for Semantic Segmetnation (FCN)

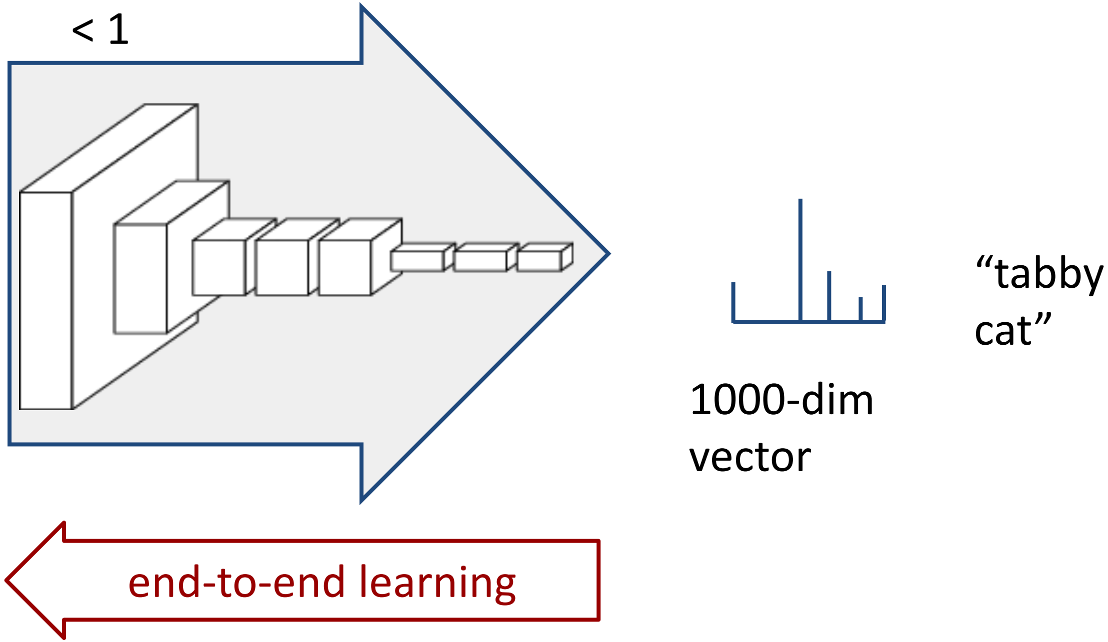
Semantic Segmentation (FCN)



Skip connections -> better results

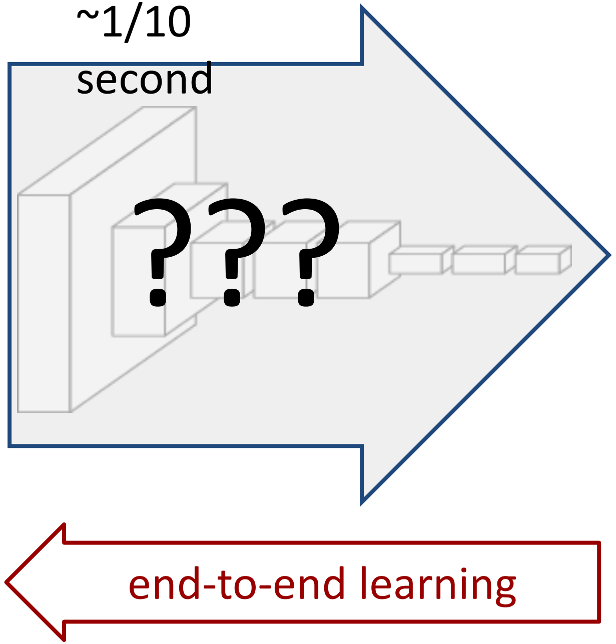
[Long et al. 15] Fully Convolutional Networks for Semantic Segmetnation (FCN)

FCN: Convnets Perform Classification



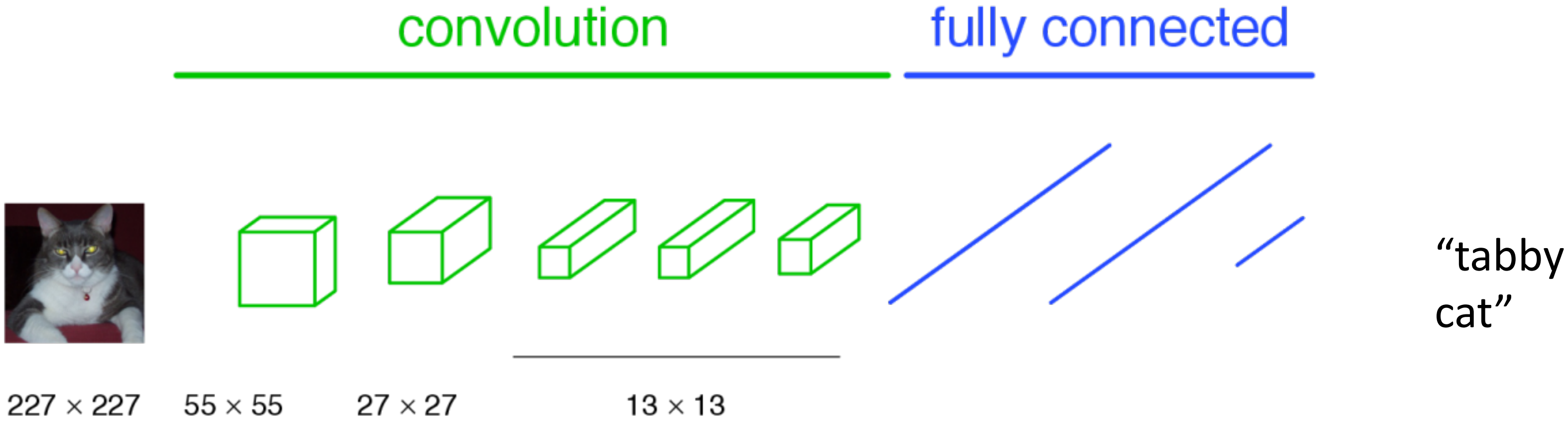
[Long et al. 15] Fully Convolutional Networks for Semantic Segmentation (FCN)

FCN: Lots of pixels, Little Time?



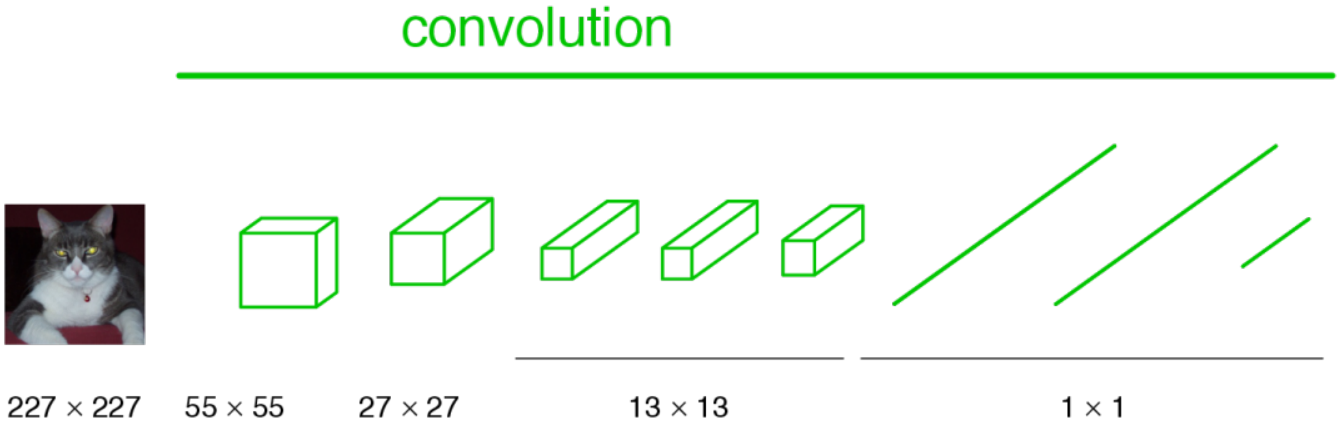
[Long et al. 15] Fully Convolutional Networks for Semantic Segmentation (FCN)

FCN: a Classification Network



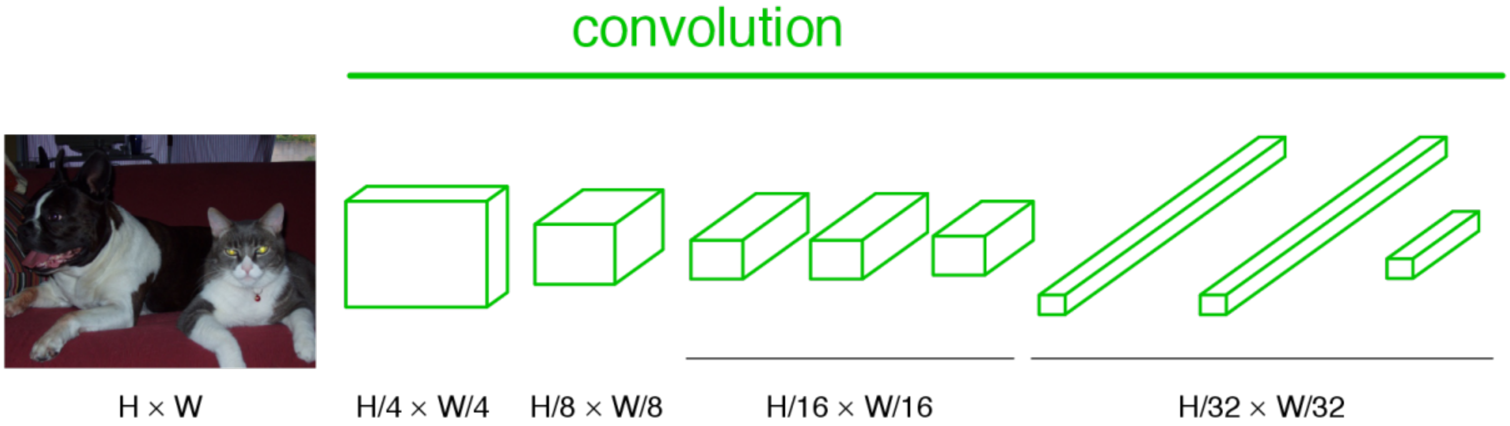
[Long et al. 15] Fully Convolutional Networks for Semantic Segmentation (FCN)

FCN: Becoming Fully Convolutional



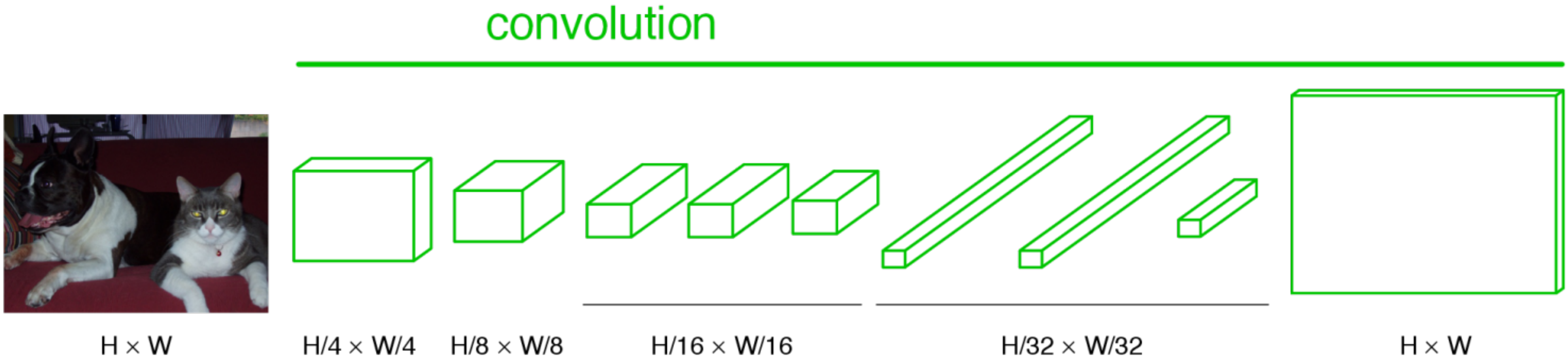
[Long et al. 15] Fully Convolutional Networks for Semantic Segmentation (FCN)

FCN: Becoming Fully Convolutional



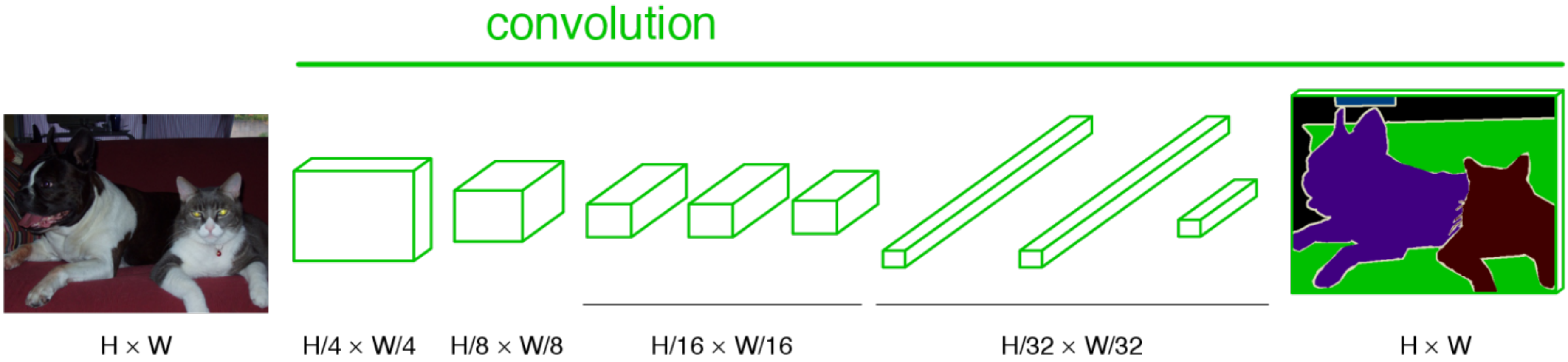
[Long et al. 15] Fully Convolutional Networks for Semantic Segmentation (FCN)

FCN: Upsampling Output



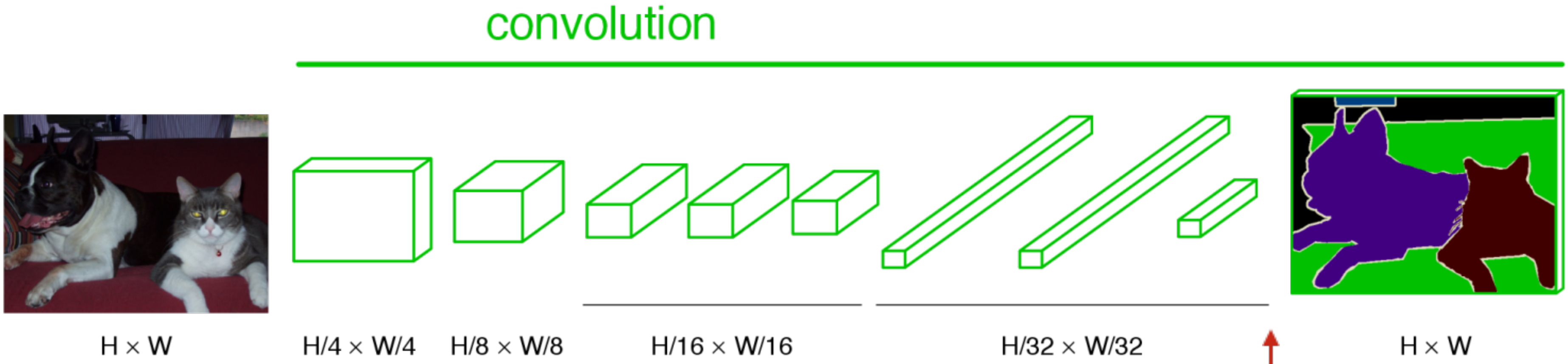
[Long et al. 15] Fully Convolutional Networks for Semantic Segmentation (FCN)

FCN: End-to-end, Pixels-to-pixels Network



[Long et al. 15] Fully Convolutional Networks for Semantic Segmentation (FCN)

FCN: End-to-end, Pixels-to-pixels Network



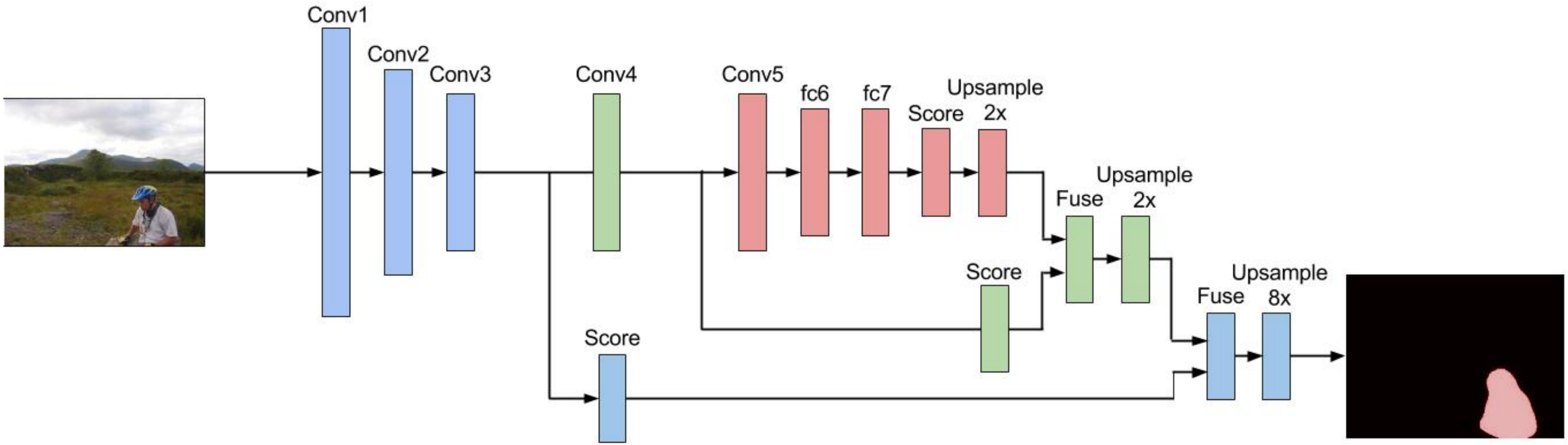
conv, pool, nonlineari ty

upsampli ng

pixelwise output + loss

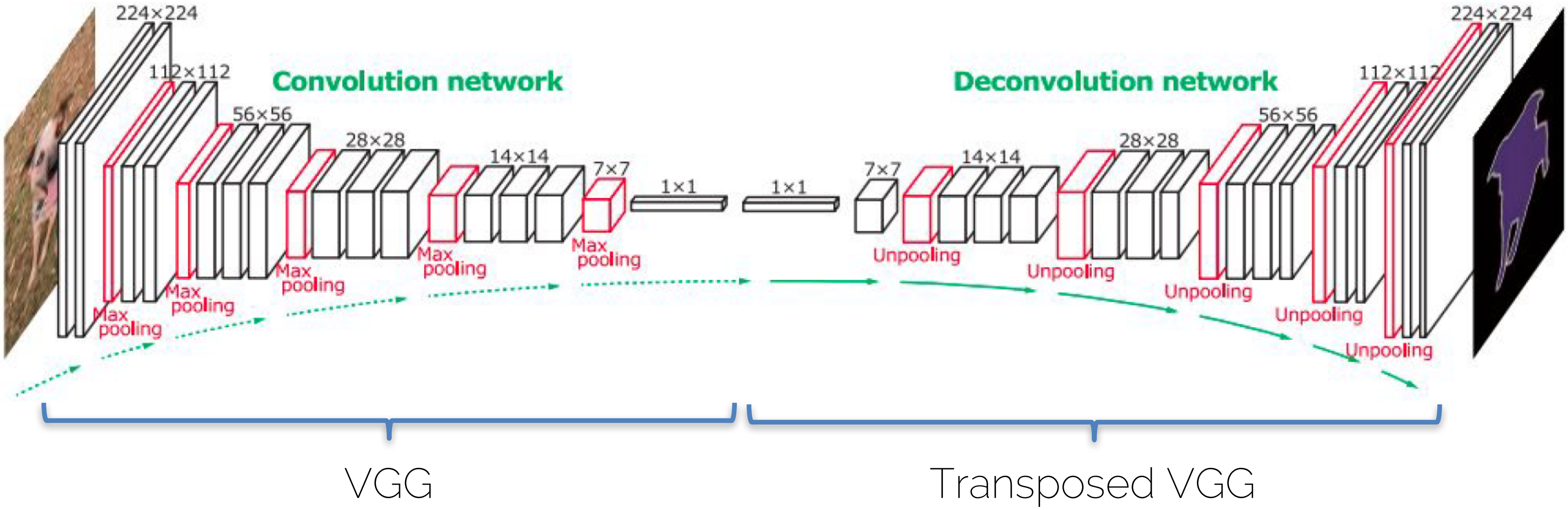
[Long et al. 15] Fully Convolutional Networks for Semantic Segmentation (FCN)

FCN: Architecture



[Long et al. 15] Fully Convolutional Networks for Semantic Segmentation (FCN)

Semantic Segmentation: Upsampling



[Noh et al. 15] Learning Deconvolution Network for Semantic Segmentation

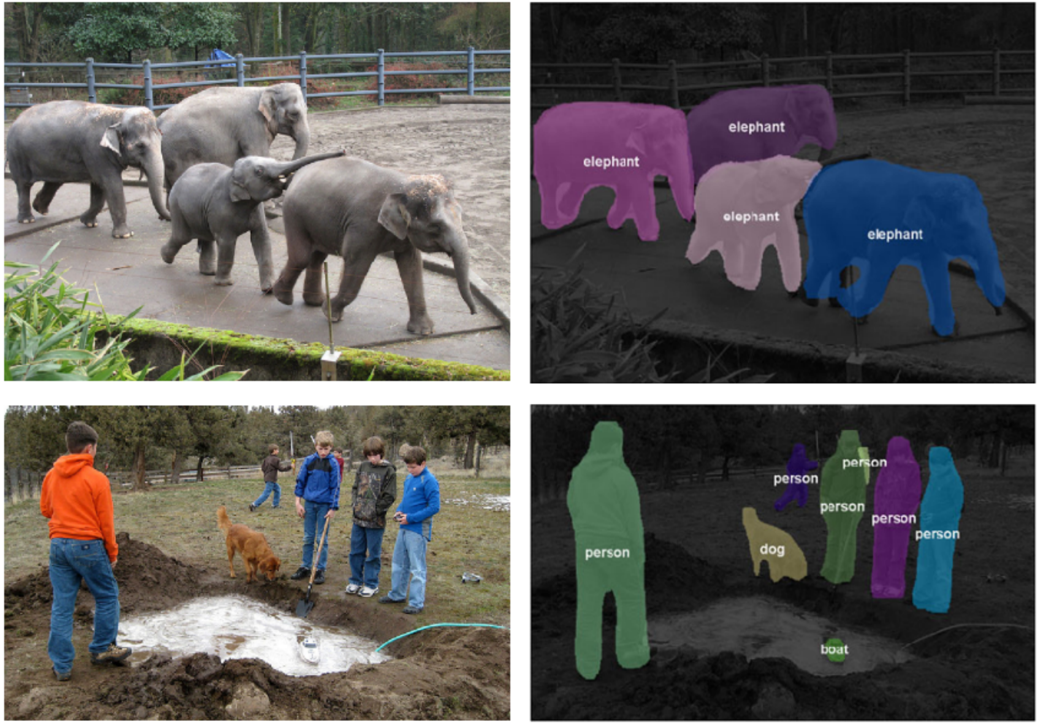
Instance Segmentation

Detect instances, classify category, label pixels of each instance;

Distinguish between instances within a category;
e.g., elephant1, elephant2, etc.

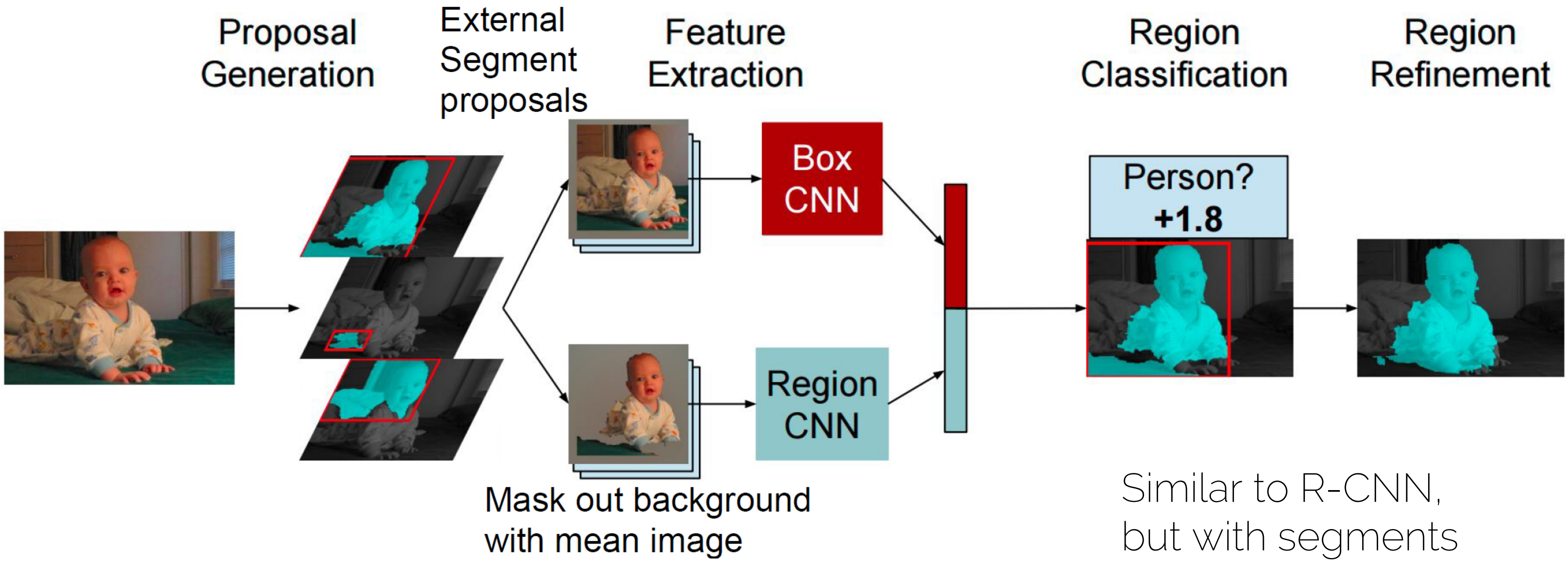
Simultaneous detection and segmentation (SDS)

MS COCO is core dataset
-> lots of work around it



[Dai et al. 15] Instance-aware Semantic Segmentation

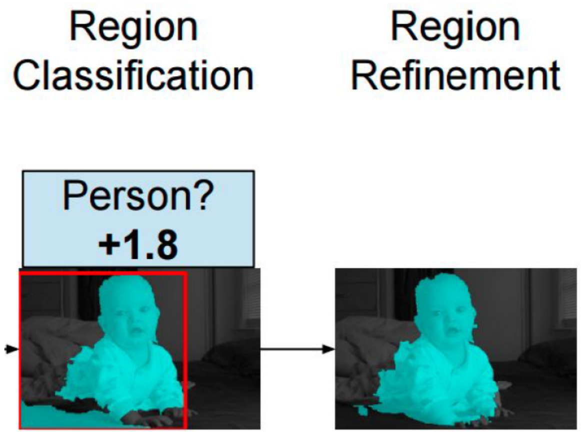
Instance Segmentation

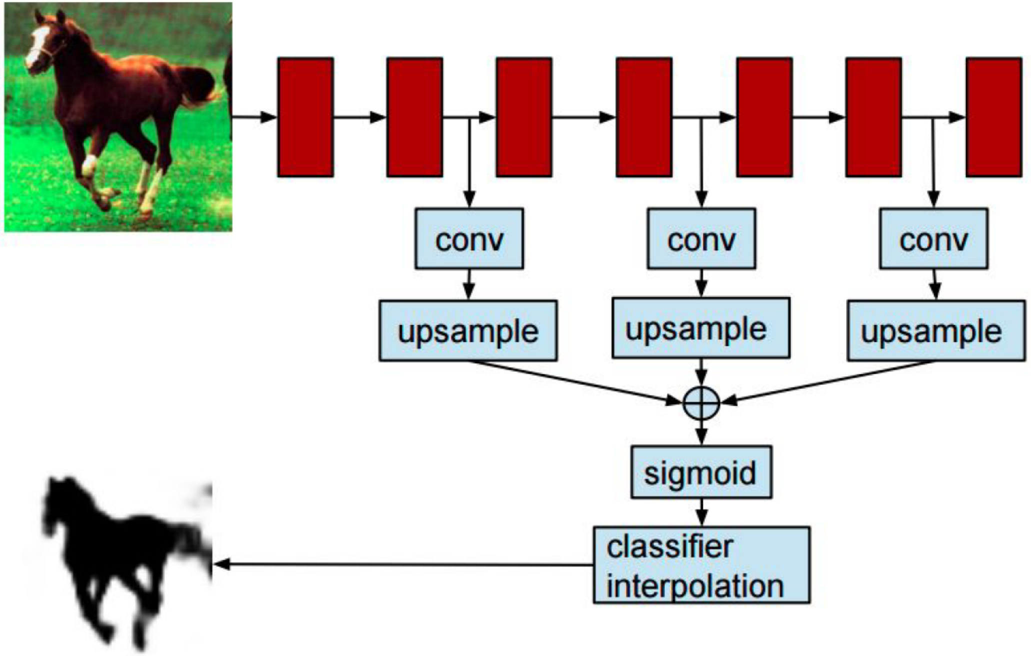


Similar to R-CNN, but with segments

[Hariharan et al. 14] Simultaneous Detection and Segmentation (Slide by Li/Karpathy/Johnson)

Instance Segmentation: Hypercolumns



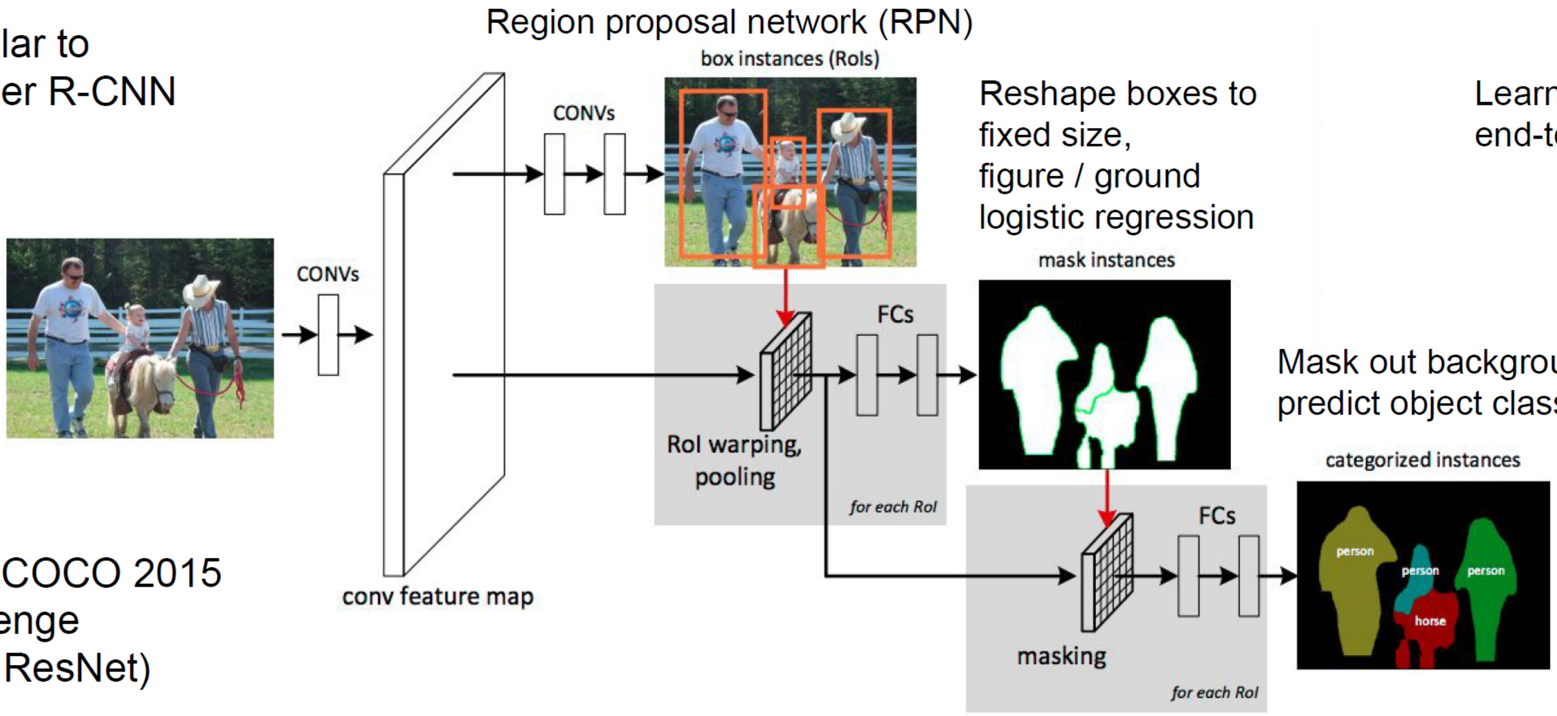


[Hariharan et al. 15] Hypercolumns for Object Segmentation and Fine-grained Localization (Slide by Li/Karpathy/Johnson)

Instance Segmentation: Cascades

Similar to Faster R-CNN

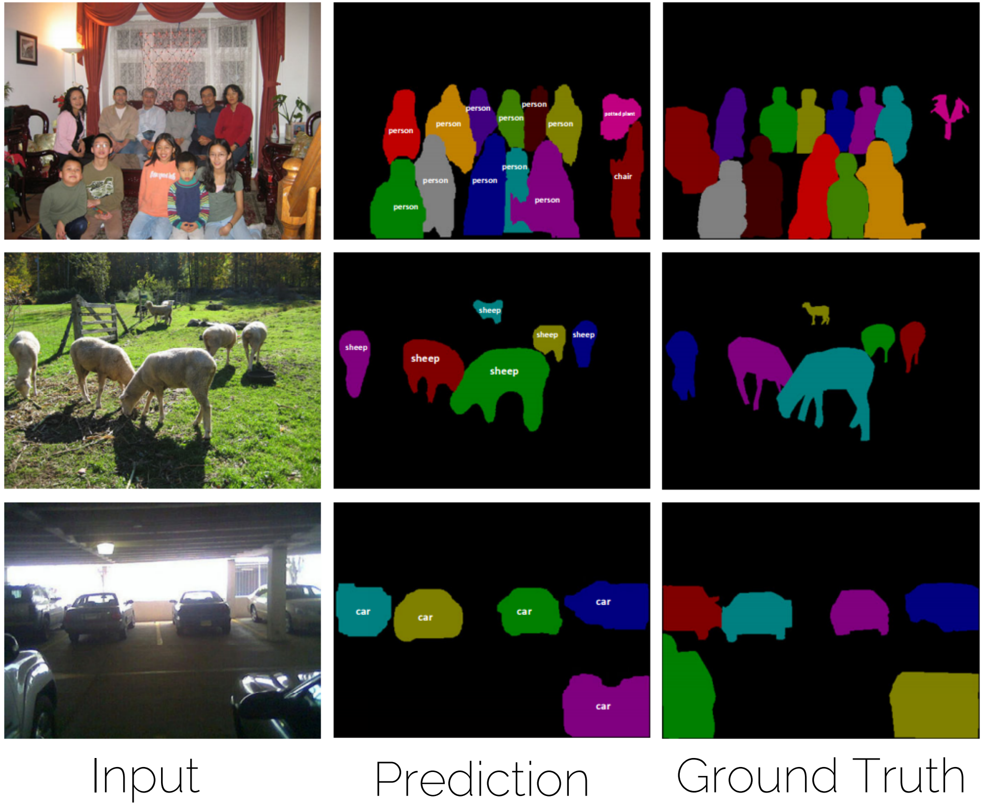
Won COCO 2015 challenge (with ResNet)



Learn entire model end-to-end!

[Dai et al. 15] Instance-aware Semantic Segmentation via Multi-task Network Cascades (Slide by Li/Karpathy/Johnson)

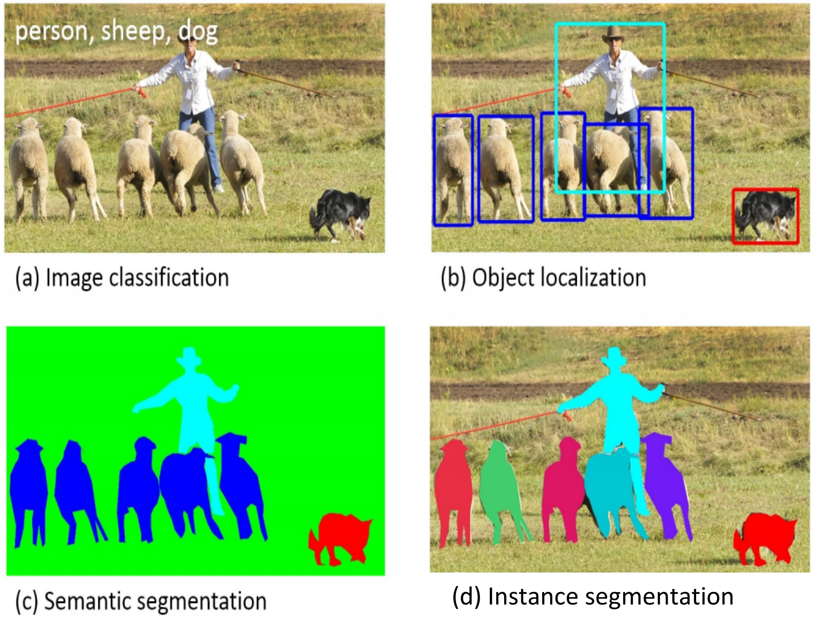
Instance Segmentation: Cascades



[Dai et al. 15] Instance-aware Semantic Segmentation via Multi-task Network Cascades (Slide by Li/Karpathy/Johnson)

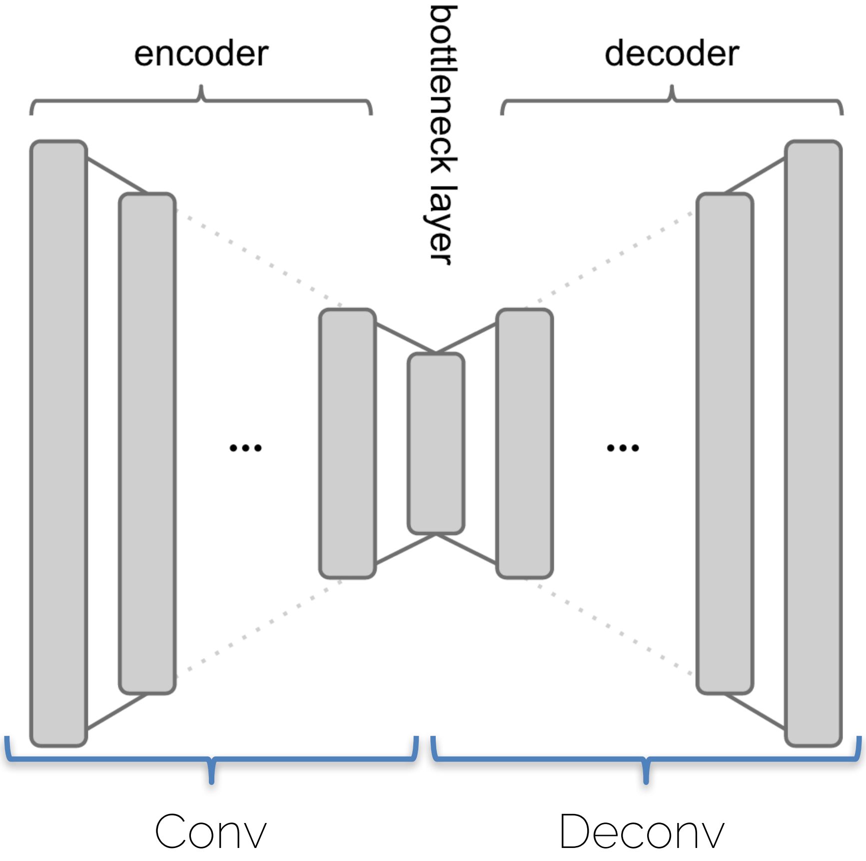
Segmentation Overview

- Semantic segmentation
  - Classify all pixels
  - Fully convolutional models, downsample, then upsample
  - Learnable upsampling (deconvolution)
  - Skip connection can help (more later)
- Instance segmentation
  - Detect instance, generate mask
  - Similar pipelines to object detection

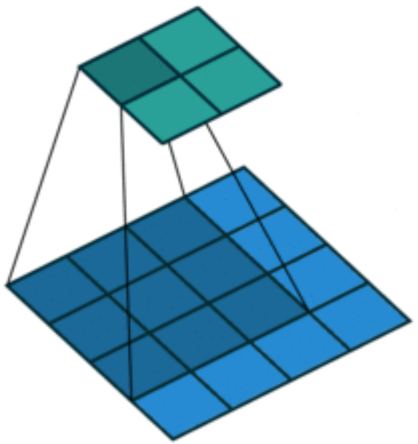


(a) Image classification (b) Object localization (c) Semantic segmentation (d) Instance segmentation

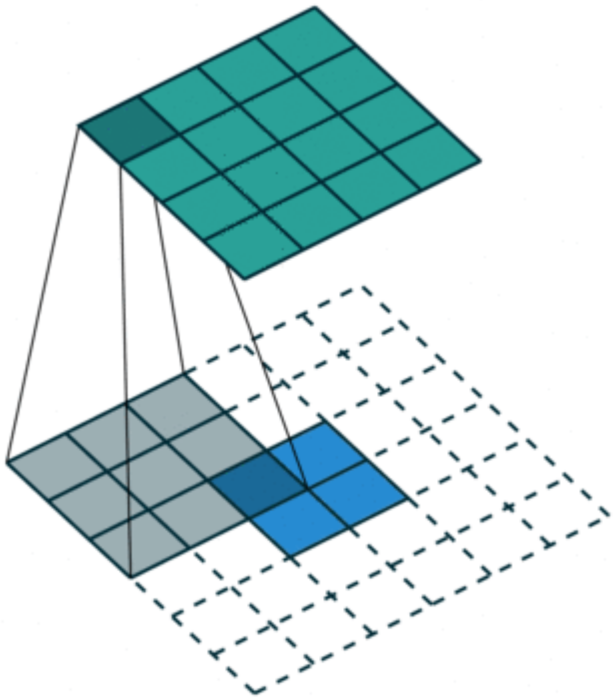
Autoencoder



Remember: Deconvolution



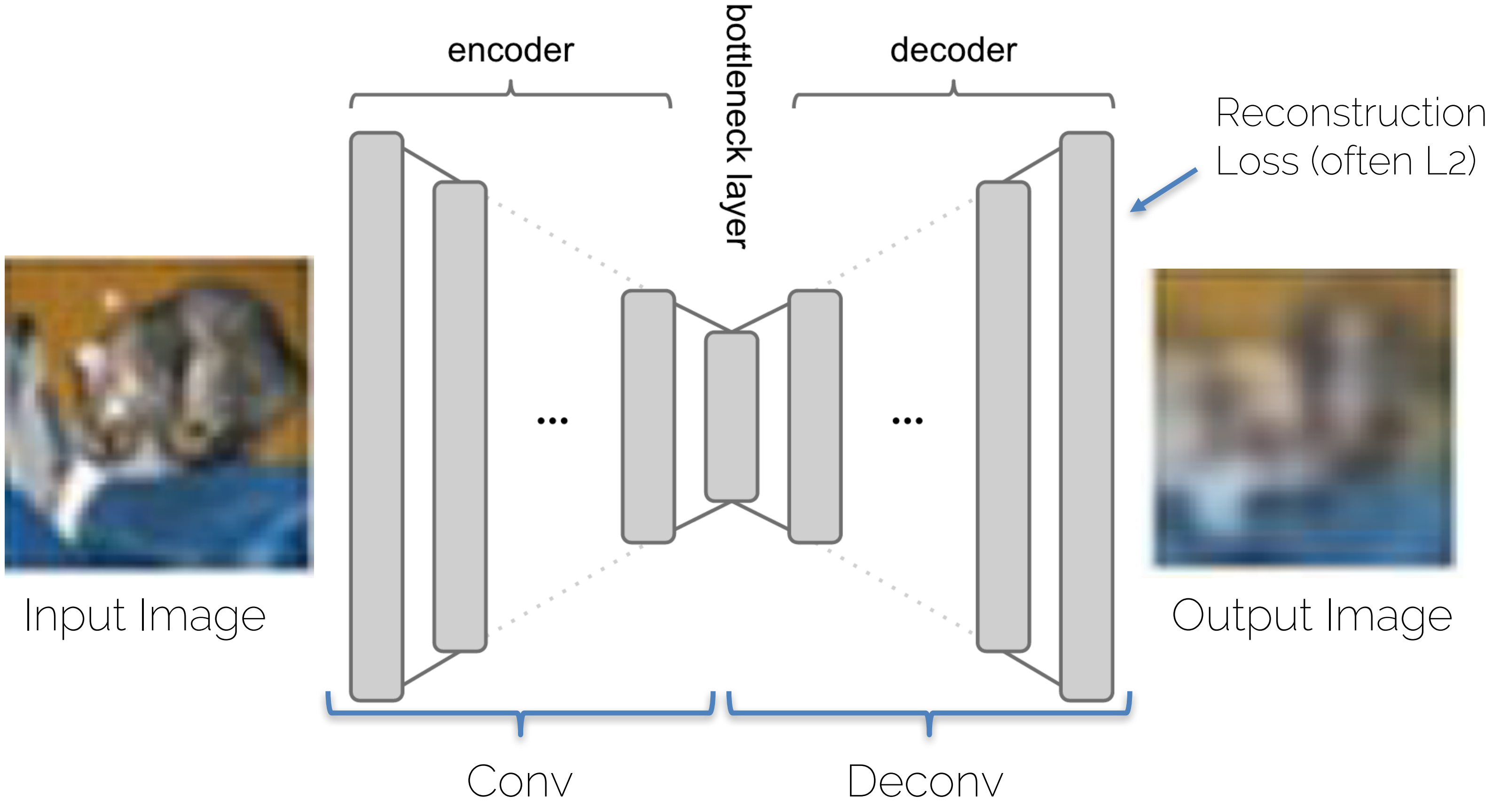
Convolution
no padding, no stride



Transposed convolution
no padding, no stride

https://github.com/vdumoulin/conv_arithmetic

Reconstruction: Autoencoder

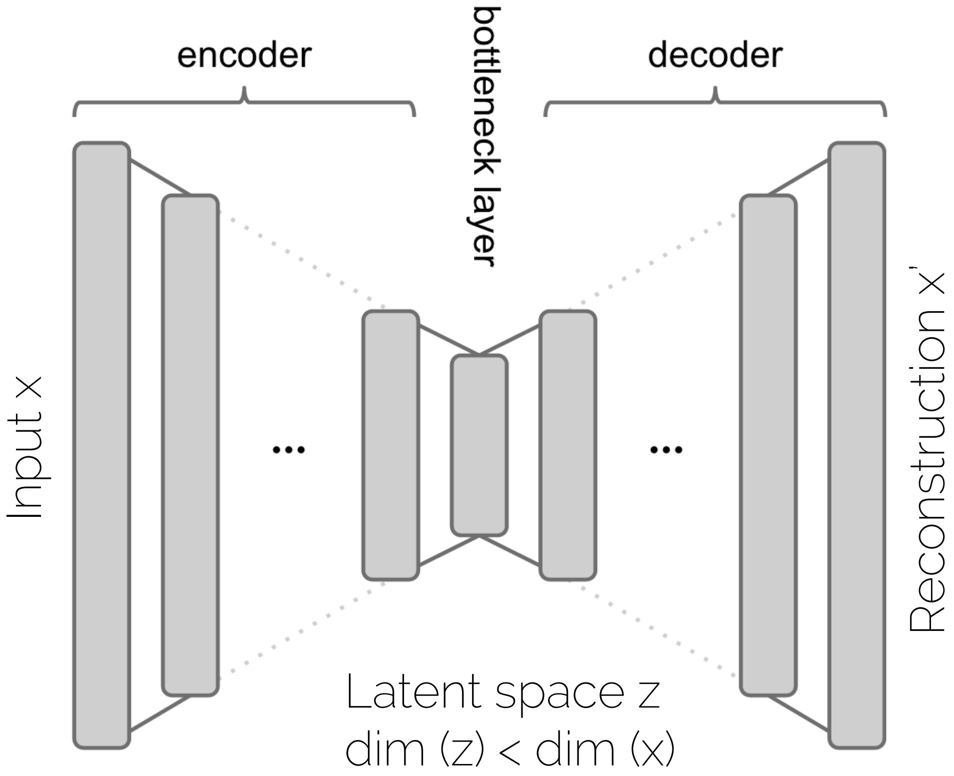


Training Classifiers vs Autoencoders

- Supervised Learning
  - Data (x, y)
    x is data, y is label
  - Goal: learn mapping x -> y
  - Example: classifier

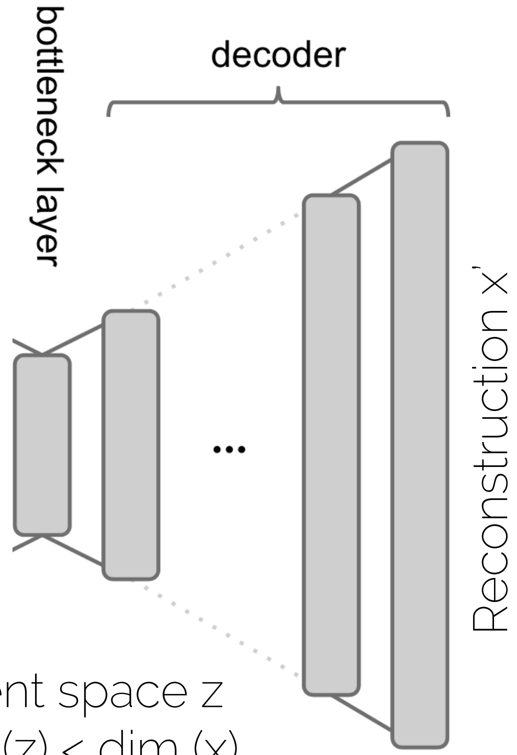
- Unsupervised Learning
  - Data (x)
    only data, no labels
  - Goal: learn structure (e.g., clustering)
  - Example: AE (autoencoder)

Training Autoencoders



Testing Autoencoders

"Test time":
-> reconstruction from
'random' vector

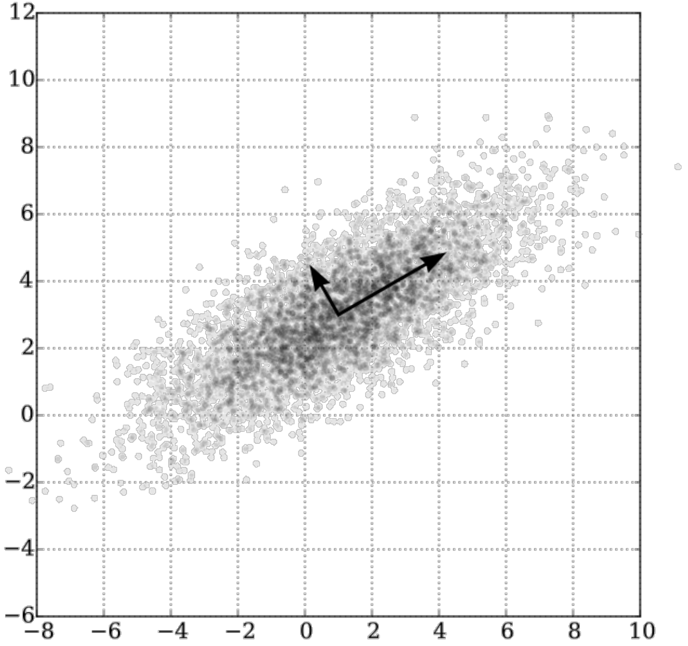


Latent space z
dim (z) < dim (x)

Reconstructed images

Typically pretty blurry… why?

Autoencoder vs PCA

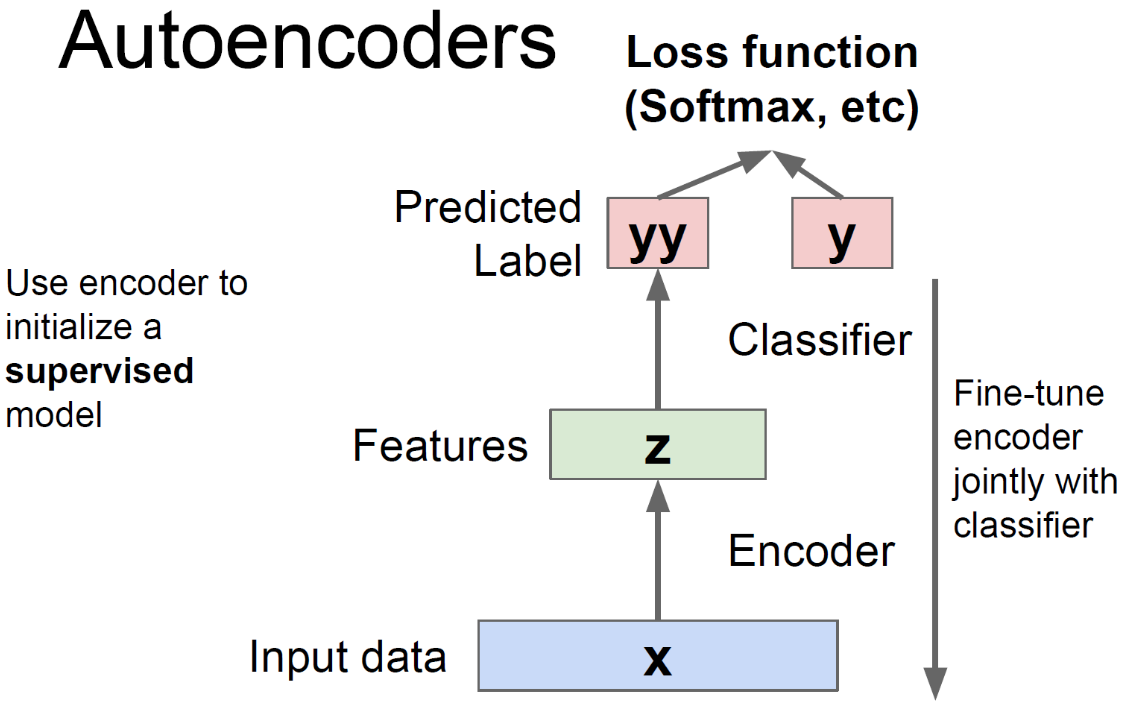


Principal Component Analysis
(low rank approximation)

What is the connection between Autoencoder and PCA?

Autoencoder: Use Cases

- Clustering
- Feature learning
- Embeddings

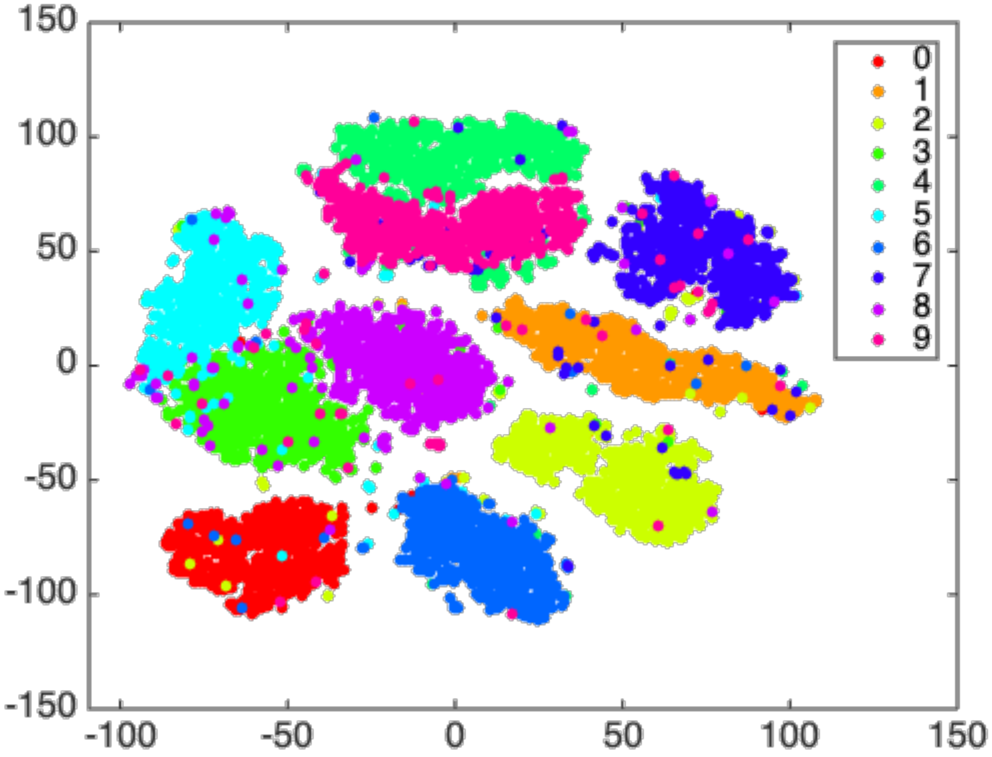


Pre-train AE -> fine-tune with small labeled data

figure by Li/Karpathy/Johnson

Autoencoder: Use Cases

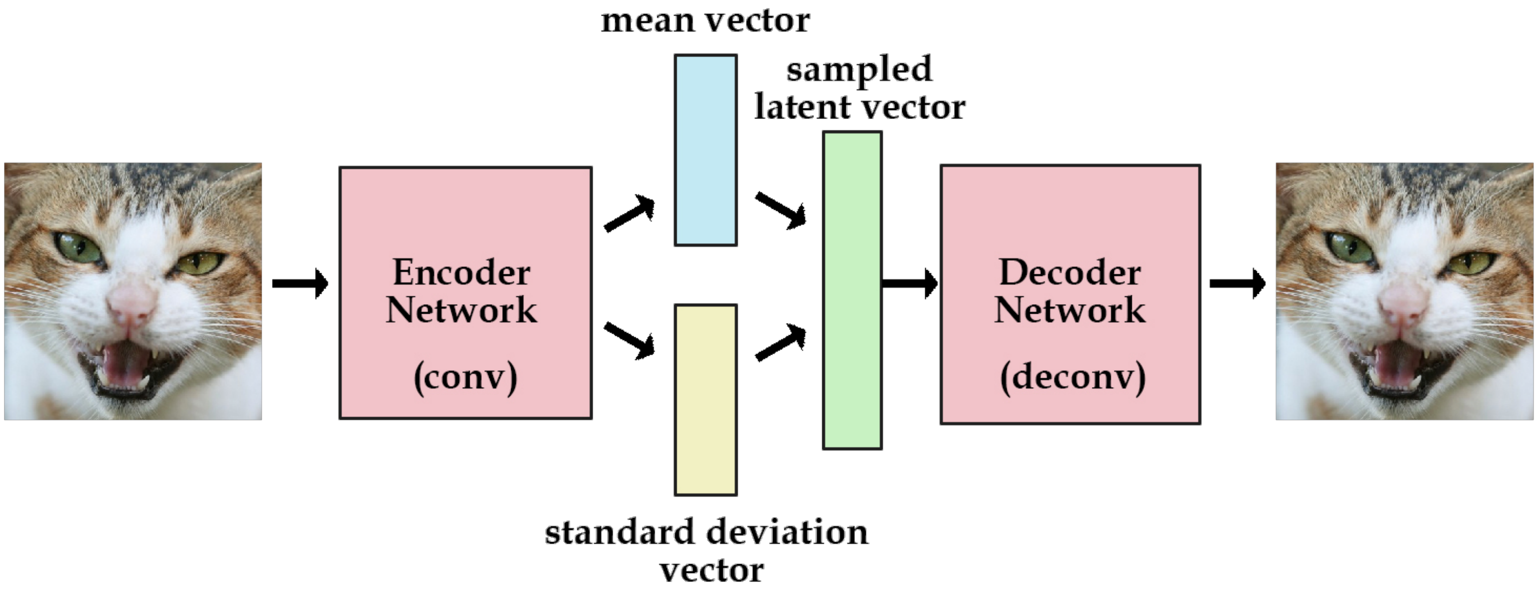
Embedding of
MNIST numbers



Autoencoder: Use Cases

3D shape embedding

Variational Autoencoders (VAE)



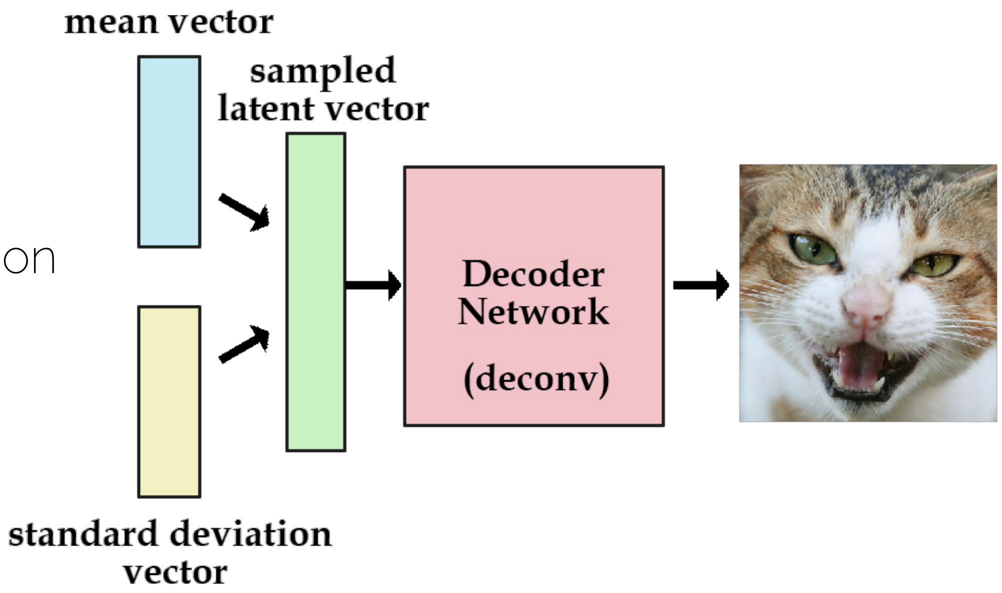
KL-Div Loss in latent space, forcing a unit Gaussian distribution
-> now the latent vector becomes a distribution

http://kvfrans.com/variational-autoencoders-explained/

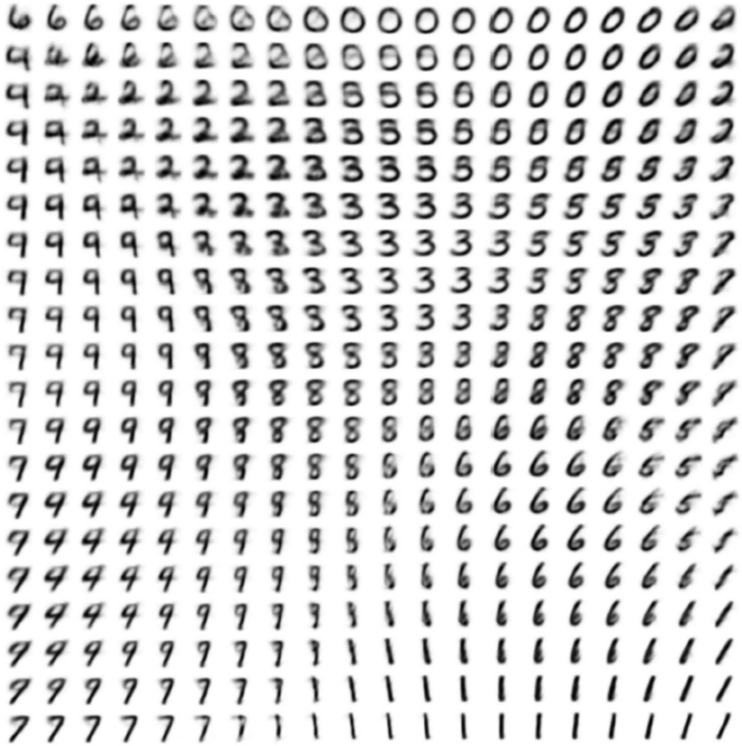
Variational Autoencoders (VAE)

- After training, generate random samples

Sample from the distribution
(e.g., unit Gaussian



Variational Autoencoders (VAE)



Autoencoder vs Variational Autoencoder

Autoencoder

Variational Autoencoder

Ground Truth

https://github.com/kvfrans/variational-autoencoder

Autoencoder Overview

- Autoencoders (AE)
  - Reconstruct input
  - Unsupervised learning
  - Latent space features are useful

- Variational Autoencoders (VAE)
  - Probability distribution in latent space (e.g., Gaussian)
  - Sample from model to generate output

Discriminative vs Generative Tasks

- Discriminative Tasks:
  - Classification
  - Localization / Detection
  - Matching
  - Low-dimensional output
- Generative Tasks (more next lecture!)
  - Generate images / videos /shapes
  - High-dimensional output

Administrative Things

- Thursday July 6th: Multi-Dimensional Convolutions (e.g., 3D), GANs, Visualization!

- Tomorrow: Short Proposal Review
  - What went right and what went wrong?
  - Michael Bronstein "Geometric Deep Learning" course

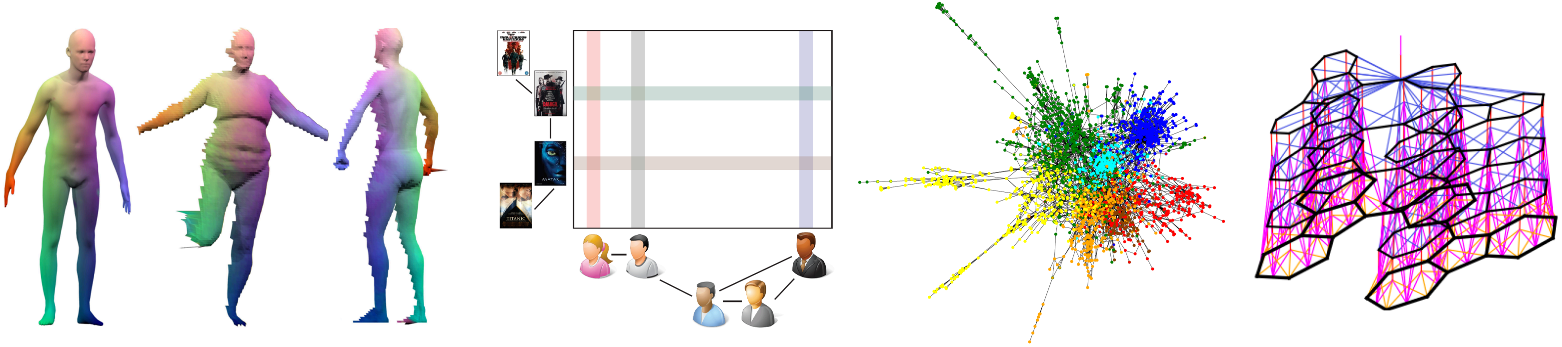
Special Course:

Geometric deep learning on graphs and manifolds

Going beyond Euclidean data

Michael Bronstein

USI Lugano / Tel Aviv University / Intel Perceptual Computing / TUM IAS



Preliminary: scheduled for Fri 30/6 and 7/7 (2pm to 4pm)

-> in our tutorial room