NANJING UNIVERSITY

计算机视觉表征与识别
Chapter 10: Recognition

王利民

媒体计算课题组

http://mcg.nju.edu.cn/

本科生大作业要求

- 综述性研究报告：针对一个专门话题，进行深入调研分析，形成一篇高质量的综述。
- 培养：查阅文献，阅读文献，撰写文献的能力，可能启发对某个方向深入研究的兴趣。
- 提交格式：**CVPR论文格式，建议英文撰写**。
- 调研范围：顶级会议和顶级期刊论文，以及最新的arXiv论文，
- 不要仅仅去看一些公众号、知乎、中文博客
- **提交时间：6月20号之前（毕业班），其他8月20号**

研究生大作业要求

- 目标：在一个方向做深入研究，最起码实现一篇现有论文，有自己的思考和理解，鼓励创新。
- 提交内容包括：**报告和代码**（评分依据）
- 报告格式：CVPR论文提交格式
  - 包含题目，摘要，引文，相关工作，具体技术路线，实验结果与分析，结论。
- 代码：可以参考网上代码，但是核心代码需要自己编写
  - 我们会认真检查，自己对自己负责。
- 提交时间：2020年8月20号

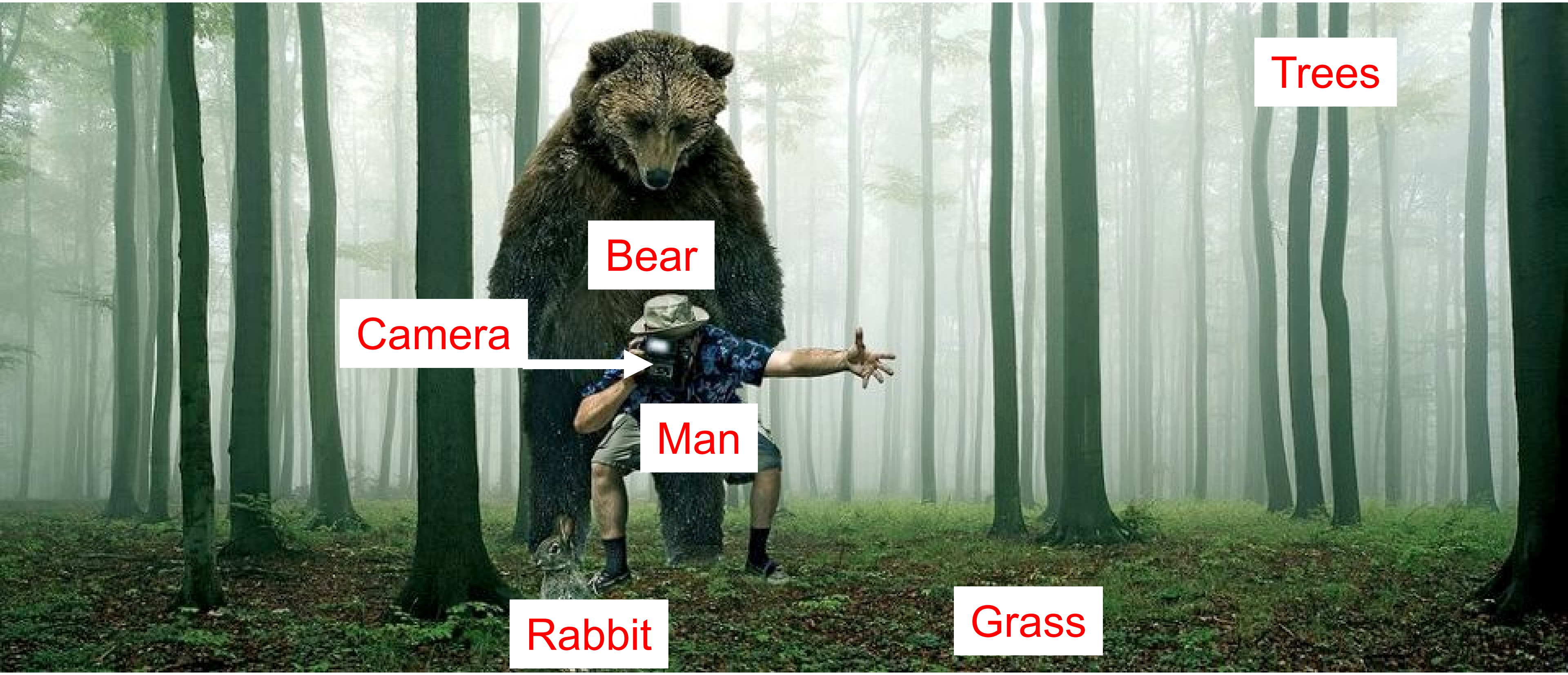
Topics

- Low-level vision: denoise, super resolution etc.
- Edge detection
- Grouping and segmentation
- Local descriptor and image matching
- Deep learning for vision
- Tracking, video analysis
- Vision and language
- Weakly/Self supervised learning
- Transformer for vision
- Etc.

Today' class

- Overview of image categorization
- Spatial pyramids bag-of-words categorizer
- Deep convolutional neural networks (CNNs)
- Object detection

What do you see in this image?



Forest

Categorization: describe, predict, or interact

Is it **dangerous**?

Is it **alive**?

How **fast** does it run?

Is it **soft**?

Does it have a **tail**?

Can I **poke with it**?

Theory of categorization

How do we determine if something is a member of a particular category?

- Definitional approach
- Prototype approach
- Exemplar approach

Definitional approach: classical view of categories

- Plato & Aristotle
  - Categories are defined by a list of properties shared by all elements in a category
  - Category membership is binary
  - Every member in the category is equal

Aristotle by Francesco Hayez

The Categories (Aristotle)

Slide Credit: A. A. Efros

Prototype Model

Exemplars Model

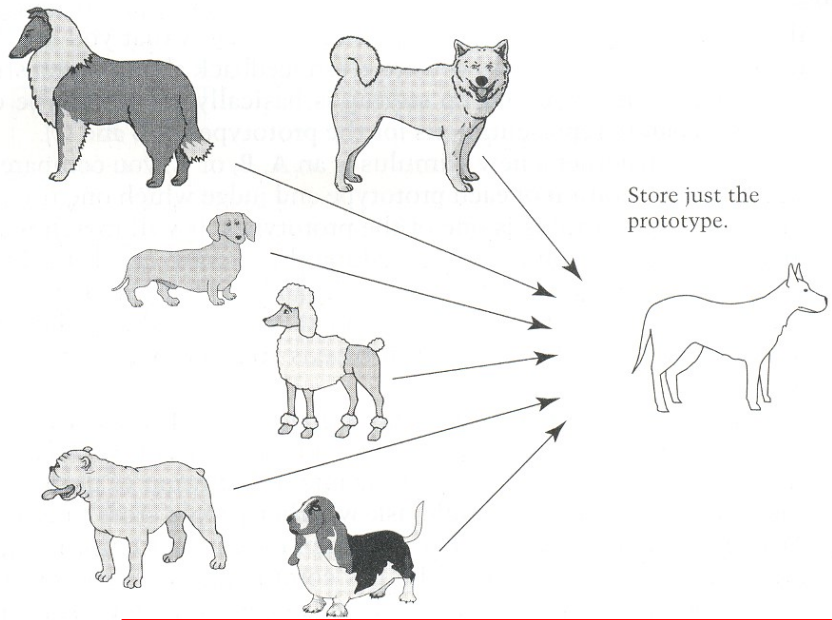


Figure 7.3. *Schematic of the prototype model. Although many exemplars are seen, only the prototype is stored. The prototype is updated continually to incorporate more experience with new exemplars.*

Category judgments are made
by comparing a new exemplar
to the prototype.

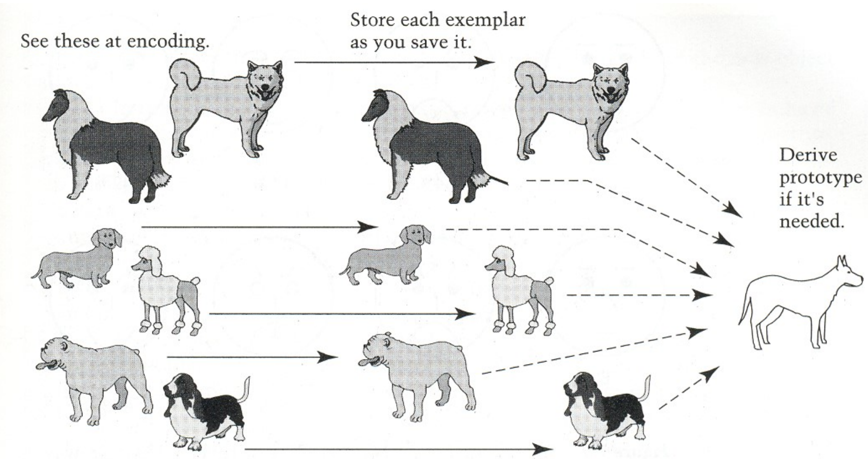


Figure 7.4. *Schematic of the exemplar model. As each exemplar is seen, it is encoded into memory. A prototype is abstracted only when it is needed, for example, when a new exemplar must be categorized.*

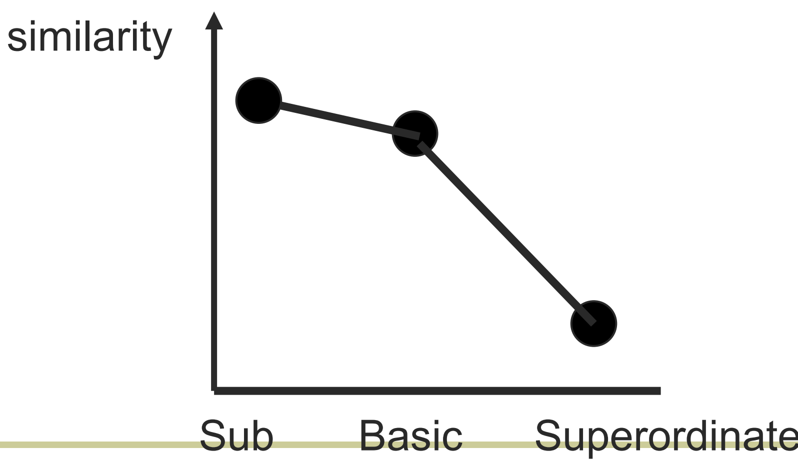
Category judgments are made
by comparing a new exemplar
to all the old exemplars of a category
or to the exemplar that is the most
appropriate

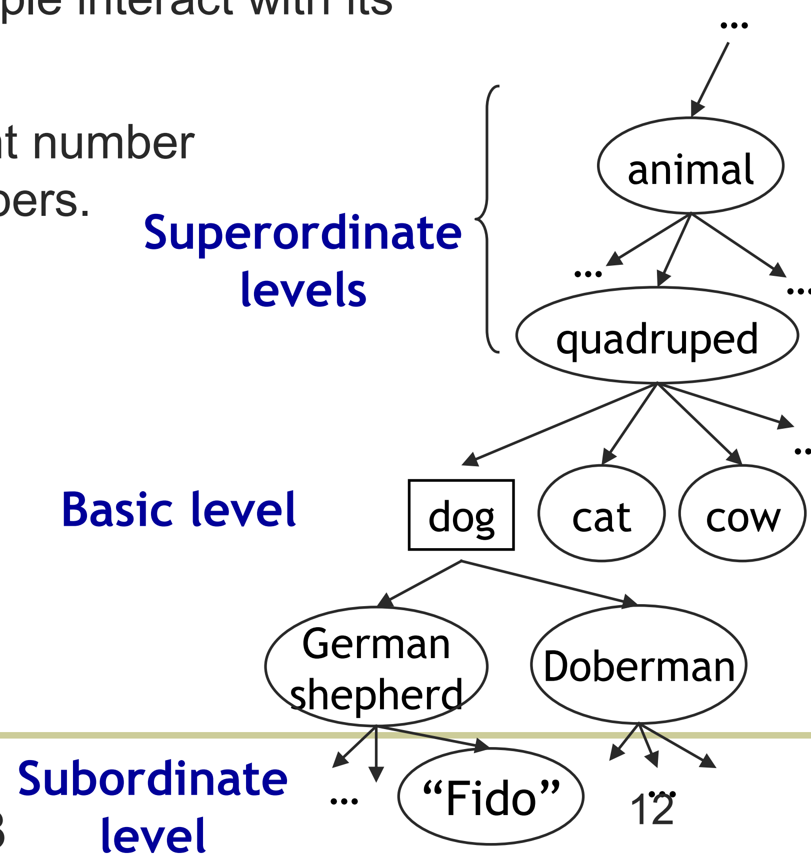
Slide Credit: Torralba

Levels of categorization [Rosch 70s]

Definition of Basic Level:

- **Similar shape**: Basic level categories are the highest-level category for which their members have similar shapes.
- **Similar motor interactions**: … for which people interact with its members using similar motor sequences.
- **Common attributes**: … there are a significant number of attributes in common between pairs of members.





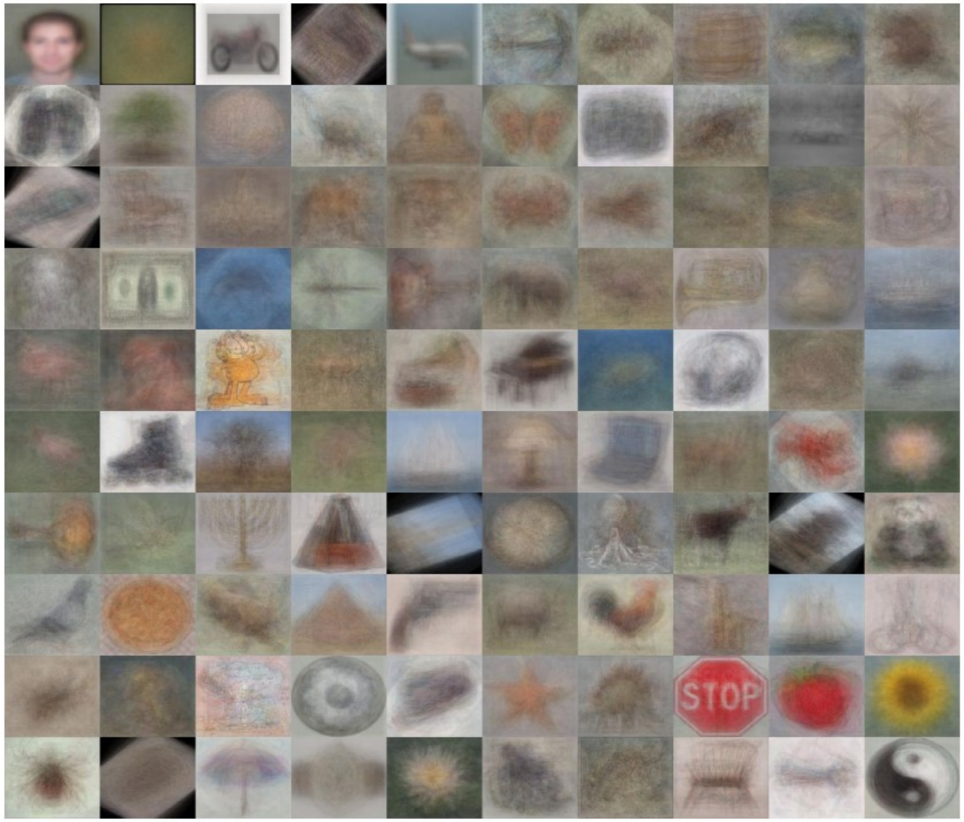
Rosch et a. Principle of categorization, 1978

Image categorization

- Cat vs Dog

Image categorization

- Object recognition



Caltech 101 Average Object Images

Image categorization

- Fine-grained recognition

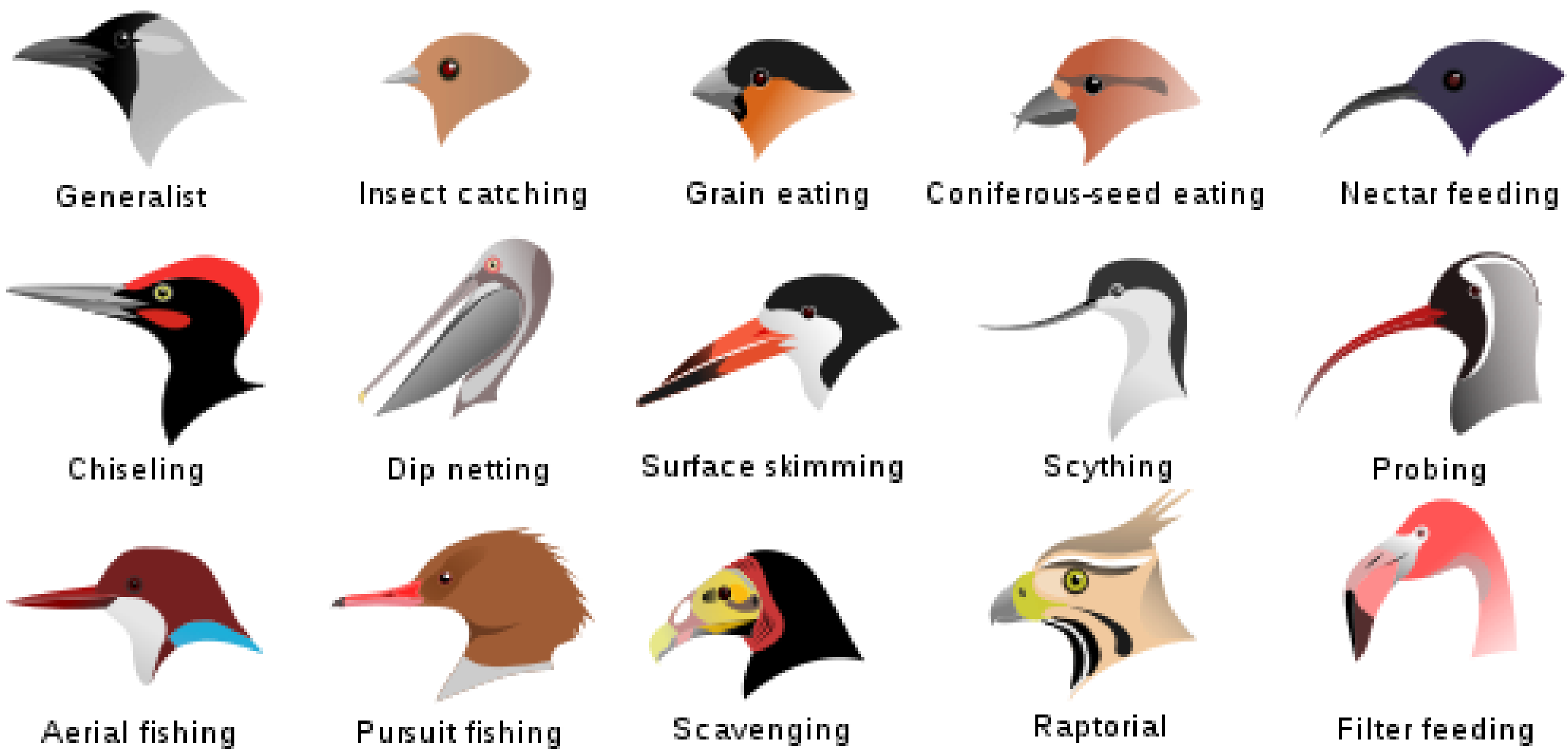


Image categorization

- Place recognition

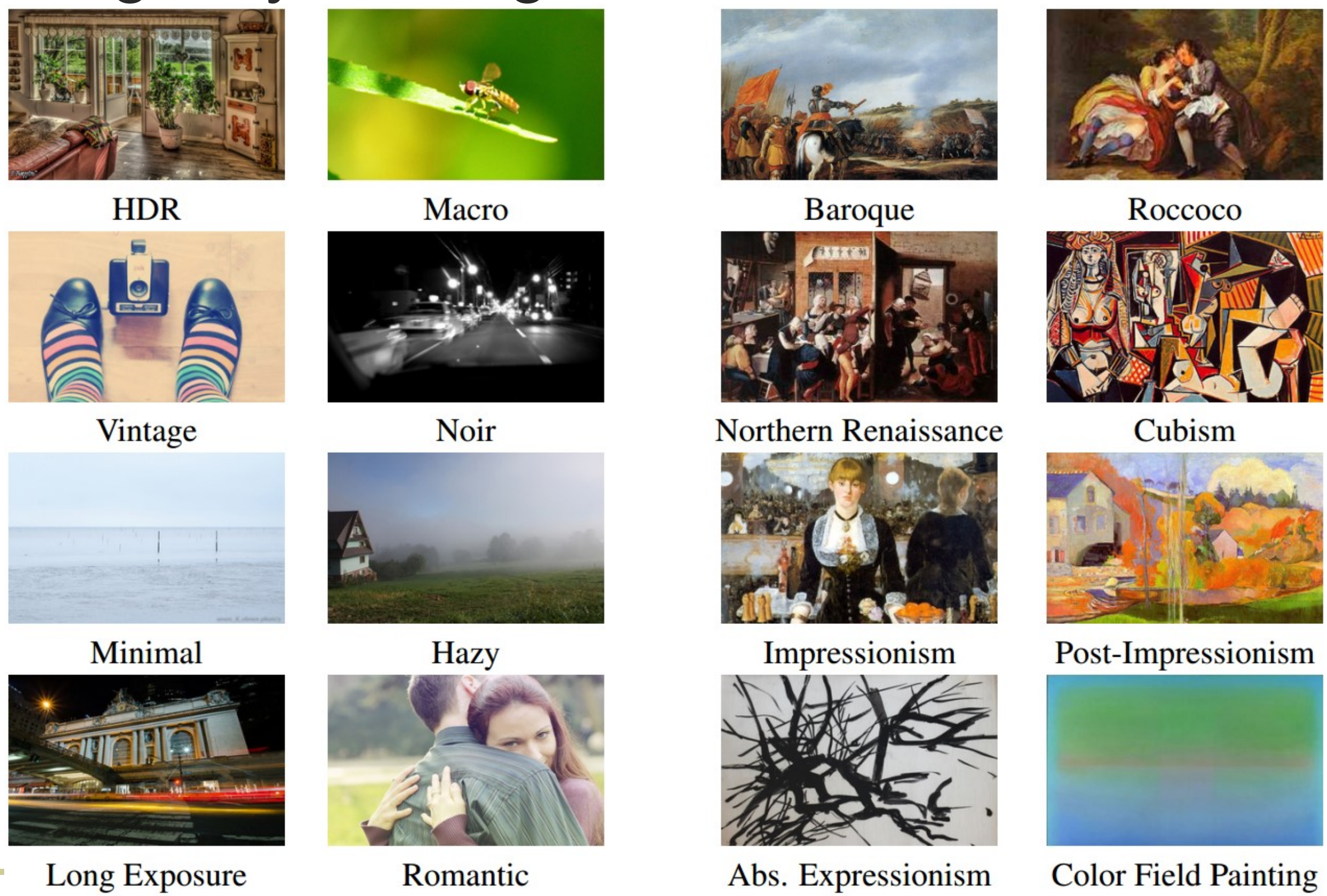
Image categorization

- Visual font recognition

[Chen et al. CVPR 2014]

Image categorization

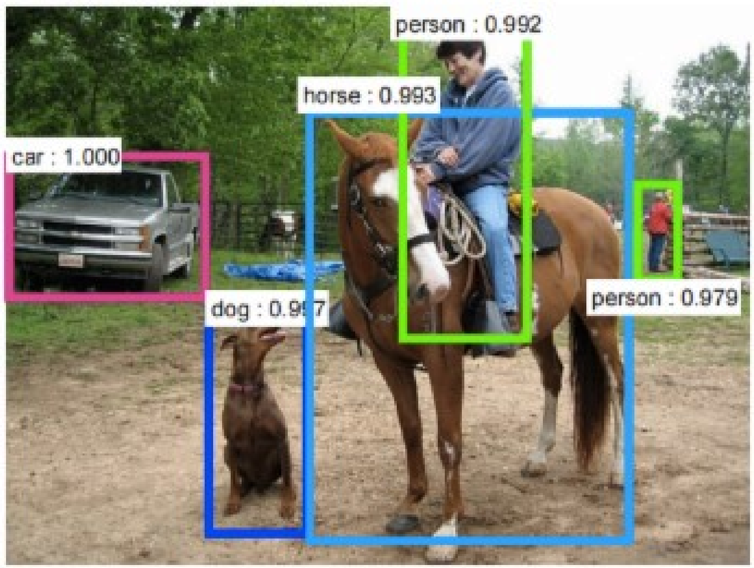
- Image style recognition

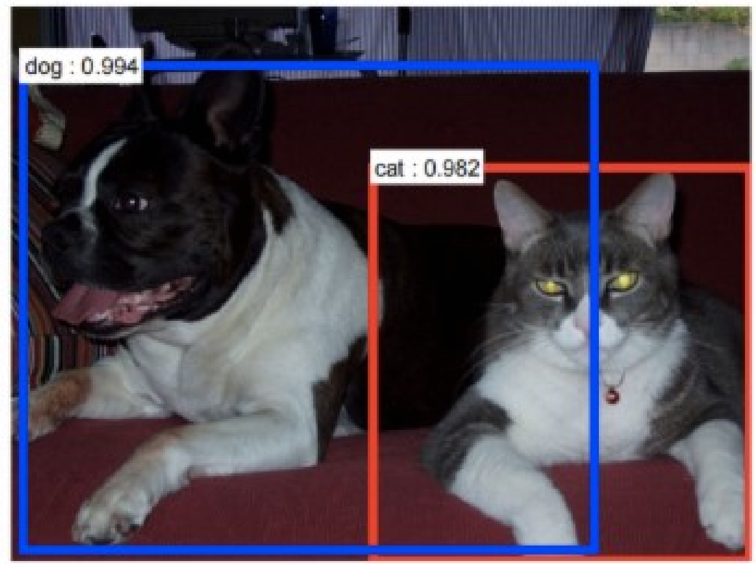


Flickr Style: 80K images covering 20 styles. Wikipaintings: 85K images for 25 art genres.

[Karayev et al. BMVC 2014]

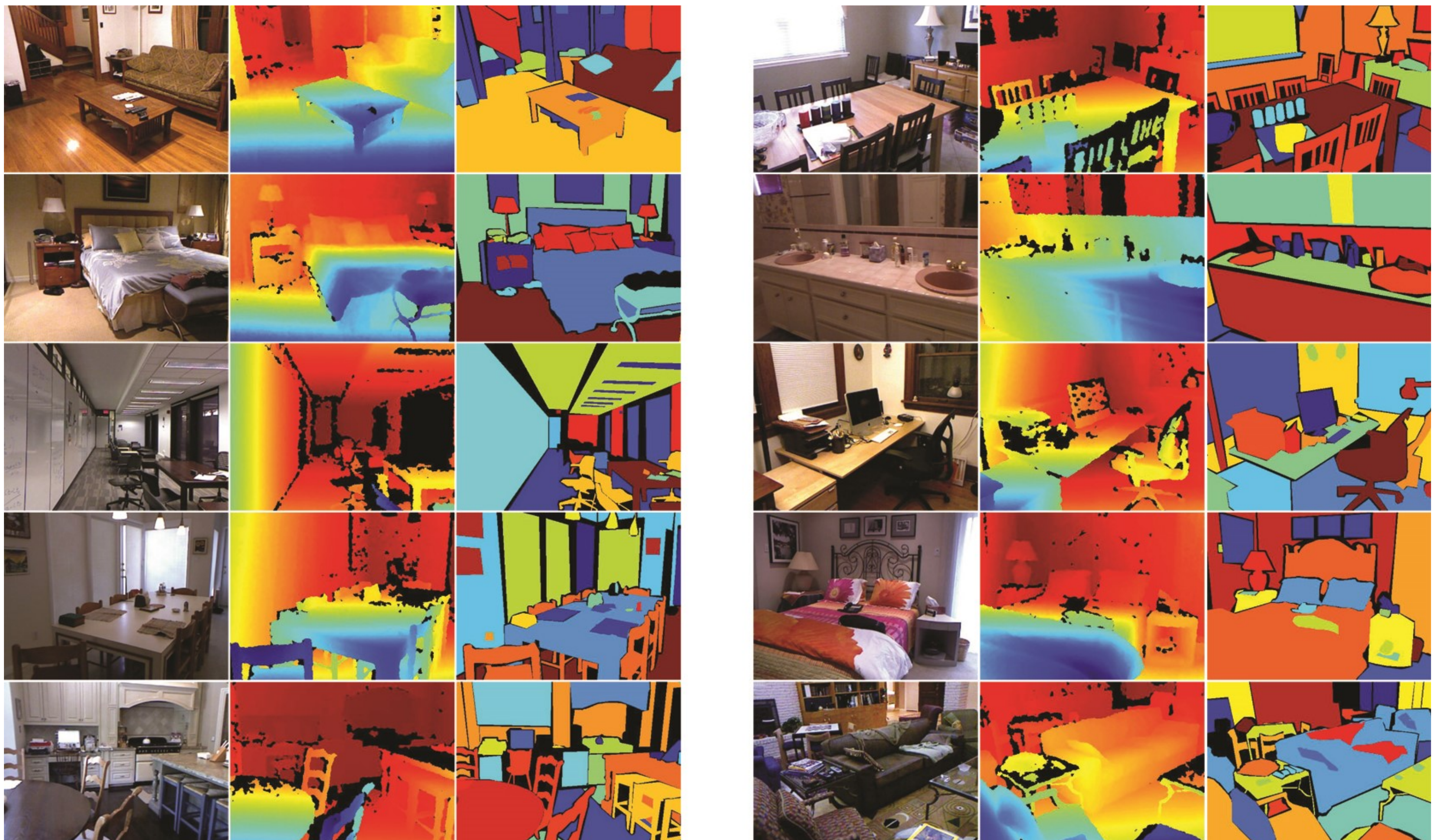
Object Category Detection





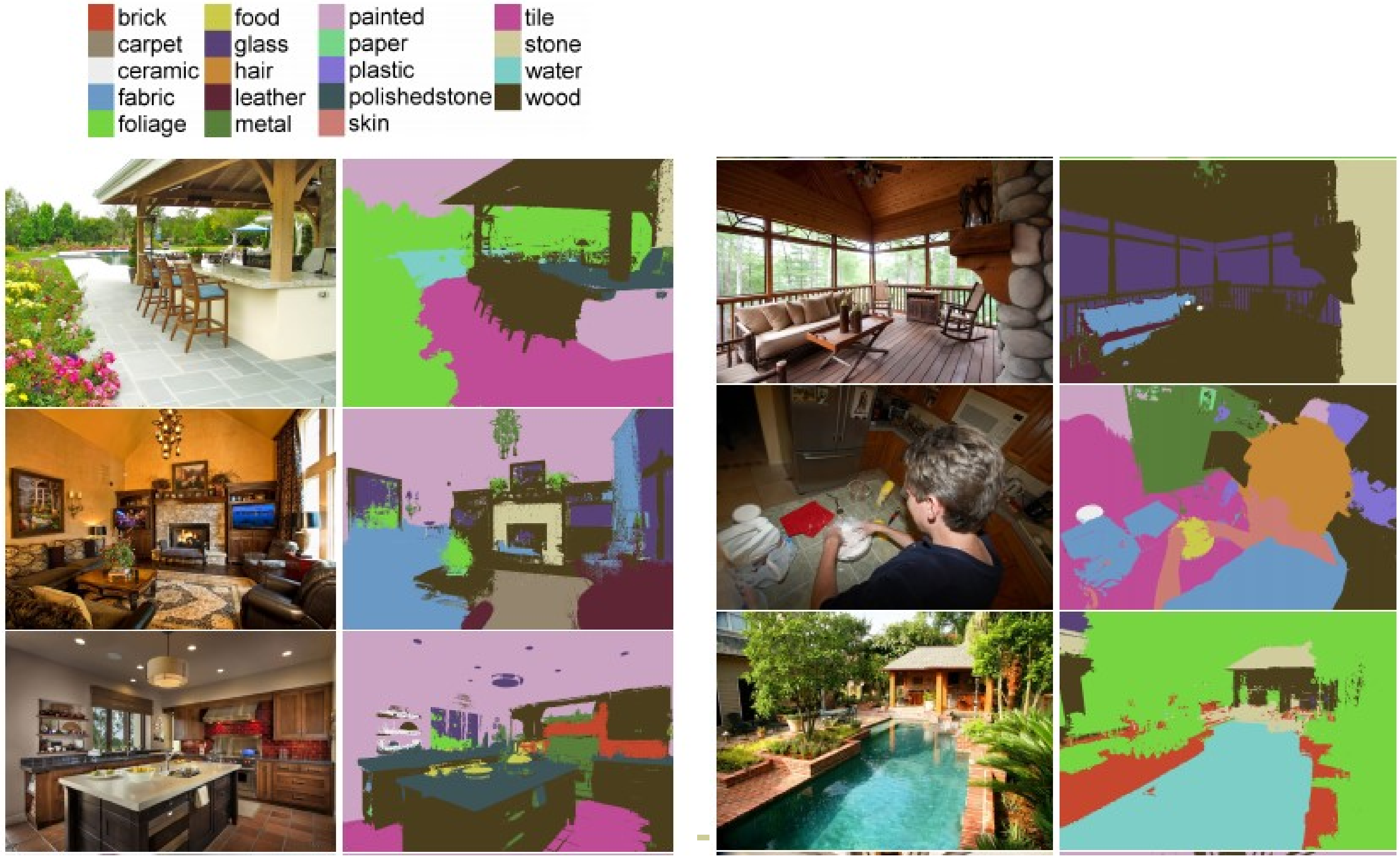
Region categorization

- Semantic segmentation from RGBD images



[Silberman et al. ECCV 2012]

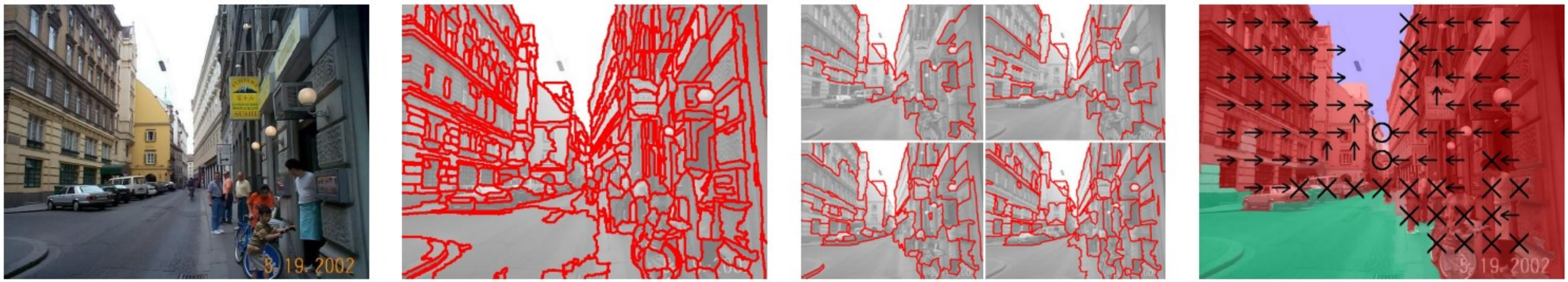
Region categorization



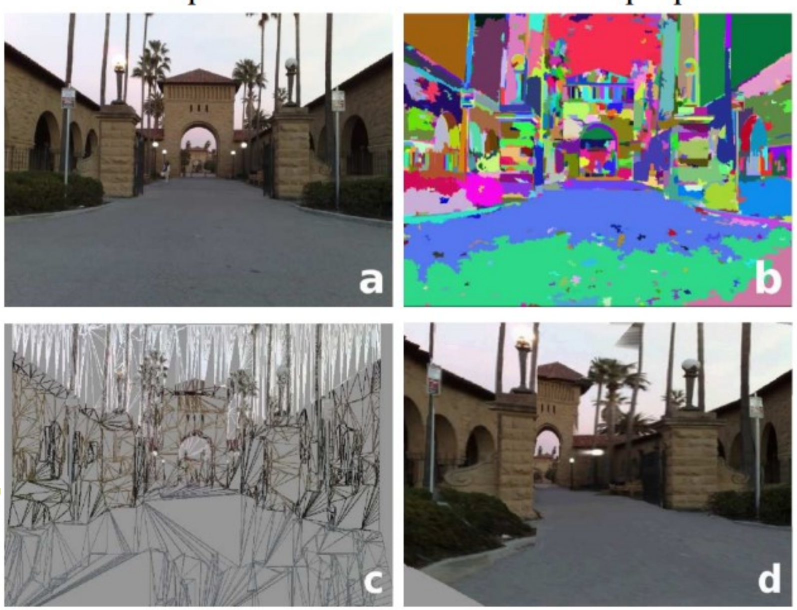
[Bell et al. CVPR 2015]

Region categorization

- Layout prediction



Input Superpixels Multiple Segmentations Surface Layout



Assign regions to orientation

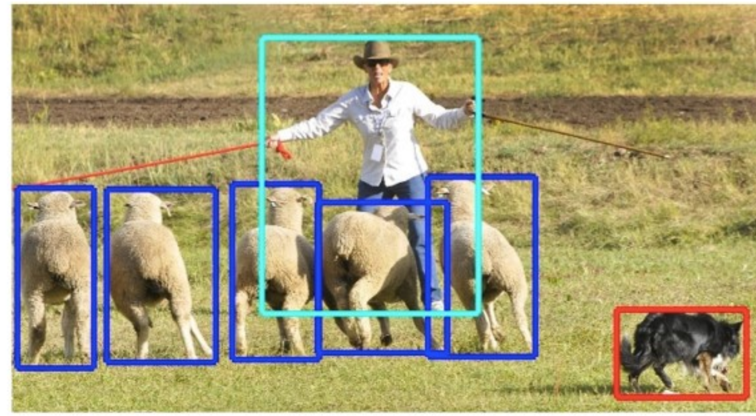
Geometric context [Hoiem et al. IJCV 2007]

Assign regions to depth

Make3D [Saxena et al. PAMI 2008]

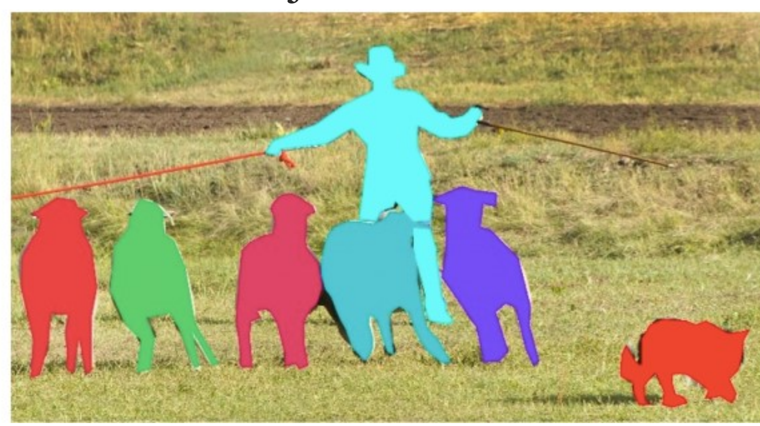
Detection, semantic segmentation, instance segmentation

image classification



object detection

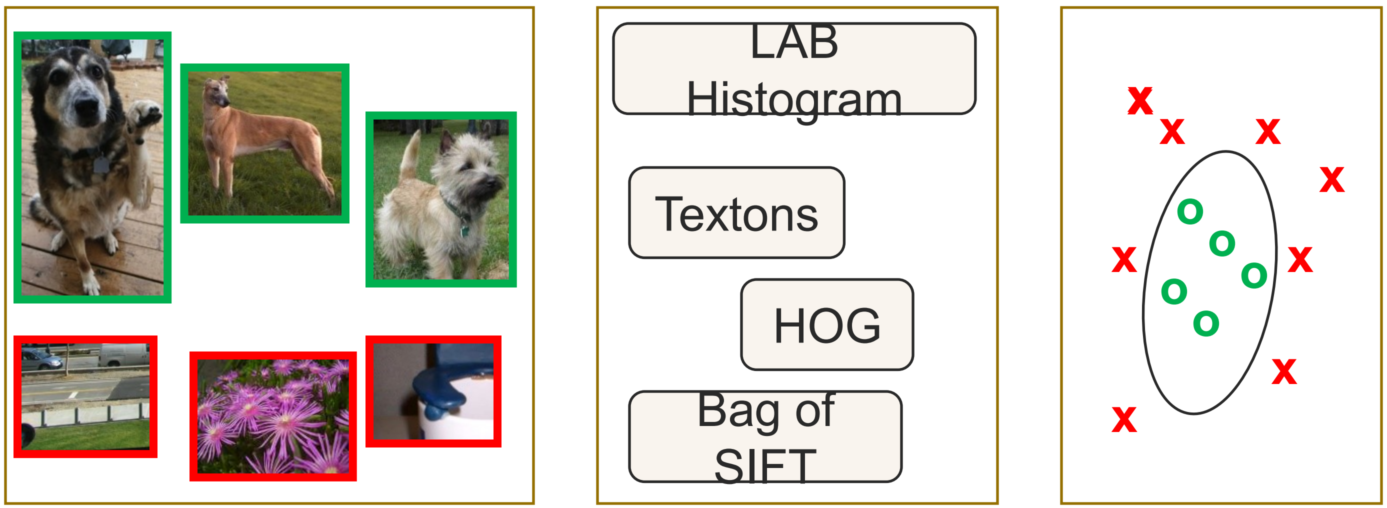
semantic segmentation



instance segmentation

Image source

Categorization from supervised learning



Examples + Image Features + Classifier = Category label

Training phase

Training

Training Labels

Training Images

Image Features

Classifier Training

Trained Classifie r

Testing phase

Training

Training Images

Image Features

Training Labels

Classifier Training

Trained Classifie r

Testing

Test Image

Image Features

Trained Classifier

Prediction

Outdoor

Today’ class

- Overview of image categorization
- **Spatial pyramids bag-of-words categorizer**
- Deep convolutional neural networks (CNNs)
- Object detection

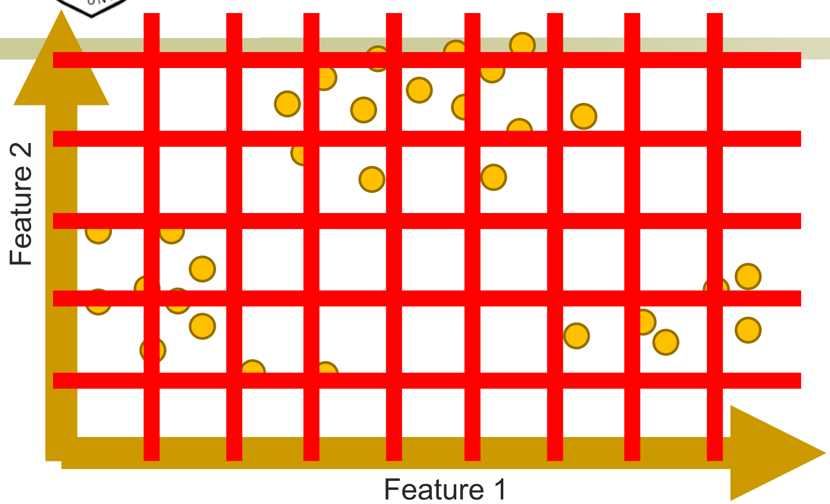
Example: Spatial Pyramid BoW Classifier

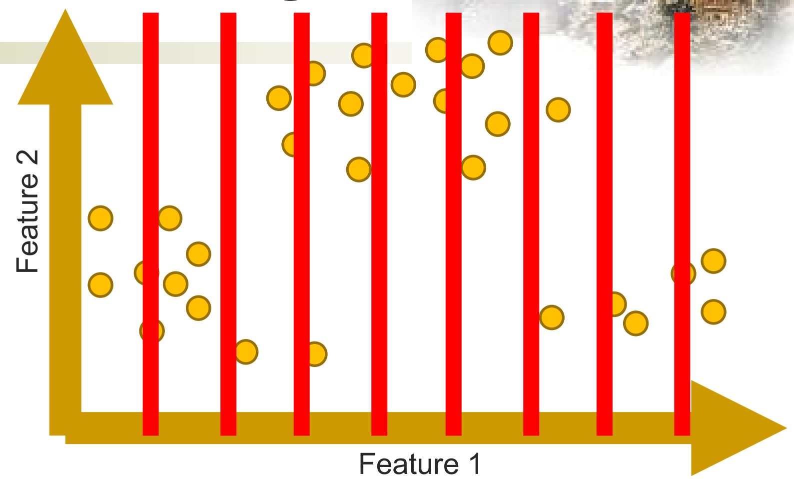
- Features: spatially binned histograms of clustered SIFT descriptors

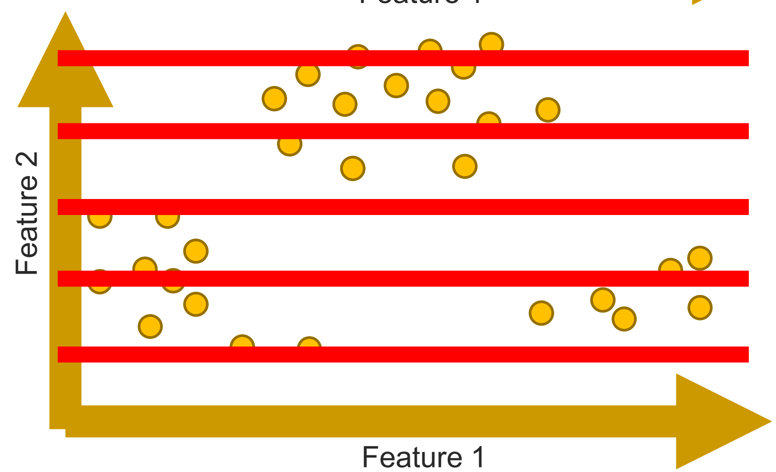
- Classifier: SVM

Lazebnik et al. CVPR 2006

Joint vs marginal binning







Joint histogram

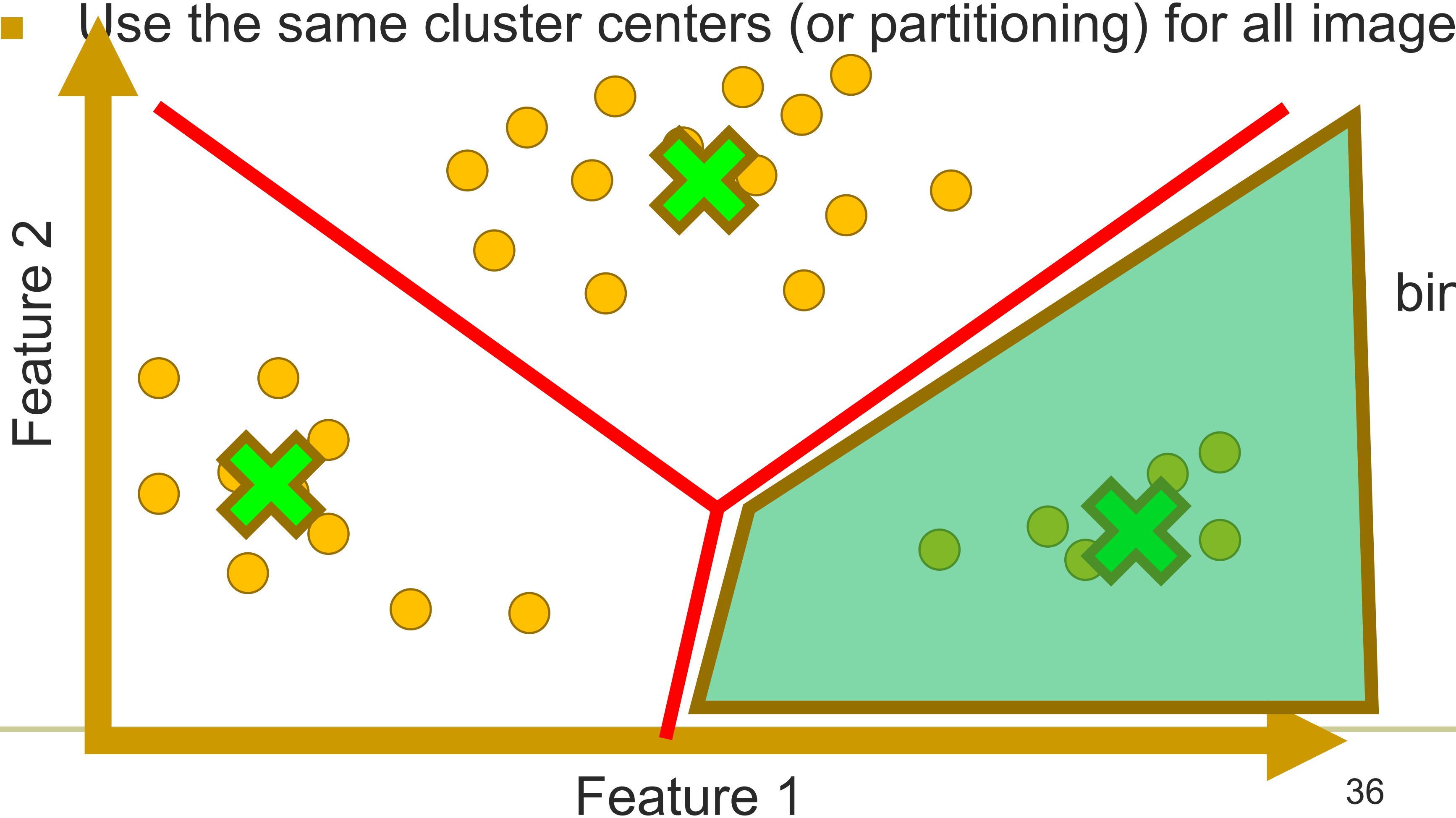
- Requires lots of data
- Loss of resolution to avoid empty bins

Marginal histogram

- Requires independent features
  - More data/bin than joint histogram

Histogram with clustering

- Cluster data (or partition) into K clusters, count how many samples appear in each cluster to get K-dim histogram
- Use the same cluster centers (or partitioning) for all images



Computing histogram distance

- Cosine similarity (dot product of normalized counts)
- Histogram intersection

$$\text{histint}(h_i, h_j) = 1 - \sum_{m=1}^{K} \min\left(h_i(m), h_j(m)\right)$$

- Chi-squared Histogram matching distance

$$\chi^2(h_i, h_j) = \frac{1}{2} \sum_{m=1}^{K} \frac{[h_i(m) - h_j(m)]^2}{h_i(m) + h_j(m)}$$

- Earth mover's distance (Cross-bin similarity measure)
  - minimal cost paid to transform one distribution into the other

[Rubner et al. The Earth Mover's Distance as a Metric for Image Retrieval, IJCV 2000]

Histograms: implementation issues

- Quantization
  - Grids: fast but applicable only with few dimensions
  - Clustering: slower but can quantize data in higher dimensions

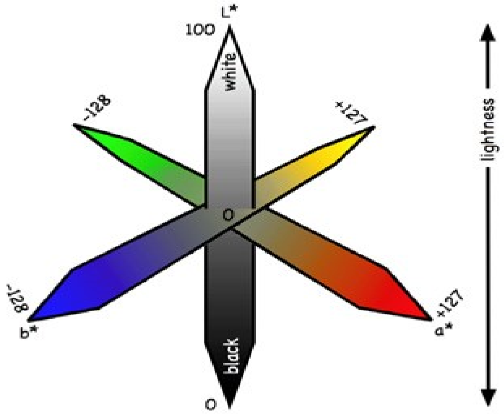
Few Bins
Need less data
Coarser representation

Many Bins
Need more data
Finer representation

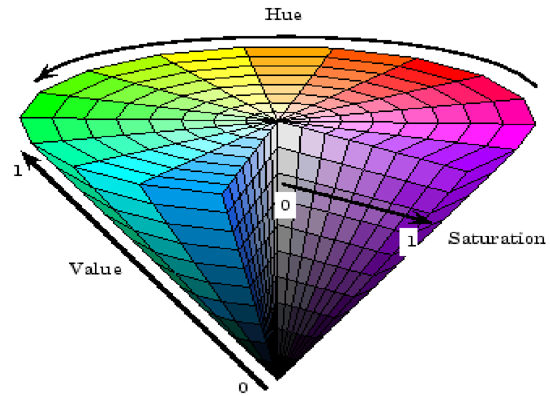
- Matching
  - Histogram intersection or Euclidean/Cosine may be faster
  - Chi-squared often works better
  - Earth mover's distance is good for when nearby bins represent similar values

What kind of things do we compute histograms of?

- Color

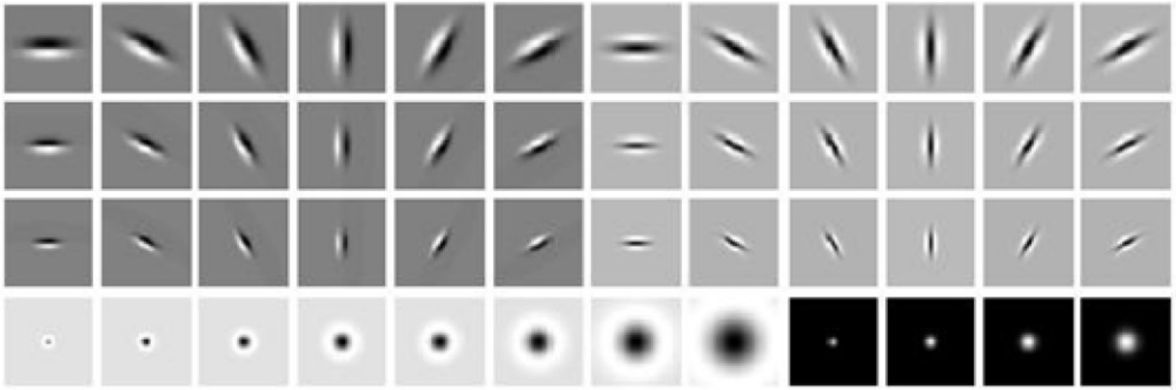


L*a*b* color space



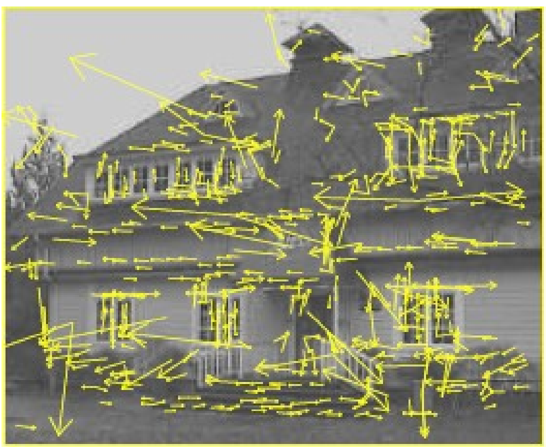
HSV color space

- Texture (filter banks or HOG over regions)



What kind of things do we compute histograms of?

- Histograms of descriptors



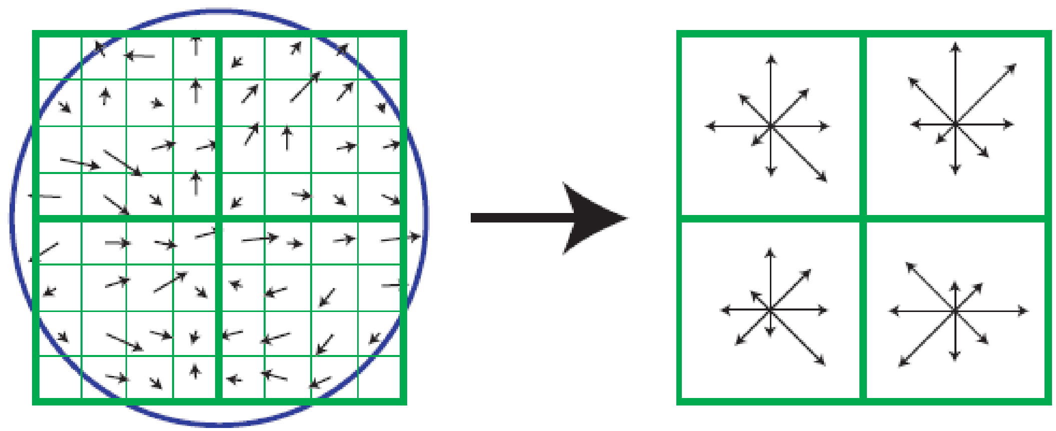


Image gradients

Keypoint descriptor

SIFT – [Lowe IJCV 2004]

- “Bag of visual words”

Image categorization with BoW

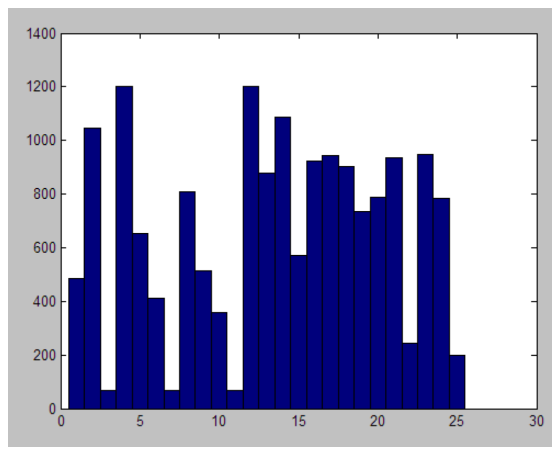
Training

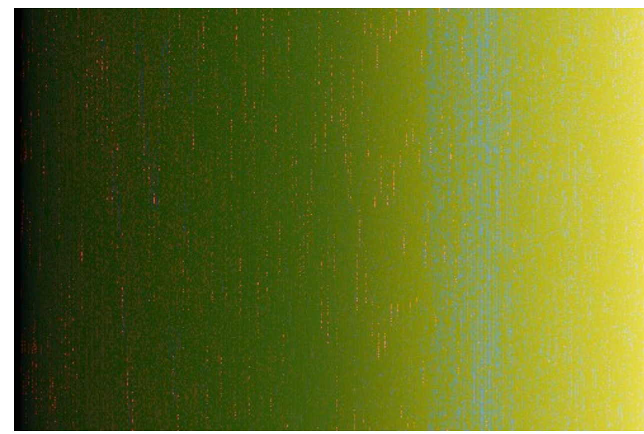
1. Extract keypoints and descriptors for all training images
2. Cluster descriptors
3. Quantize descriptors using cluster centers to get “visual words”
4. Represent each image by normalized counts of “visual words”
5. Train classifier on labeled examples using histogram values as features

Testing

1. Extract keypoints/descriptors and quantize into visual words
2. Compute visual word histogram
3. Compute label or confidence using classifier

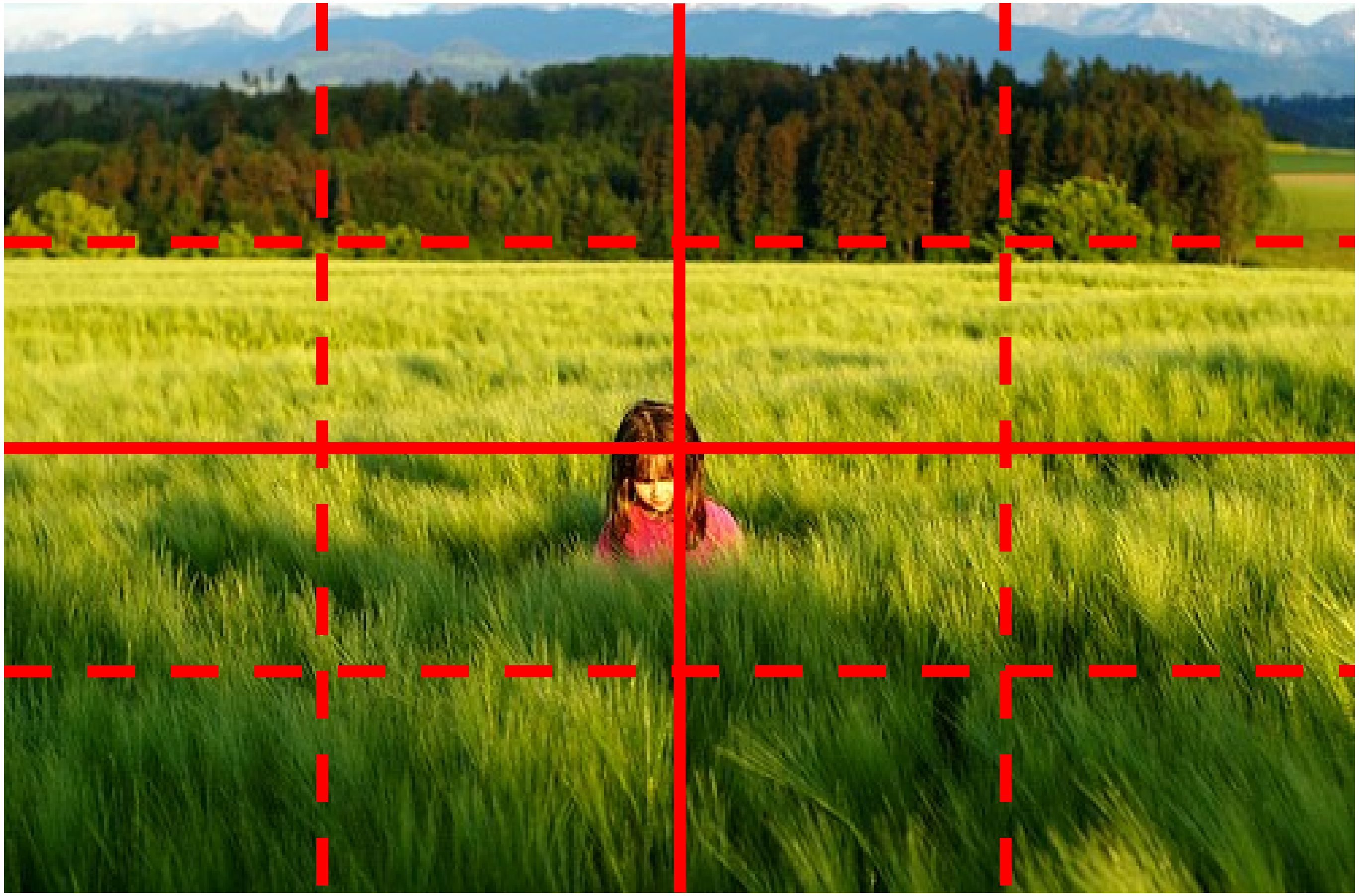
But what about spatial layout?





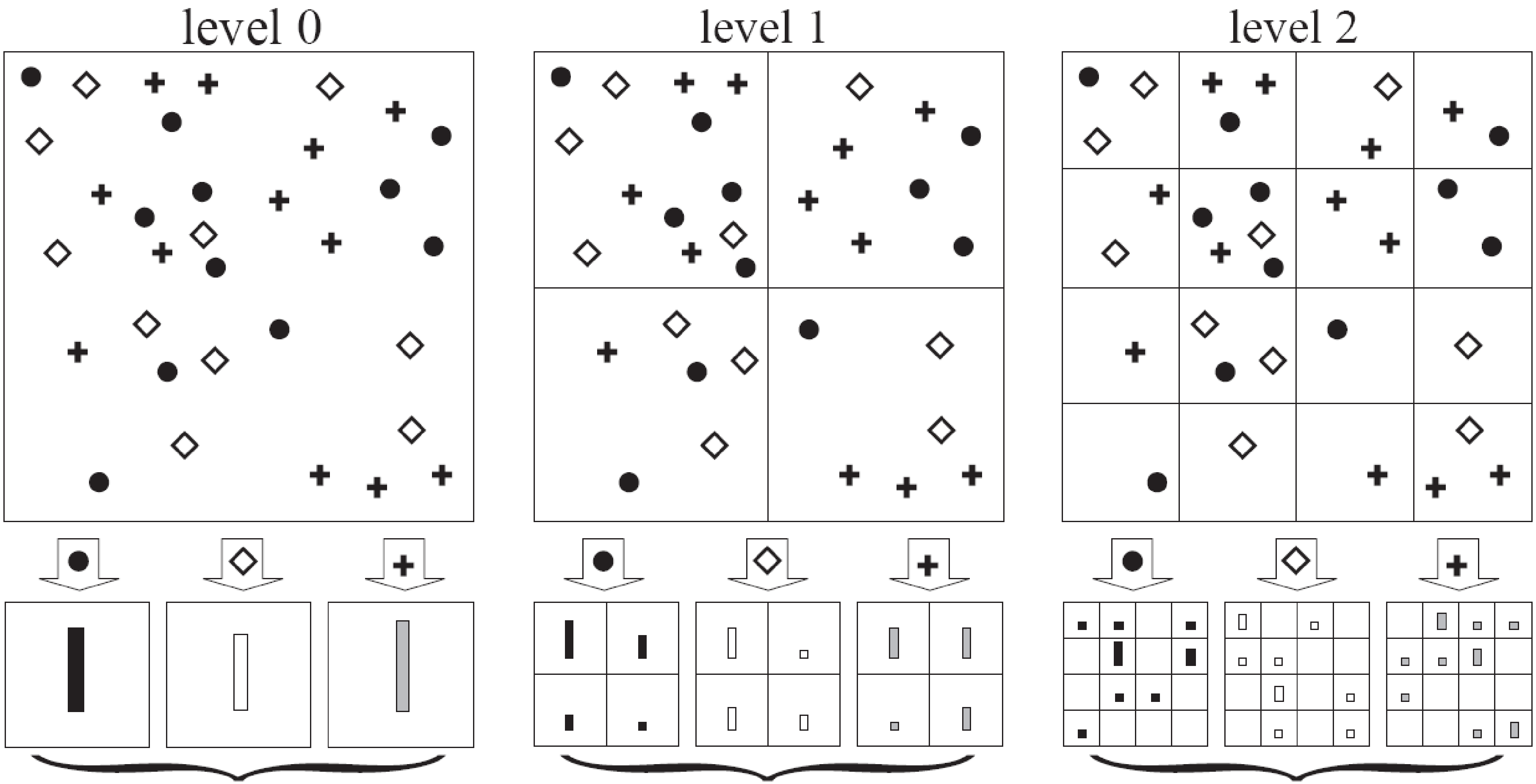
All of these images have the same color histogram

Spatial pyramid



Compute histogram in each spatial bin

Spatial pyramid



[Lazebnik et al. CVPR 2006]

Training

Training Images

→ **Spatial Pyramid BoW** → Classifier Training → Trained Classifier

Training Labels → Classifier Training

Testing

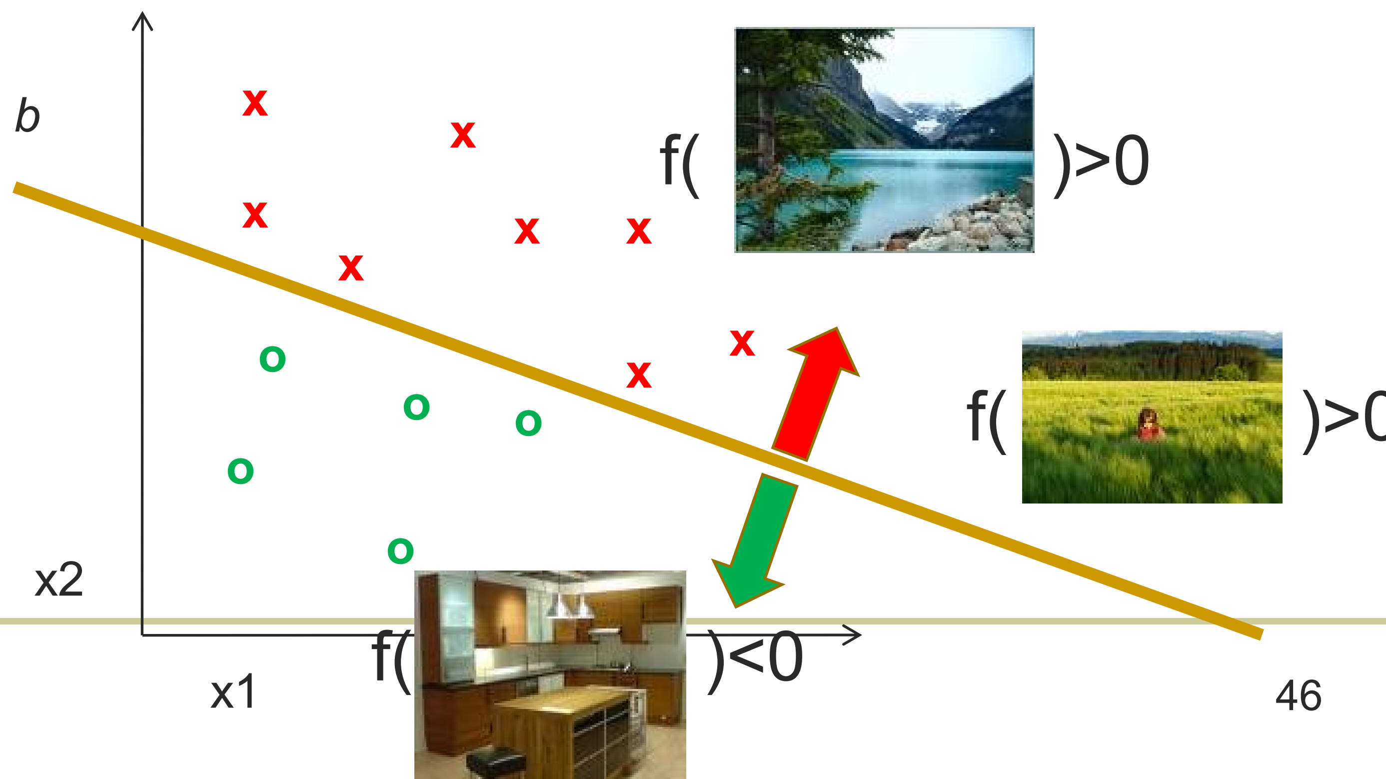
Test Image

→ **Spatial Pyramid BoW** → Trained Classifier → Prediction **Outdoor**

Linear SVM classifier

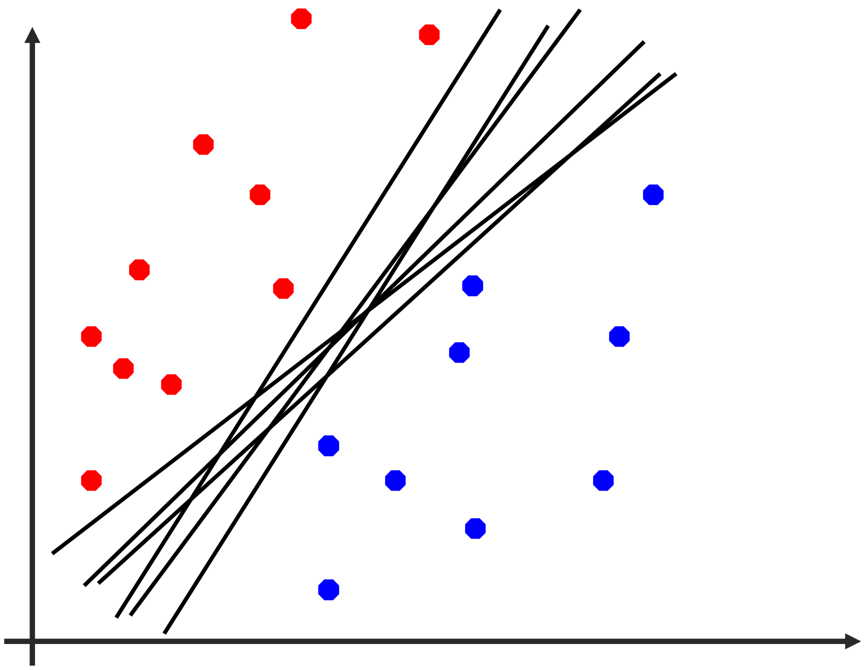
Find the hyperplane that separate examples of different categories

$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$



Linear Separators

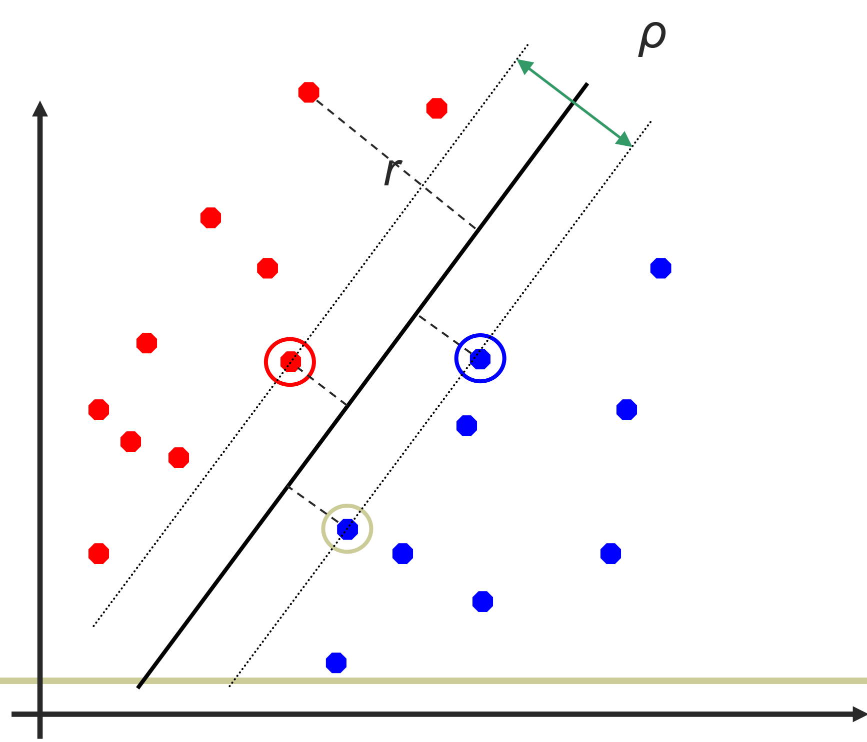
- Which of the linear separators is best?



http://www.cs.utexas.edu/~mooney/cs391L/slides/**svm.ppt**
http://www1.cs.columbia.edu/~belhumeur/courses/biometrics/2010/svm.ppt

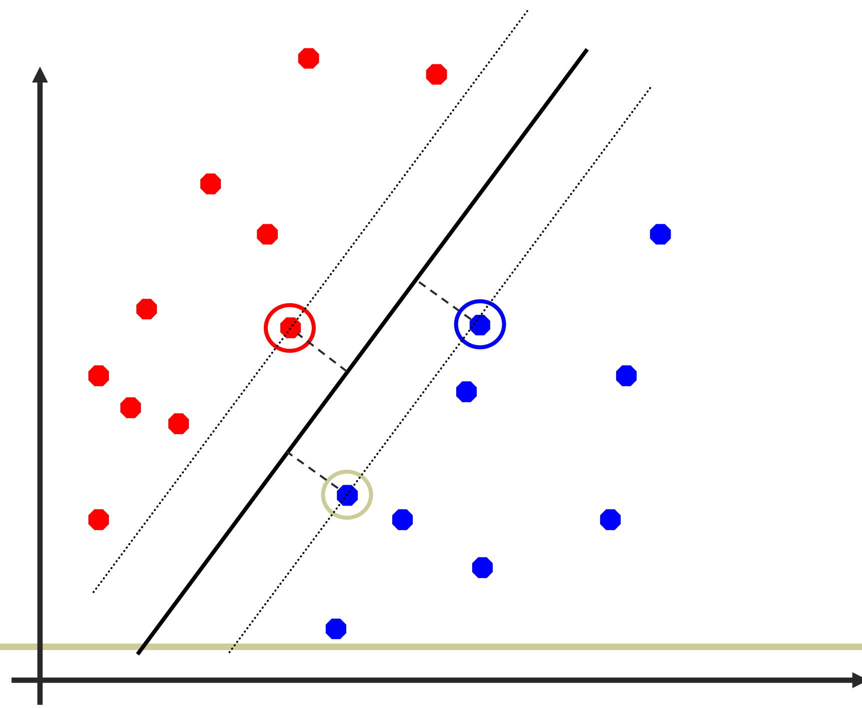
Classification Margin

- Distance from example $\mathbf{x}_i$ to the separator is $r = \frac{\mathbf{w}^T \mathbf{x}_i + b}{\|\mathbf{w}\|}$
- Examples closest to the hyperplane are ***support vectors***.
- ***Margin*** *ρ* of the separator is the distance between support vectors.



Maximum Margin Classification

- Implies that only support vectors matter; other training examples are ignorable.



Spatial Pyramids Results

minaret (97.6%) windsor chair (94.6%) joshua tree (87.9%) okapi (87.8%)

cougar body (27.6%) beaver (27.5%) crocodile (25.0%) ant (25.0%)

	Weak features		Strong features (200)	
L	Single-level	Pyramid	Single-level	Pyramid
0	15.5 ±0.9		41.2 ±1.2	
1	31.4 ±1.2	32.8 ±1.3	55.9 ±0.9	57.0 ±0.8
2	47.2 ±1.1	49.3 ±1.4	63.6 ±0.9	**64.6** ±0.8
3	52.2 ±0.8	**54.0** ±1.1	60.3 ±0.9	64.6 ±0.7

Table 2. Classification results for the Caltech-101 database.

Recap: Spatial Pyramid BoW Classifier

- Features
  1. Extract dense SIFT (spatially pooled and normalized histograms of gradients)
  2. Assign each SIFT vector to a cluster number
  3. Compute histograms of spatially pooled clustered SIFT vectors
  - Variations like Fisher vectors and 2nd order pooling shown to improve performance

- Classifier
  - Linear SVM (or slightly better performance with chi-squared or histint SVMs)

Today’ class

- Overview of image categorization
- Spatial pyramids bag-of-words categorizer
- **Deep convolutional neural networks (CNNs)**
- Object detection

New training setup with moderate sized datasets

Training Labels

Training Images

Tune CNN features and Neural Network classifier

Trained Classifier

Initialize CNN Features

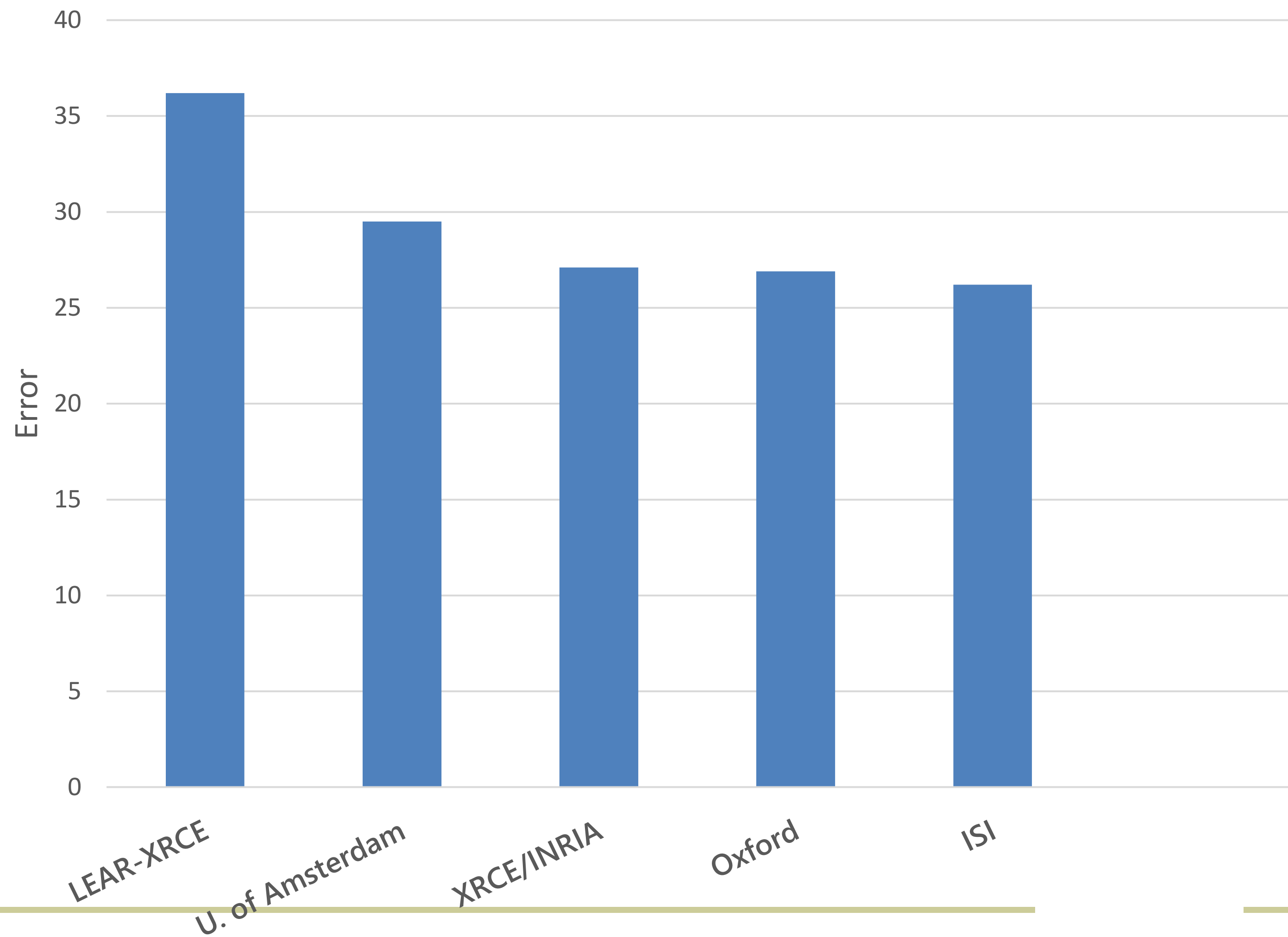
Dataset similar to task with millions of labeled examples

History of deep convolutional nets

- 1950's: neural nets (perceptron) invented by Rosenblatt
- 1980's/1990's: Neural nets are popularized and then abandoned as being interesting idea but impossible to optimize or "unprincipled"
- 1990's: LeCun achieves state-of-art performance on character recognition with convolutional network (main ideas of today's networks)
- 2000's: Hinton, Bottou, Bengio, LeCun, Ng, and others keep trying stuff with deep networks but without much traction/acclaim in vision
- 2010-2011: Substantial progress in some areas, but vision community still unconvinced
  - Some neural net researchers get ANGRY at being ignored/rejected
- 2012: shock at ECCV 2012 with ImageNet challenge

2012 ImageNet 1K

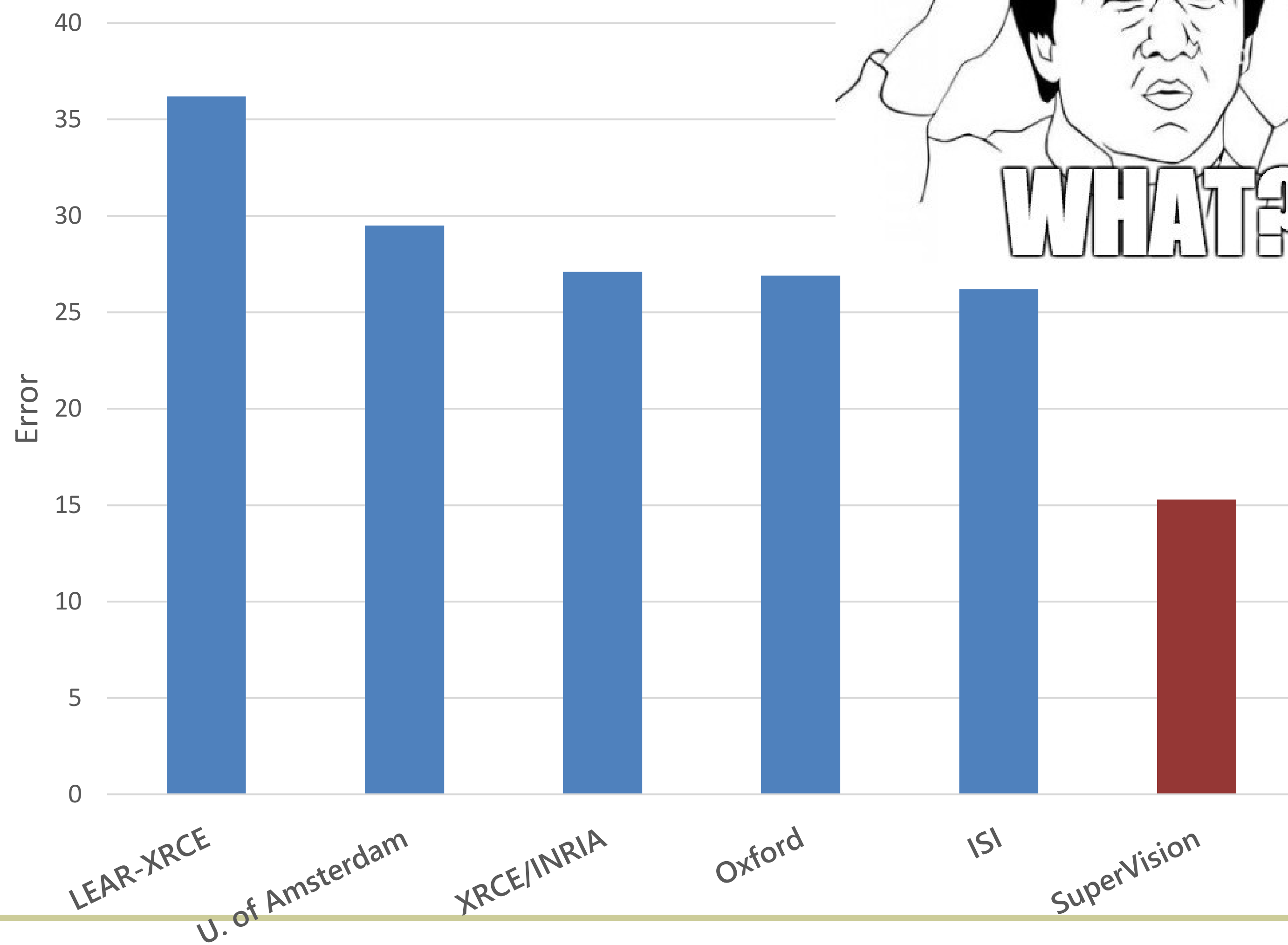
(Fall 2012)

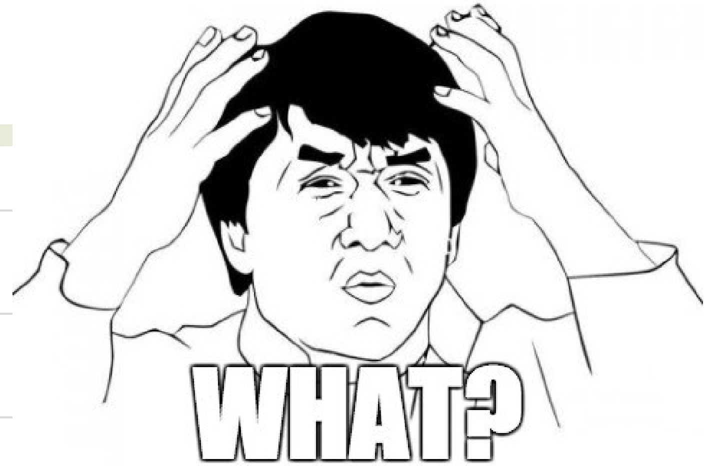


Slide: Jia-bin Huang

2012 ImageNet 1K

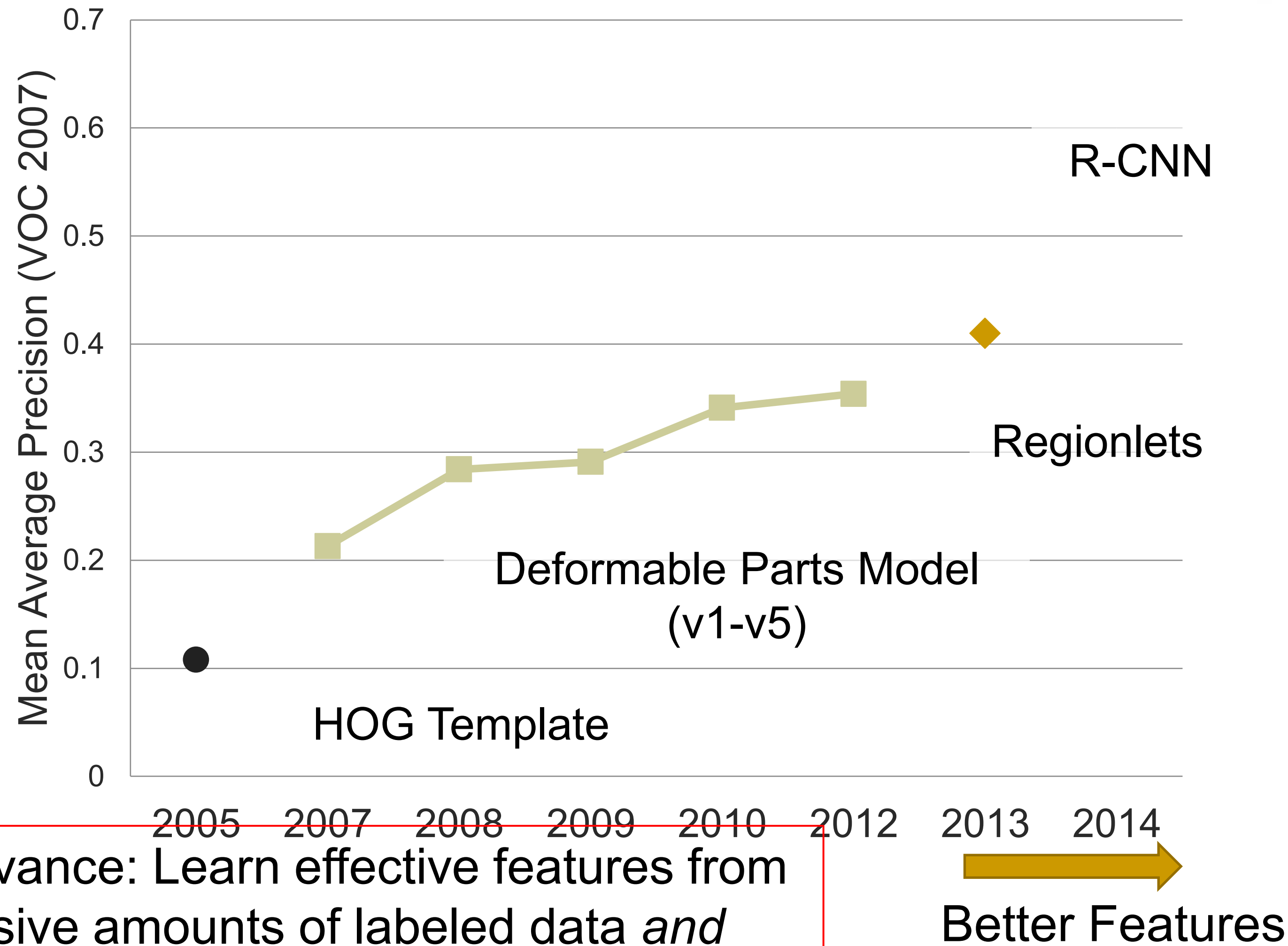
(Fall 2012)





Slide: Jia-bin Huang

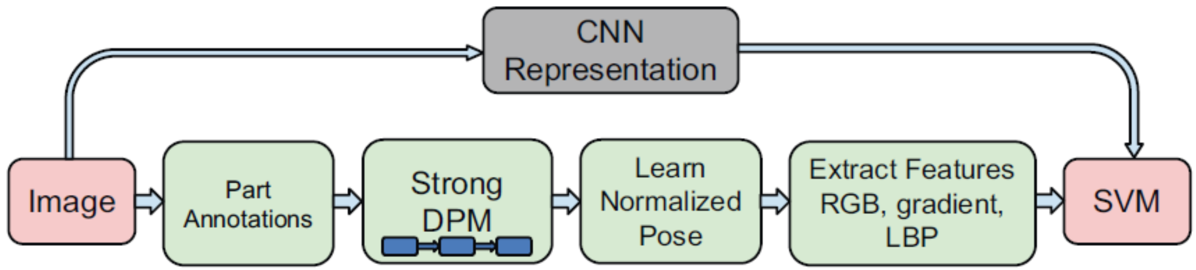
Improvements in Object Detection

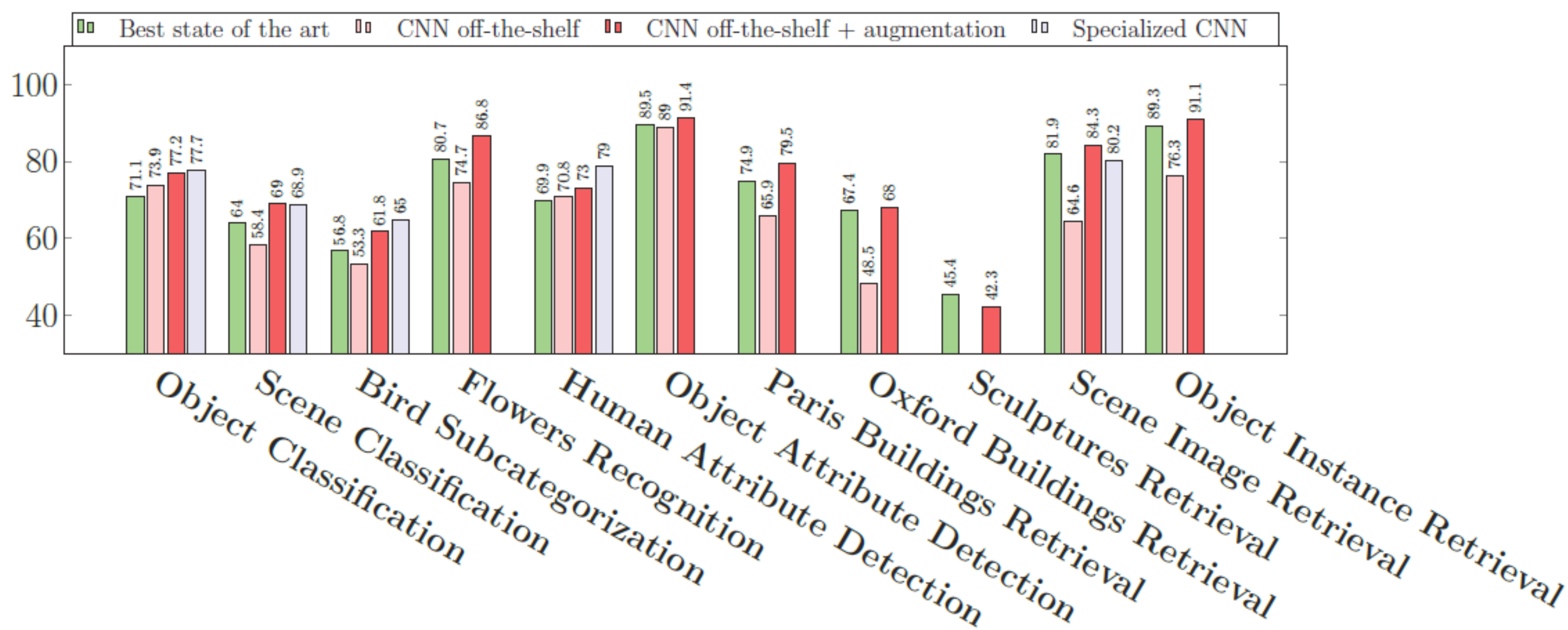


Key Advance: Learn effective features from massive amounts of labeled data *and* adapt to new tasks with less data

HOG: Dalal-Triggs 2005 DPM: Felzenszwalb et al. 2008-2012 Regionlets: Wang et al. 2013 R-CNN: Girshick et al. 2014

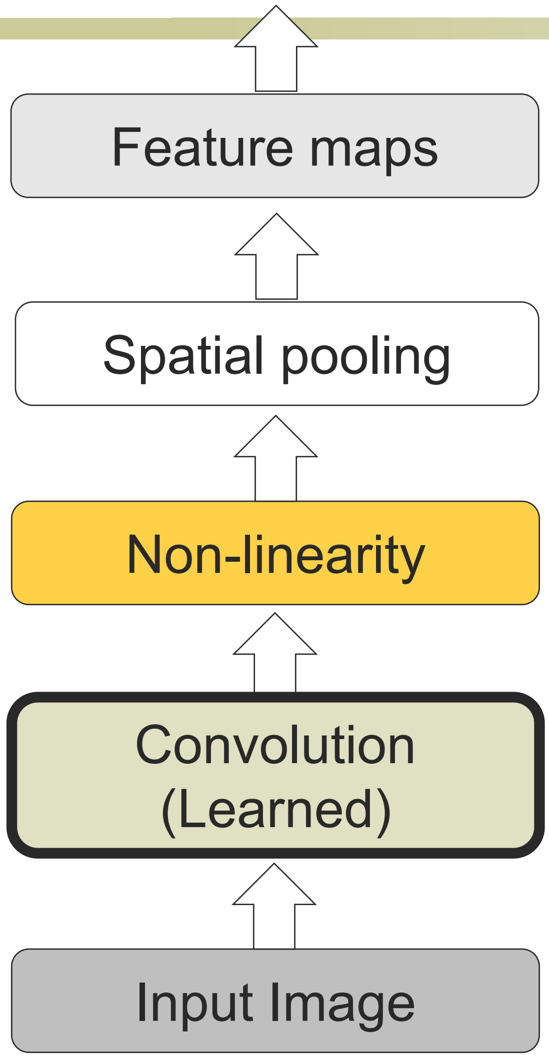
"CNN Features off-the-shelf: an Astounding Baseline for Recognition"

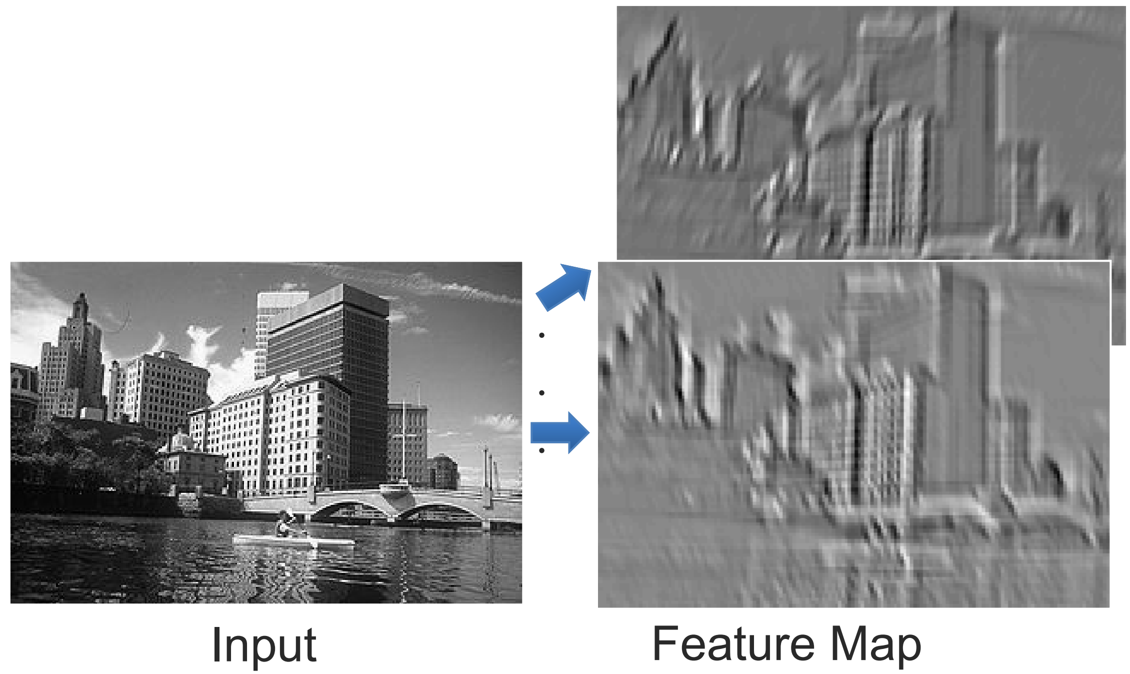




Razavian et al. CVPR 2014

Key operations in a CNN



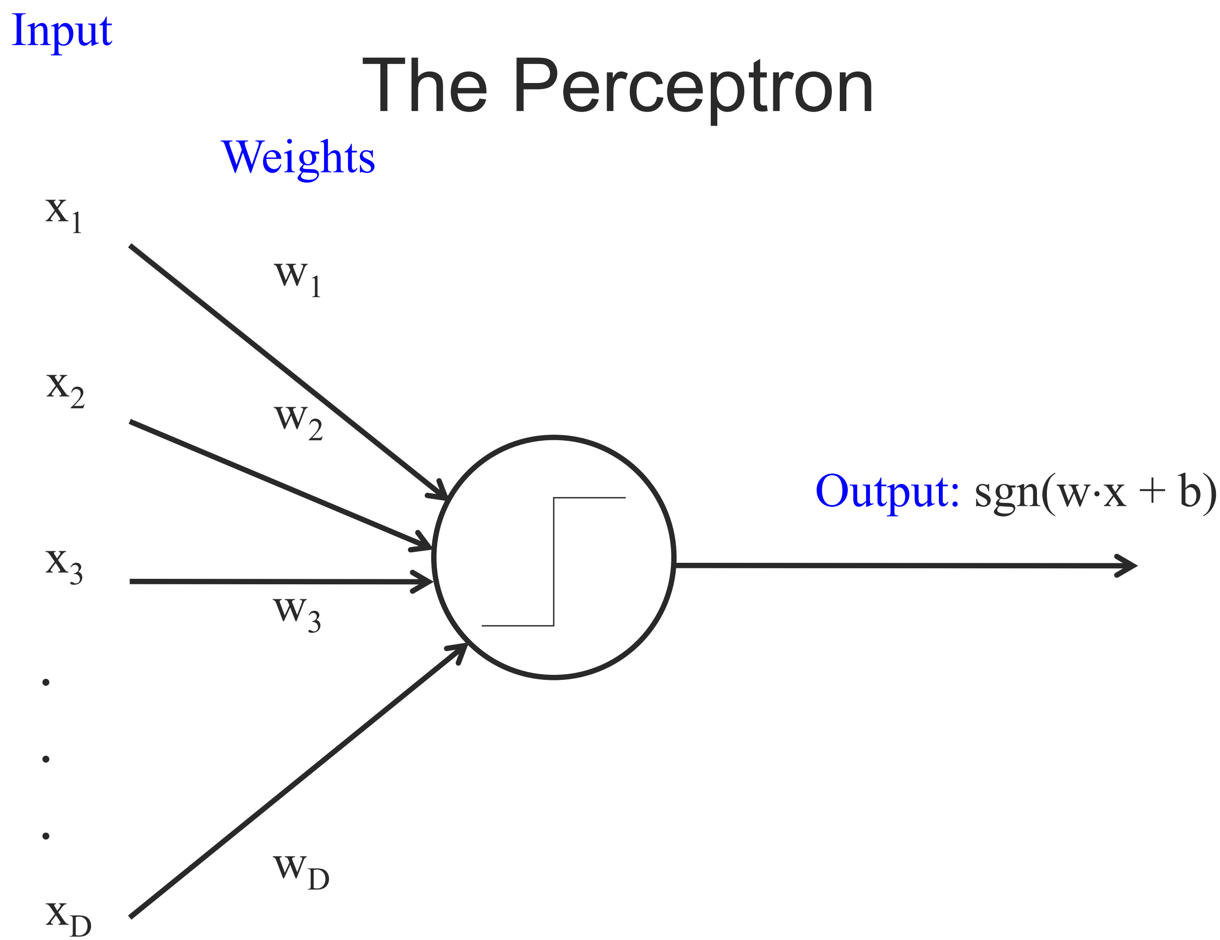


Source: R. Fergus, Y. LeCun

Slide: Lazebnik

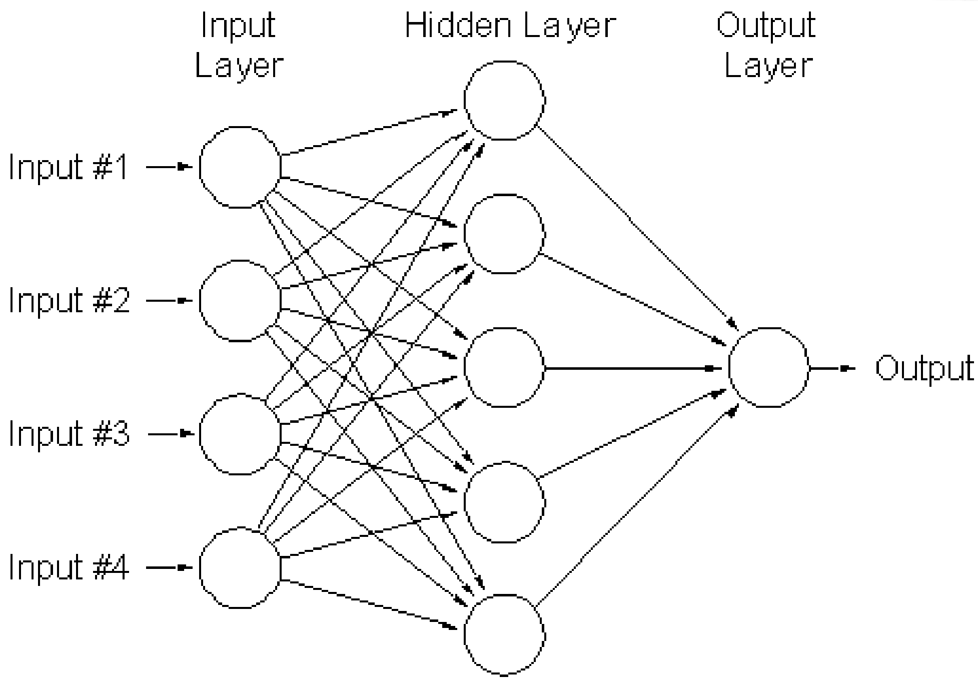
Rewind…

The Perceptron

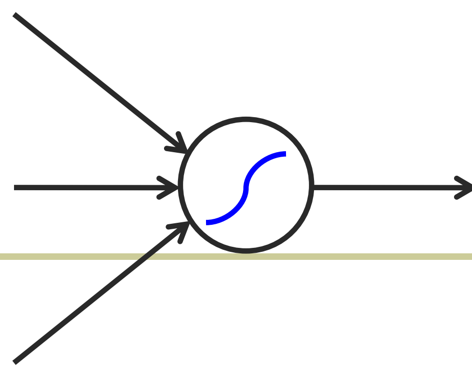


Rosenblatt, Frank (1958), The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Cornell Aeronautical Laboratory, Psychological Review, v65, No. 6, pp. 386–408.

Two-layer neural network



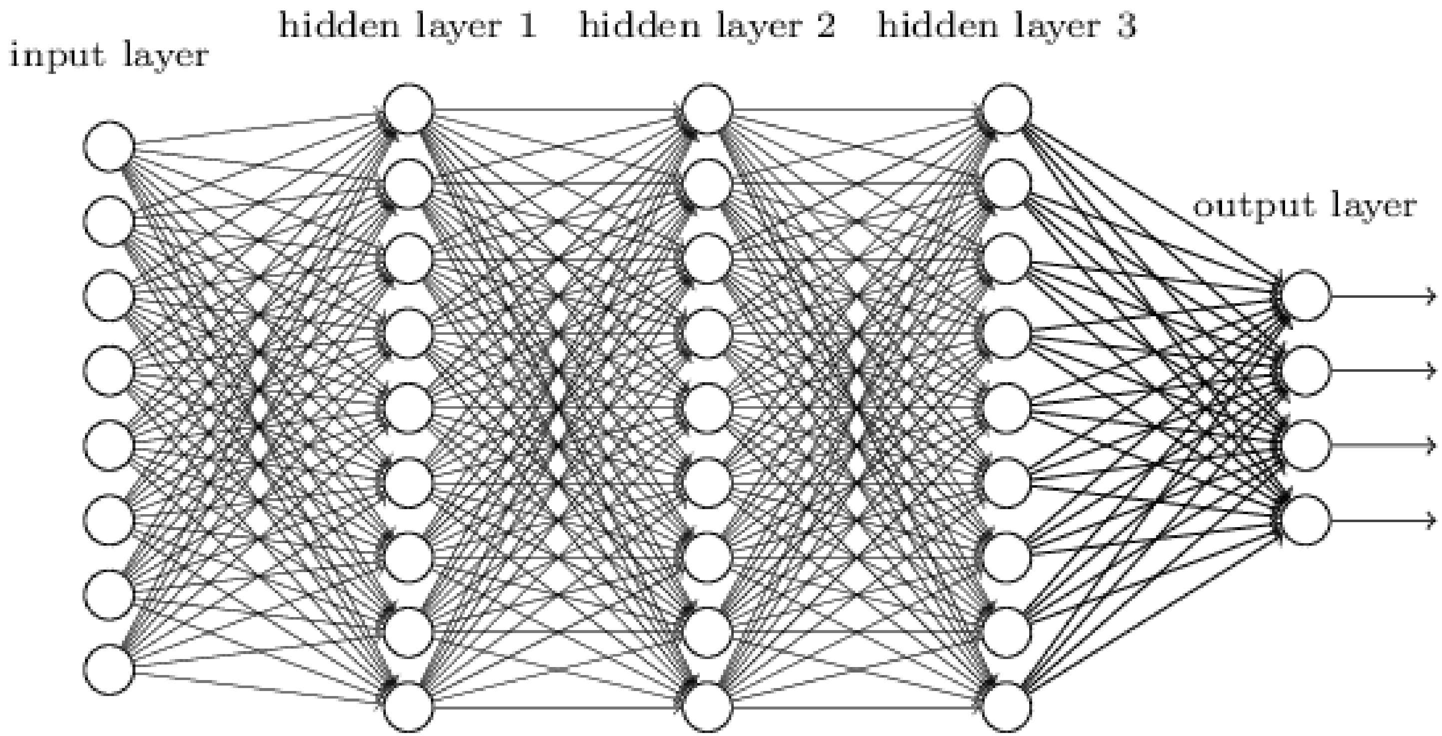
- Can learn nonlinear functions provided each perceptron has a differentiable nonlinearity



Sigmoid: $g(t) = \frac{1}{1+e^{-t}}$

Slide: Lazebnik

Multi-layer neural network



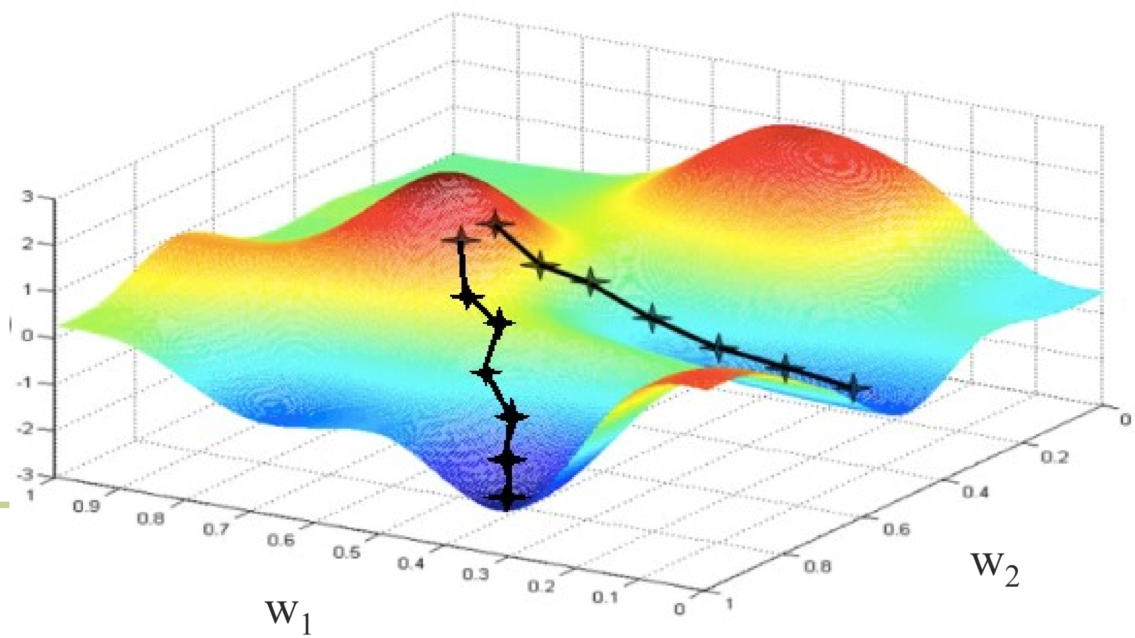
Slide: Lazebnik

Training of multi-layer networks

- Find network weights to minimize the *training error* between true and estimated labels of training examples, e.g.:

$$E(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - f_{\mathbf{w}}(\mathbf{x}_i) \right)^2$$

- Update weights by **gradient descent:** $\mathbf{w} \leftarrow \mathbf{w} - \alpha \frac{\partial E}{\partial \mathbf{w}}$



Slide: Lazebnik

Training of multi-layer networks

- Find network weights to minimize the *training error* between true and estimated labels of training examples, e.g.:

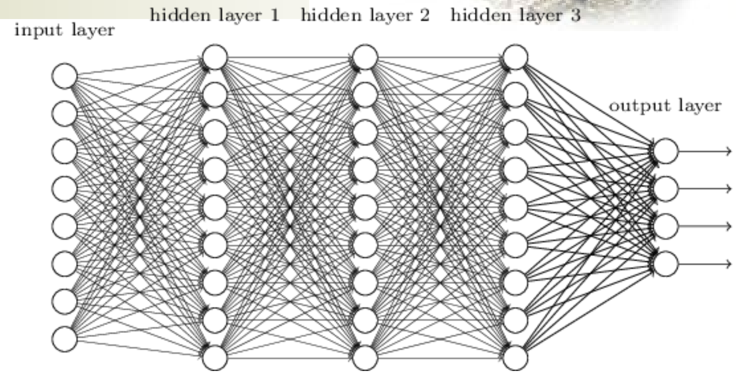
$$E(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - f_{\mathbf{w}}(\mathbf{x}_i)\right)^2$$

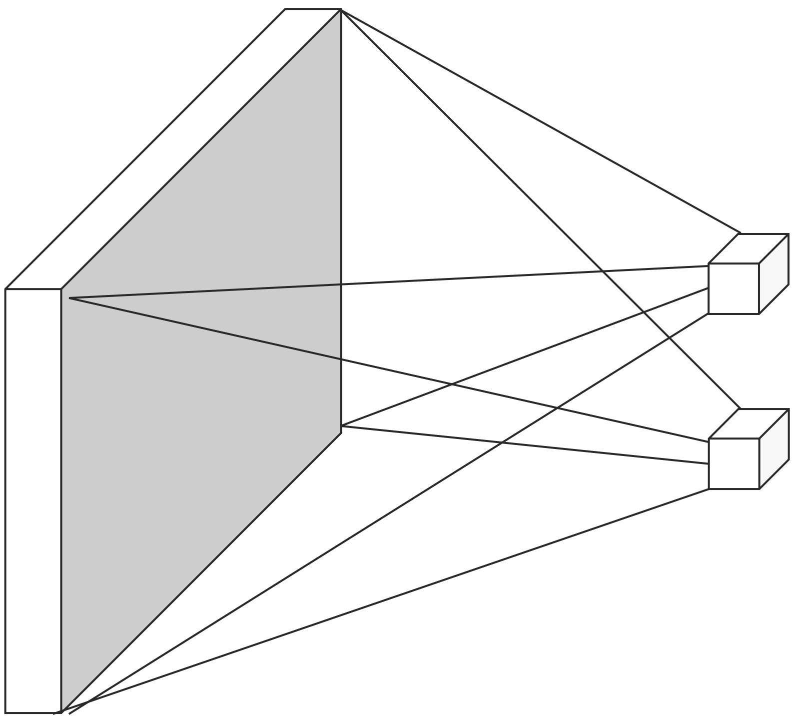
- Update weights by **gradient descent:** $\mathbf{w} \leftarrow \mathbf{w} - \alpha \dfrac{\partial E}{\partial \mathbf{w}}$

- **Back-propagation:** gradients are computed in the direction from output to input layers and combined using chain rule
- **Stochastic gradient descent:** compute the weight update w.r.t. a small batch of examples at a time, cycle through training examples in random order in multiple epochs

Slide: Lazebnik

From fully connected to convolutional networks



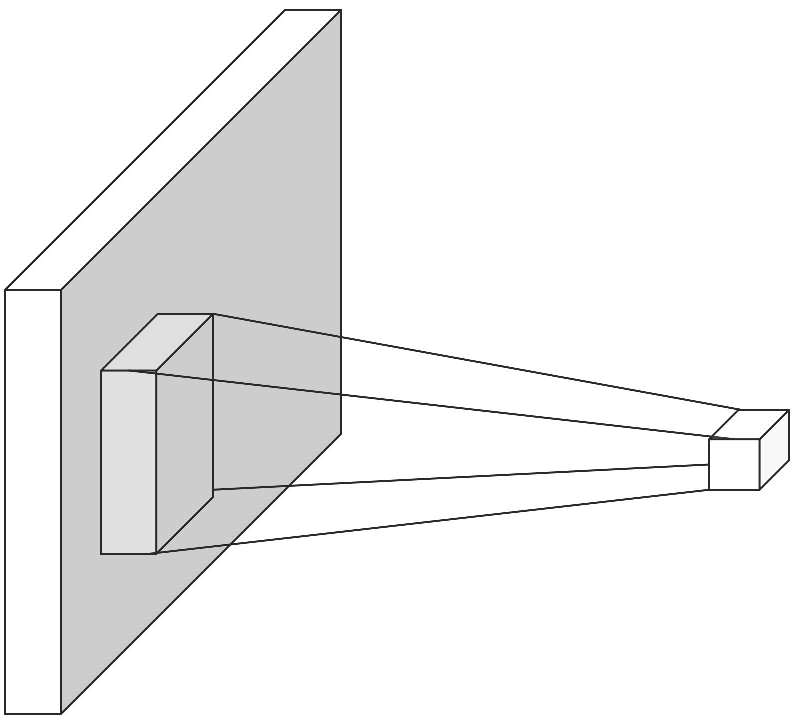


image

Fully connected layer

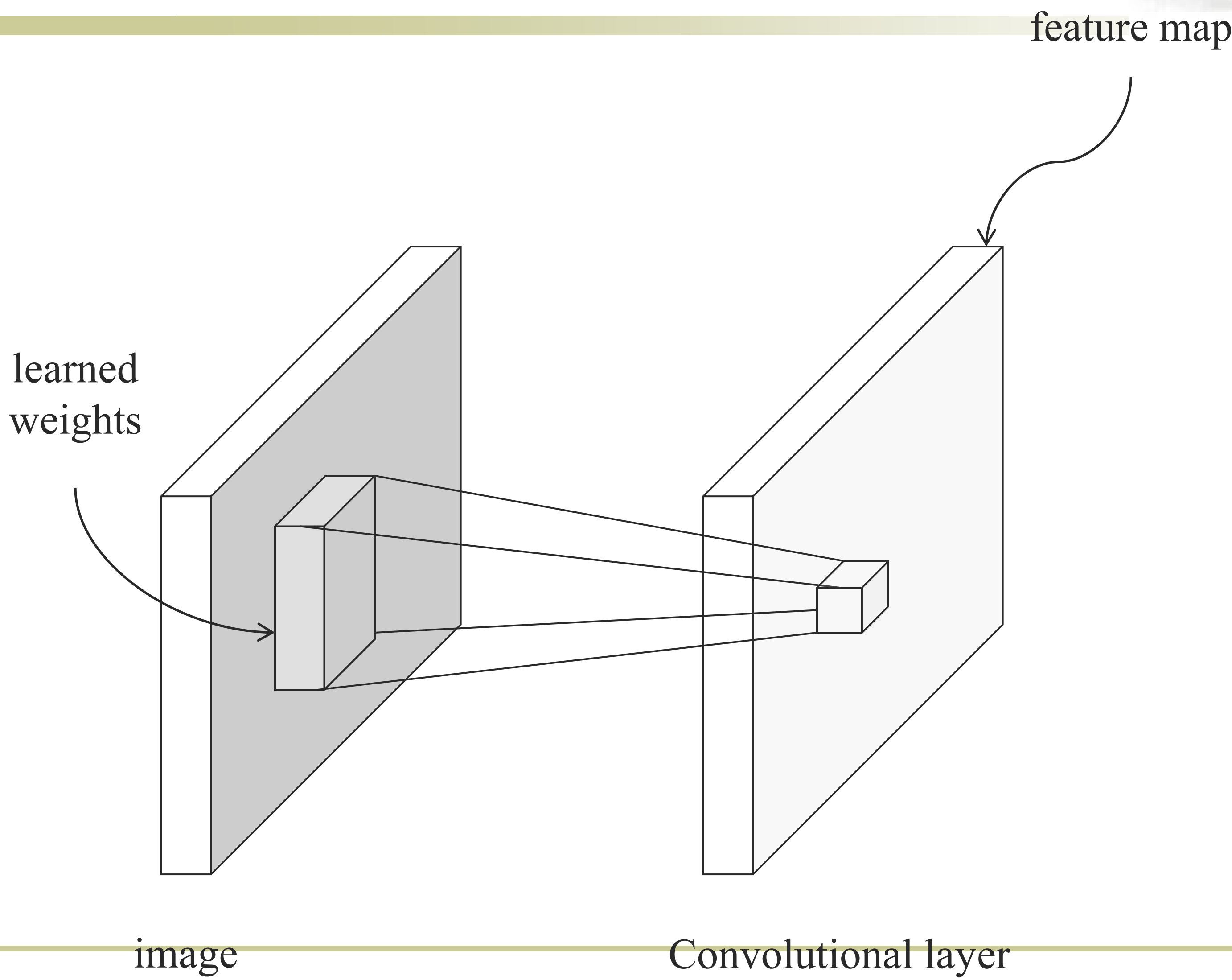
Slide: Lazebnik

From fully connected to convolutional networks



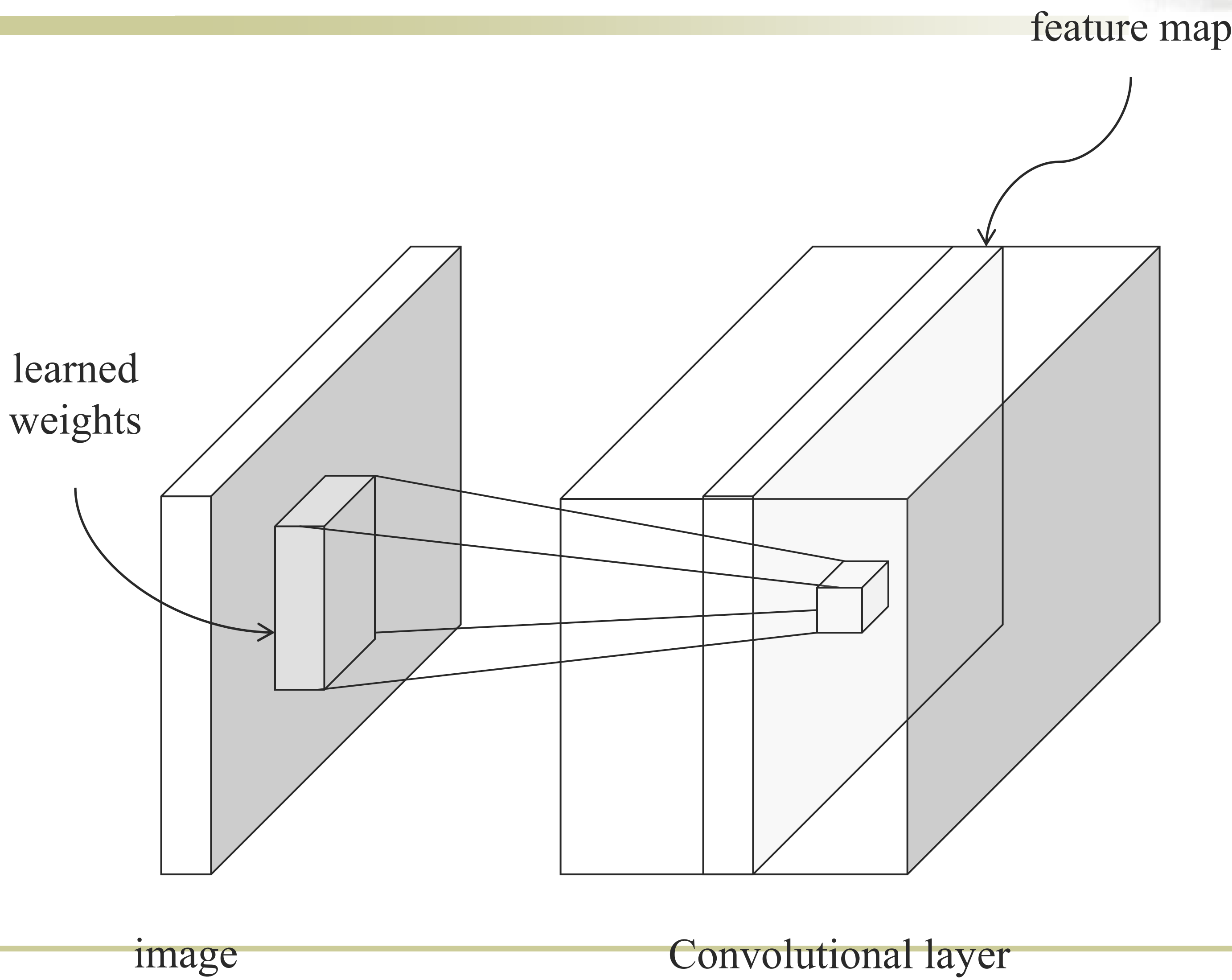
Slide: Lazebnik

From fully connected to convolutional networks



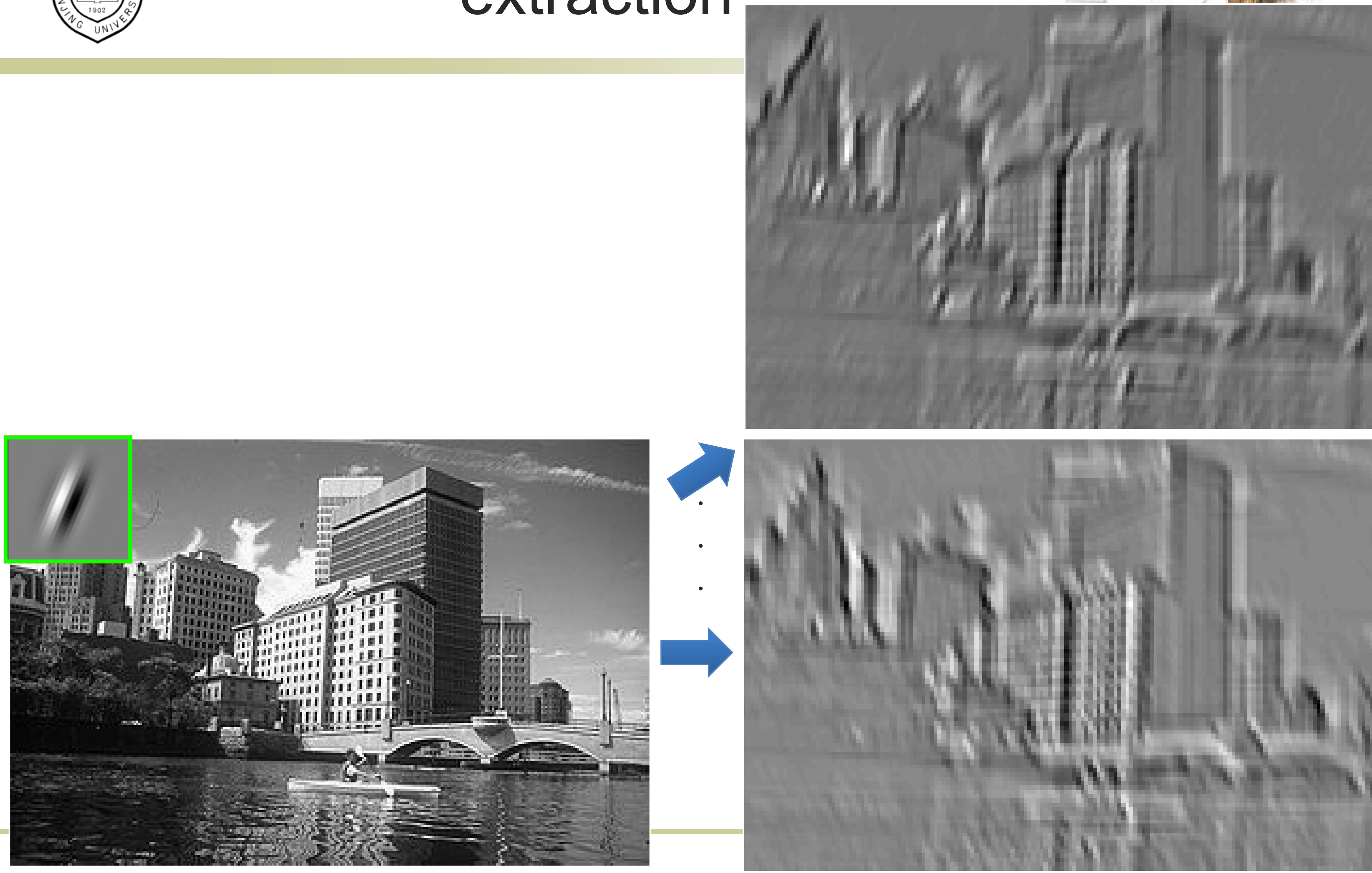
Slide: Lazebnik

From fully connected to convolutional networks



Slide: Lazebnik

Convolution as feature extraction

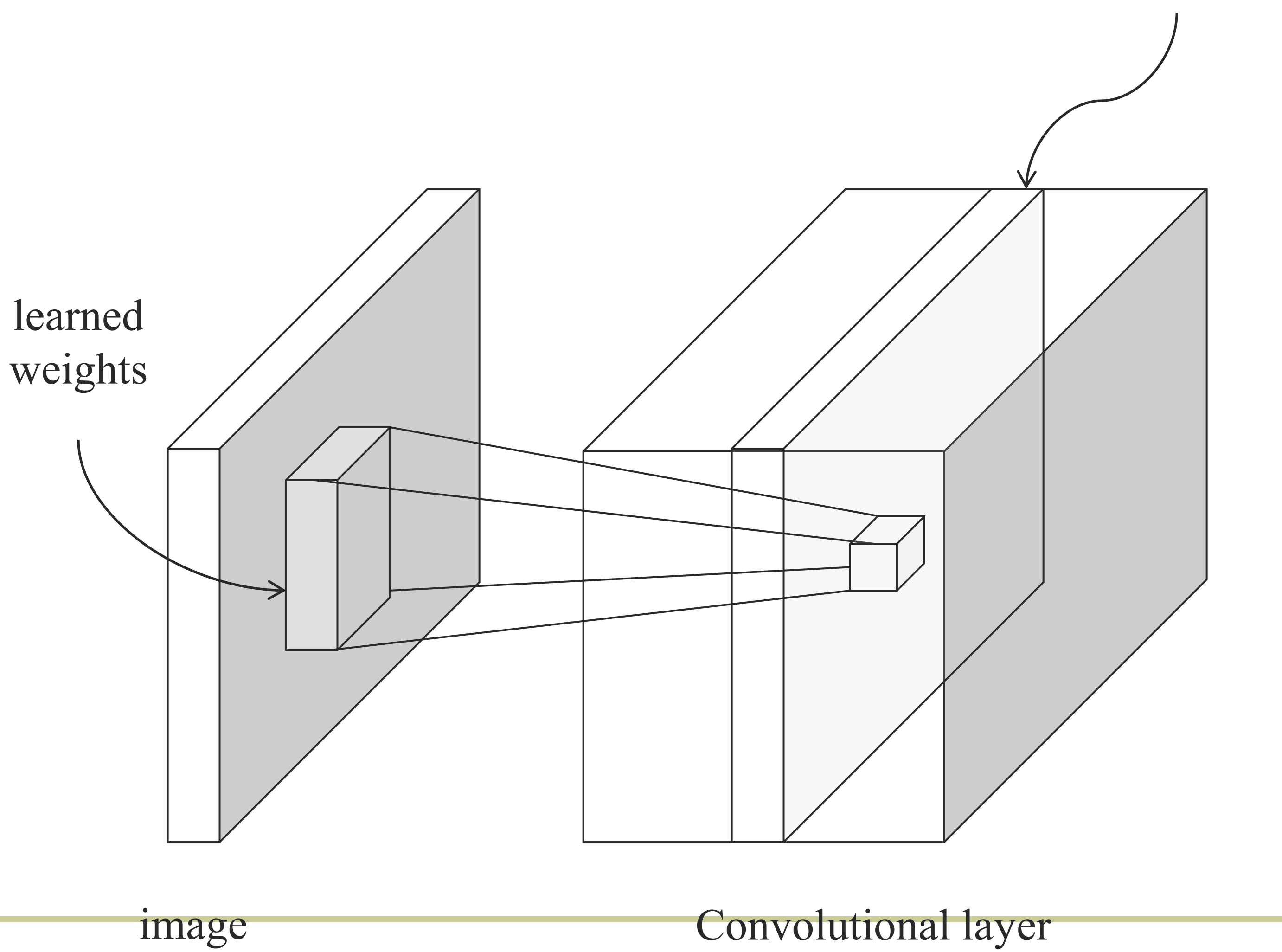


Input

Feature Map

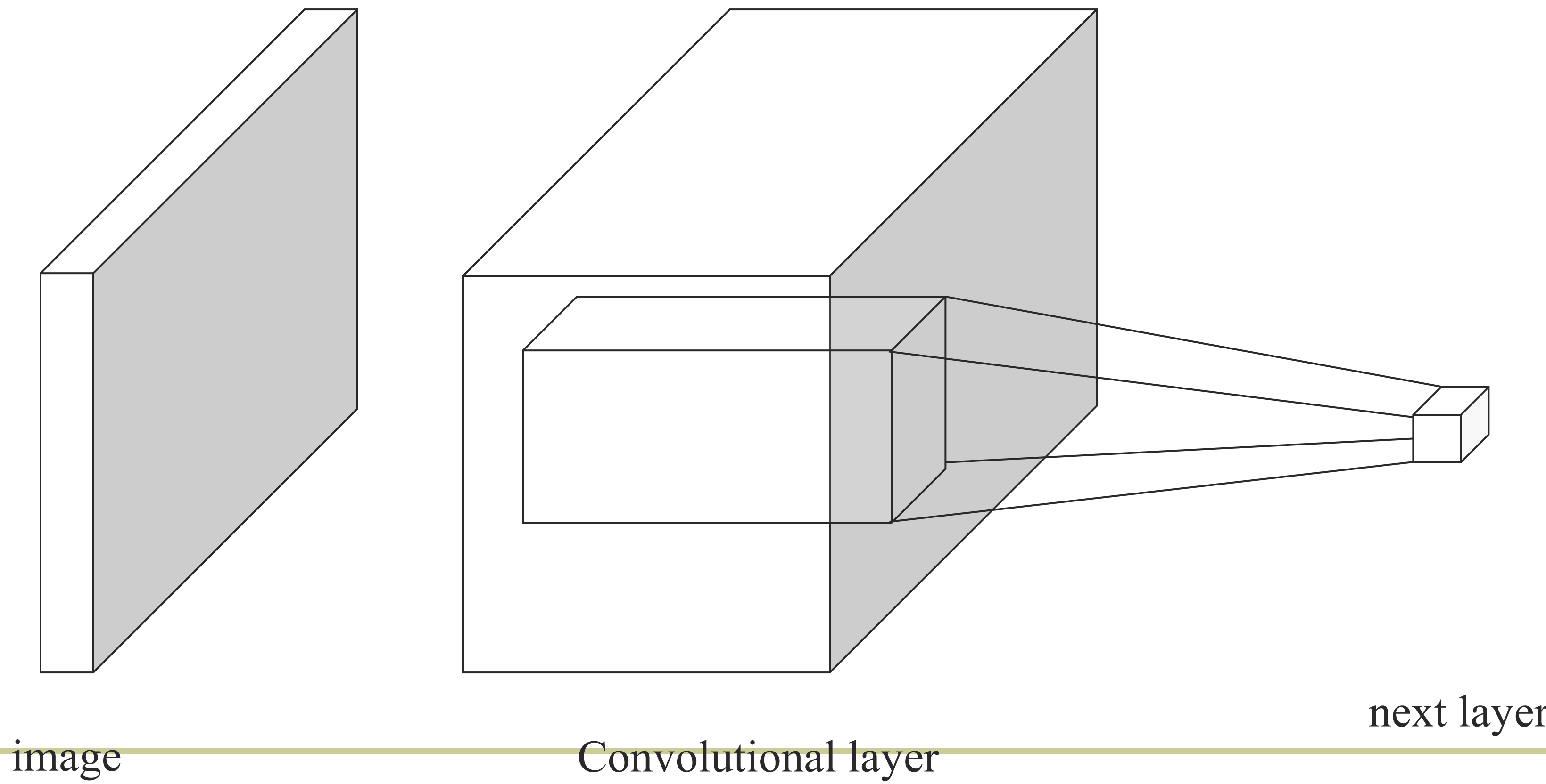
Slide: Lazebnik

From fully connected to convolutional networks



Slide: Lazebnik

From fully connected to convolutional networks



Slide: Lazebnik

Key operations in a CNN

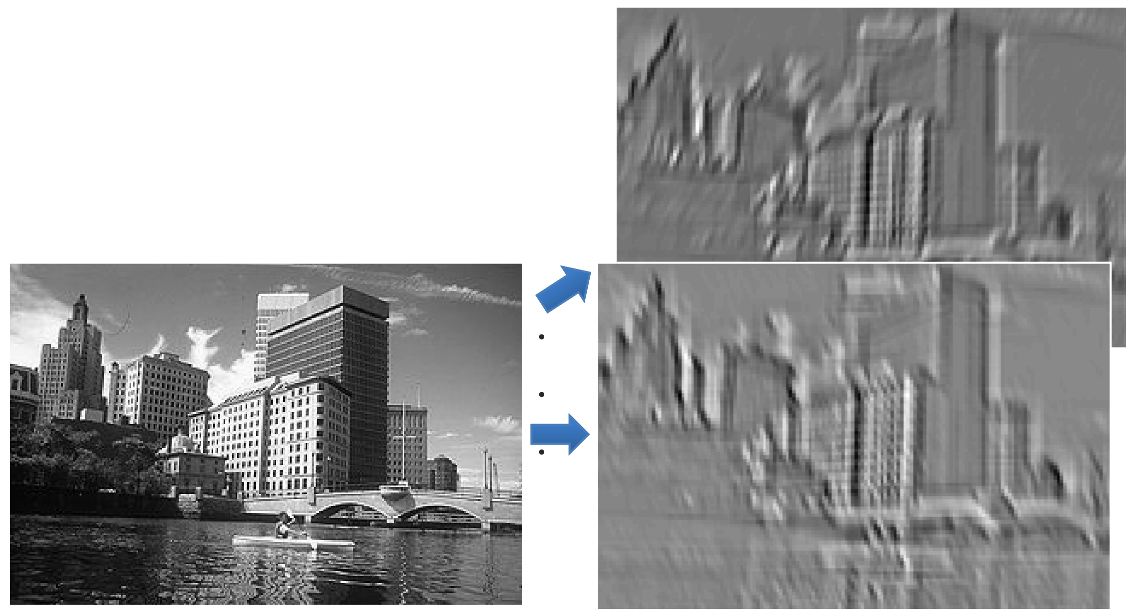
Feature maps

Spatial pooling

Non-linearity

Convolution (Learned)

Input Image

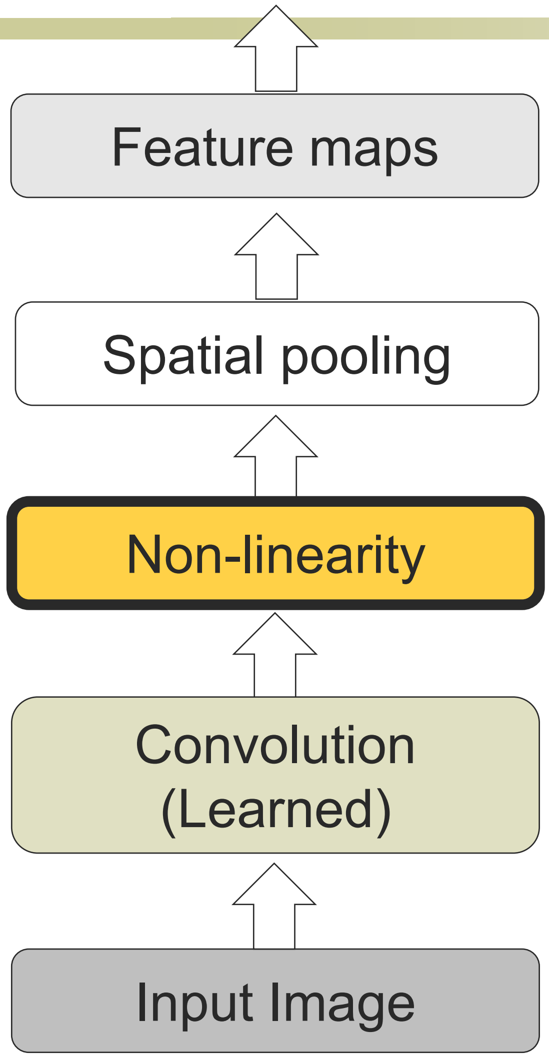


Input — Feature Map

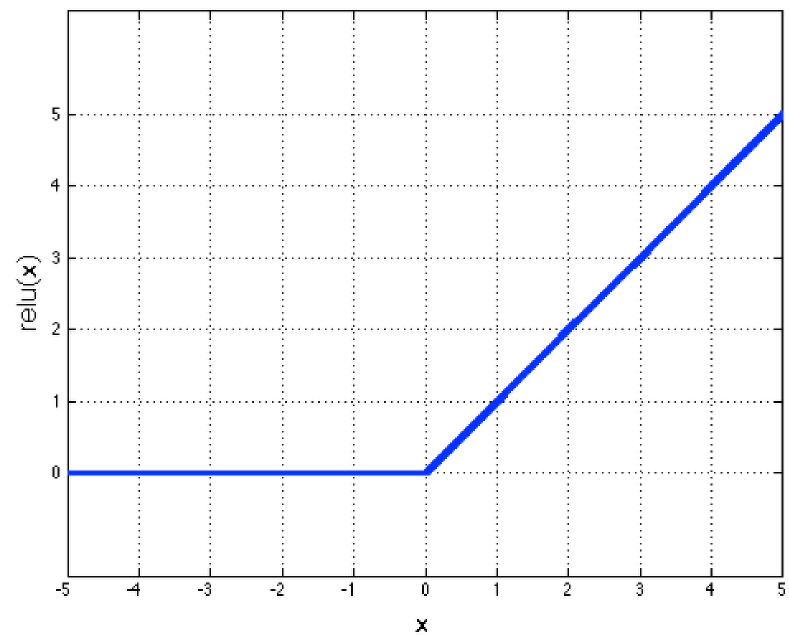
Source: R. Fergus, Y. LeCun

Slide: Lazebnik

Key operations



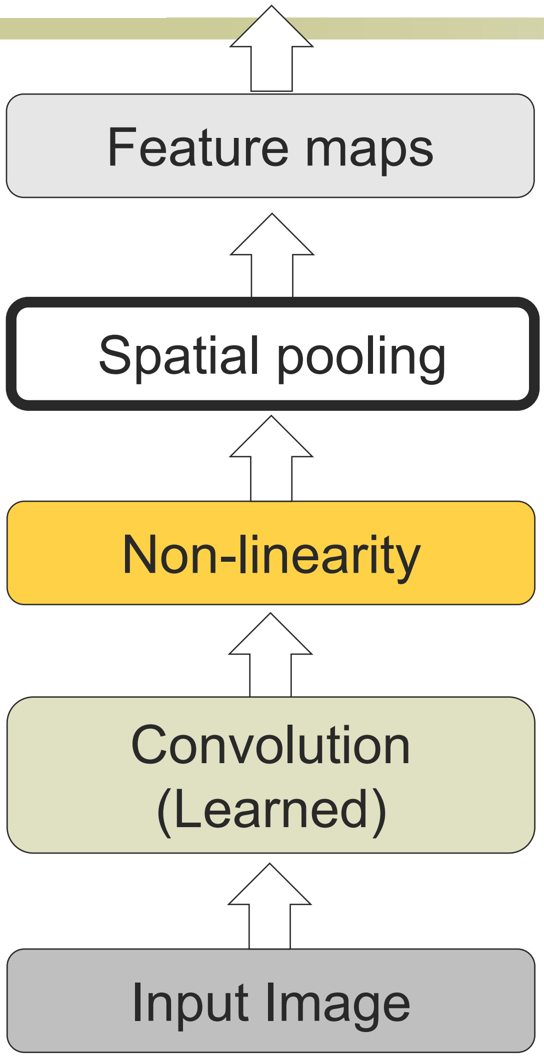
Rectified Linear Unit (ReLU)

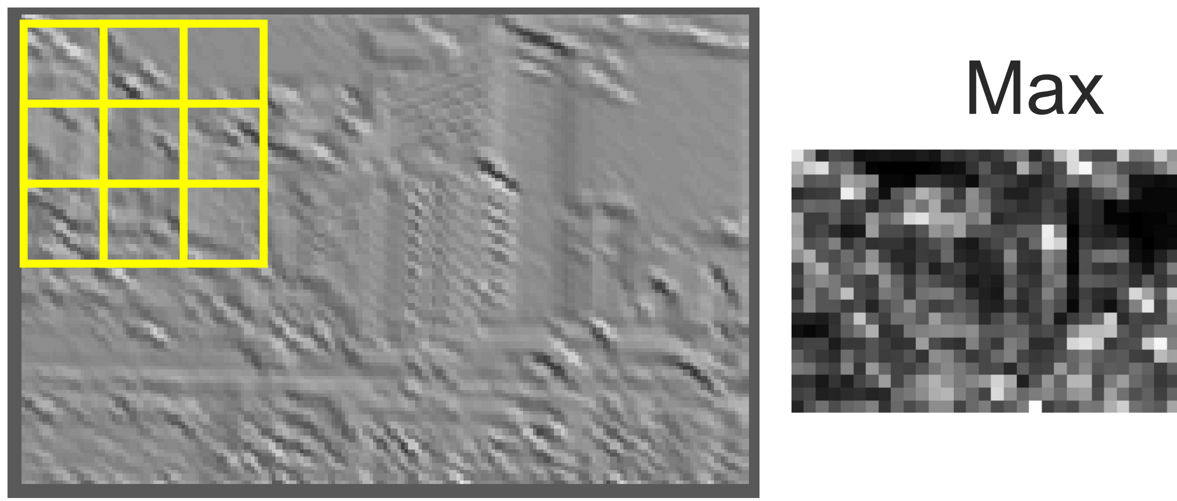


Source: R. Fergus, Y. LeCun

Slide: Lazebnik

Key operations





Source: R. Fergus, Y. LeCun

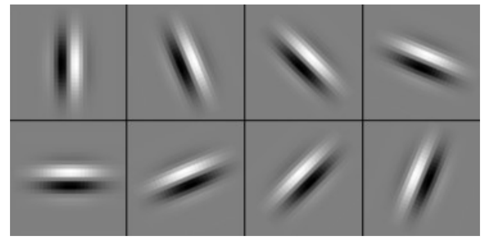
Slide: Lazebnik

Comparison to Pyramids with SIFT

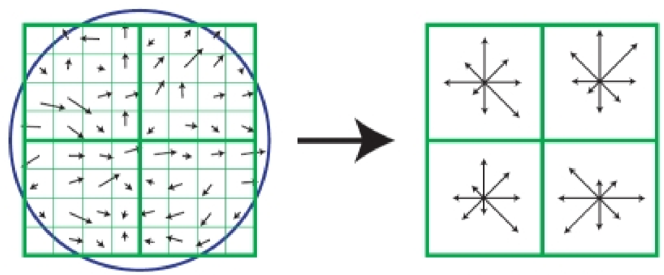
Lowe [IJCV 2004]

Image Pixels

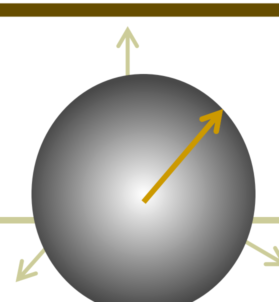
Apply oriented filters



Spatial pool (Sum)



Normalize to unit length



Feature Vector

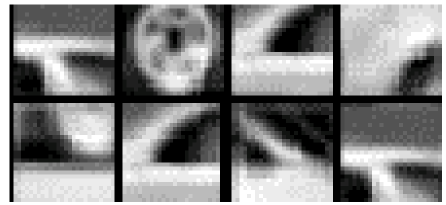
slide credit: R. Fergus

Comparison to Pyramids with SIFT

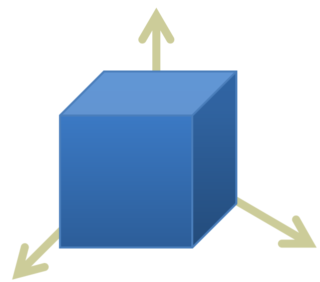
Lazebnik, Schmid, Ponce [CVPR 2006]

SIFT Features

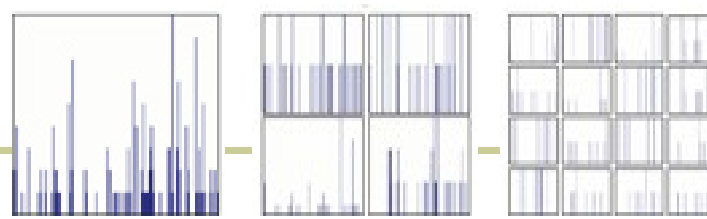
Filter with Visual Words



Max



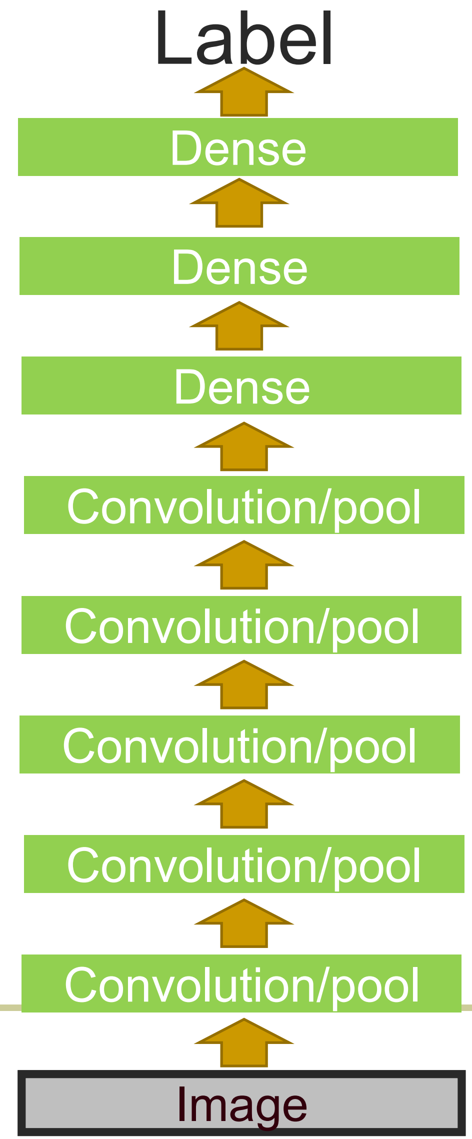
Multi-scale spatial pool (Sum)



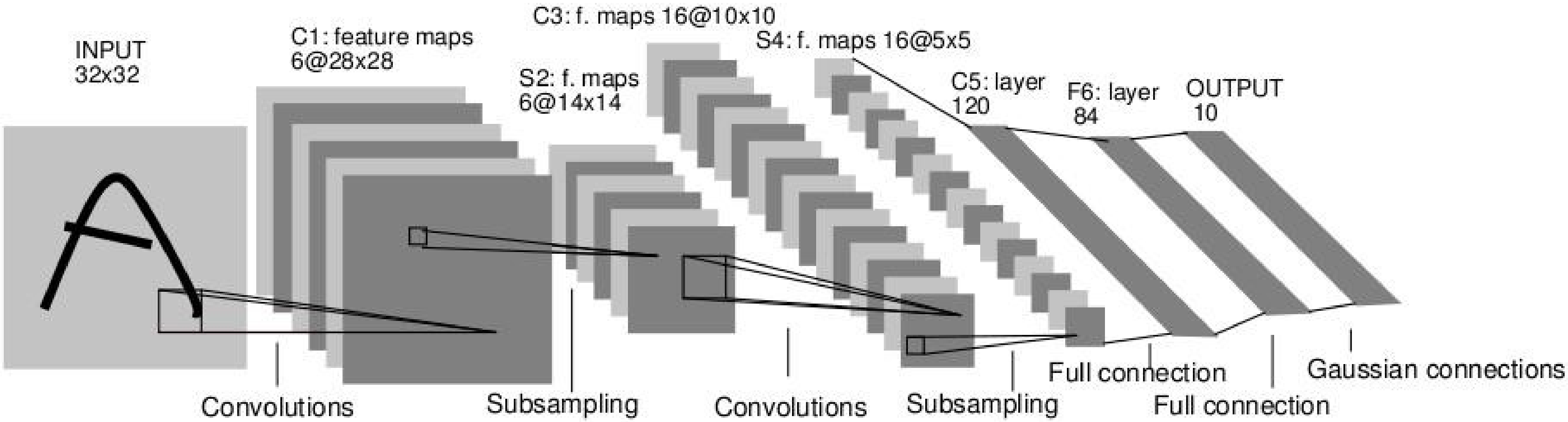
Classifier

slide credit: R. Fergus

Key idea: learn features and classifier that work well together ("end-to-end training")



LeNet-5

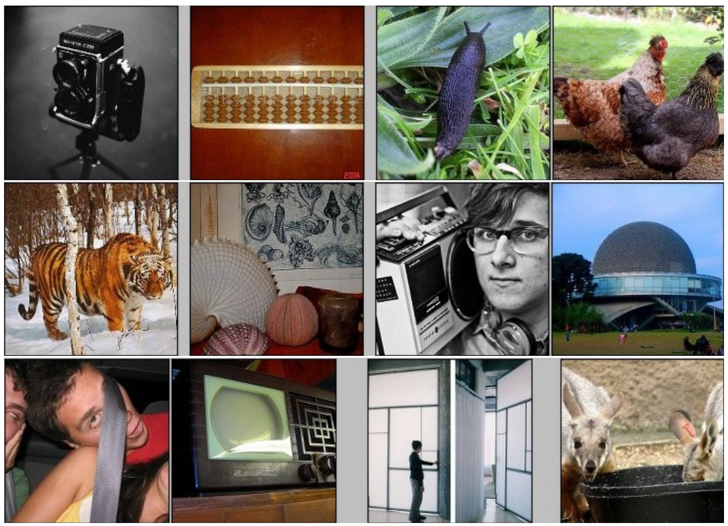


- Average pooling
- Sigmoid or tanh nonlinearity
- Fully connected layers at the end
- Trained on MNIST digit dataset with 60K training examples

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86(11): 2278–2324, 1998.

Fast forward to the arrival of big visual data…

IMAGENET

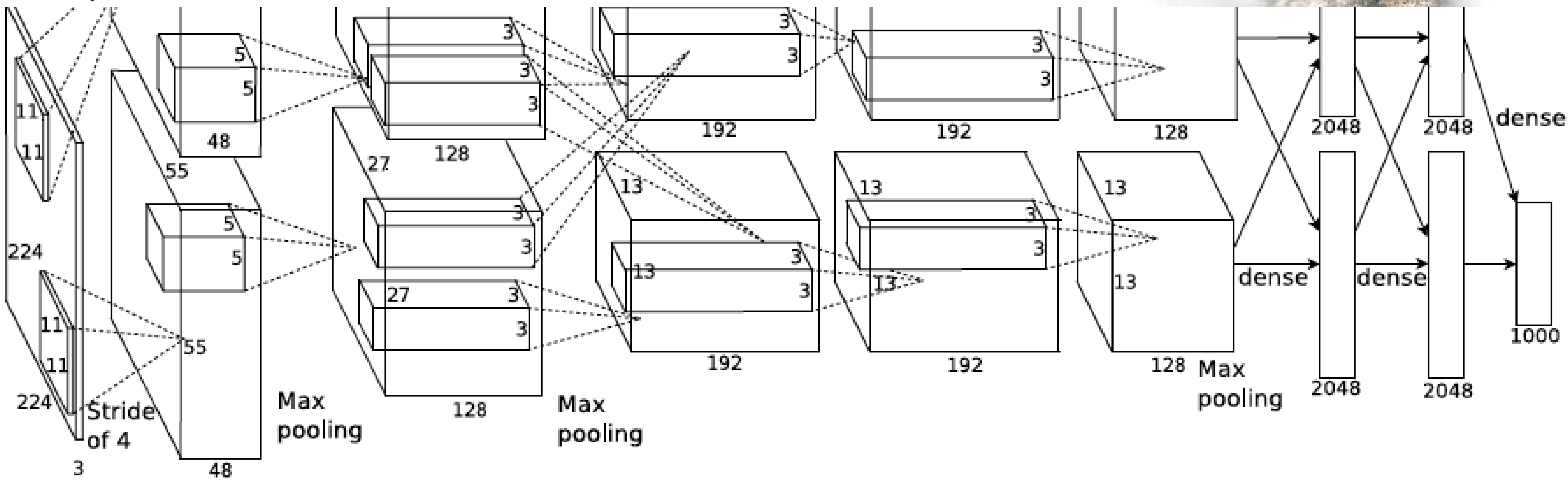


- ~14 million labeled images, 20k classes
- Images gathered from Internet
- Human labels via Amazon MTurk
- ImageNet Large-Scale Visual Recognition Challenge (ILSVRC): 1.2 million training images, 1000 classes

www.image-net.org/challenges/LSVRC/

Slide: Lazebnik

AlexNet: ILSVRC 2012 winner

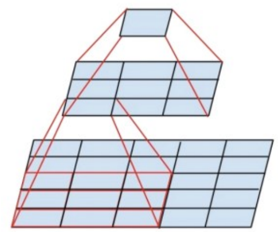


- Similar framework to LeNet but:
  - Max pooling, **ReLU nonlinearity**
  - **More data** and **bigger model** (7 hidden layers, 650K units, 60M params)
  - GPU implementation (**50x speedup** over CPU)
    - Trained on two GPUs for a week
  - Dropout regularization

A. Krizhevsky, I. Sutskever, and G. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012

VGGNet

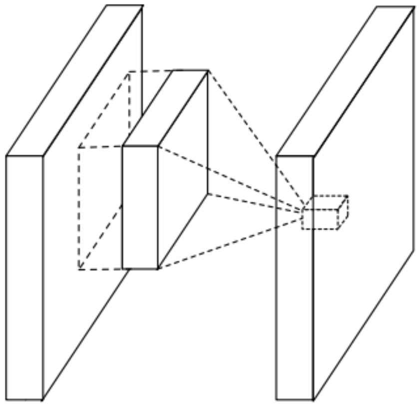
- Sequence of deeper networks trained progressively
- Large receptive fields replaced by successive layers of 3x3 convolutions (with ReLU in between)



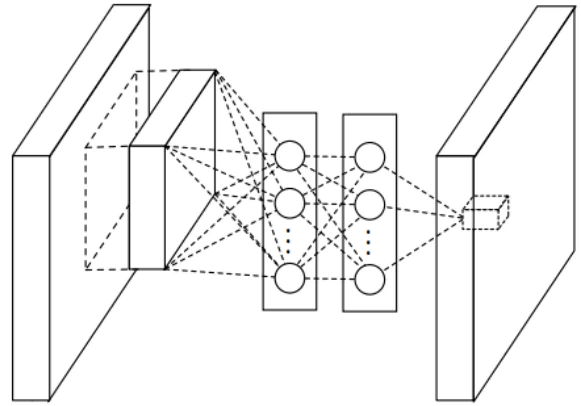
  - One 7x7 conv layer with C feature maps needs $49C^2$ weights, three 3x3 conv layers need only $27C^2$ weights

K. Simonyan and A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015

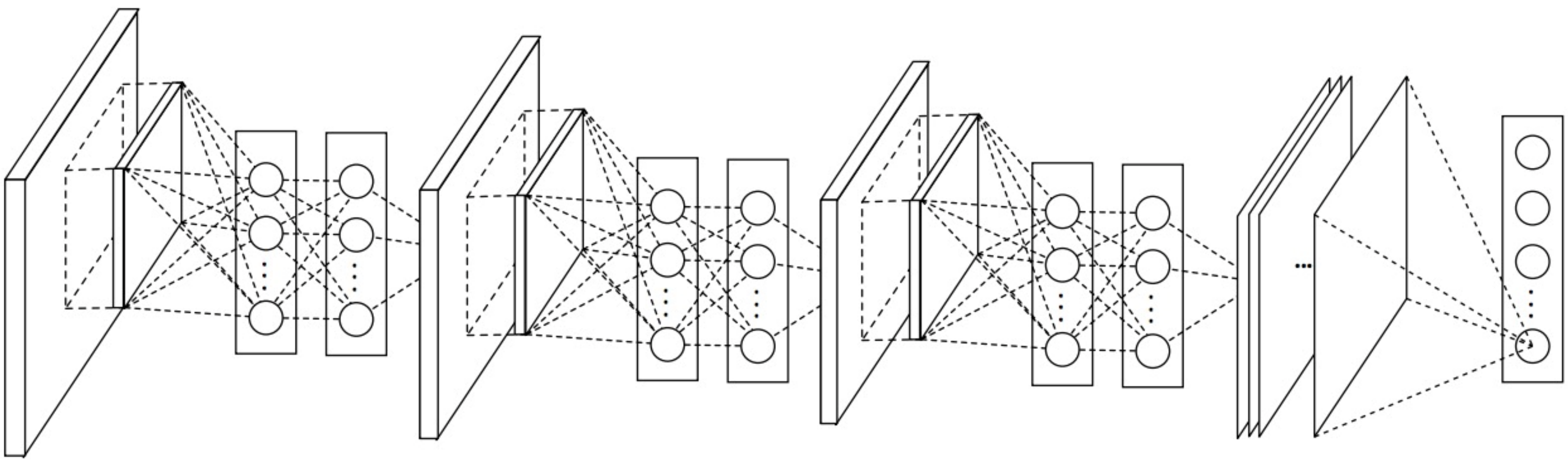
Network in network



(a) Linear convolution layer



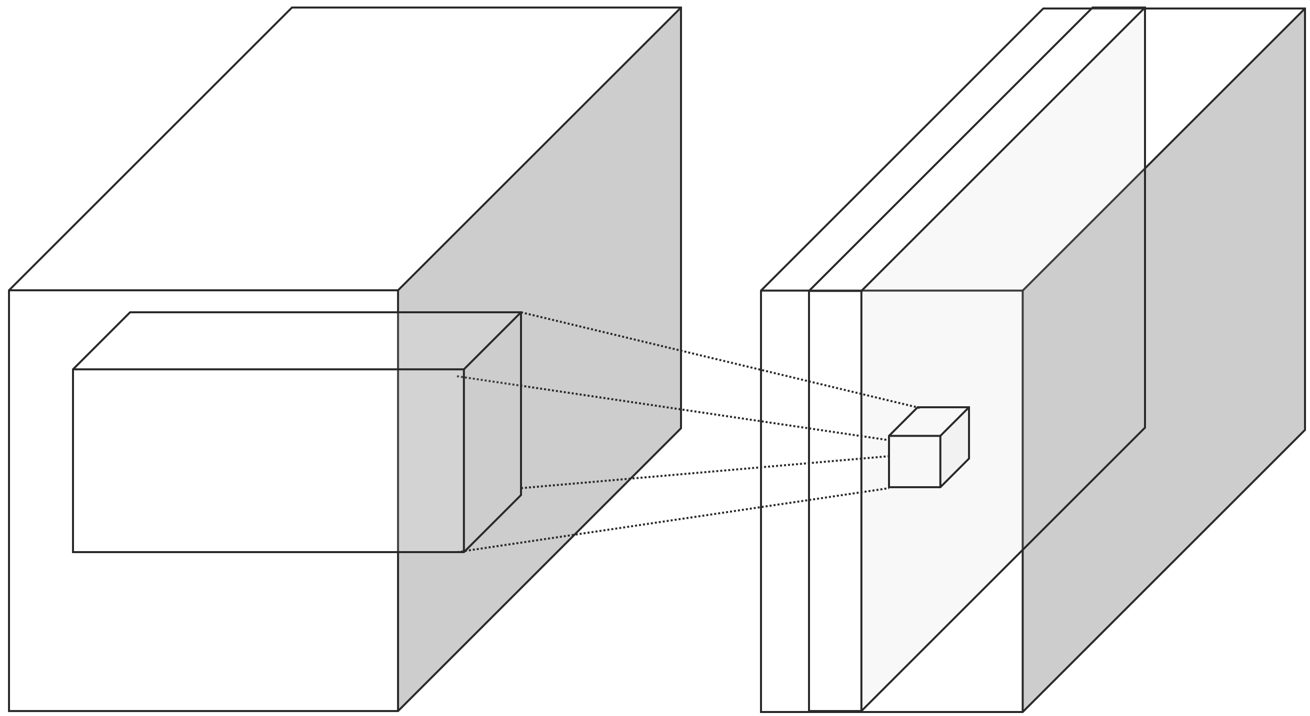
(b) Mlpconv layer



M. Lin, Q. Chen, and S. Yan, Network in network, ICLR 2014

Slide: Lazebnik

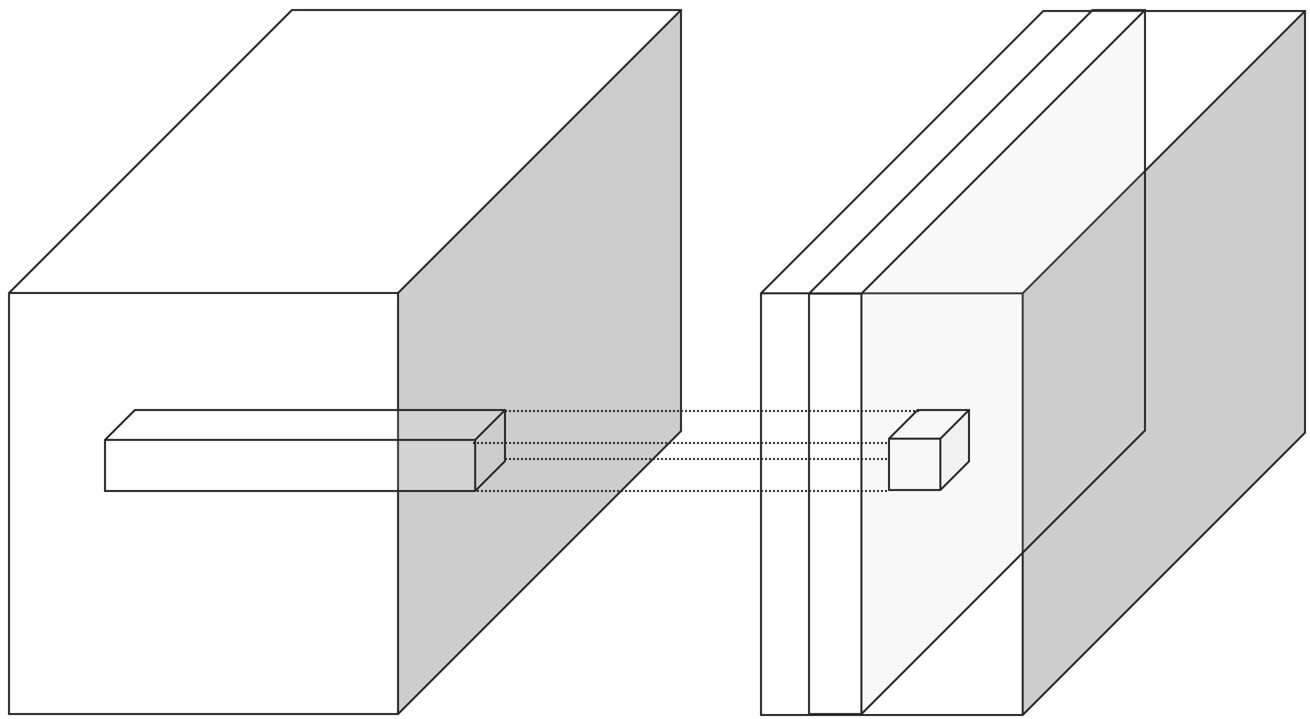
1x1 convolutions



conv layer

Slide: Lazebnik

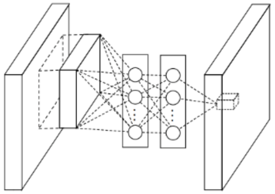
1x1 convolutions

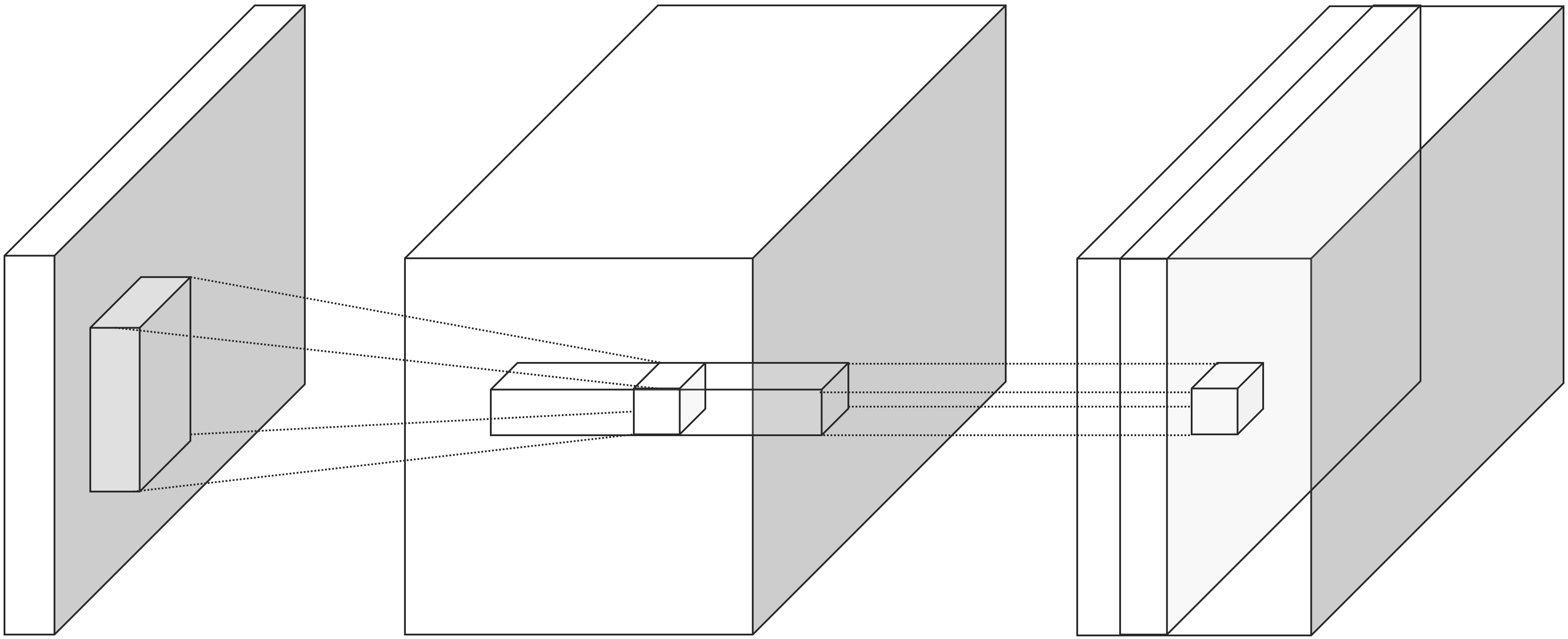


1x1 conv layer

Slide: Lazebnik

1x1 convolutions



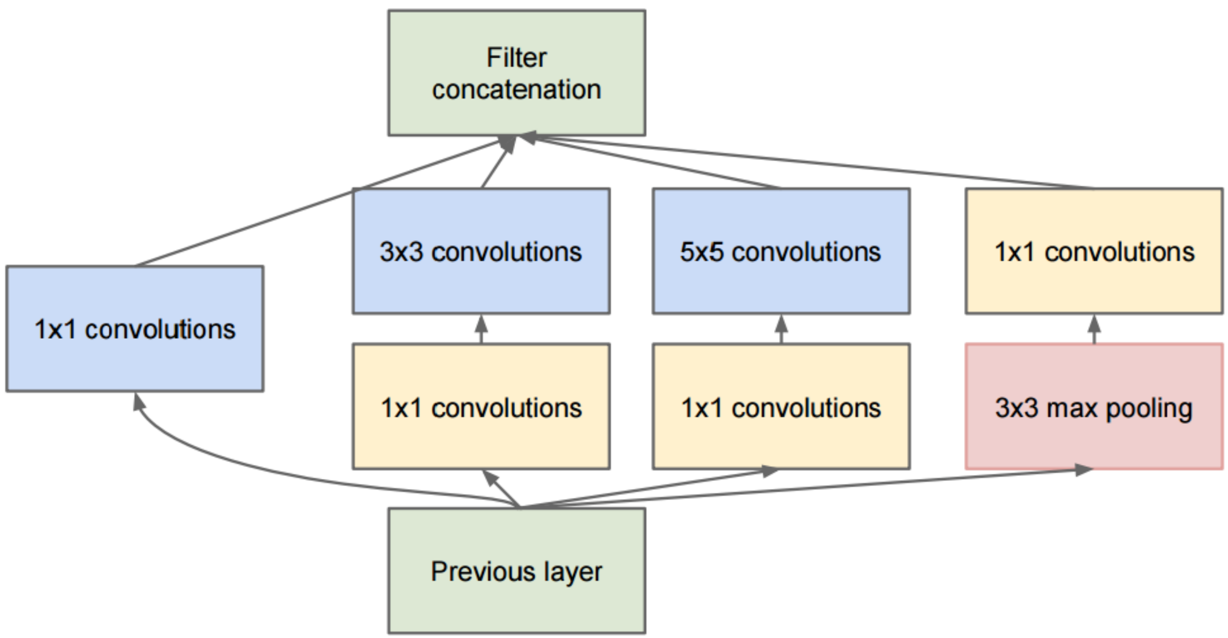


1x1 conv layer

Slide: Lazebnik

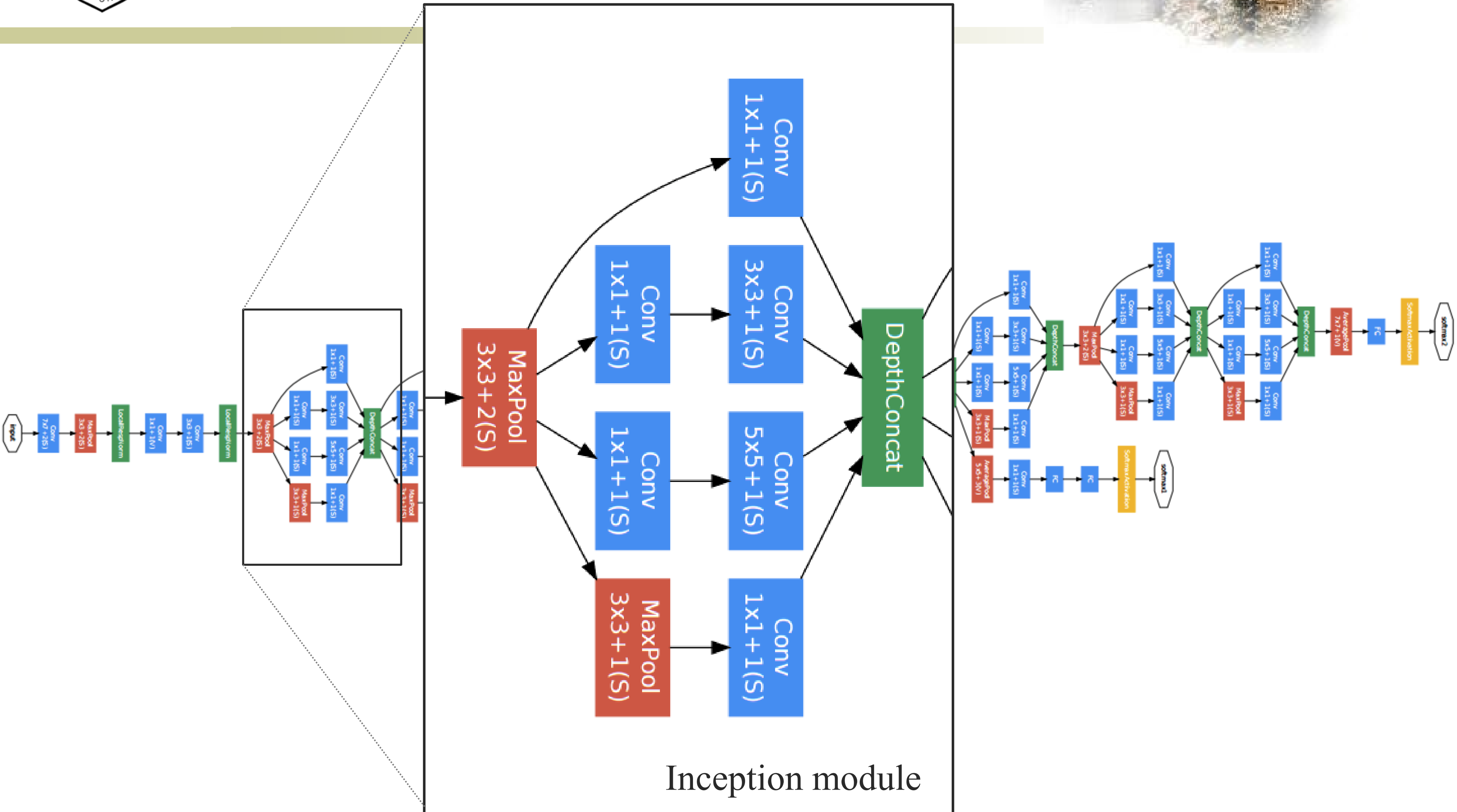
GoogLeNet: Inception module

- Parallel paths with different receptive field sizes and operations to capture sparse patterns of correlations
- 1x1 convolutions for dimensionality reduction before expensive convolutions



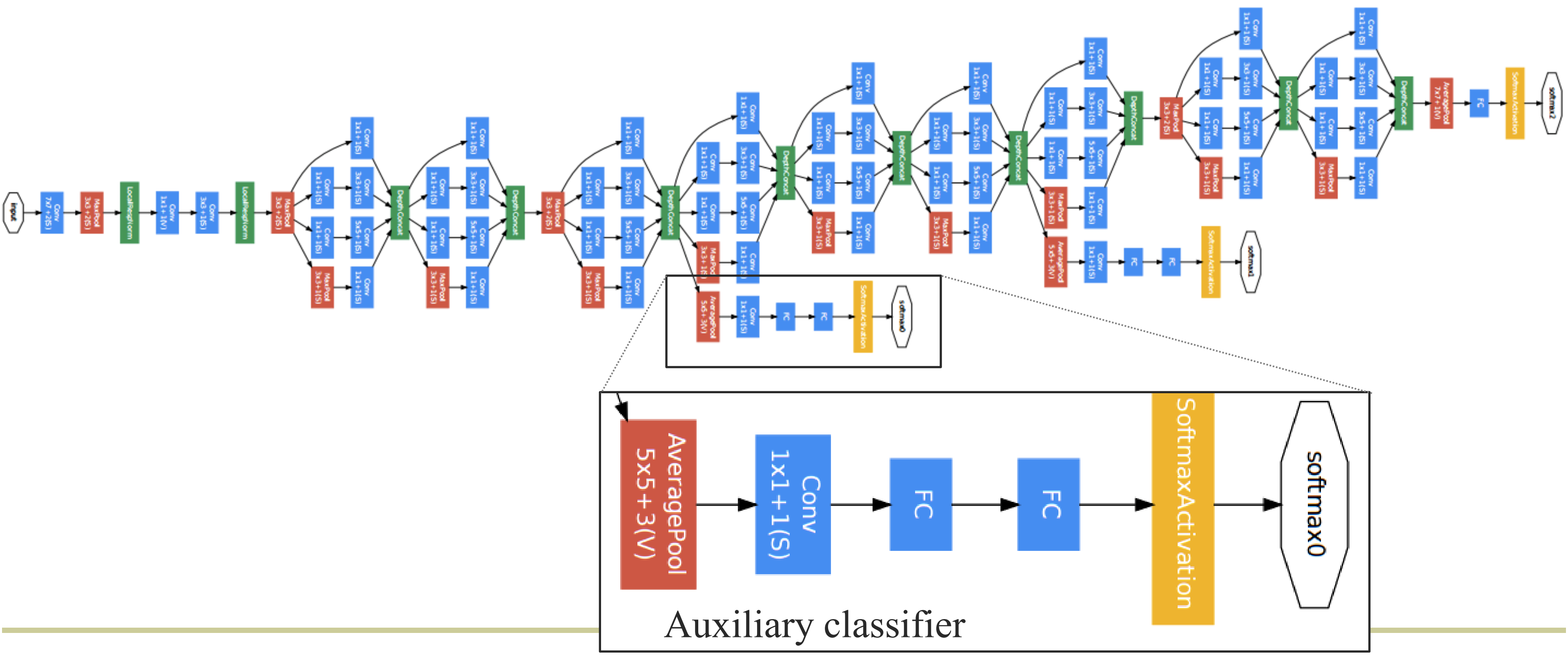
C. Szegedy et al., Going deeper with convolutions, CVPR 2015

GoogLeNet



C. Szegedy et al., Going deeper with convolutions, CVPR 2015

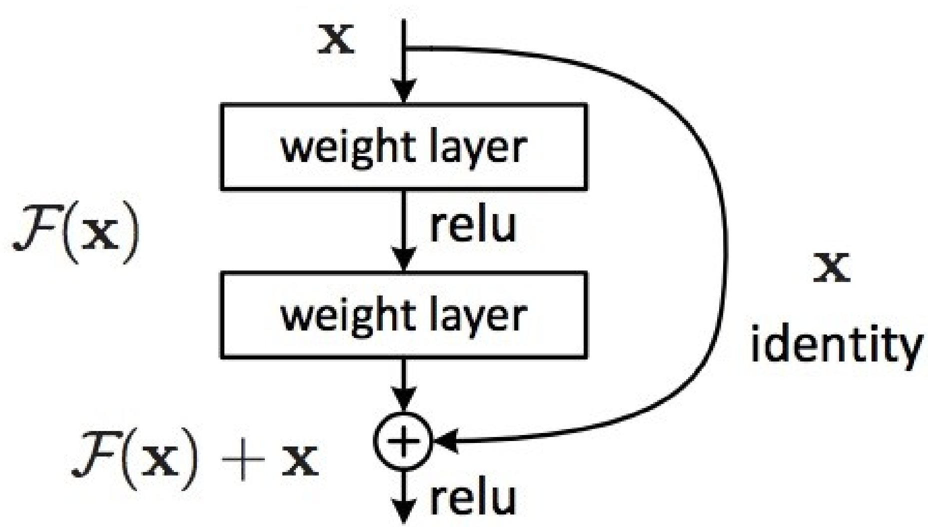
GoogLeNet



C. Szegedy et al., Going deeper with convolutions, CVPR 2015

ResNet: the residual module

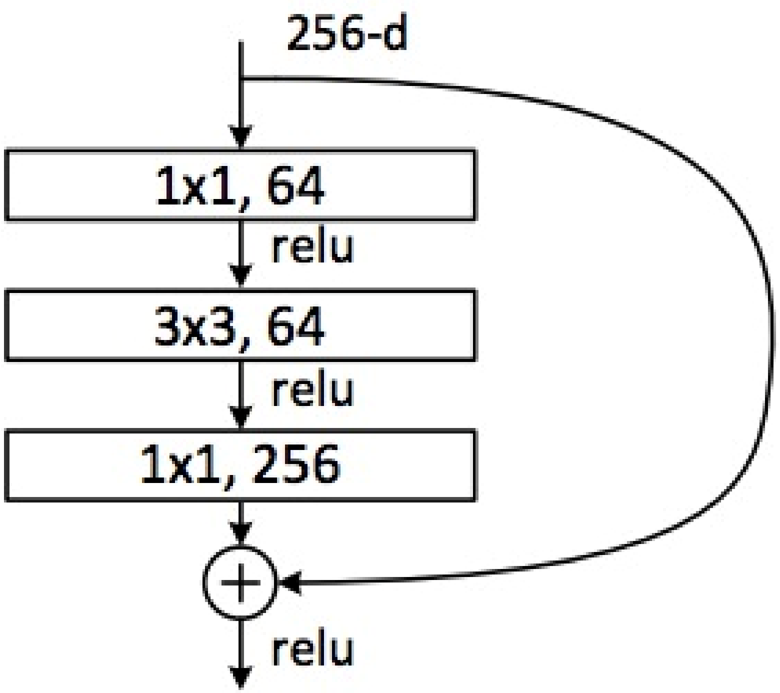
- Introduce *skip* or *shortcut* connections (existing before in various forms in literature)
- Make it easy for network layers to represent the identity mapping
- For some reason, need to skip at least two layers



Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, Deep Residual Learning for Image Recognition, CVPR 2016 (Best Paper)

ResNet

Deeper residual module (bottleneck)



- Directly performing 3x3 convolutions with 256 feature maps at input and output:
  256 x 256 x 3 x 3 ~ 600K operations
- Using 1x1 convolutions to reduce 256 to 64 feature maps, followed by 3x3 convolutions, followed by 1x1 convolutions to expand back to 256 maps:
  256 x 64 x 1 x 1 ~ 16K
  64 x 64 x 3 x 3 ~ 36K
  64 x 256 x 1 x 1 ~ 16K
  Total: ~70K

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, Deep Residual Learning for Image Recognition, CVPR 2016 (Best Paper)

Slide: Lazebnik

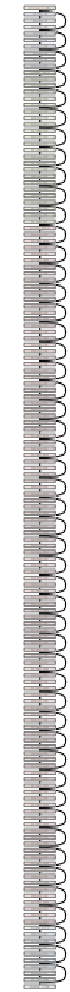
ResNet: going real deep

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)

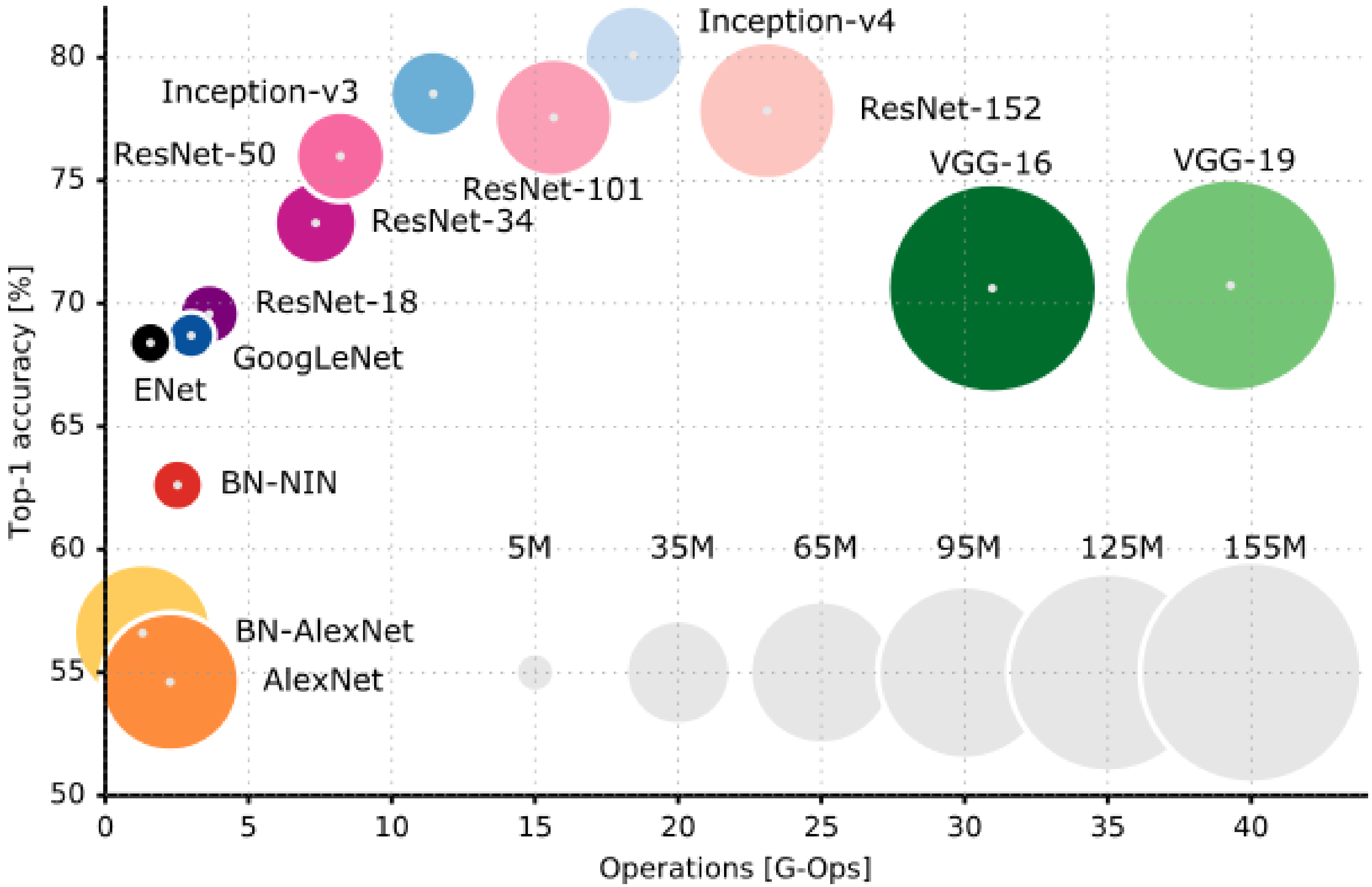
VGG, 19 layers
(ILSVRC 2014)

ResNet, 152 layers
(ILSVRC 2015)



Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, Deep Residual Learning for Image Recognition, CVPR 2016

Bigger not better: innovations typically reduce parameters, despite deeper nets



Key ideas of CNN Architectures

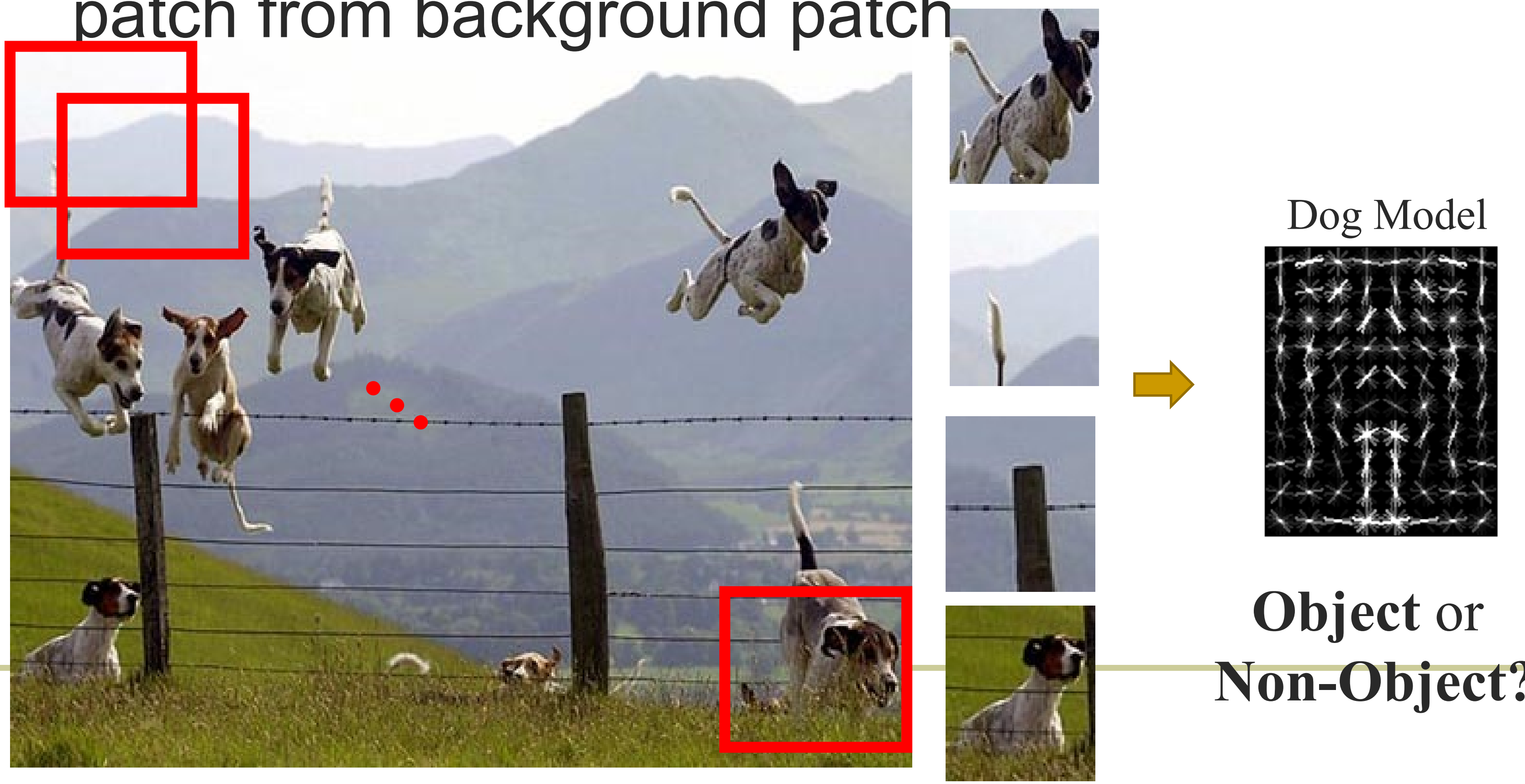
- Convolutional layers
  - Same local functions evaluated everywhere → much fewer parameters
- Pooling
  - Larger receptive field and translational invariance
- ReLU: maintain a gradient signal over large portion of domain
- Limit parameters
  - Sequence of 3x3 filters instead of large filters (also encodes that local pixels are more relevant)
  - 1x1 convs to reduce feature dimensions
- Skip network
  - Prevents having to maintain early layers (just add residual)
  - Acts as ensemble

Today’ class

- Overview of image categorization
- Spatial pyramids bag-of-words categorizer
- Deep convolutional neural networks (CNNs)
- **Object detection**

Object Category Detection

- Focus on object search: “Where is it?”
- Build templates that quickly differentiate object patch from background patch



General Process of Object Recognition

Specify Object Model

↓

Generate Hypotheses

↓

Score Hypotheses

↓

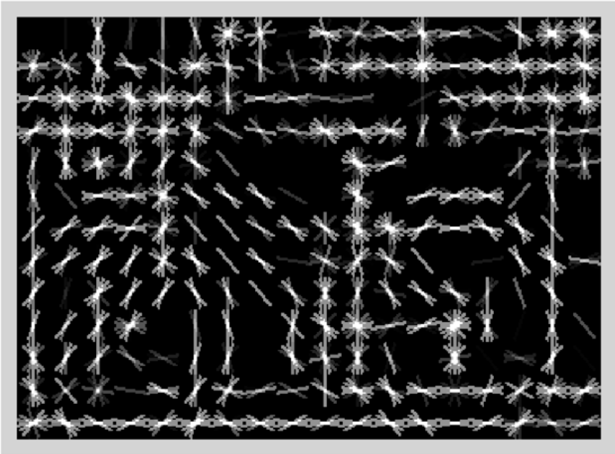
Resolve Detections

What are the object parameters?

Specifying an object model

1. Statistical Template in Bounding Box
   - Object is some (x,y,w,h) in image
   - Features defined wrt bounding box coordinates

Image

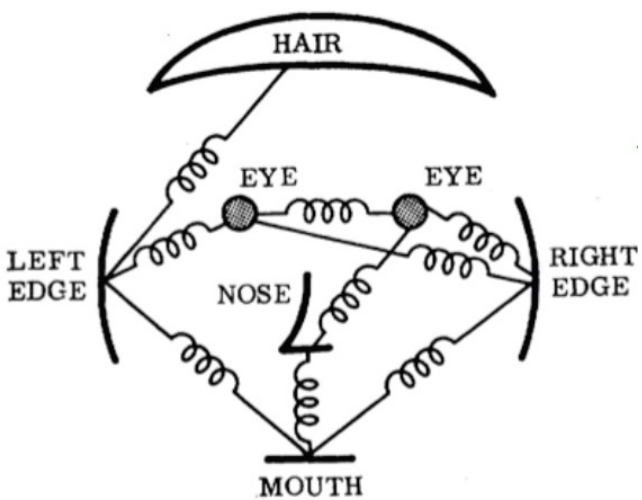


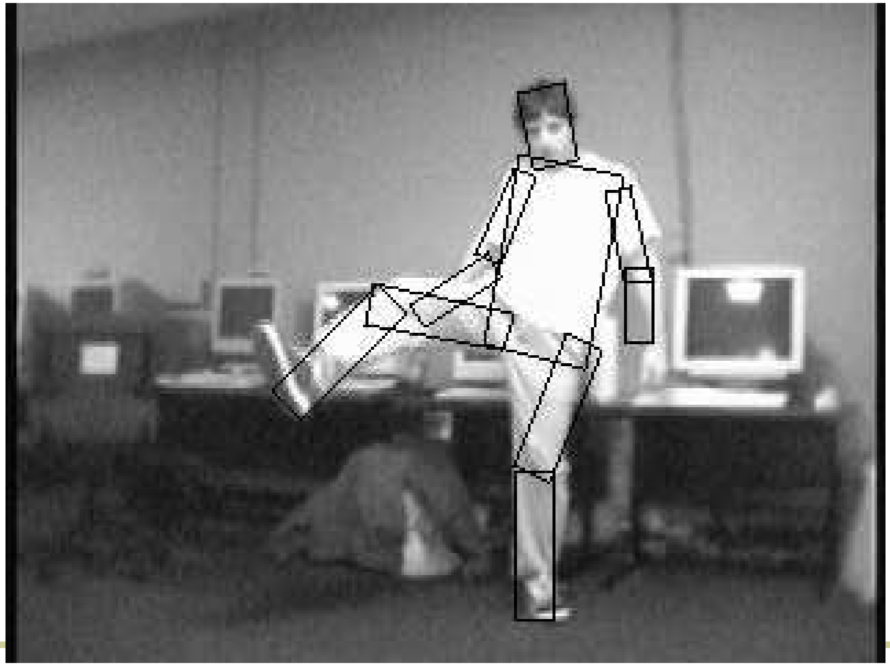
Template Visualization

Images from Felzenszwalb

Specifying an object model

2. Articulated parts model
   - Object is configuration of parts
   - Each part is detectable



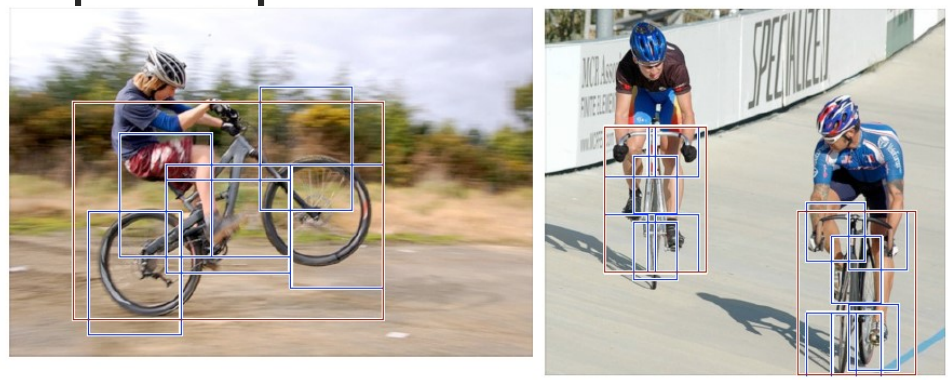


Images from Felzenszwalb

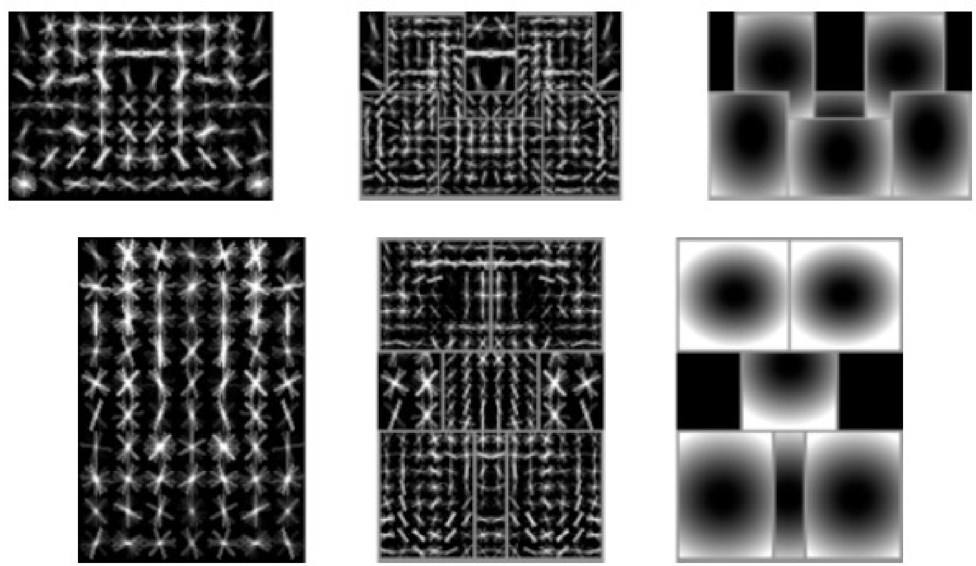
Specifying an object model

3. Hybrid template/parts model

Detections



Template Visualization



root filters
coarse resolution

part filters
finer resolution

deformation
models

Felzenszwalb et al. 2008

General Process of Object Recognition

Specify Object Model

↓

Generate Hypotheses

↓

Score Hypotheses

↓

Resolve Detections

Propose an alignment of the model to the image

Generating hypotheses

1. Sliding window
   - Test patch at each location and scale

Generating hypotheses

1. Sliding window
   - Test patch at each location and scale

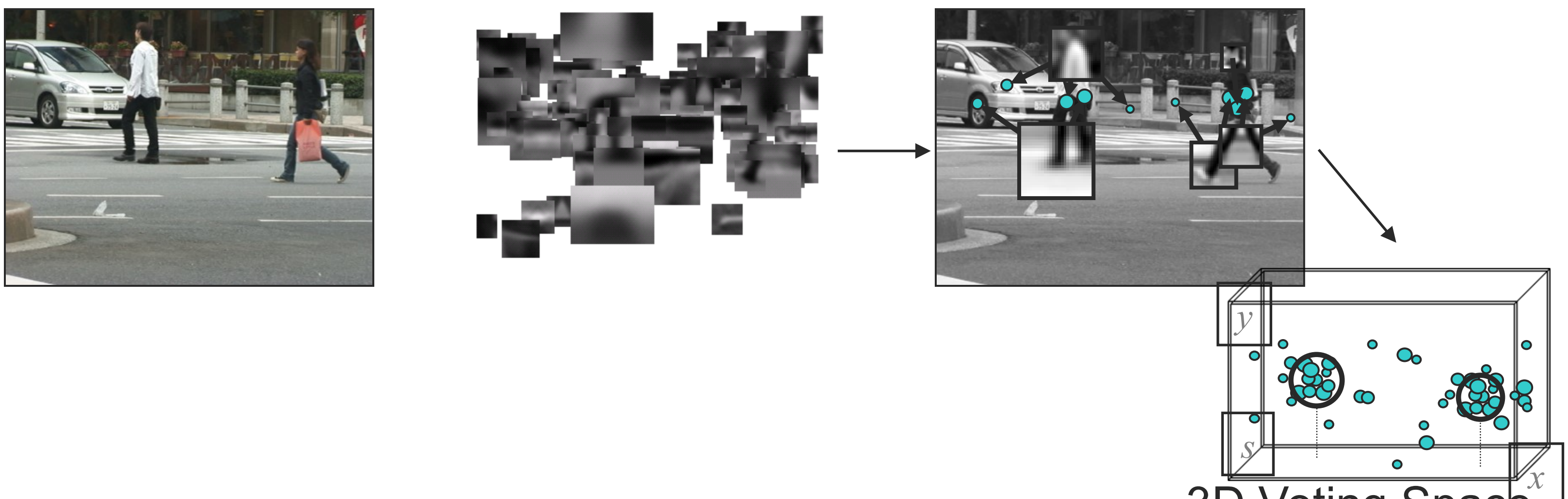
Generating hypotheses

2. Voting from patches/keypoints

Interest Points

Matched Codebook Entries

Probabilistic Voting

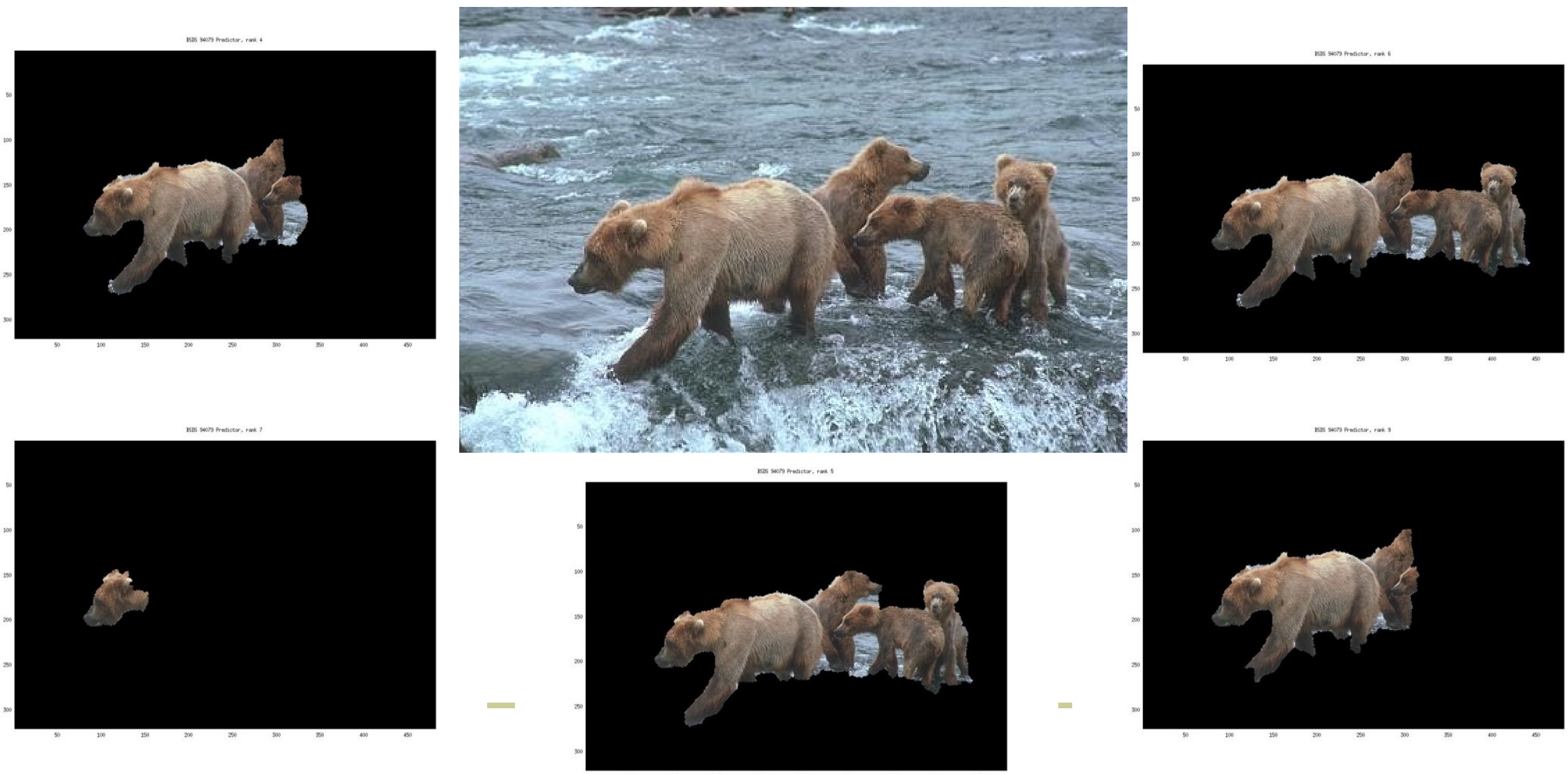


3D Voting Space (continuous)

ISM model by Leibe et al.

Generating hypotheses

3. Region-based proposal



Endres Hoiem 2010

Sliding window vs. region proposals

Sliding window

- Comprehensive search over position, scale (sometimes aspect, though expensive)
- Typically 100K candidates
- Simple
- Speed boost through convolution often possible
- Repeatable
- Even with many candidates, may not be a good fit to object

Region proposals

- Search over regions guided by image contours/patterns with varying aspect/size
- Typically 2-10K candidates
- Random (not repeatable)
- Requires a preprocess (currently 1-5s)
- Often requires resizing patch to fit fixed size
- More likely to provide candidates with very good object fit

General Process of Object Recognition

Specify Object Model

↓

Generate Hypotheses

↓

Score Hypotheses

↓

Resolve Detections

Currently CNN features and classifiers

General Process of Object Recognition

Specify Object Model

↓

Generate Hypotheses

↓

Score Hypotheses

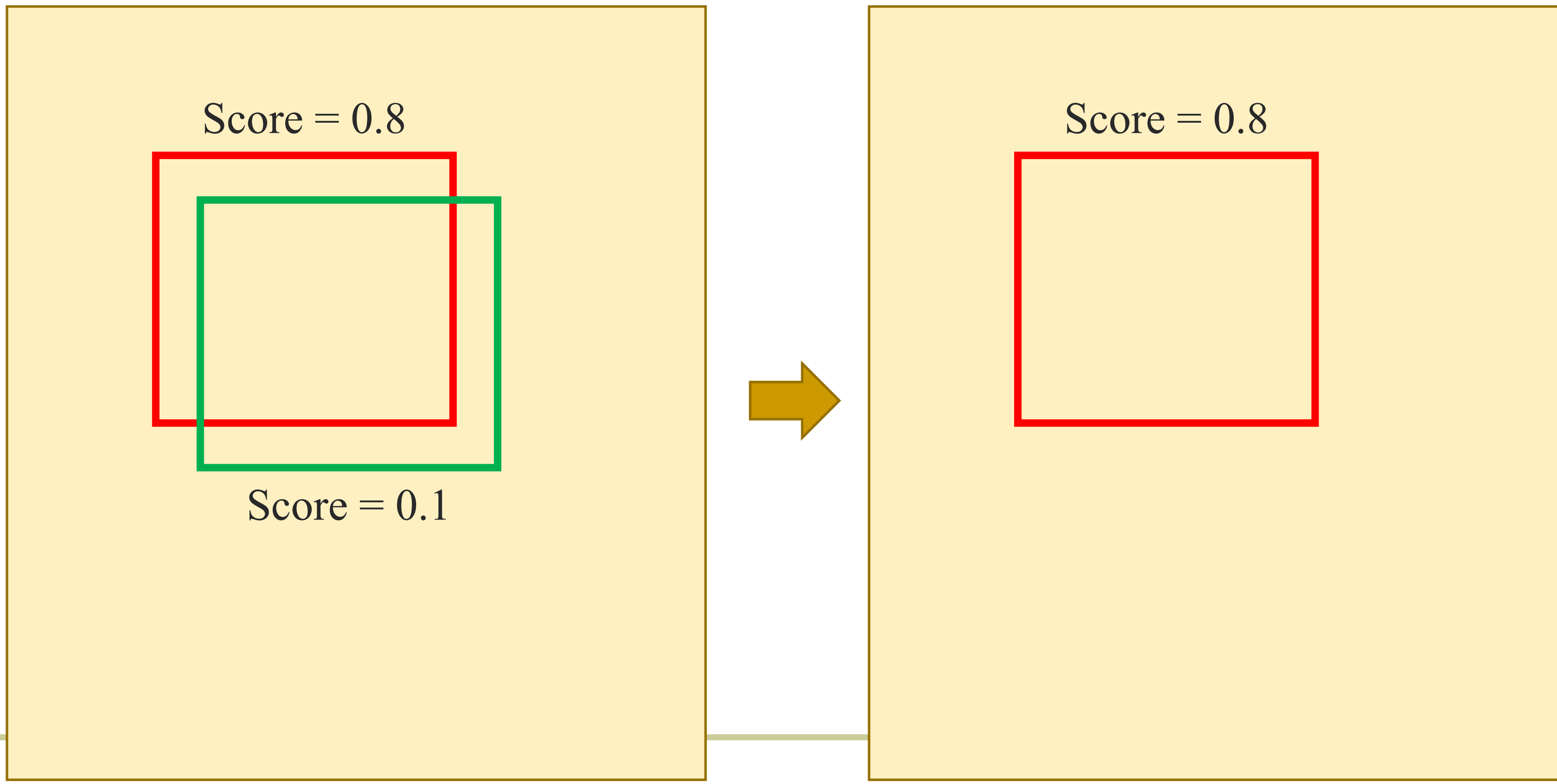
↓

Resolve Detections

Optionally, rescore each proposed object based on whole set

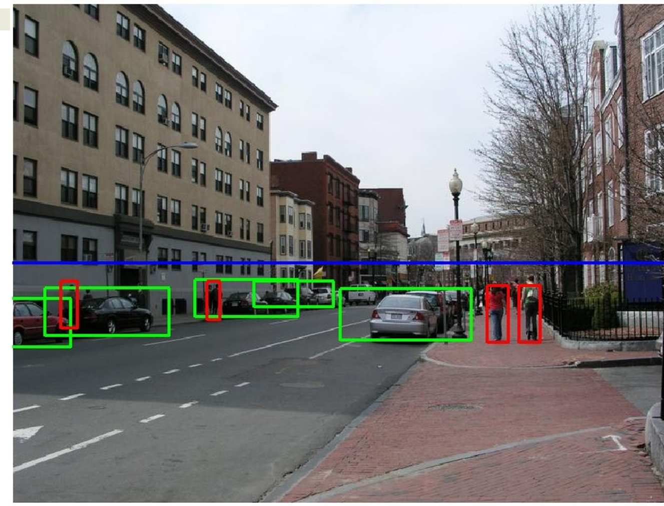
Resolving detection scores

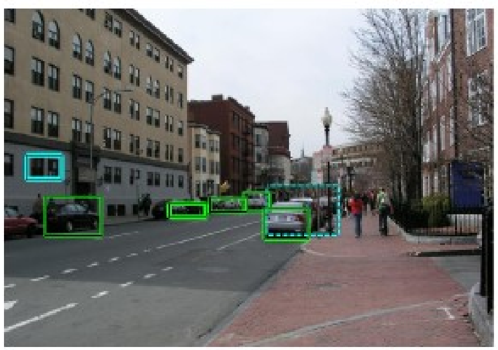
1. Non-max suppression



Resolving detection scores

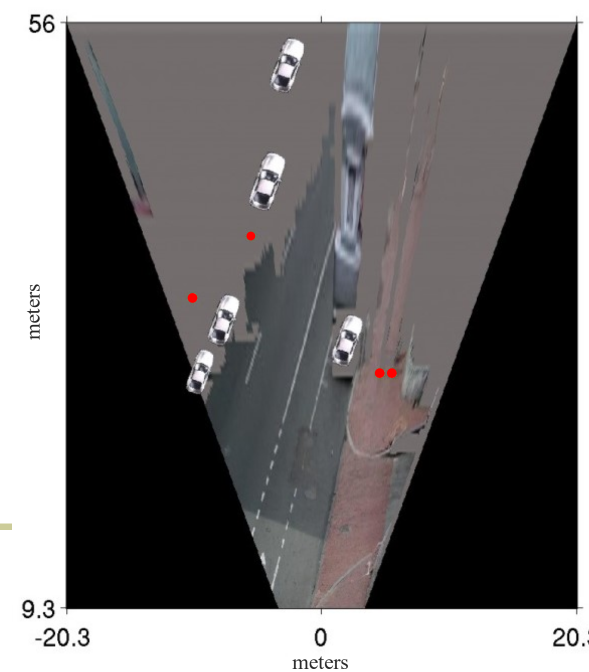
2. Context/reasoning





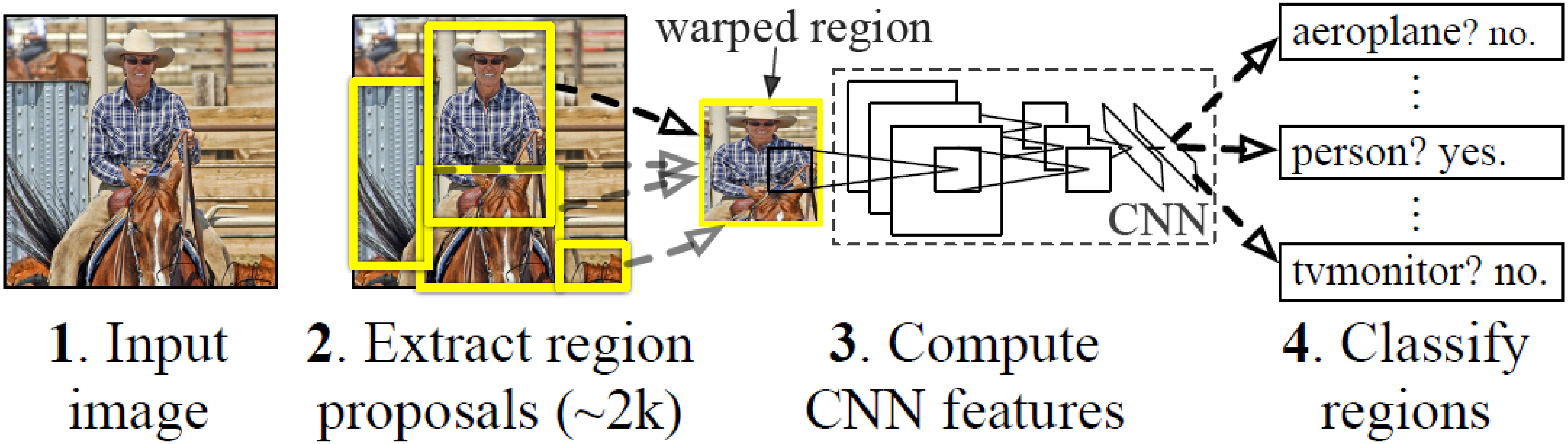
(g) Car Detections: Local

(h) Ped Detections: Local



Hoiem et al. 2006

R-CNN (Girshick et al. CVPR 2014)



- Replace sliding windows with "selective search" region proposals (Uijilings et al. IJCV 2013)
- Extract rectangles around regions and resize to 227x227
- Extract features with fine-tuned CNN (that was initialized with network trained on ImageNet before training)
- Classify last layer of network features with SVM

http://arxiv.org/pdf/1311.2524.pdf

Fine-tuning example: ImageNet->VOC

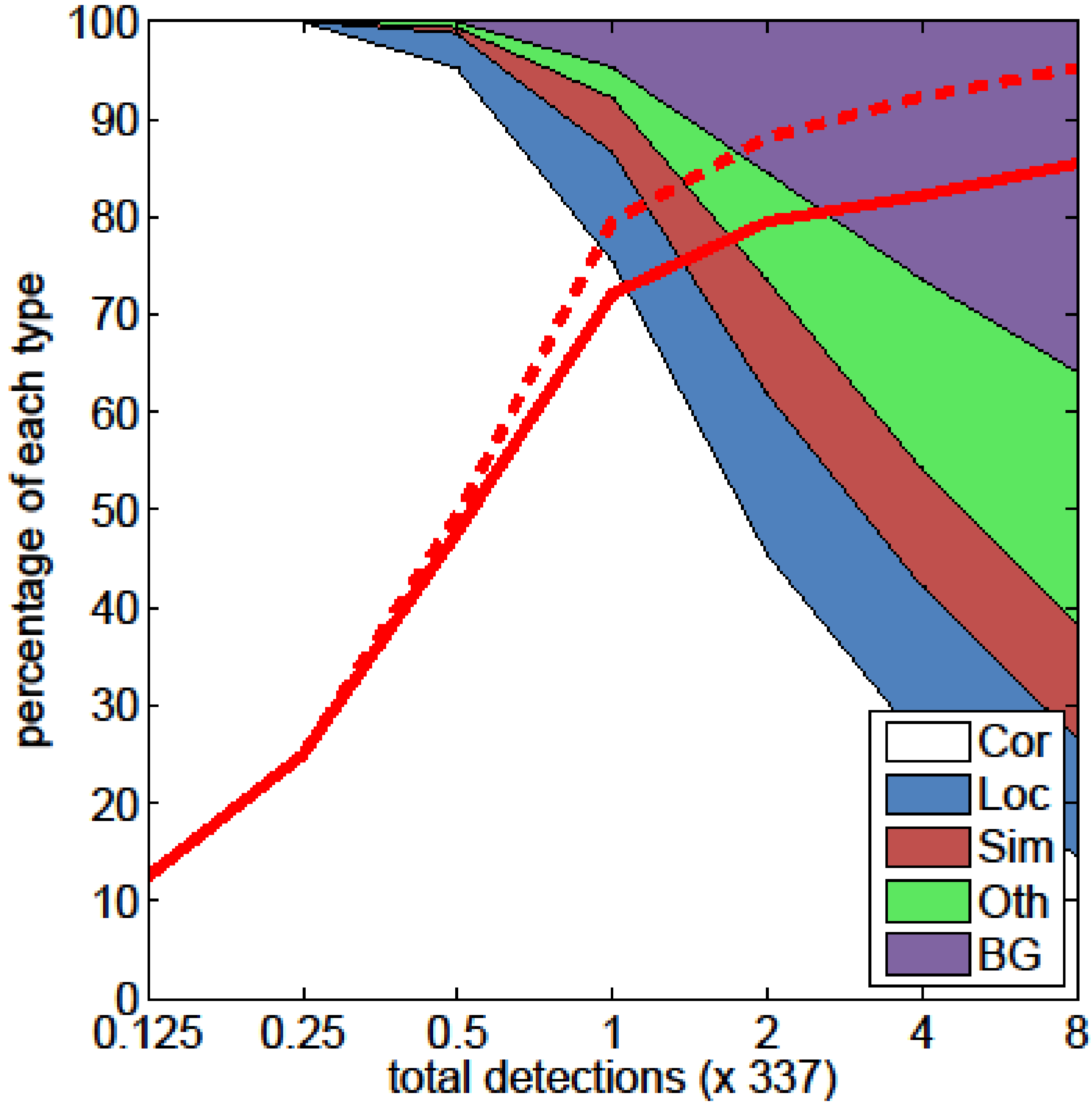
1. Train full network on ImageNet 1000-class classification
2. Replace classification layer with output layer for VOC (e.g. confidences for 20 classes)
3. Train on VOC pos/neg examples with low initial learning rate (1/10th what is used for new network)

Notes

- This usually works well if the “big data” task and target task are similar (object classification vs detection)
  - 0.45 AP without fine-tuning → 0.54 AP with fine tuning; training only on VOC does much worse
- Not necessary if target task is also very big

Mistakes are often reasonable

Bicycle: AP = 0.73



Confident Mistakes

bicycle (loc): ov=0.44 1−r=0.70

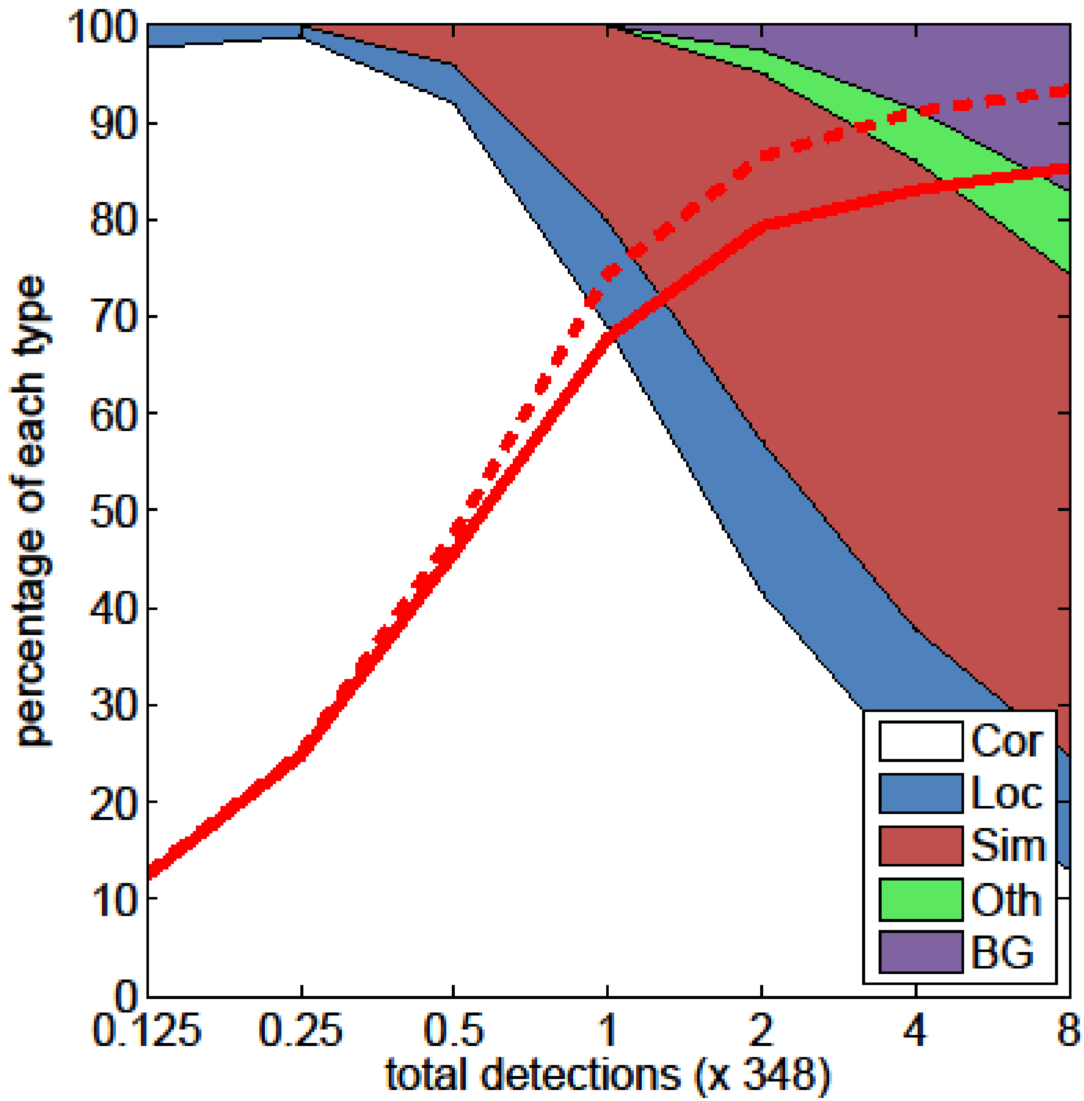
bicycle (sim): ov=0.00 1−r=0.56

bicycle (bg): ov=0.00 1−r=0.47

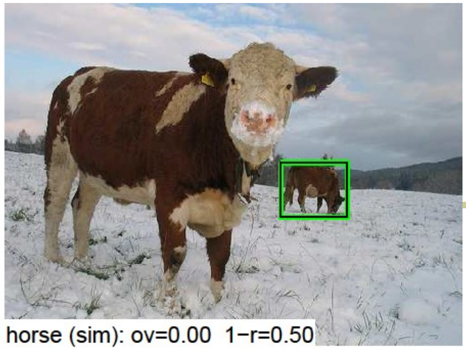
R-CNN results

Mistakes are often reasonable

Horse: AP = 0.69



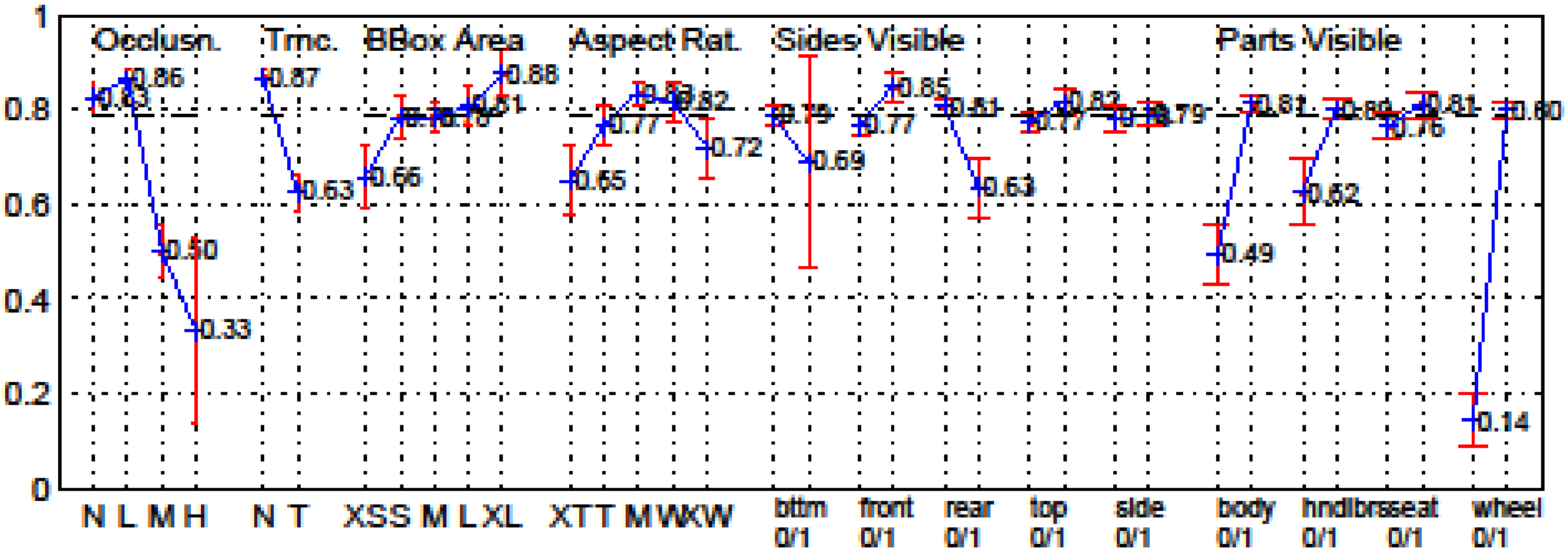
Confident Mistakes



R-CNN results

Misses are often predictable

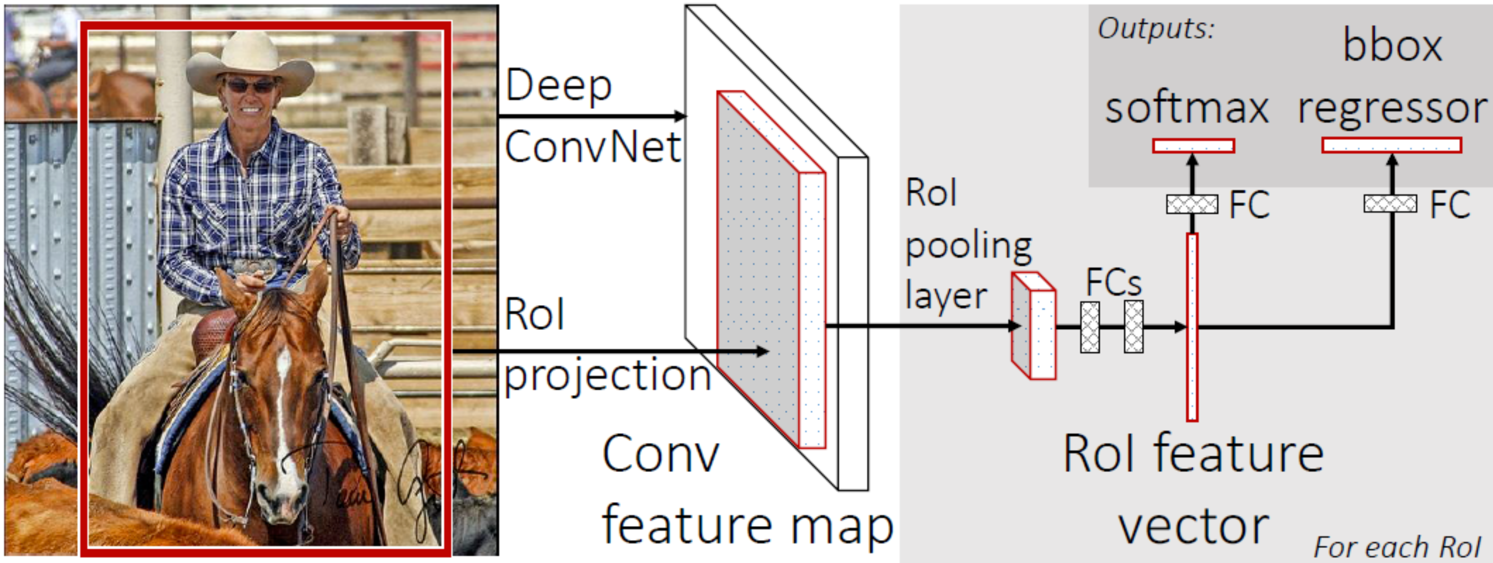
Bicycle



Small objects, distinctive parts absent or occluded, unusual views

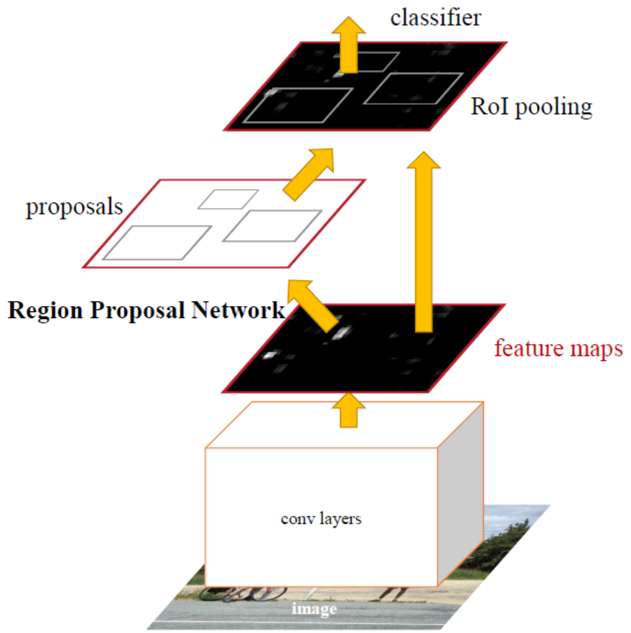
R-CNN results

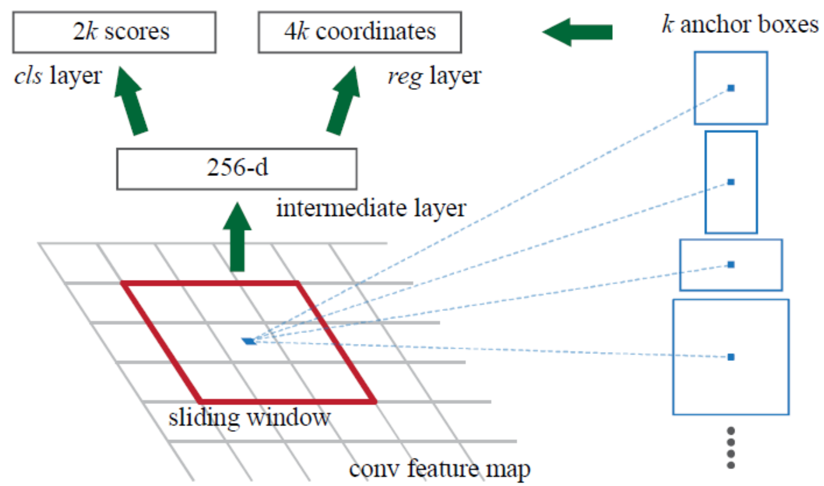
Fast R-CNN – Girshick 2015



- Compute CNN features for image once
- Pool into 7x7 spatial bins for each region proposal, output class scores and regressed bboxes
- 100x speed up of R-CNN (0.02 – 0.1 FPS → 0.5-20 FPS) with similar accuracy

Faster R-CNN – Ren et al. 2016





- Convolutional features used for generating proposals and scoring
  - Generate proposals with “objectness” scores and refined bboxes for each of k “anchors”
  - Score proposals in same way as Fast R-CNN
- Similar accuracy to Fast R-CNN with 10x speedup

- Faster R-CNN slightly better accuracy than Fast R-CNN
- More data improves results considerably

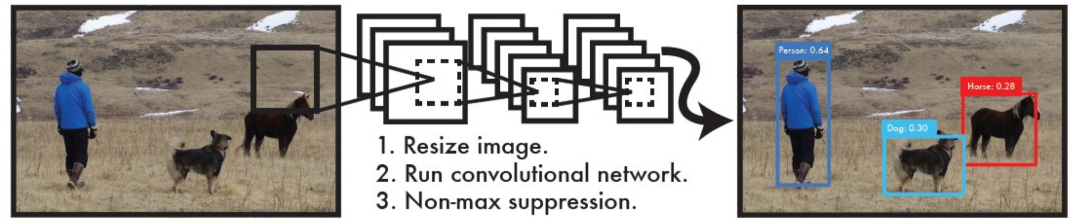
Table 6: Results on PASCAL VOC 2007 test set with Fast R-CNN detectors and VGG-16. For RPN, the train-time proposals for Fast R-CNN are 2000. RPN* denotes the unsharing feature version.

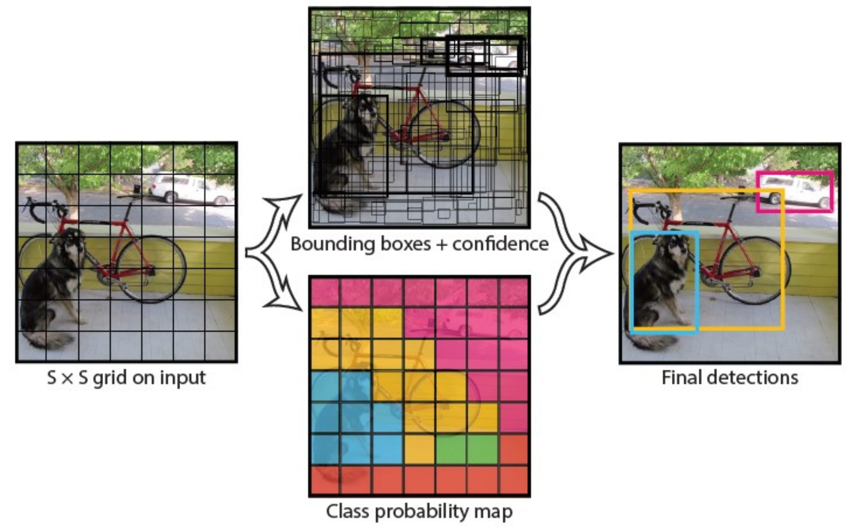
method	# box	data	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
SS	2000	07	66.9	74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8
SS	2000	07+12	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
RPN*	300	07	68.5	74.1	77.2	67.7	53.9	51.0	75.1	79.2	78.9	50.7	78.0	61.1	79.1	81.9	72.2	75.9	37.2	71.4	62.5	77.4	66.4
RPN	300	07	69.9	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3	67.2	80.3	79.8	75.0	76.3	39.1	68.3	67.3	81.1	67.6
RPN	300	07+12	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
RPN	300	COCO+07+12	78.8	84.3	82.0	77.7	68.9	65.7	88.1	88.4	88.9	63.6	86.3	70.8	85.9	87.6	80.1	82.3	53.6	80.4	75.8	86.6	78.9

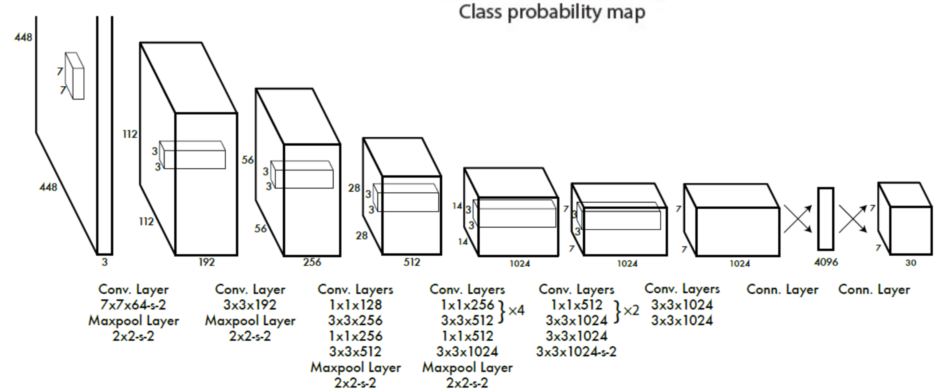
YOLO – Redmon et al. 2016

1. CNN produces 4096 features for 7x7 grid on image (fully conv.)
2. Each cell produces a score for each object and 2 bboxes w/ conf
3. Non-max suppression

- 7x speedup over Faster RCNN (45-155 FPS vs. 7-18 FPS)
- Some loss of accuracy due to lower recall, poor localization







[Yolo v2](https://youtu.be/VOC3huqHrss) – Redmon et al. 2017

- Batch normalization
- Pre-train on higher resolution ImageNet
- Use and improve anchor box idea from Faster RCNN
- Train at multiple resolutions
- Very good accuracy, very fast

	YOLO								YOLOv2
batch norm?		✓	✓	✓	✓	✓	✓	✓	✓
hi-res classifier?			✓	✓	✓	✓	✓	✓	✓
convolutional?				✓	✓	✓	✓	✓	✓
anchor boxes?				✓	✓				
new network?					✓	✓	✓	✓	✓
dimension priors?						✓	✓	✓	✓
location prediction?						✓	✓	✓	✓
passthrough?							✓	✓	✓
multi-scale?								✓	✓
hi-res detector?									✓
VOC2007 mAP	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8	**78.6**

Detection Frameworks	Train	mAP	FPS
Fast R-CNN [5]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[15]	2007+2012	73.2	7
Faster R-CNN ResNet[6]	2007+2012	76.4	5
YOLO [14]	2007+2012	63.4	45
SSD300 [11]	2007+2012	74.3	46
SSD500 [11]	2007+2012	76.8	19
YOLOv2 288×288	2007+2012	69.0	91
YOLOv2 352×352	2007+2012	73.7	81
YOLOv2 416×416	2007+2012	76.8	67
YOLOv2 480×480	2007+2012	77.8	59
YOLOv2 544×544	2007+2012	**78.6**	40

https://youtu.be/VOC3huqHrss

Reading list

- https://culurciello.github.io/tech/2016/06/04/nets.html
- Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86(11): 2278–2324, 1998.
- A. Krizhevsky, I. Sutskever, and G. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012
- D. Kingma and J. Ba, Adam: A Method for Stochastic Optimization, ICLR 2015
- M. Zeiler and R. Fergus, Visualizing and Understanding Convolutional Networks, ECCV 2014 (best paper award)
- K. Simonyan and A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015
- M. Lin, Q. Chen, and S. Yan, Network in network, ICLR 2014
- C. Szegedy et al., Going deeper with convolutions, CVPR 2015
- C. Szegedy et al., Rethinking the inception architecture for computer vision, CVPR 2016
- K. He, X. Zhang, S. Ren, and J. Sun, Deep Residual Learning for Image Recognition, CVPR 2016 (best paper award)

Influential Works in Detection

- Sung-Poggio (1994, 1998) : ~2412 citations
  - Basic idea of statistical template detection (I think), bootstrapping to get "face-like" negative examples, multiple whole-face prototypes (in 1994)
- Rowley-Baluja-Kanade (1996-1998) : ~4953
  - "Parts" at fixed position, non-maxima suppression, simple cascade, rotation, pretty good accuracy, fast
- Schneiderman-Kanade (1998-2000,2004) : ~2600
  - Careful feature/classifier engineering, excellent results, cascade
- Viola-Jones (2001, 2004) : ~27,000
  - Haar-like features, Adaboost as feature selection, hyper-cascade, very fast, easy to implement
- Dalal-Triggs (2005) : ~18000
  - Careful feature engineering, excellent results, HOG feature, online code
- Felzenszwalb-Huttenlocher (2000): ~2100
  - Efficient way to solve part-based detectors
- Felzenszwalb-McAllester-Ramanan (2008,2010): ~10788
  - Excellent template/parts-based blend
- Girshick-Donahue-Darrell-Malik (2014): ~19788
  - Region proposals + fine-tuned CNN features (marks significant advance in accuracy over hog-based methods)
- Redmon, Divvala, Girshick, Farhadi (2016): ~16386
  - Refine and simplify RCNN++ approach to predict directly from last conv layer