Course number: 80240743

Deep Learning

Xiaolin Hu (胡晓林) & Jun Zhu (朱军)

Dept. of Computer Science and Technology

Tsinghua University

Topic 5: Applications of CNN in Computer Vision

Xiaolin Hu
Dept. of Computer Science and Technology
Tsinghua University

Outline

- Image classification
- Object detection
- Image segmentation
- Image segmentation+object detection
- Image style transformation

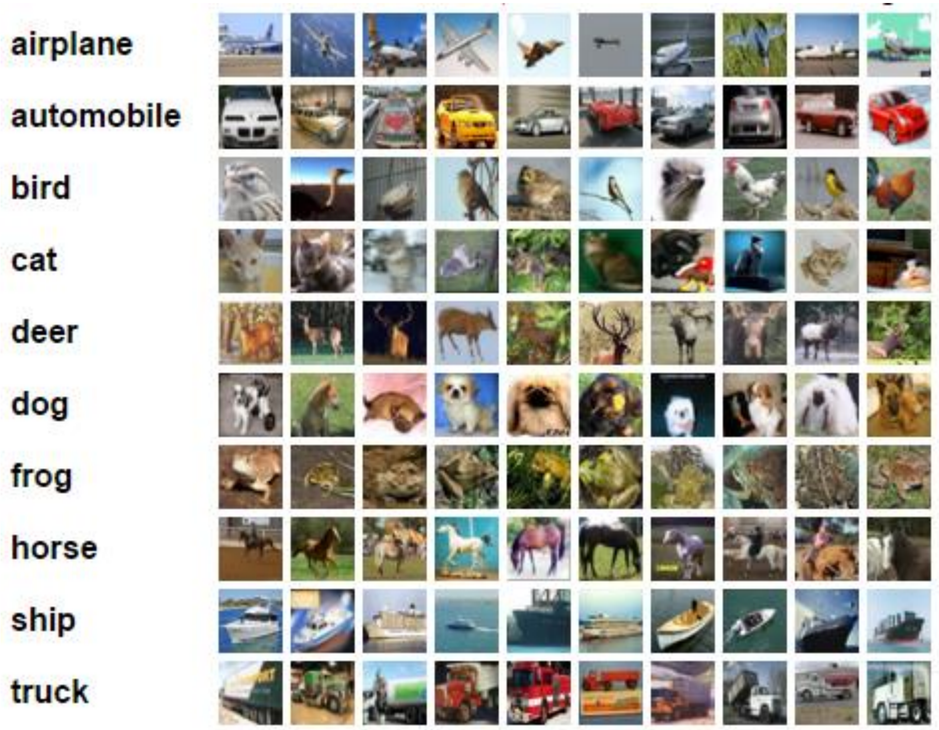
Small image datasets

MNIST

- 60,000 training images and 10,000 test images
- 28x28 black and white images

CIFAR-10 & CIFAR100

- 50,000 training images and 10,000 test images
- 32x32 colour images



ImageNet competition (ILSVRC)

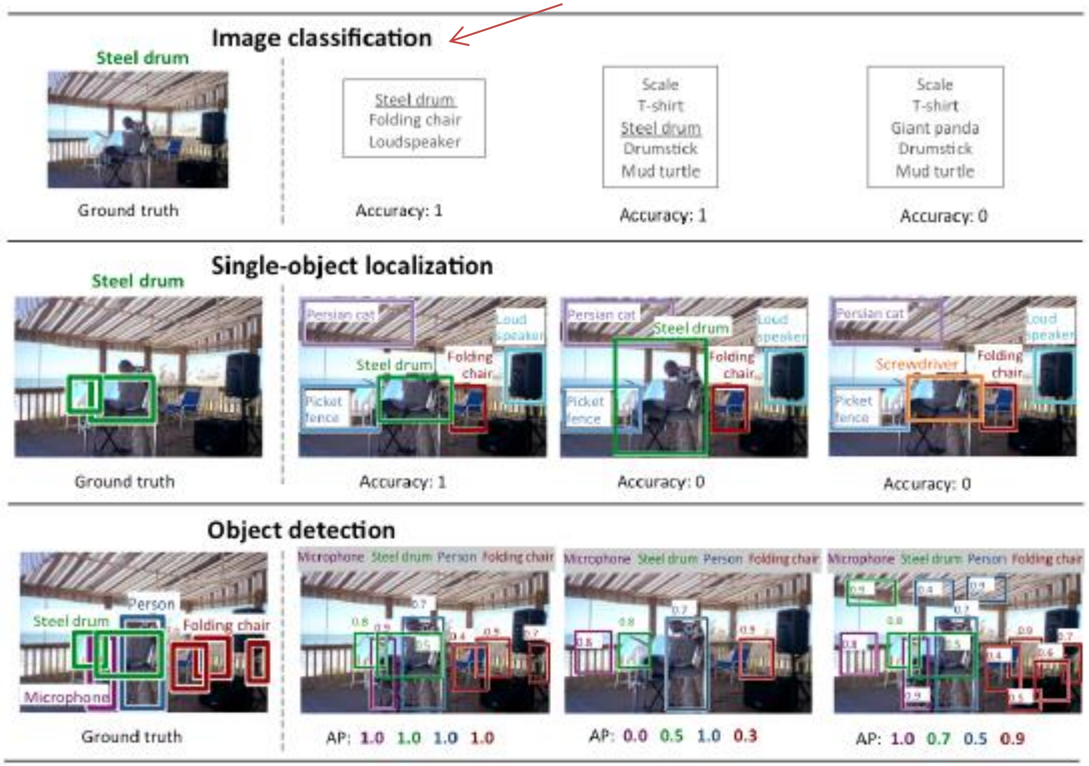
1M training images

Tasks

2010-

2011-

2013-



Top-1

Top-5 (preferred)

Two human expert: 5.1%, 12%

The first column shows the ground truth labeling on an example image, and the next three show three sample outputs with the corresponding evaluation score.

Russakovsky, et al., 2014

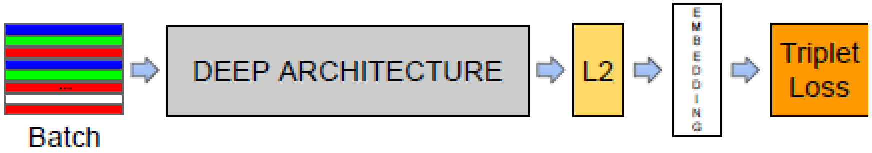
Specific image classification

Face identification

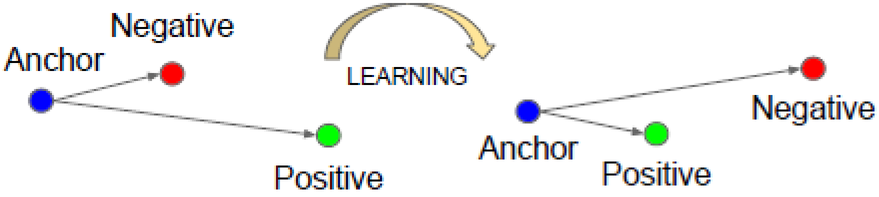
Model	Accuracy (%)
DeepFace (2014)	97.25
DeepID (2014)	97.45
DeepID2 (2014)	99.15
DeepID2+ (2014)	99.47
DeepID3 (2014)	99.53
FaceNet (2015)	99.63

FaceNet

- Architecture



- Triplet loss



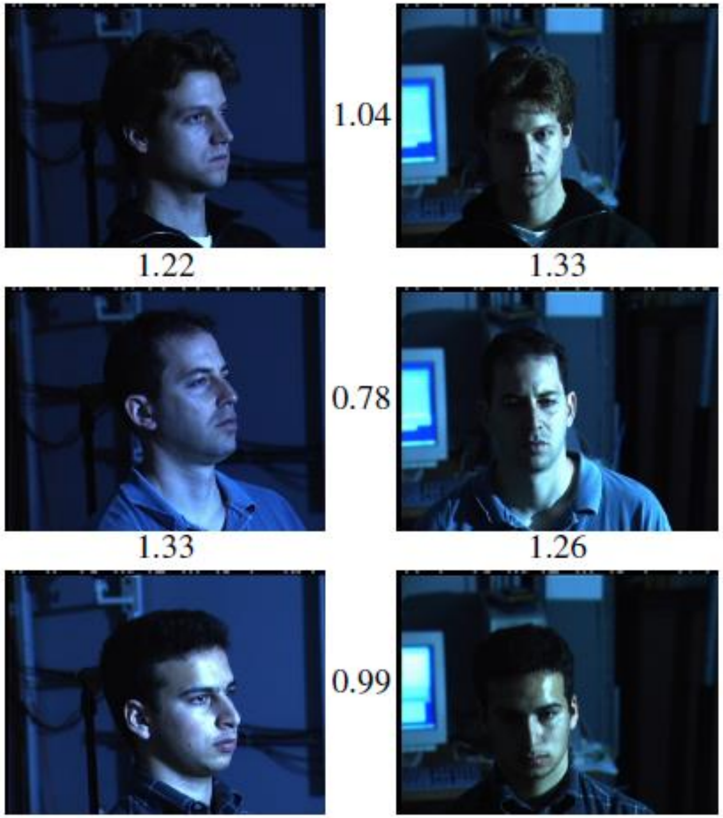
Goal:

$$\|x_i^a - x_i^p\|_2^2 + \alpha < \|x_i^a - x_i^n\|_2^2, \ \forall \left(x_i^a, x_i^p, x_i^n\right) \in \mathcal{T}$$

Loss:

$$\sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha\right]_+$$

- 100M-200M training faces of about 8M different identities

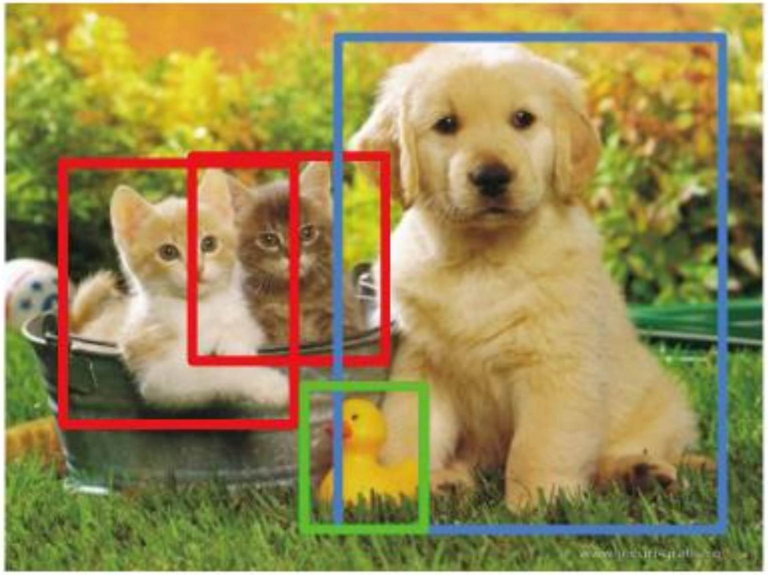


Schroff et al., CVPR 2015

Outline

- Image classification
- Object detection
- Image segmentation
- Image segmentation+object detection
- Image style transformation

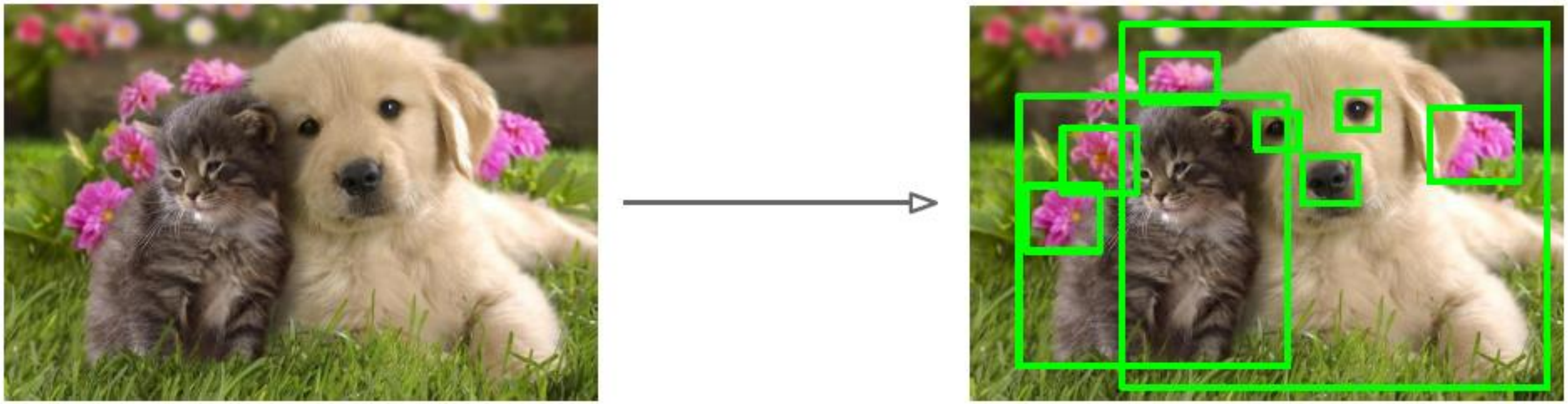
Task



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

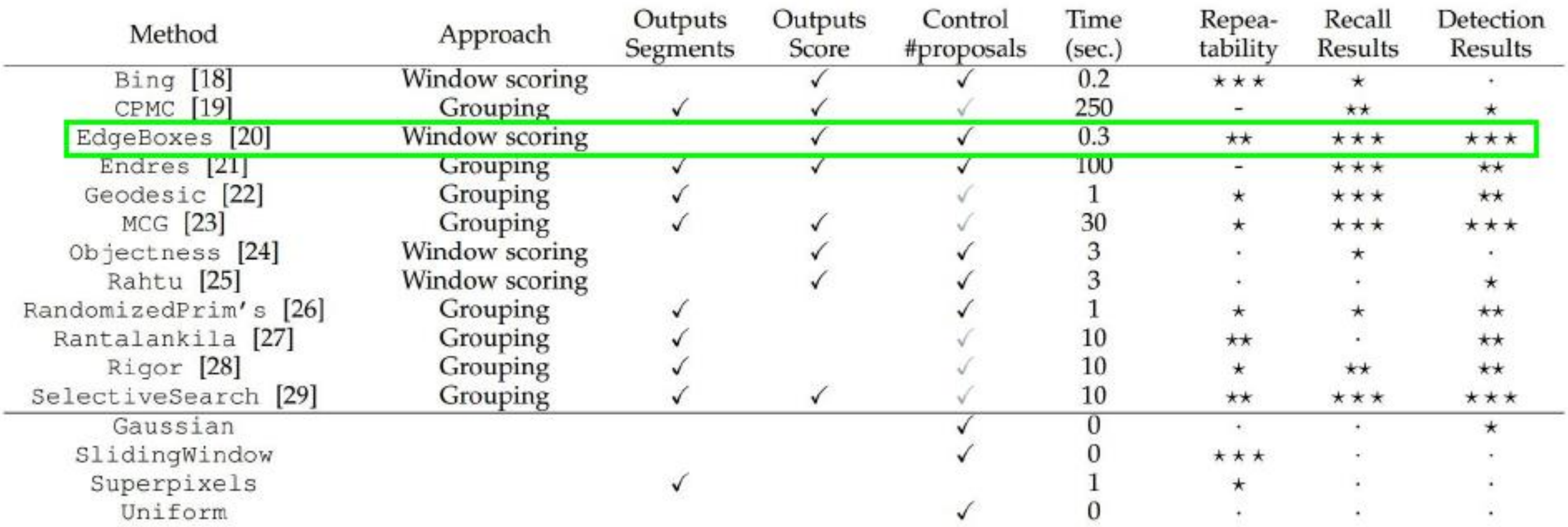
Region proposals

- Find “blobby” image regions that are likely to contain objects
- “Class-agnostic” object detector
- Look for “blob-like” regions



Slide credit: Fei-Fei Li & Andrej Karpathy & Justin Johnson

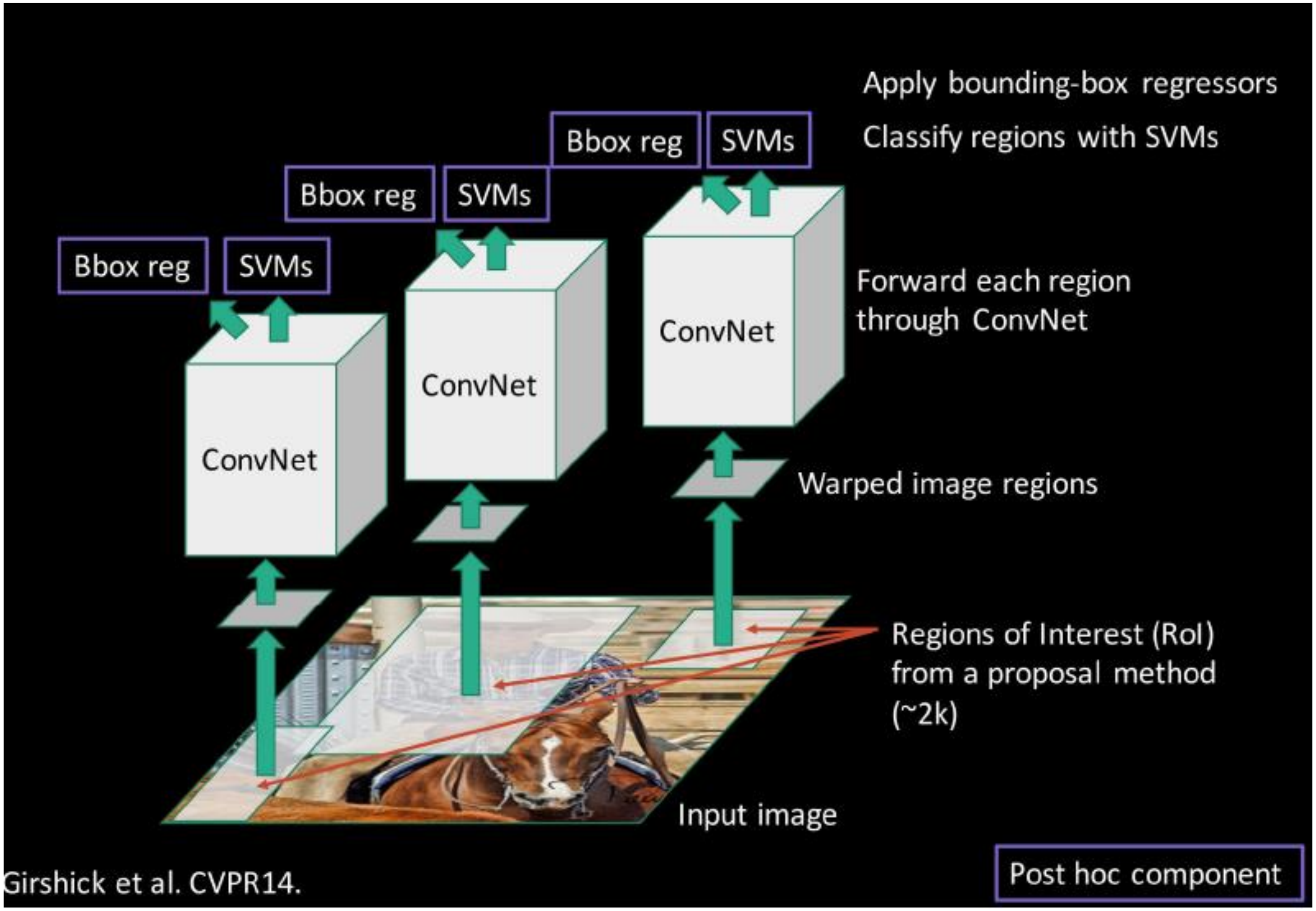
Region proposal: many choices



Method	Approach	Outputs Segments	Outputs Score	Control #proposals	Time (sec.)	Repea-tability	Recall Results	Detection Results
`Bing` [18]	Window scoring		✓	✓	0.2	★★★	★	·
`CPMC` [19]	Grouping	✓	✓	✓	250	-	★★	★
`EdgeBoxes` [20]	Window scoring		✓	✓	0.3	★★	★★★	★★★
`Endres` [21]	Grouping	✓	✓	✓	100	-	★★★	★★
`Geodesic` [22]	Grouping	✓		✓	1	★	★★★	★★
`MCG` [23]	Grouping	✓	✓	✓	30	★	★★★	★★★
`Objectness` [24]	Window scoring		✓	✓	3	·	★	·
`Rahtu` [25]	Window scoring		✓	✓	3	·	·	★
`RandomizedPrim's` [26]	Grouping	✓		✓	1	★	★	★★
`Rantalankila` [27]	Grouping	✓		✓	10	★★	·	★★
`Rigor` [28]	Grouping	✓		✓	10	★	★★	★★
`SelectiveSearch` [29]	Grouping	✓	✓	✓	10	★★	★★★	★★★
`Gaussian`				✓	0	·	·	★
`SlidingWindow`				✓	0	★★★	·	·
`Superpixels`		✓			1	★	·	·
`Uniform`				✓	0	·	·	·

Hosang et al, PAMI 2015

R-CNN



Girshick et al. CVPR14.

Slide credit: Ross Girschick

Steps

1. Train (or download) a classification model for ImageNet (AlexNet)
2. Fine-tune model for detection
3. Extract features
4. Train one binary SVM per class to classify region features
5. For each class, train a linear regression model to map from cached features to offsets to GT boxes to make up for “slightly wrong” proposals

Training image regions

Cached region features

Regression targets
(dx, dy, dw, dh)
Normalized coordinates

(0, 0, 0, 0)
Proposal is good

(.25, 0, 0, 0)
Proposal too far to left

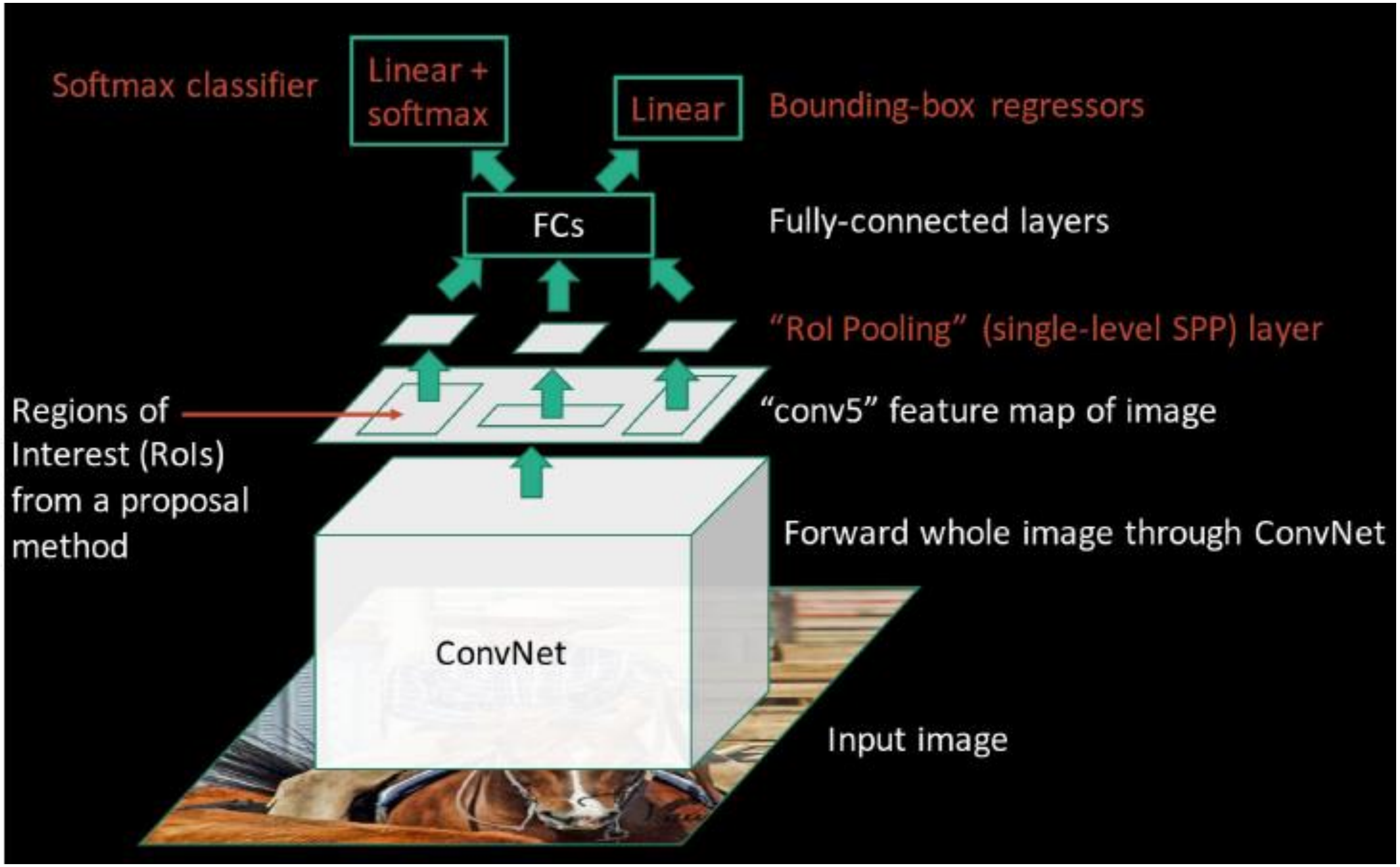
(0, 0, -0.125, 0)
Proposal too wide

Slide credit: Fei-Fei Li & Andrej Karpathy & Justin Johnson

R-CNN problems

- Slow at test-time: need to run full forward pass of CNN for each region proposal
- SVMs and regressors are post-hoc: CNN features not updated in response to SVMs and regressors
- Complex multistage training pipeline

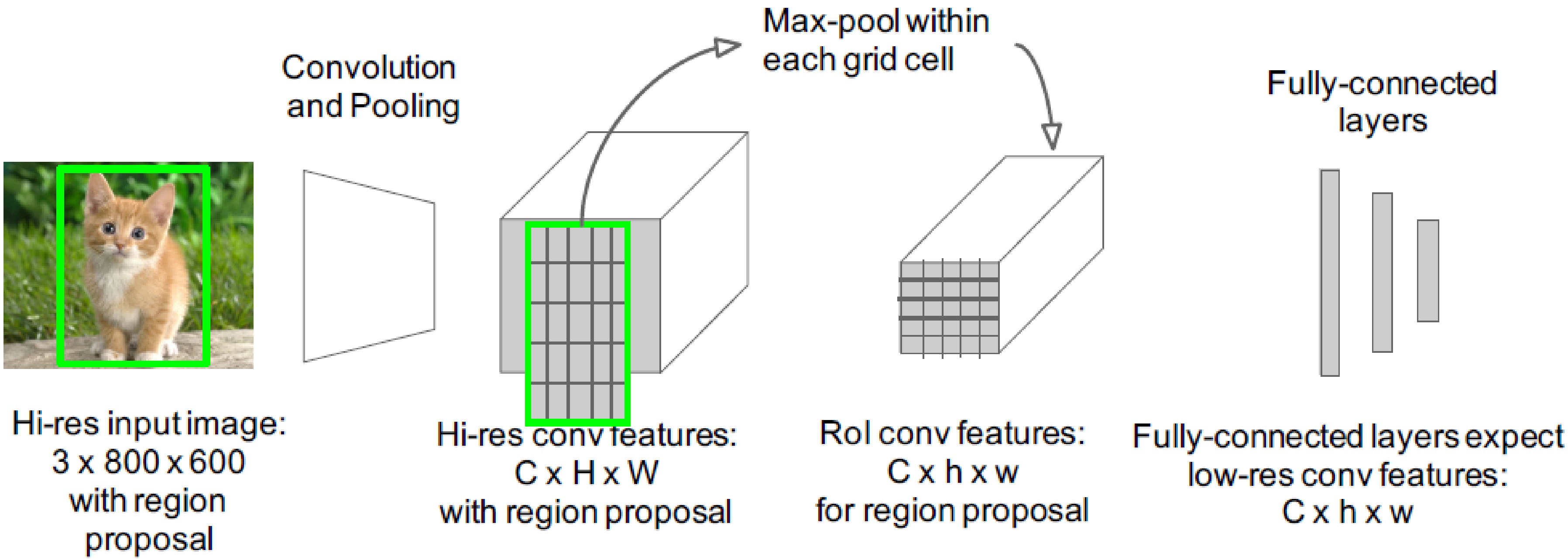
Fast R-CNN



Girshick, ICCV 2015

Slide credit: Ross Girschick

Region of interest region pooling



Convolution and Pooling

Max-pool within each grid cell

Fully-connected layers

Hi-res input image:
3 x 800 x 600
with region
proposal

Hi-res conv features:
C x H x W
with region proposal

RoI conv features:
C x h x w
for region proposal

Fully-connected layers expect
low-res conv features:
C x h x w

Slide credit: Fei-Fei Li & Andrej Karpathy & Justin Johnson

Fast R-CNN Results

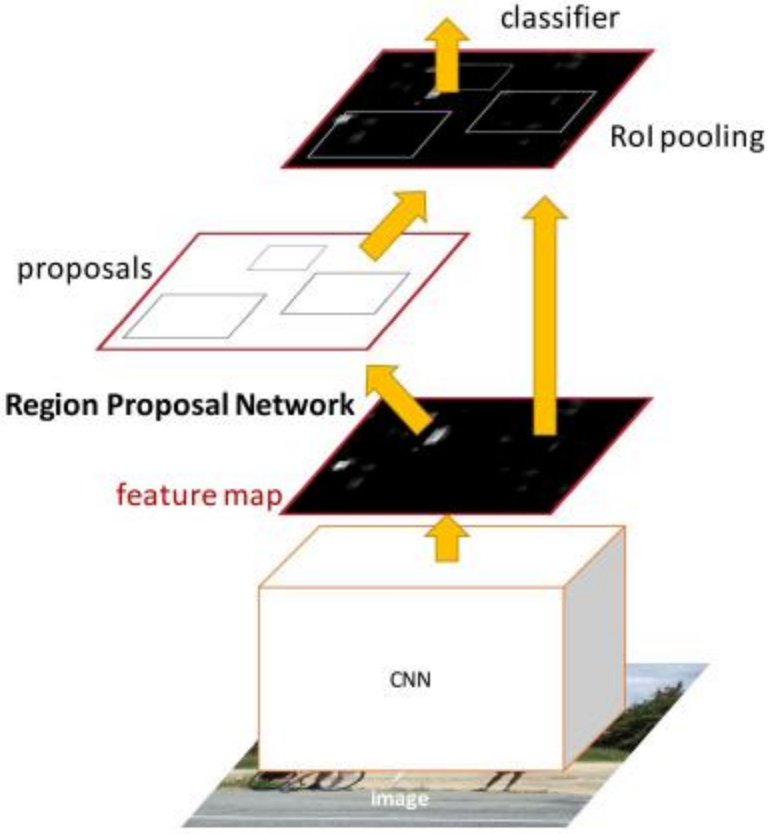
		R-CNN	**Fast R-CNN**
Faster!	Training Time:	84 hours	**9.5 hours**
	(Speedup)	1x	**8.8x**
FASTER!	Test time per image	47 seconds	**0.32 seconds**
	(Speedup)	1x	**146x**
Better!	mAP (VOC 2007)	66.0	**66.9**

Using VGG-16 CNN on Pascal VOC 2007 dataset

Slide credit: Fei-Fei Li & Andrej Karpathy & Justin Johnson

Faster R-CNN

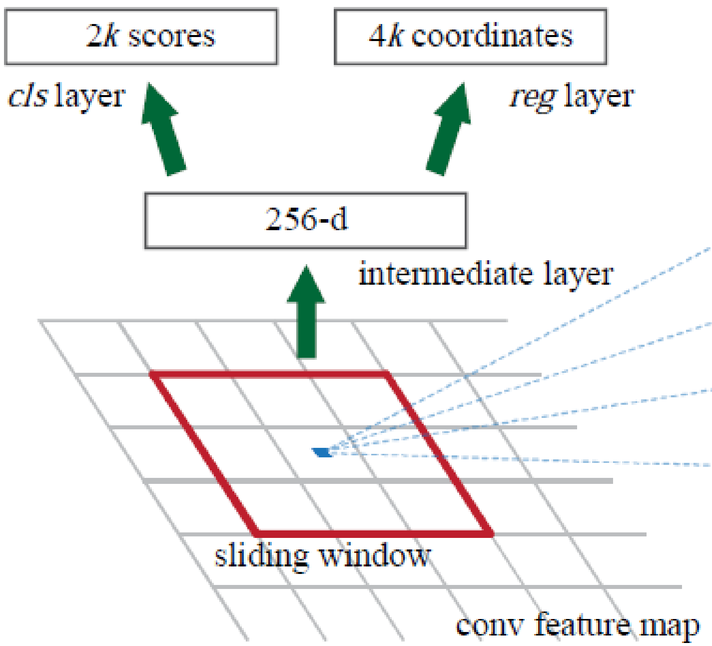
- Insert a **Region Proposal Network (RPN)** after the last convolutional layer
- RPN trained to produce region proposals directly; no need for external region proposals!
- After RPN, use RoI Pooling and an upstream classifier and bbox regressor just like Fast R-CNN



Ren et al., NIPS 2015

Slide credit: Ross Girschick

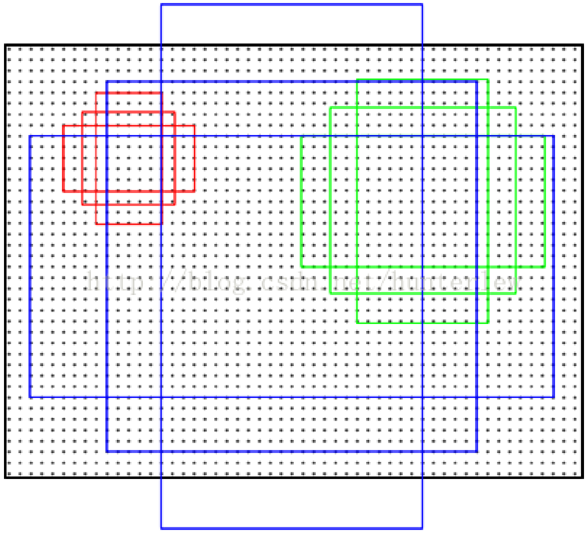
Region proposal network



Obj or not

Ren et al., NIPS 2015

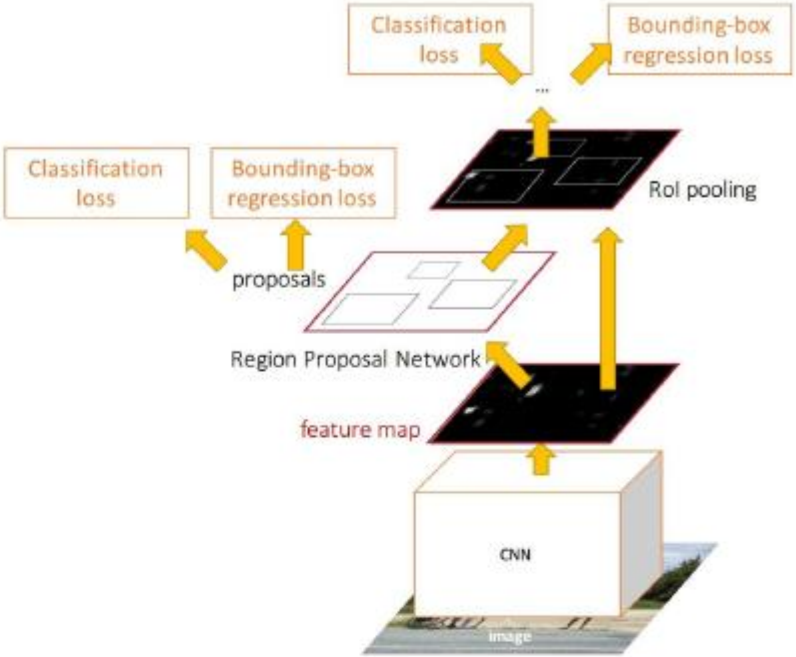
At each location, k=9 boxes are generated in the input image



http://blog.csdn.NET/shenxiaolu1984/article/details/51152614

Faster R-CNN: training

- In the paper: Ugly pipeline
  - Use alternating optimization to train RPN, then Fast R-CNN with RPN proposals, etc.
  - More complex than it has to be
- Since publication: Joint training! One network, four losses
  - RPN classification (anchor good / bad)
  - RPN regression (anchor -> proposal)
  - Fast R-CNN classification (over classes)
  - Fast R-CNN regression (proposal -> box)



Slide credit: Ross Girschick

Faster R-CNN results

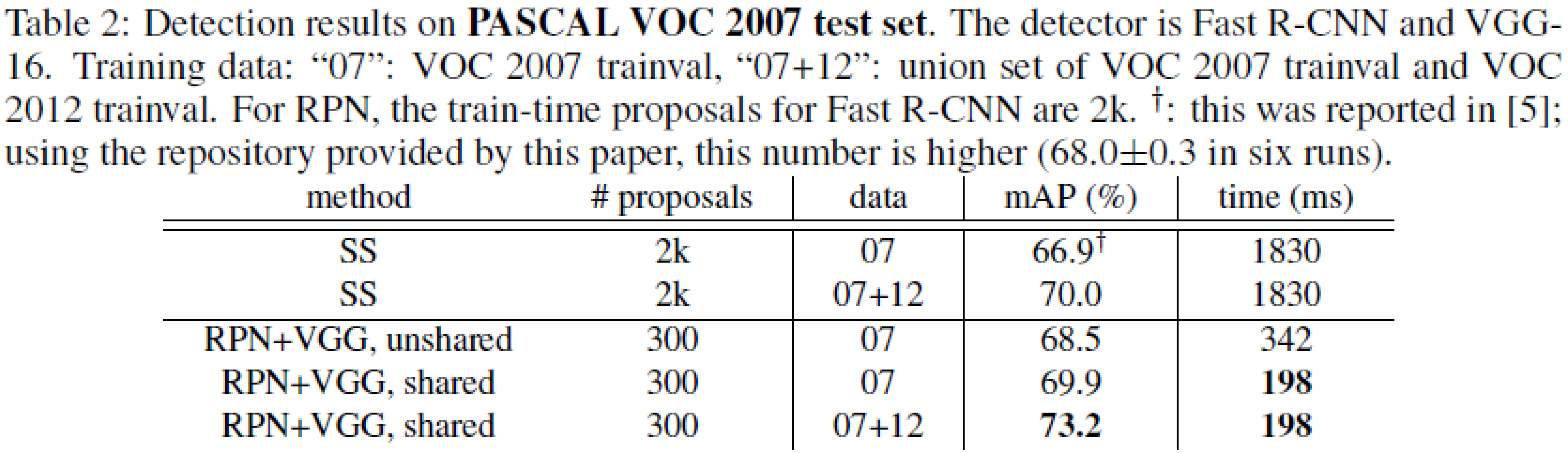
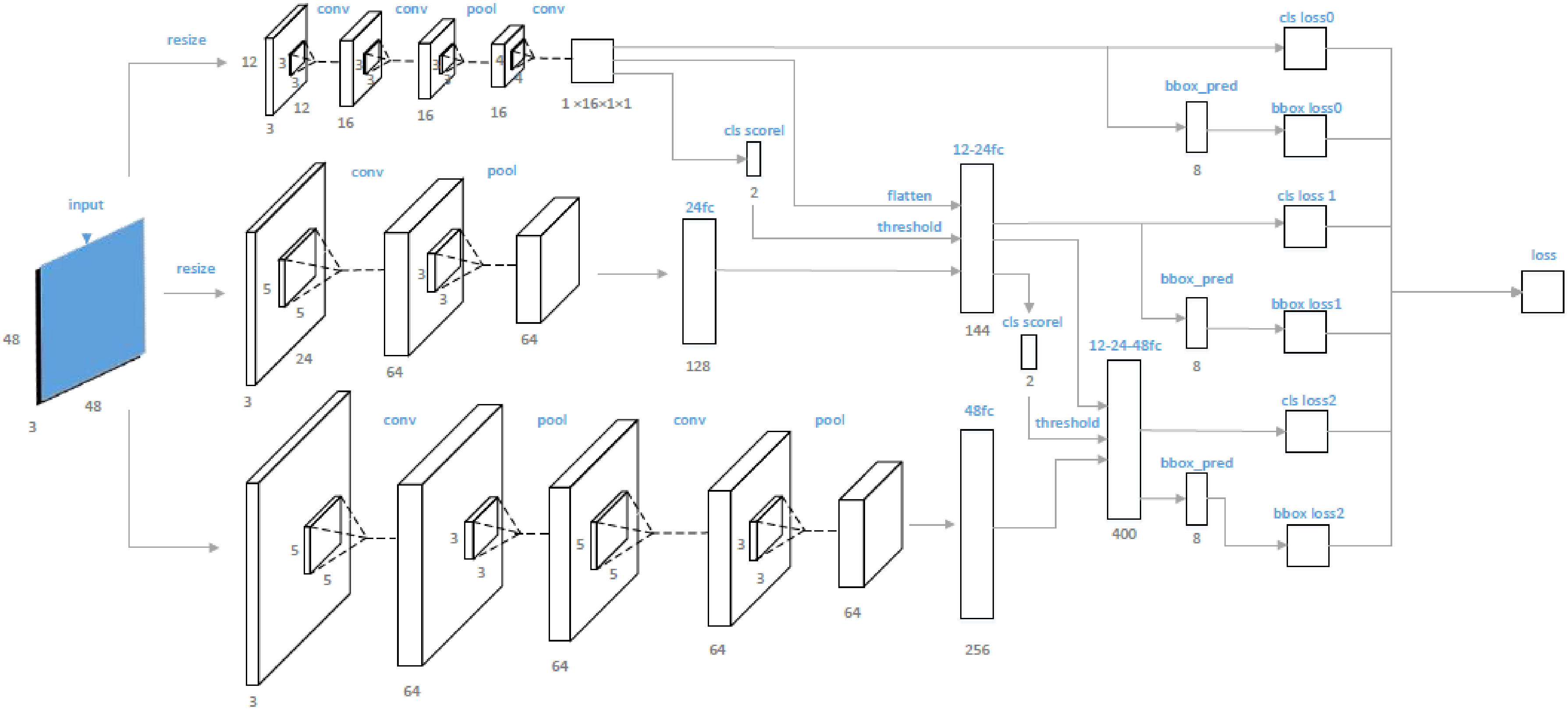


Table 2: Detection results on **PASCAL VOC 2007 test set**. The detector is Fast R-CNN and VGG-16. Training data: "07": VOC 2007 trainval, "07+12": union set of VOC 2007 trainval and VOC 2012 trainval. For RPN, the train-time proposals for Fast R-CNN are 2k. †: this was reported in [5]; using the repository provided by this paper, this number is higher (68.0±0.3 in six runs).

method	# proposals	data	mAP (%)	time (ms)
SS	2k	07	66.9†	1830
SS	2k	07+12	70.0	1830
RPN+VGG, unshared	300	07	68.5	342
RPN+VGG, shared	300	07	69.9	**198**
RPN+VGG, shared	300	07+12	**73.2**	**198**

Specific object detection

Cascaded CNN for face detection



Qin et al., CVPR 2017

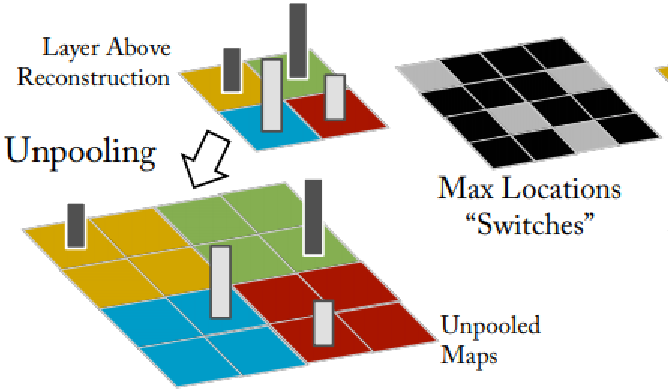
Outline

- Image classification
- Object detection
- Image segmentation
- Image segmentation+object detection
- Image style transformation

Task

How to enlarge the feature maps

- Most CNNs need to enlarge the feature maps in certain layers. What would you do?
- Upsampling (resampling and interpolation)
  - Take an input image, rescale it to the desired size and then calculate the pixel values at each point using a interpolation method such as bilinear interpolation
- Unpooling (reverse max pooling)
  - Record the locations of the maxima within each pooling region in a set of switch variables. Then place the reconstructions from the layer above into appropriate locations



Zeiler, Fergus, 2013

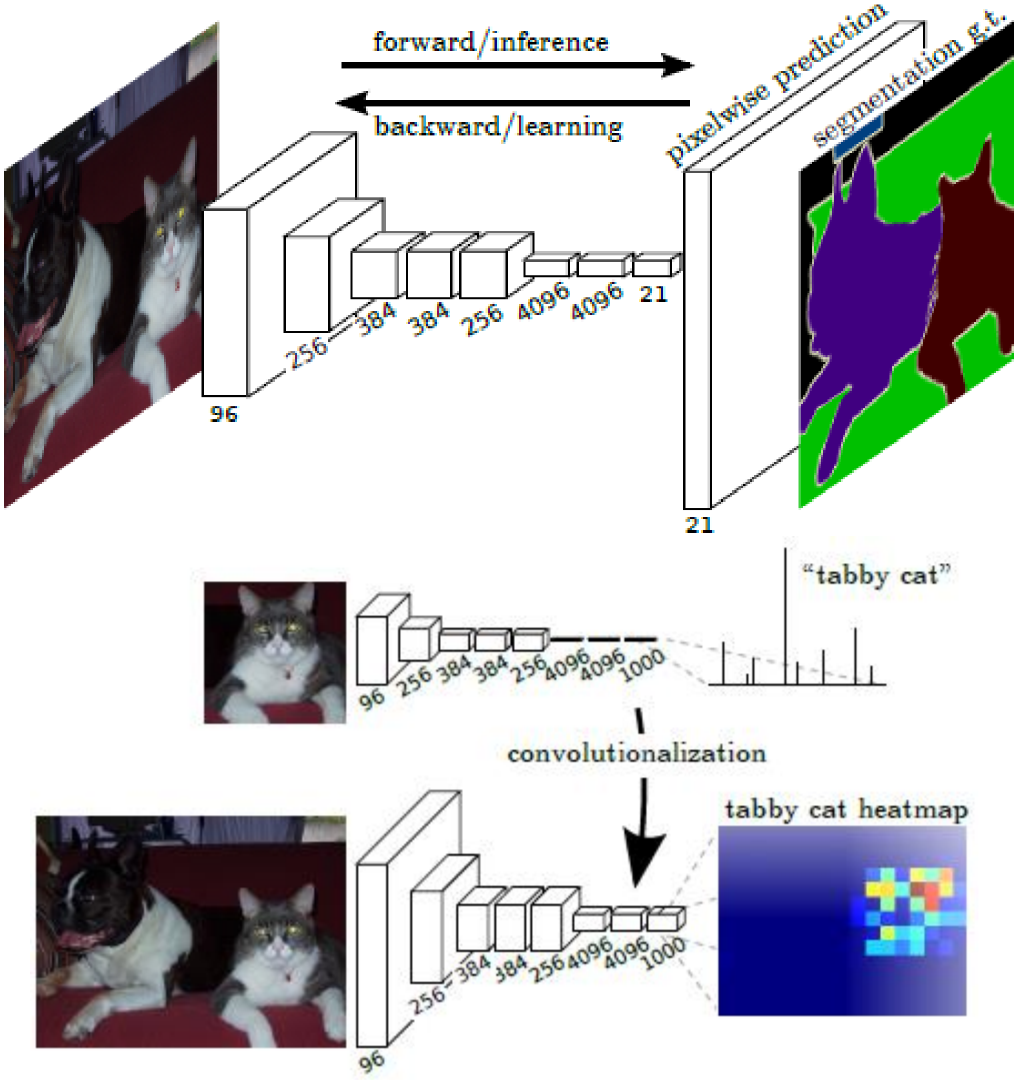
- Transposed convolution (wrongly called *deconvolution*)

  http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

Fully Convolutional Networks

Long et al., CVPR 2015

AlexNet
VGG-16
GoogLeNet



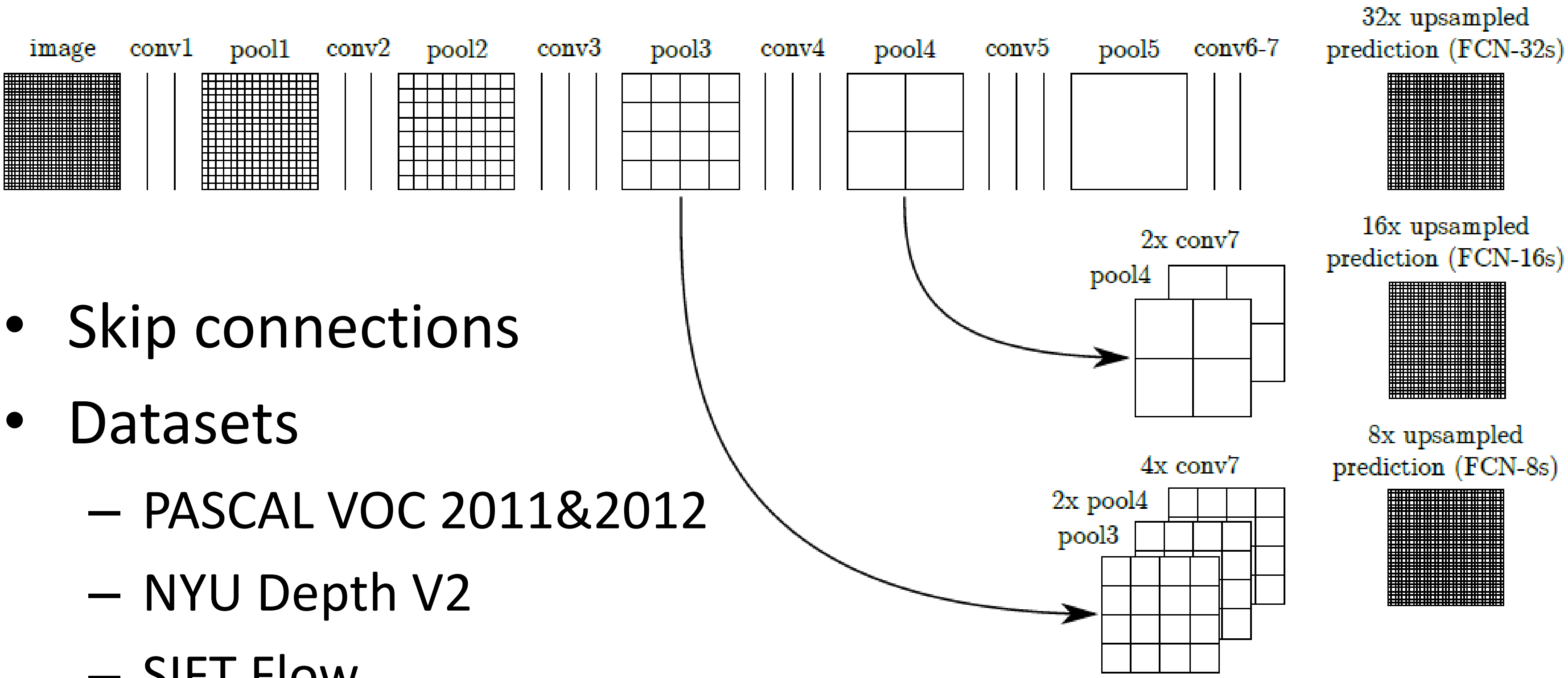
Fully Convolutional Networks

Long et al., CVPR 2015

- Upsample (assume factor *f*)
  - Bilinear interpolation
    - It seems this is the method utilized in the paper
  - Backwards convolution (wrongly called deconvolution) with an output stride of *f*
  - A stack of backward convolution layers and activation functions can even learn a nonlinear upsampling
    - This was not performed in this paper but performed in (Badrinarayanan et al., 2015; Noh, et al., 2015)

Fully Convolutional Networks

Long et al., CVPR 2015



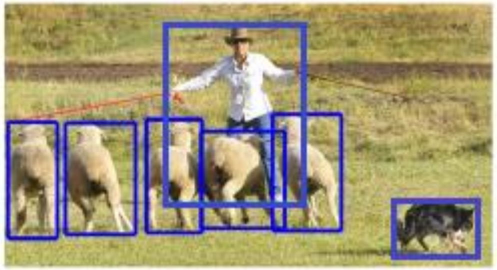
- Skip connections
- Datasets
  - PASCAL VOC 2011&2012
  - NYU Depth V2
  - SIFT Flow

Outline

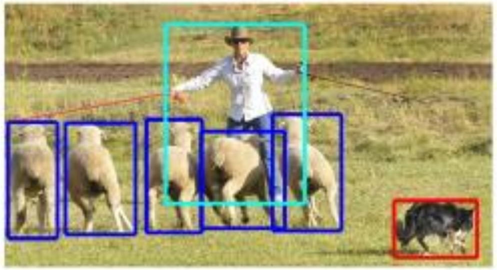
- Image classification
- Object detection
- Image segmentation
- Image segmentation+object detection
- Image style transformation

Types of problems

Image classification



Object proposal with box



Object detection or localization with box

Semantic segmentation

(Historically, those datasets about scenes are called “scene labeling” data sets)

Object proposal with segmentation

Individual instance segmentation

Learning to Segment Object Candidates

“DeepMask”

Pinheiro et al., NIPS 2015

- Full Scene Inference
  - Apply the model densely at multiple locations and scales, which gives a segmentation mask and object score at each image location

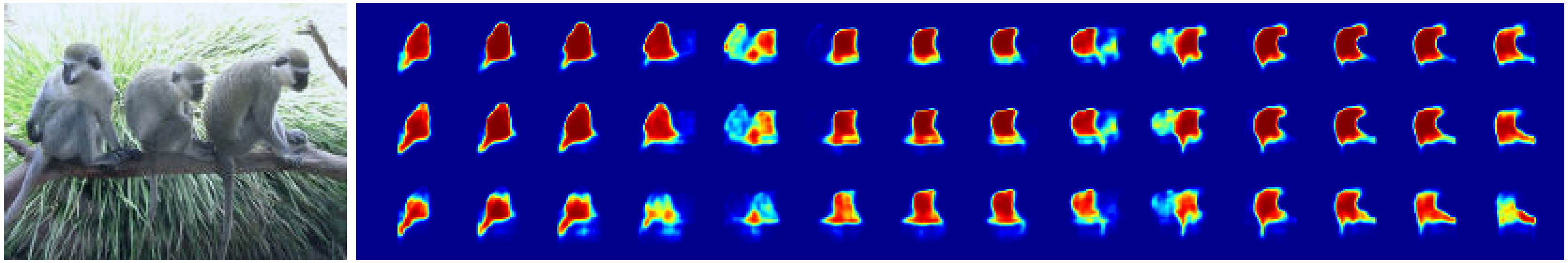


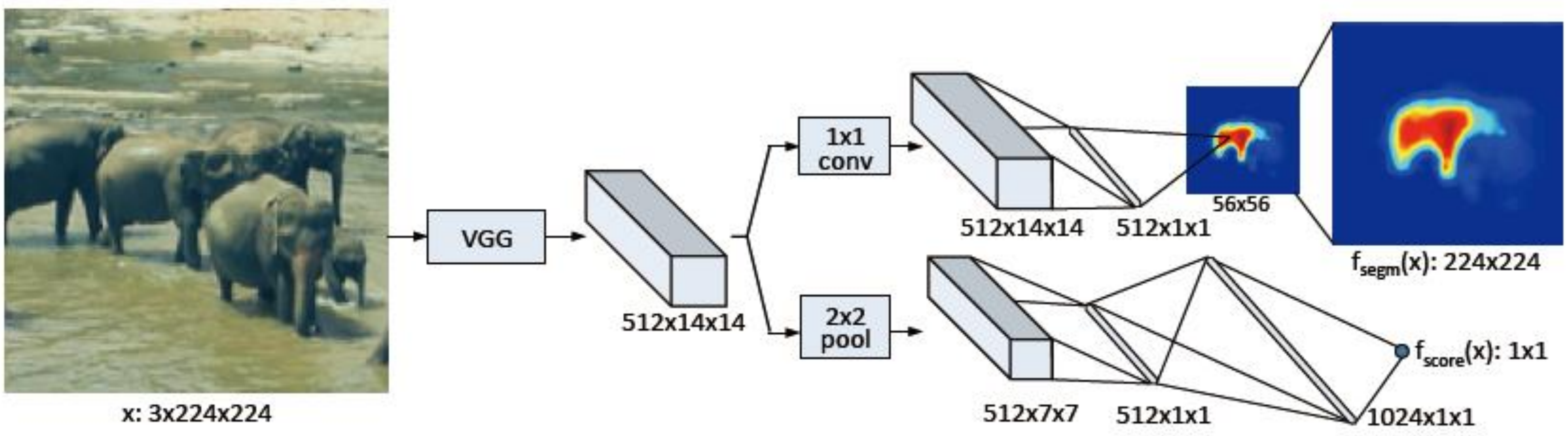
Figure 2: Output of segmentation model applied densely to a full image with a 16 pixel stride (at a

- Datasets
  - VOC 2007 & MS COCO

Learning to Segment Object Candidates

“DeepMask”

Pinheiro et al., NIPS 2015



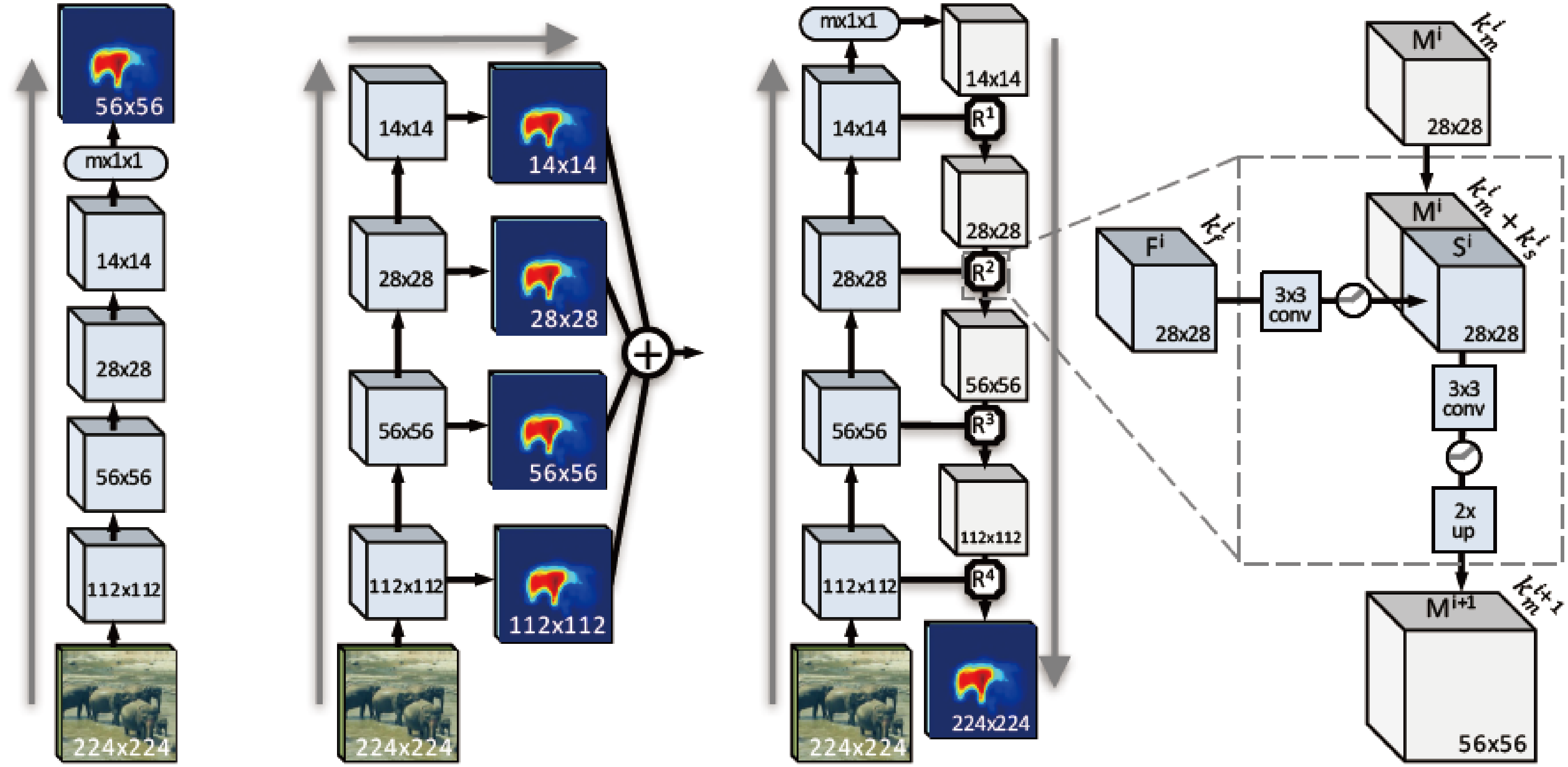
- training triplets: input patch x, mask m and label y

$$\mathcal{L}(\theta) = \sum_{k} \left(\frac{1+y_k}{2w^o h^o} \sum_{ij} \log(1+e^{-m_k^{ij} f_{segm}^{ij}(x_k)}) + \lambda \log(1+e^{-y_k f_{score}(x_k)}) \right)$$

Learning to Refine Object Segments

“SharpMask”

Pinheiro et al., 2016



(a) feedforward (b) feedforward + skip (c) proposed network (d) refinement module

Learning to Refine Object Segments

"SharpMask"

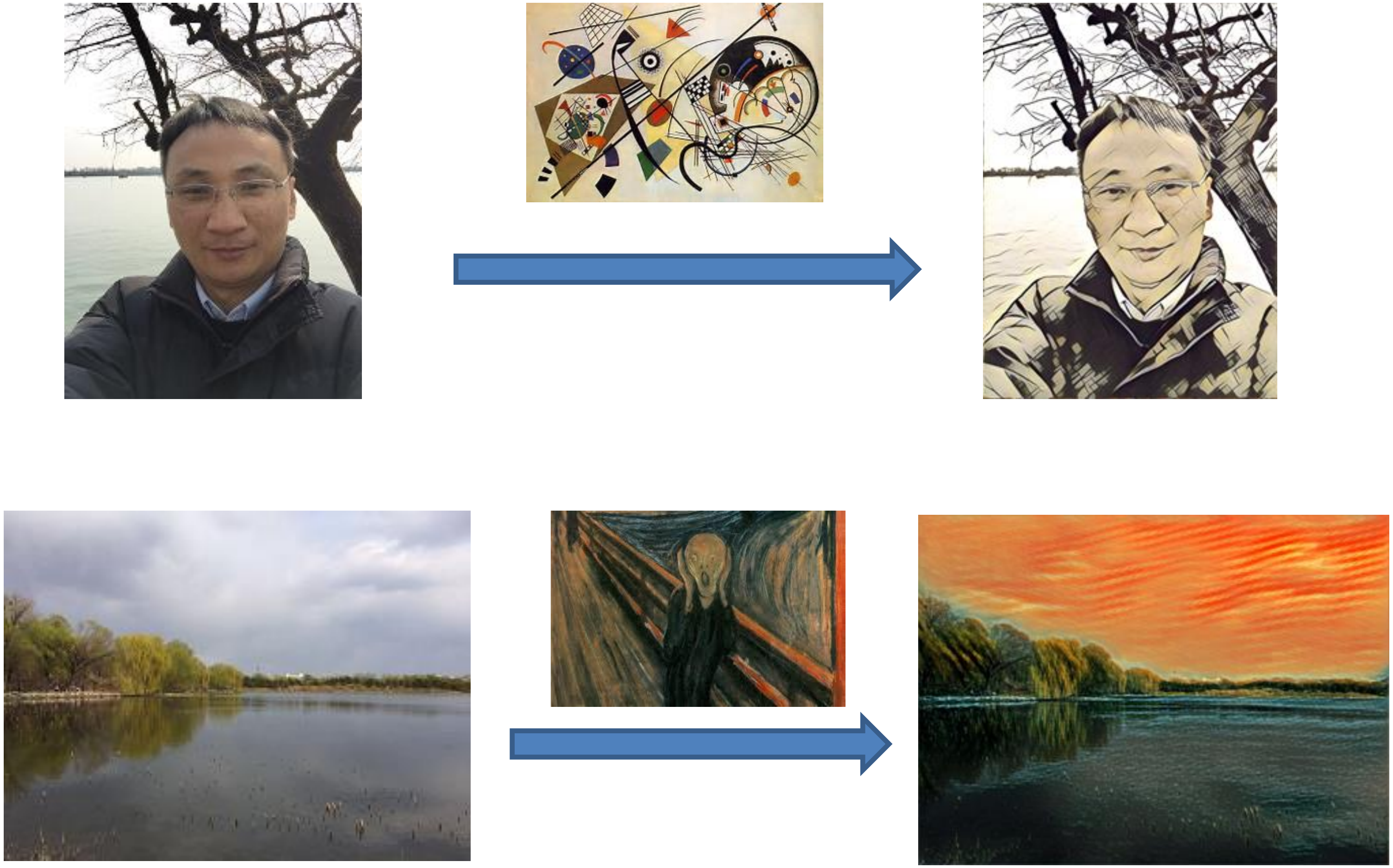
Pinheiro et al., 2016

- Two-stage training
  - First: train the feedforward path (identical to DeepMask)
  - Second: frozen the feedforward path and replace its prediction layer with a linear layer, then train the feedback path
- Select the top N scoring proposal windows and apply the refinement in a batch mode to these top N locations
- Pretrained Residual Net was used
- Dataset: MS COCO

Outline

- Image classification
- Object detection
- Image segmentation
- Image segmentation+object detection
- Image style transformation

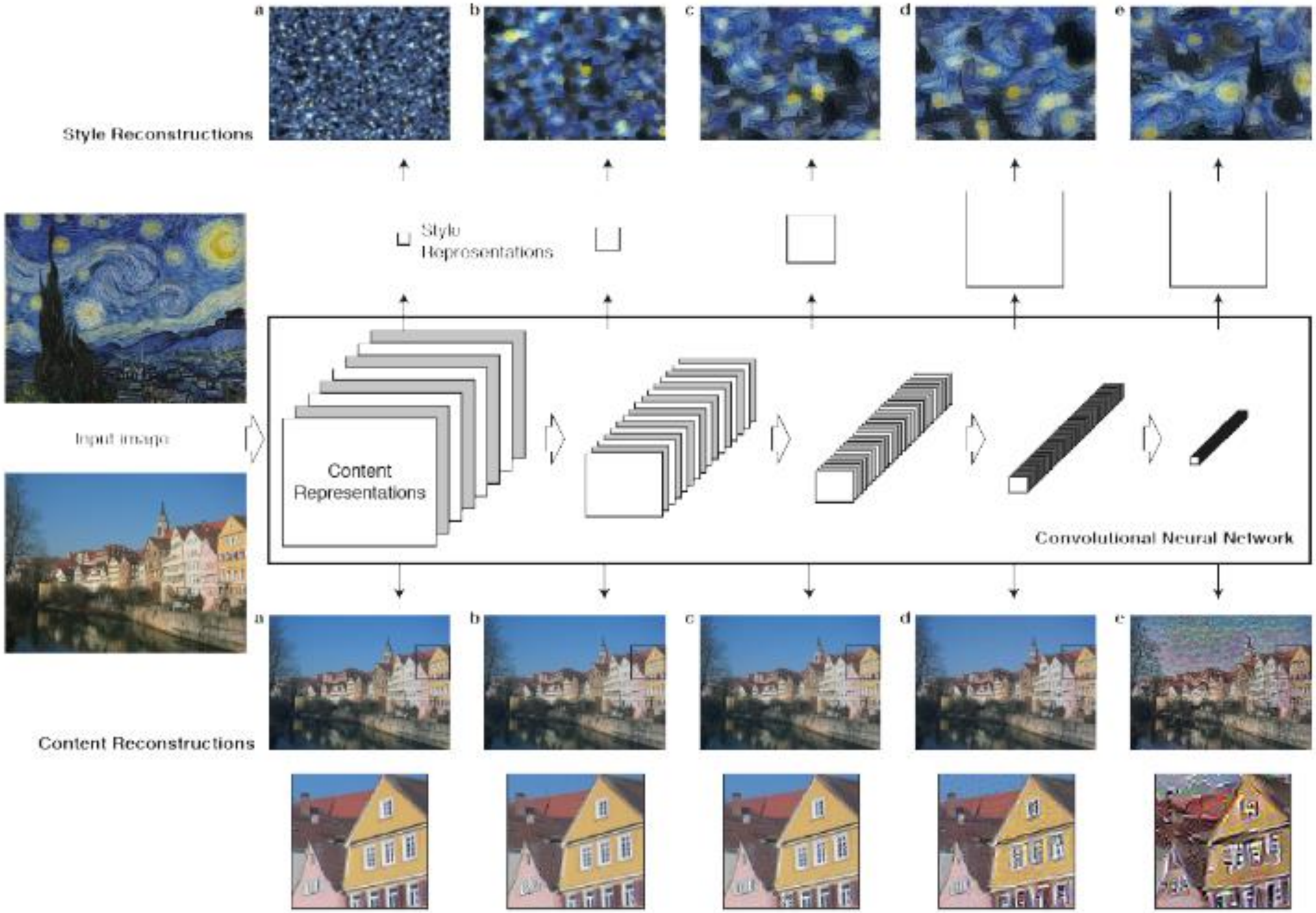
Task



Created by Prisma

A Neural Algorithm of Artistic Style

Gatys et al., 2016



A Neural Algorithm of Artistic Style

- Content loss

$$\mathcal{L}_{content}(\vec{p},\vec{x},\overset{\substack{\text{layer index}\\ \downarrow}}{l}) = \frac{1}{2}\sum_{i,j}(\overset{\substack{\text{photo}\\ \downarrow}}{F^l_{ij}} - P^l_{ij})^2$$

- $\vec{p}$: original photo
- $\vec{x}$: generated image
- F_{ij}&P_{ij}: j-th element in i-th feature map

- Style loss

$$E_l = \frac{1}{4N_l^2M_l^2}\sum_{i,j}(G^l_{ij} - \overset{\substack{\text{painting}\\ \downarrow}}{A^l_{ij}})^2$$

where G and A are gram matrices, e.g.,

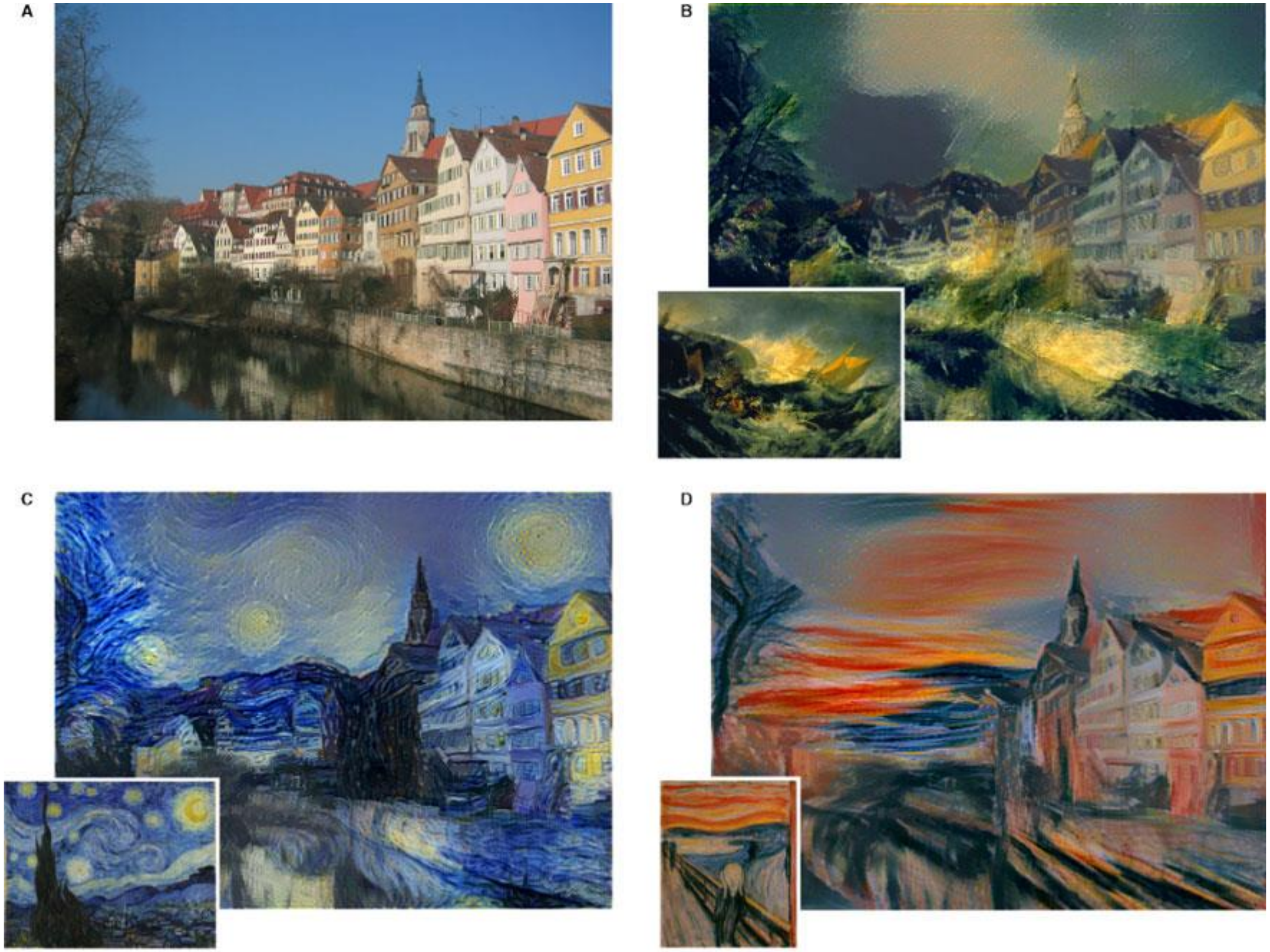
$$G^l_{ij} = \sum_k F^l_{ik}F^l_{jk}$$

$$\mathcal{L}_{style}(\vec{a},\vec{x}) = \sum_{l=0}^{L} w_l E_l$$

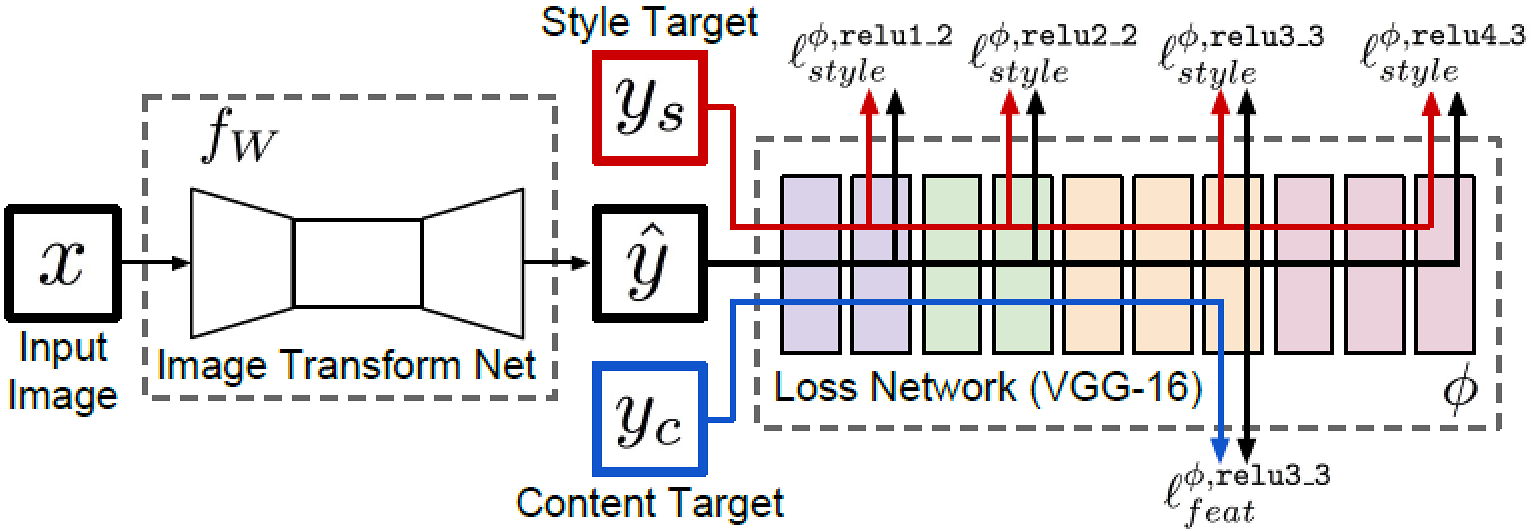
- Optimize the total loss w.r.t. pixels x (not w)

$$\mathcal{L}_{total}(\vec{p},\vec{a},\vec{x}) = \alpha\mathcal{L}_{content}(\vec{p},\vec{x}) + \beta\mathcal{L}_{style}(\vec{a},\vec{x})$$

Results



Feedforward generation



$$\ell_{feat}^{\phi,j}(\hat{y},y) = \frac{1}{C_j H_j W_j} \|\phi_j(\hat{y}) - \phi_j(y)\|_2^2$$

$$\ell_{style}^{\phi,j}(\hat{y},y) = \|G_j^{\phi}(\hat{y}) - G_j^{\phi}(y)\|_F^2$$

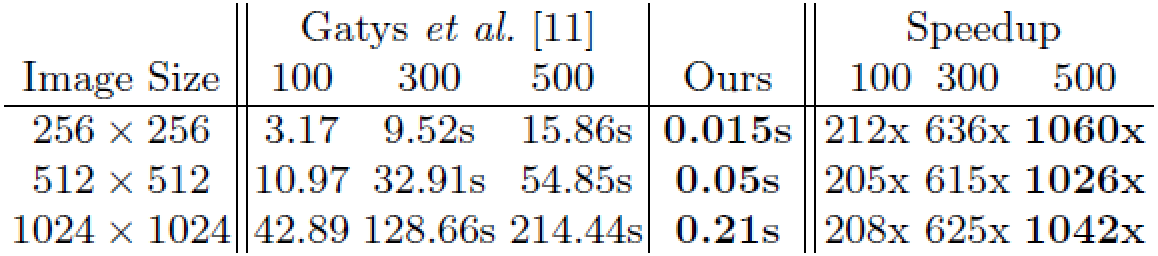
where G is the gram matrix

Johnson, Alahi, Fei-Fei, ECCV 2016

Results

Style
The Starry Night,
Vincent van Gogh,
1889

Style
The Muse,
Pablo Picasso,
1935



	Gatys *et al.* [11]				Speedup		
Image Size	100	300	500	Ours	100	300	500
256 × 256	3.17	9.52s	15.86s	**0.015s**	212x	636x	**1060x**
512 × 512	10.97	32.91s	54.85s	**0.05s**	205x	615x	**1026x**
1024 × 1024	42.89	128.66s	214.44s	**0.21s**	208x	625x	**1042x**

Summary

- Image classification
- Object detection
- Image segmentation
  - upsample
- Image segmentation+object detection
- Image style transformation
  - Optimization-based approach
  - Generative network

Further reading

- Schroff, Kalenichenko, Philbin (2015)
  FaceNet: A Unified Embedding for Face Recognition and Clustering
  CVPR
- Girshick, Donahue, Darrell, Malik (2014)
  Rich feature hierarchies for accurate object detection and semantic segmentation
  CVPR
- Girshick (2015)
  Fast R-CNN
  ICCV
- Ren, He, Girshick, Sun (2015)
  Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks
  NIPS

Further reading

- Long, Shelhamer, Darrell (2015)
  Fully Convolutional Networks for Semantic Segmentation
  CVPR
- Pinheiro, Collobert, Dollar (2015)
  Learning to Segment Object Candidates
  NIPS
- Pinheiro, Lin, Collobert, Dollar (2016)
  Learning to Refine Object Segments
  ECCV
- Gatys, Ecker, Bethge (2016)
  Image Style Transfer Using Convolutional Neural Networks
  CVPR
- Johnson, Alahi, Fei-Fei (2016)
  Perceptual Losses for Real-Time Style Transfer and Super-Resolution
  ECCV

Prepare for the next lecture

- Form groups of 2 and every group prepares a 5-minute presentation with slides for one of the following papers
  - Santurkar, Tsipras, Ilyas, Madry (2018) How does batch normalization help optimization? NeurIPS
  - Transposed convolution: http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html