Recognition and Learning

Dr. Sander Ali Khowaja,
Assistant Professor, Department of Telecommunication Engineering
Faculty of Engineering and Technology, University of Sindh, Pakistan
Senior Member, IEEE – Member, ACM
https://sander-ali.github.io

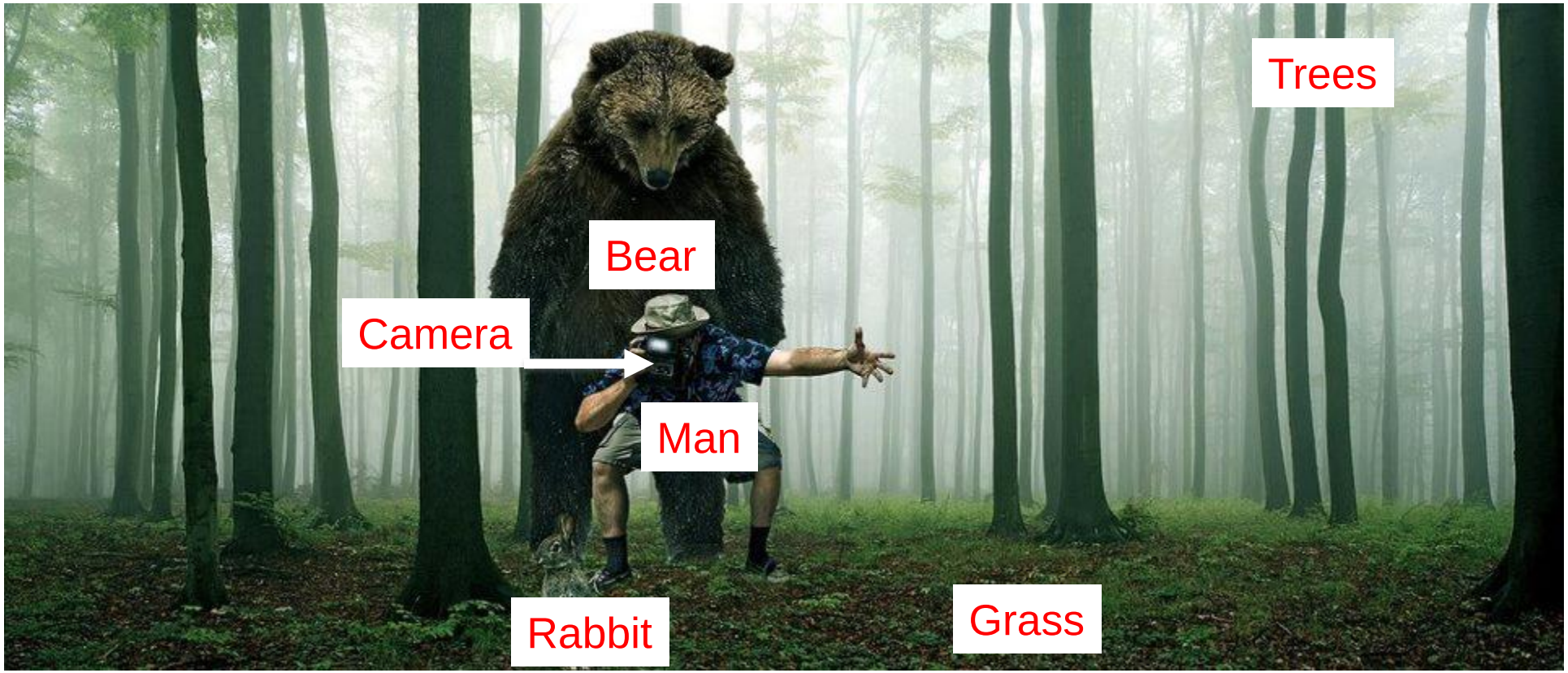
Image Features and Categorization

Recognition and Learning

- Image Features and Categorization
- Convolutional Neural Networks
- Object Detection
- Part and Pixel Labeling
- Action Recognition
- Vision and Language


AIPOLY
Mashable
patio
Patio.

What do you see in this image?



Forest

Describe, Predict, or Interact with the object based on visual cues

Is it **dangerous**?

Is it **alive**?

How **fast** does it run?

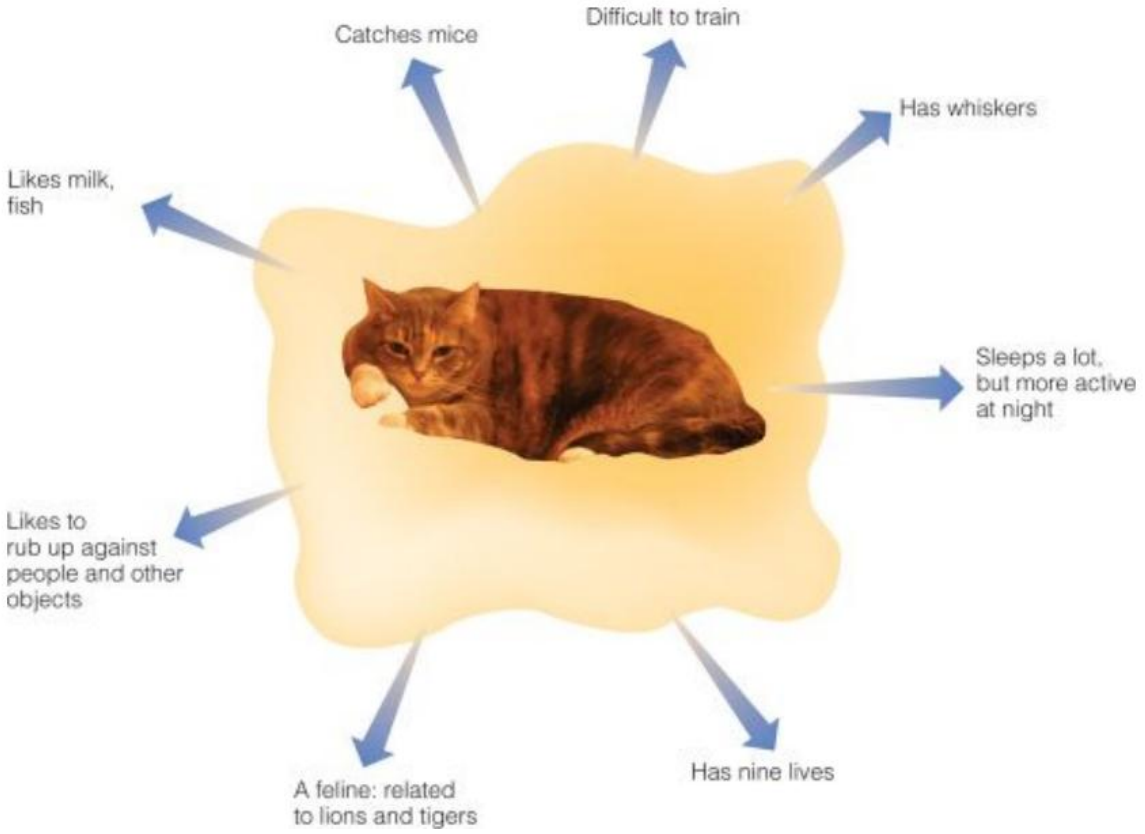
Is it **soft**?

Does it have a **tail**?

Can I **poke with it**?

Why do we care about categories?

- From an object's category, we can make predictions about its behavior in the future, beyond of what is immediately perceived.
- Pointers to knowledge
  - Help to understand individual cases not previously encountered
- Communication



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

The Categories (Aristotle)

Aristotle by Francesco Hayez

Slide Credit: A. A. Efros

Prototype or sum of exemplars?

Prototype Model

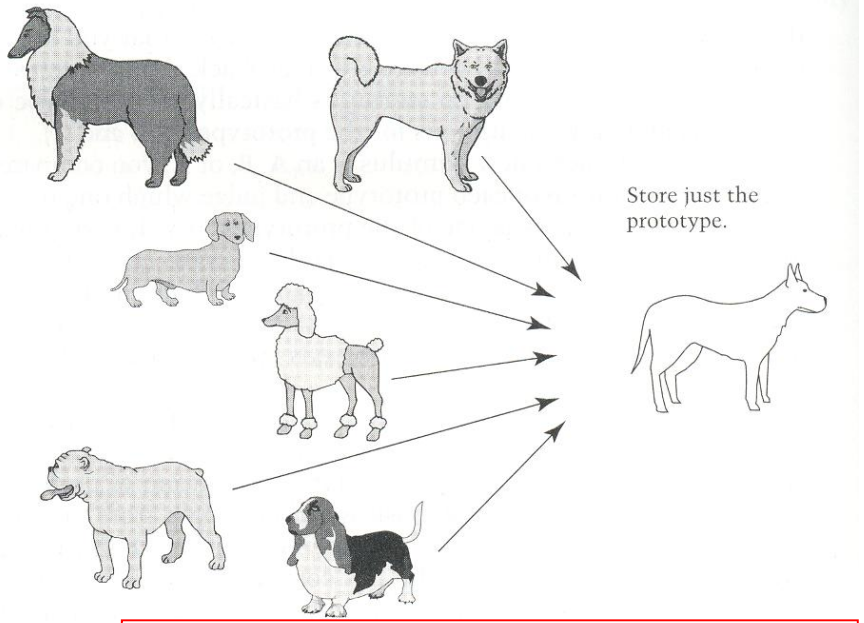


Figure 7.3. *Schematic of the prototype model. Although many exemplars are seen, only the prototype is stored. The prototype is updated continually to incorporate more experience with new exemplars.*

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

Exemplars Model

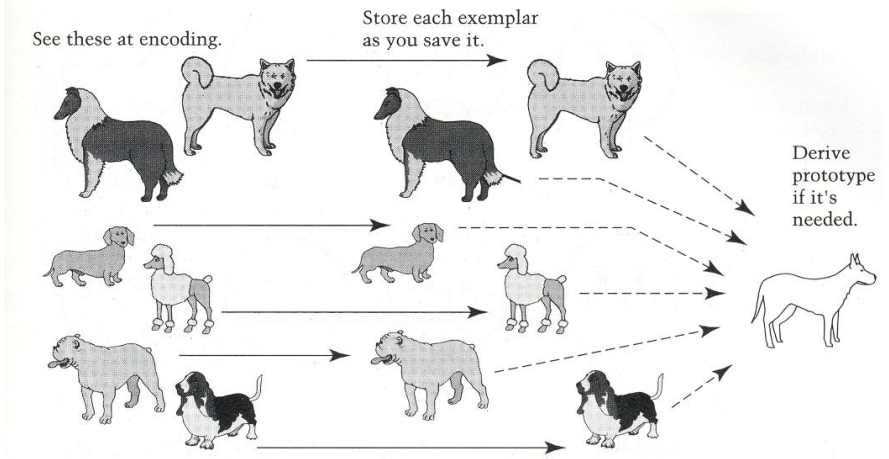


Figure 7.4. *Schematic of the exemplar model. As each exemplar is seen, it is encoded into memory. A prototype is abstracted only when it is needed, for example, when a new exemplar must be categorized.*

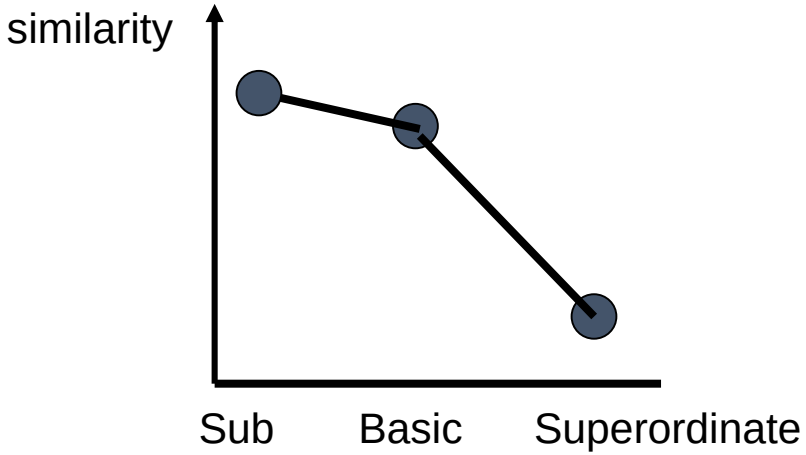
Category judgments are made
by comparing a new exemplar
to all the old exemplars of a category
or to the exemplar that is the most
appropriate

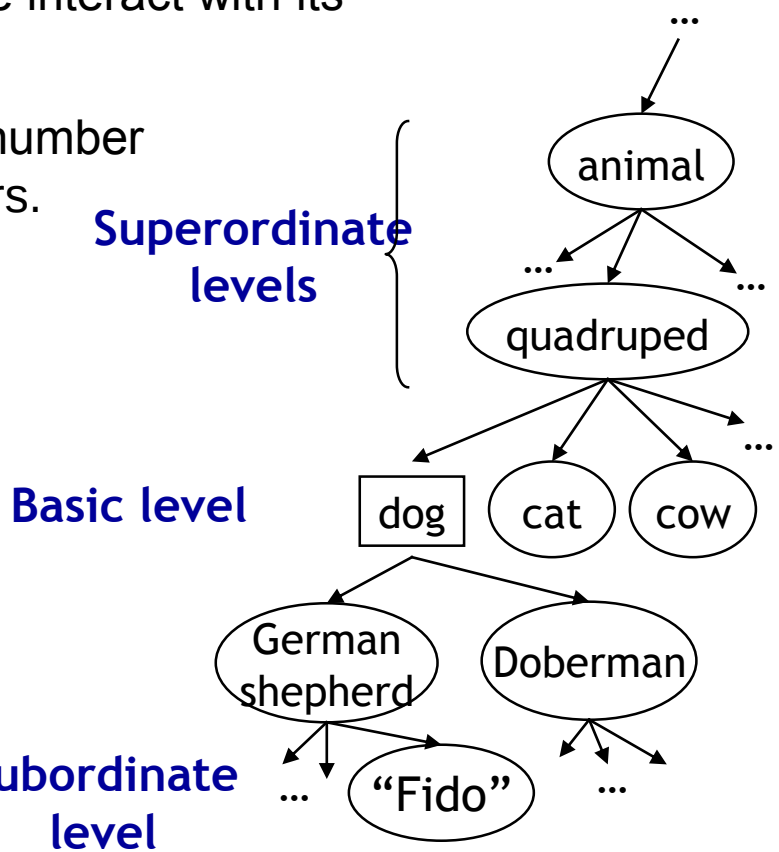
Slide Credit: Torralba

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.





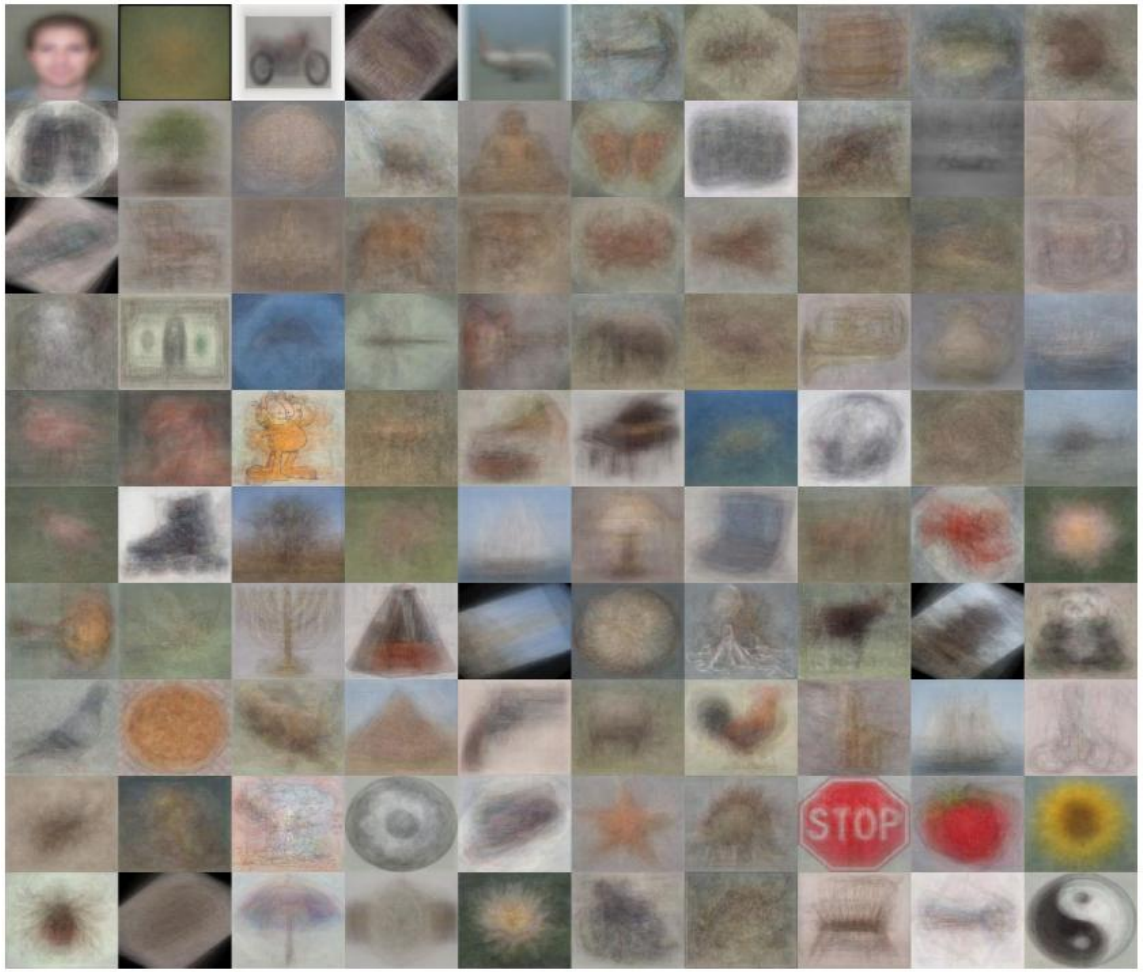
Rosch et a. Principle of categorization, 1978

Image Categorization

- Cat vs Dog

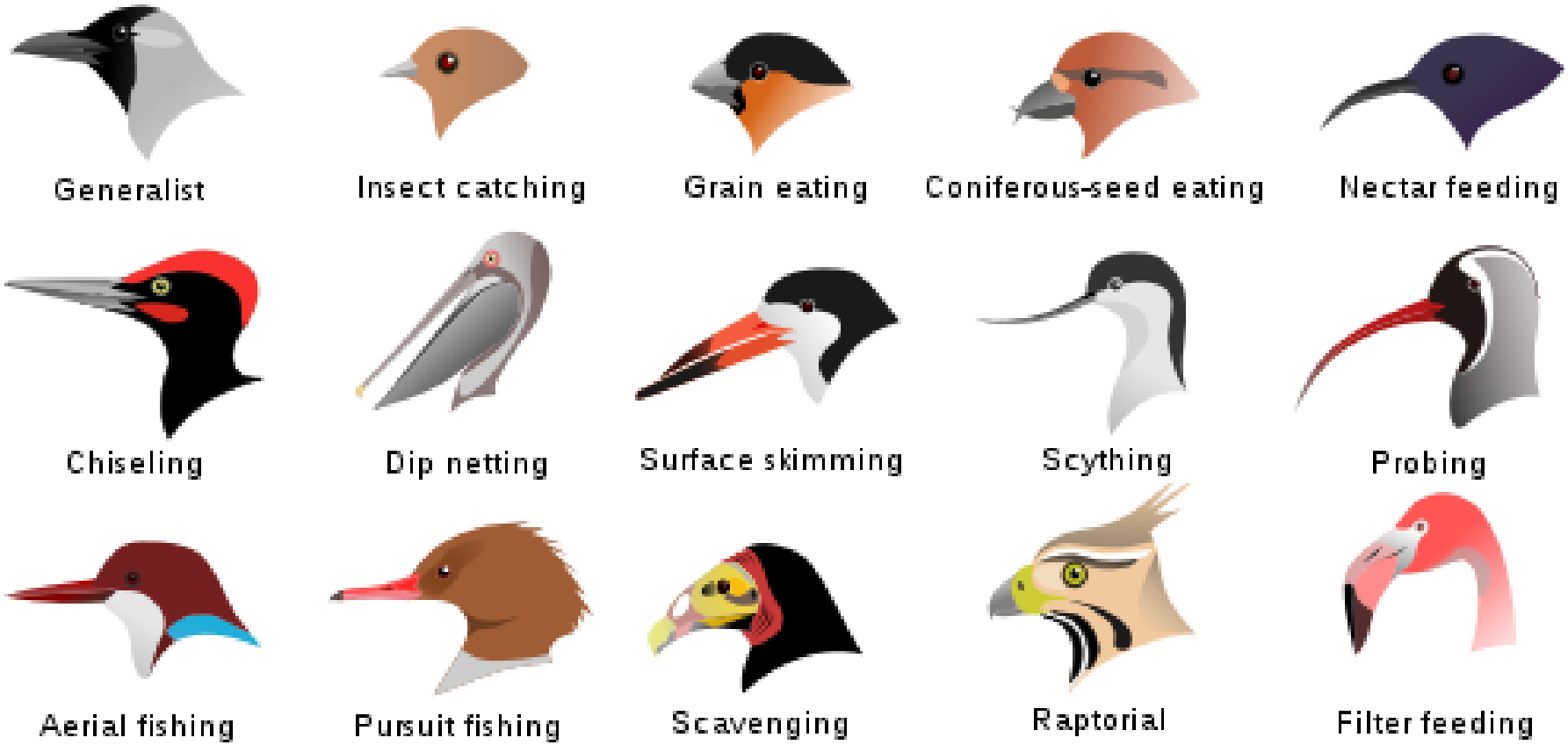
Image Categorization

- Object recognition



Caltech 101 Average Object Images

Image Categorization



[Visipedia Project]

Image Categorization

- Place recognition

Places Database [Zhou et al. NIPS 2014]

Image Categorization

- Visual font recognition

[Chen et al. CVPR 2014]

Image Categorization

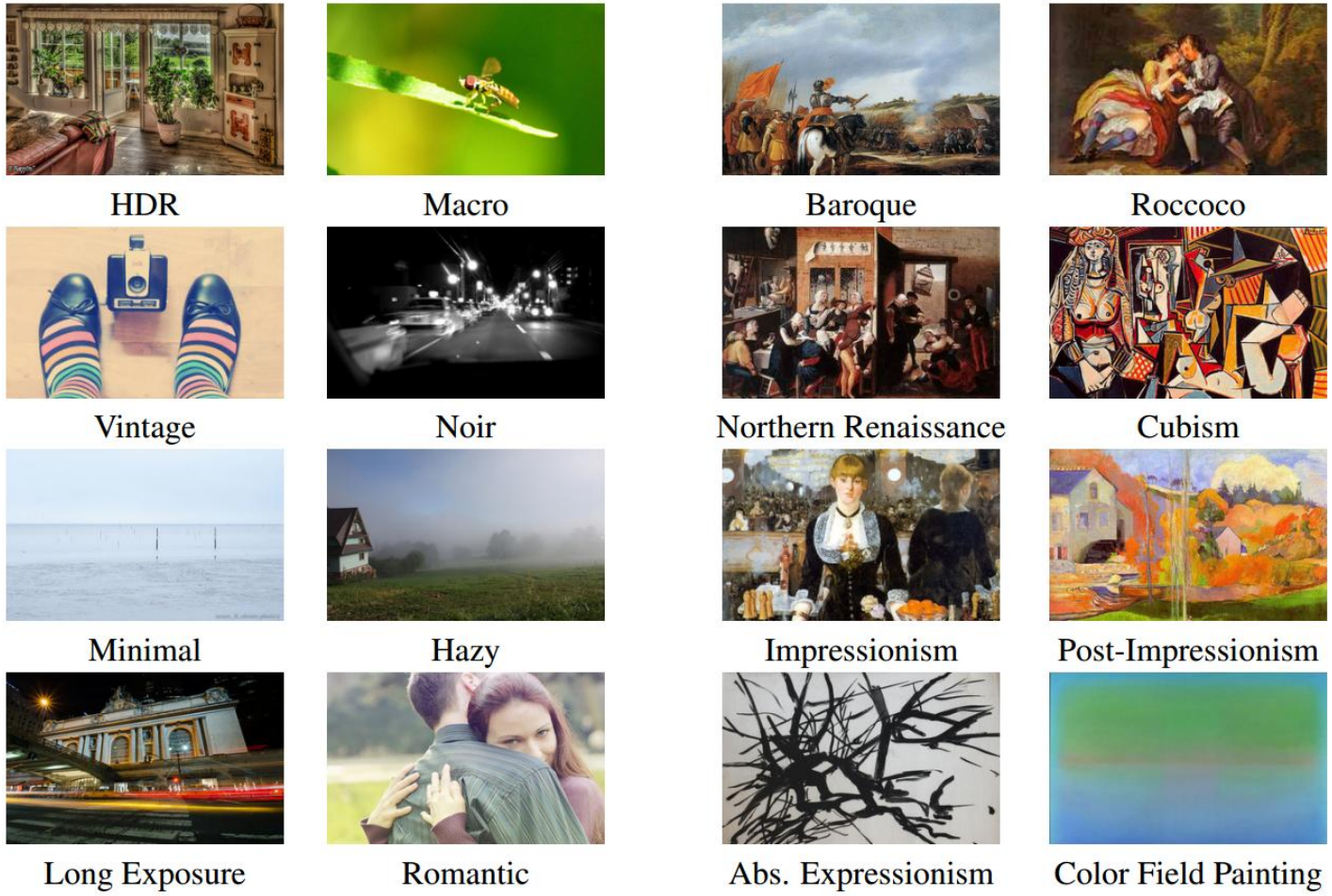
- Dating historical photos

1940 1953 1966 1977

[Palermo et al. ECCV 2012]

Image Categorization

- Image style recognition

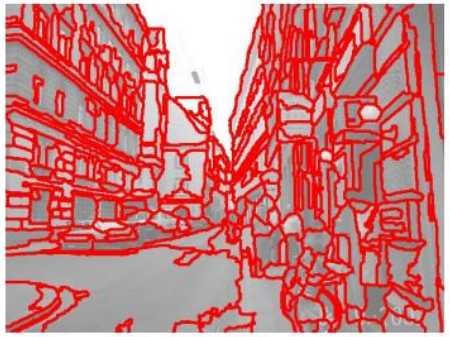


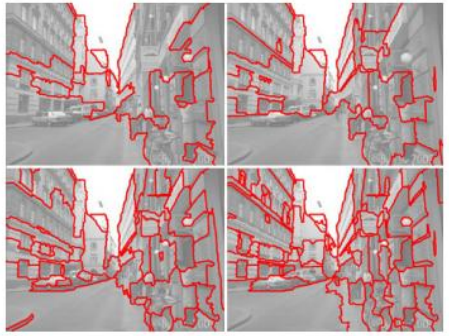
[Karayev et al. BMVC 2014]

Region Categorization

- Layout prediction


Input

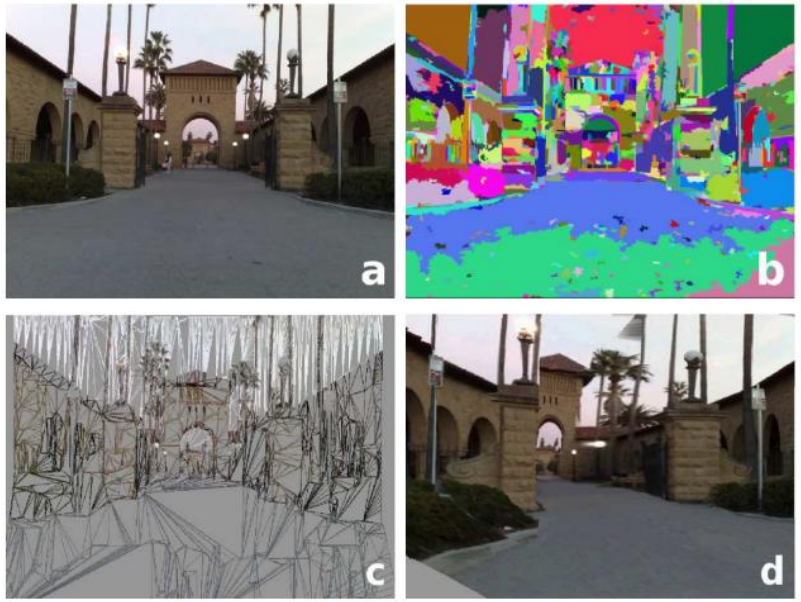

Superpixels


Multiple Segmentations


Surface Layout

Assign regions to orientation

Geometric context [Hoiem et al. IJCV 2007]

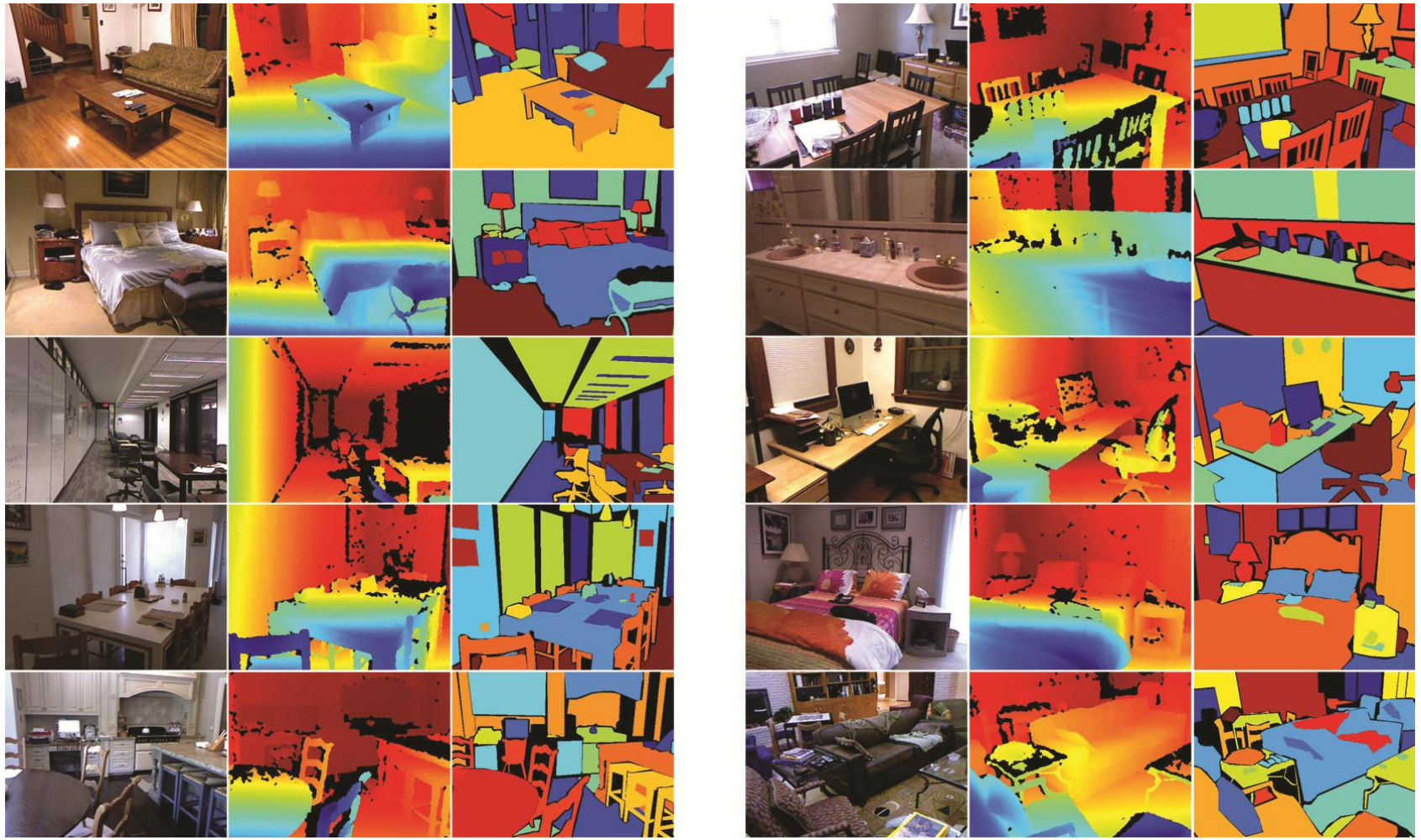


Assign regions to depth

Make3D [Saxena et al. PAMI 2008]

Region Categorization

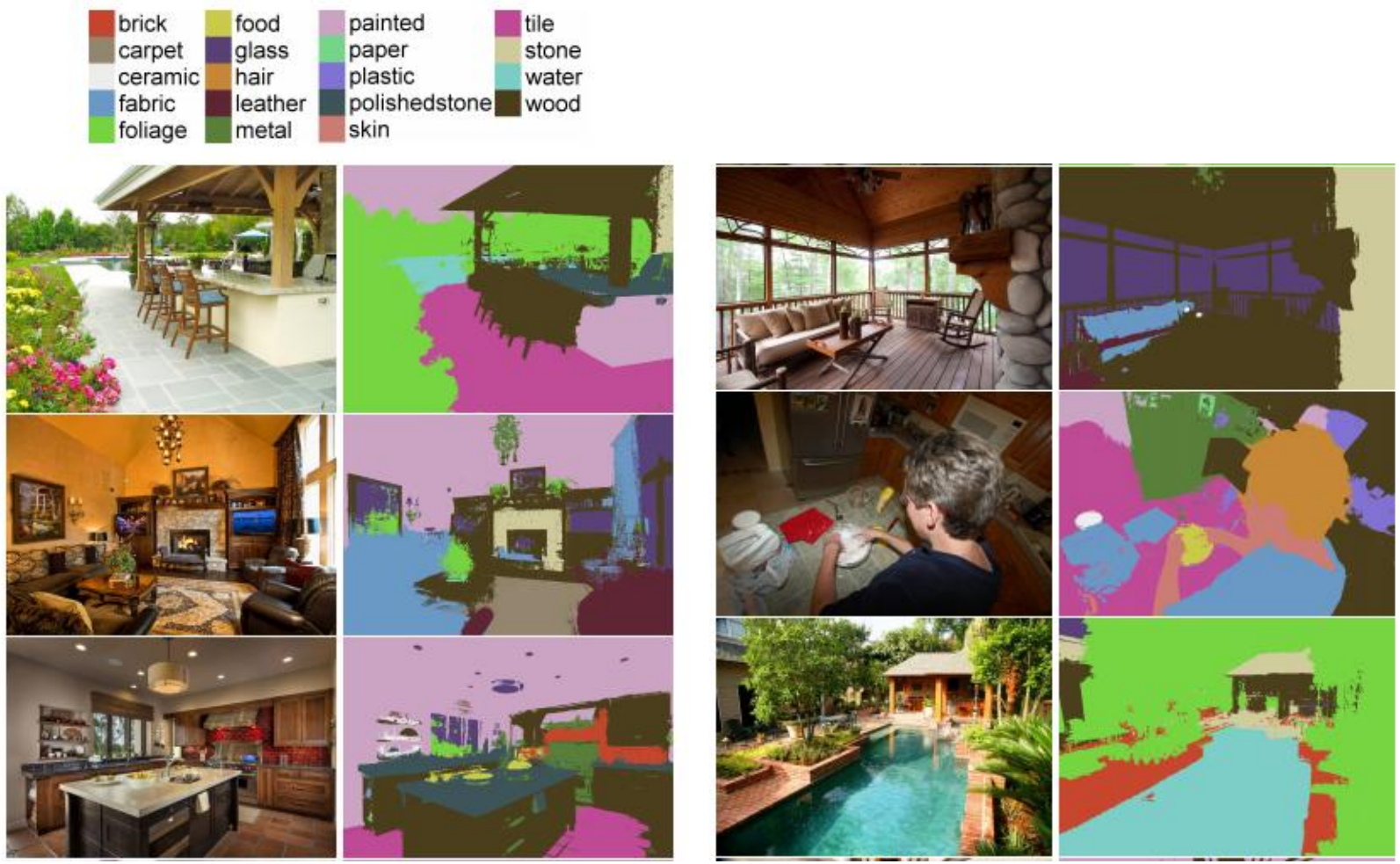
- Semantic segmentation from RGBD images



[Silberman et al. ECCV 2012]

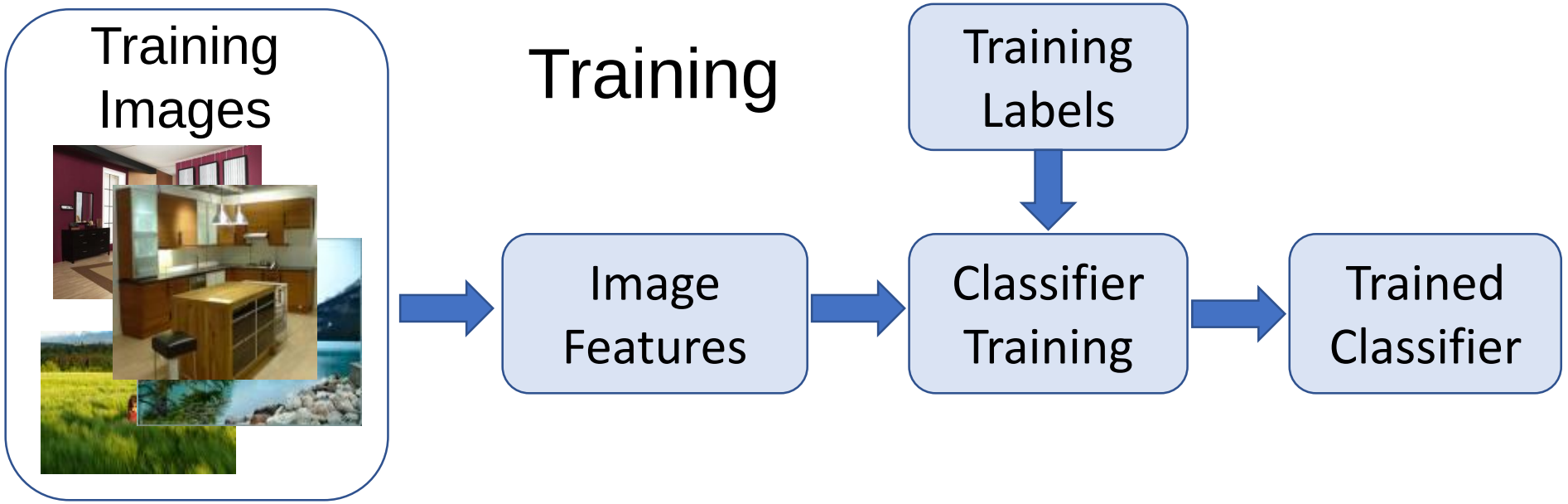
Region Categorization

- Material recognition

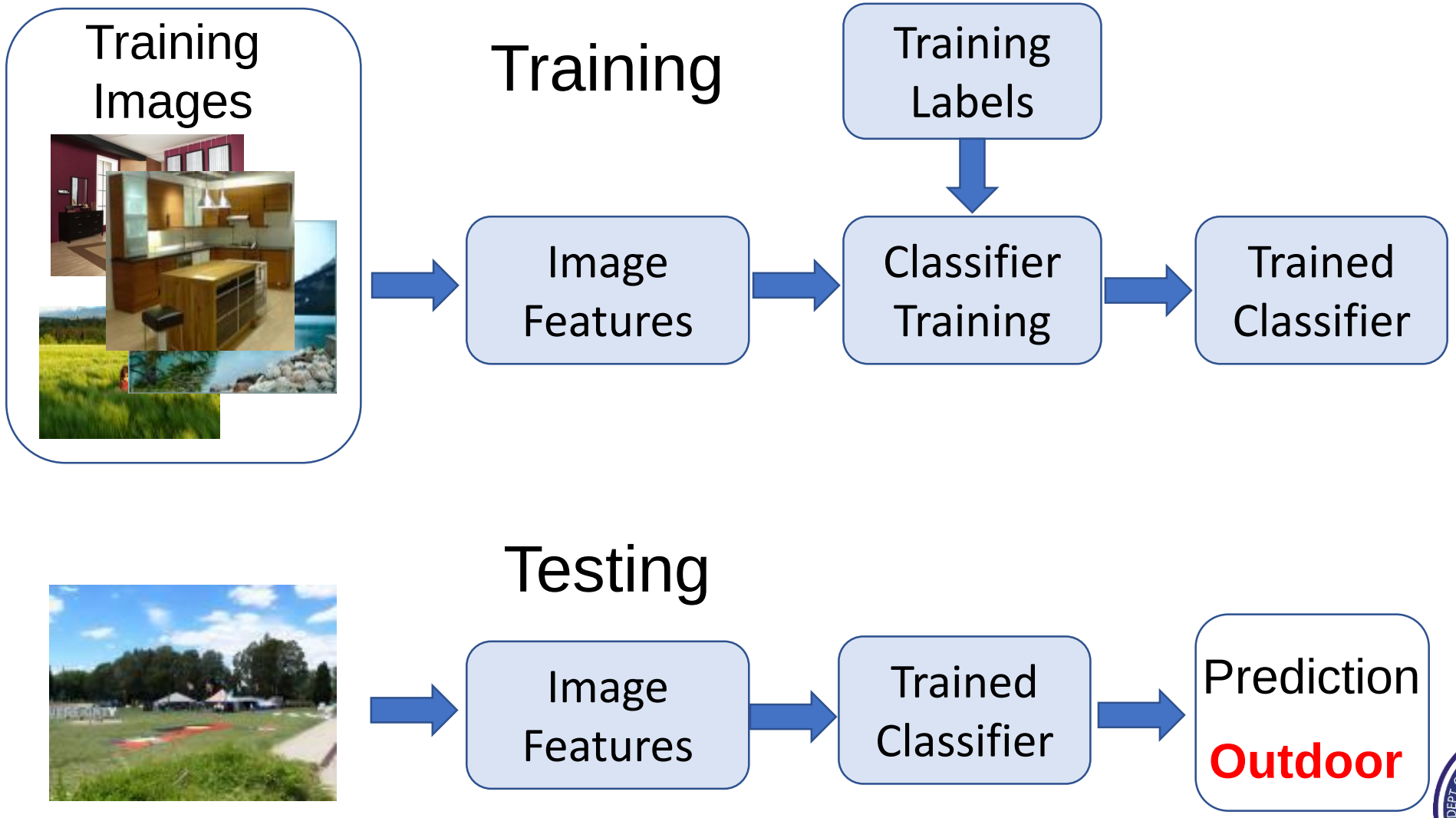


[Bell et al. CVPR 2015]

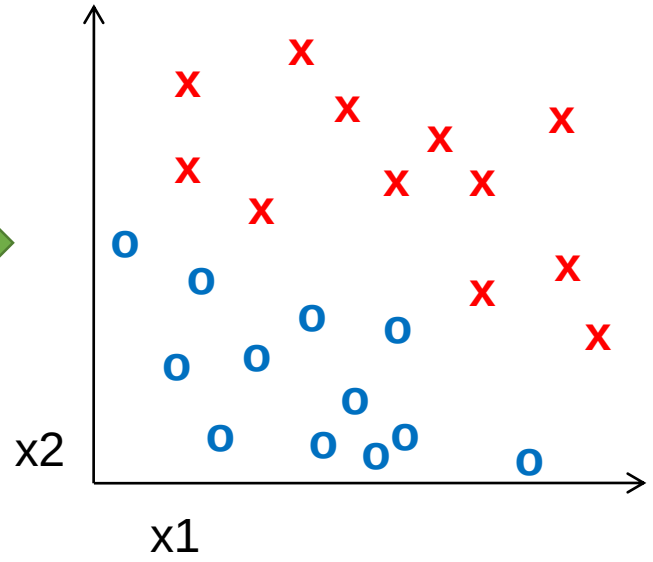
Training Phase



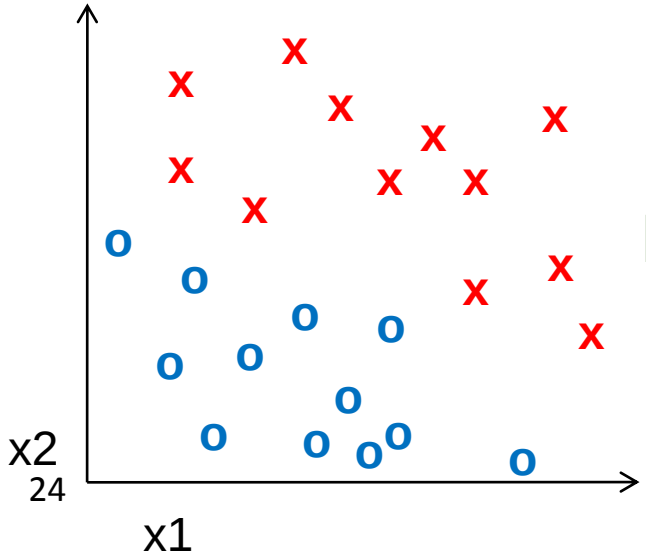
Testing Phase

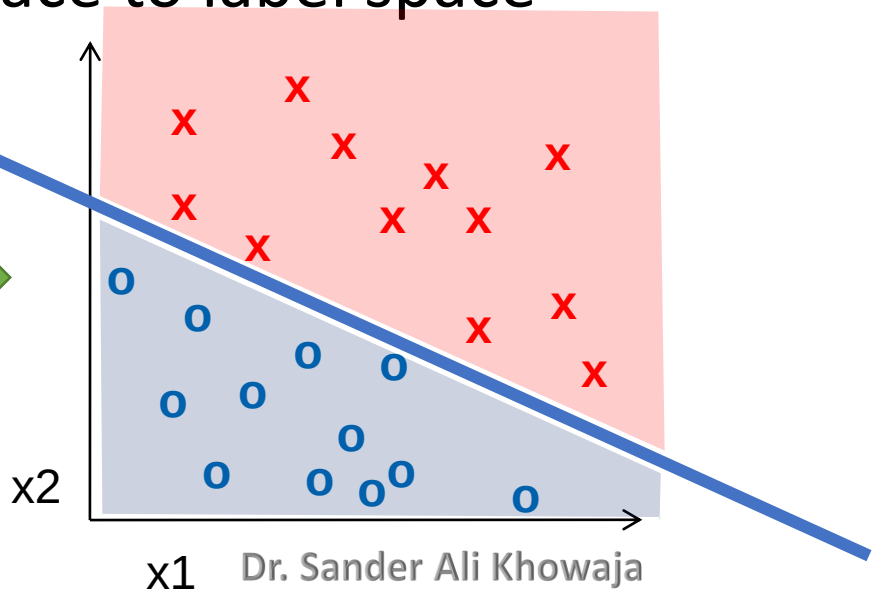


- **Image features**: map images to feature space



- **Classifiers**: map feature space to label space



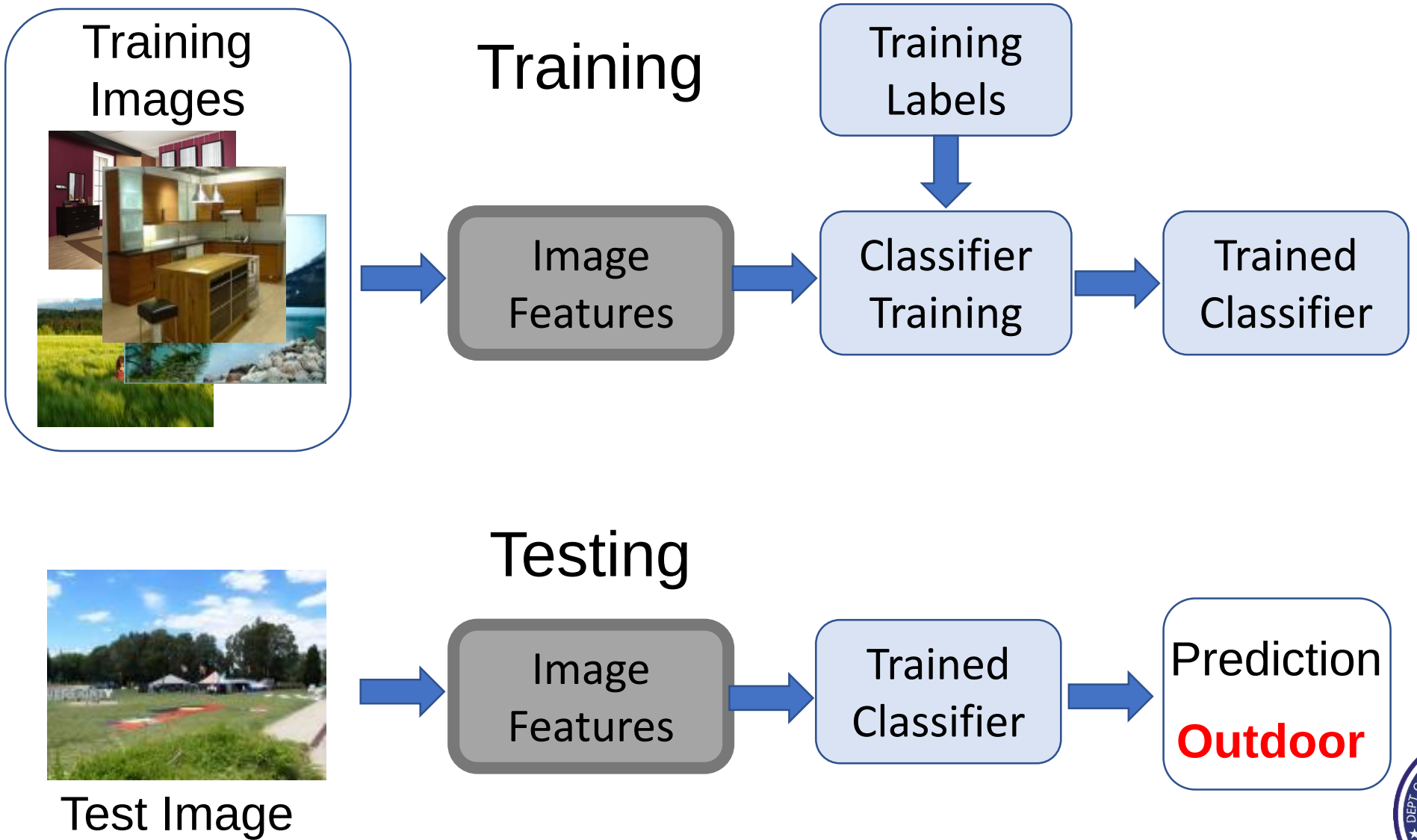


Different types of classification

- **Exemplar-based**: transfer category labels from examples with most similar features
  - What similarity function? What parameters?
- **Linear classifier**: confidence in positive label is a weighted sum of features
  - What are the weights?
- **Non-linear classifier**: predictions based on more complex function of features
  - What form does the classifier take? Parameters?
- **Generative classifier**: assign to the label that best explains the features (makes features most likely)
  - What is the probability function and its parameters?

Note: You can always fully design the classifier by hand, but usually this is too difficult. Typical solution: learn from training examples.

Testing Phase



What are good features for

- recognizing a beach?






What are good features for…

- recognizing cloth fabric?






What are good features for…

- recognizing a mug?






What are the right features?

Depend on what you want to know!

- Object: shape
  - Local shape info, shading, shadows, texture
- Scene : geometric layout
  - linear perspective, gradients, line segments
- Material properties: albedo, feel, hardness
  - Color, texture
- Action: motion
  - Optical flow, tracked points

General principles of representation

- **Coverage**
  - Ensure that all relevant info is captured
- **Concision**
  - Minimize number of features without sacrificing coverage
- **Directness**
  - Ideal features are independently useful for prediction

Image Representations

- Templates
  - Intensity, gradients, etc.
- Histograms
  - Color, texture, SIFT descriptors, etc.
- Average of features

Image Intensity

Gradient template

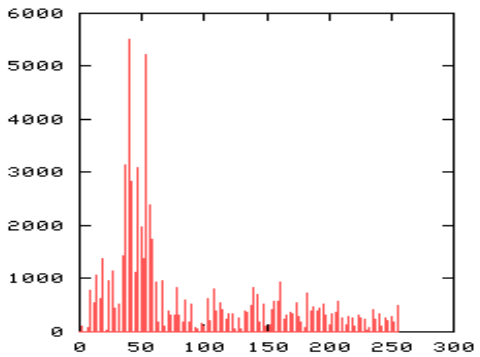
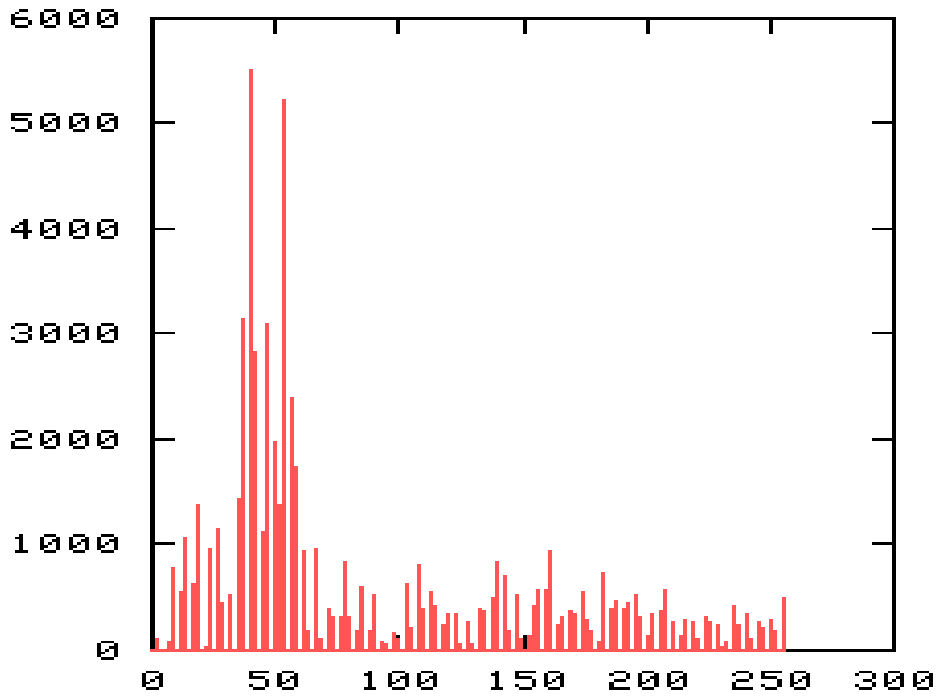
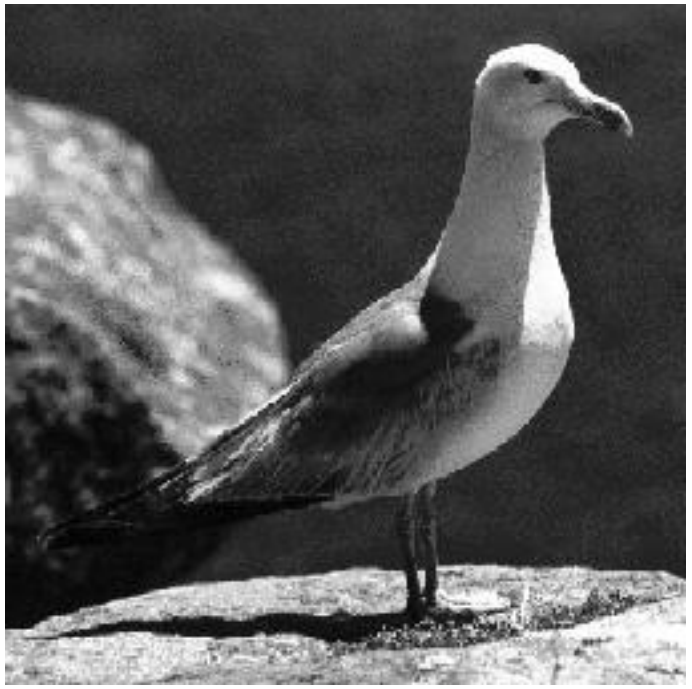


Image Representations: histograms





Global histogram

- Represent distribution of features
  - Color, texture, depth, …

Images from Dave Kauchak

Image Representations: Histograms

- Data samples in 2D

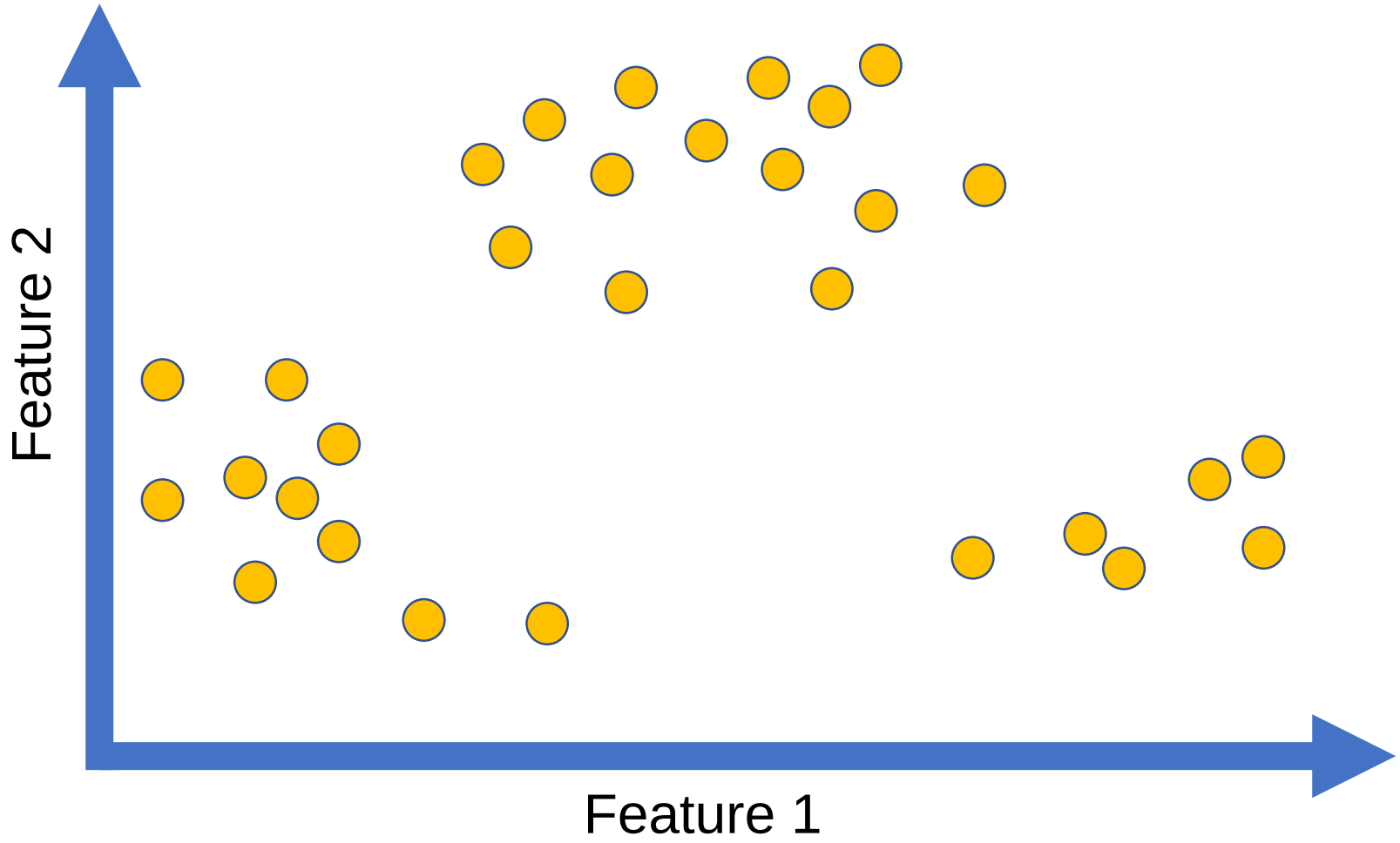


Image Representations: Histograms

- Probability or count of data in each bin
- Marginal histogram on feature 1

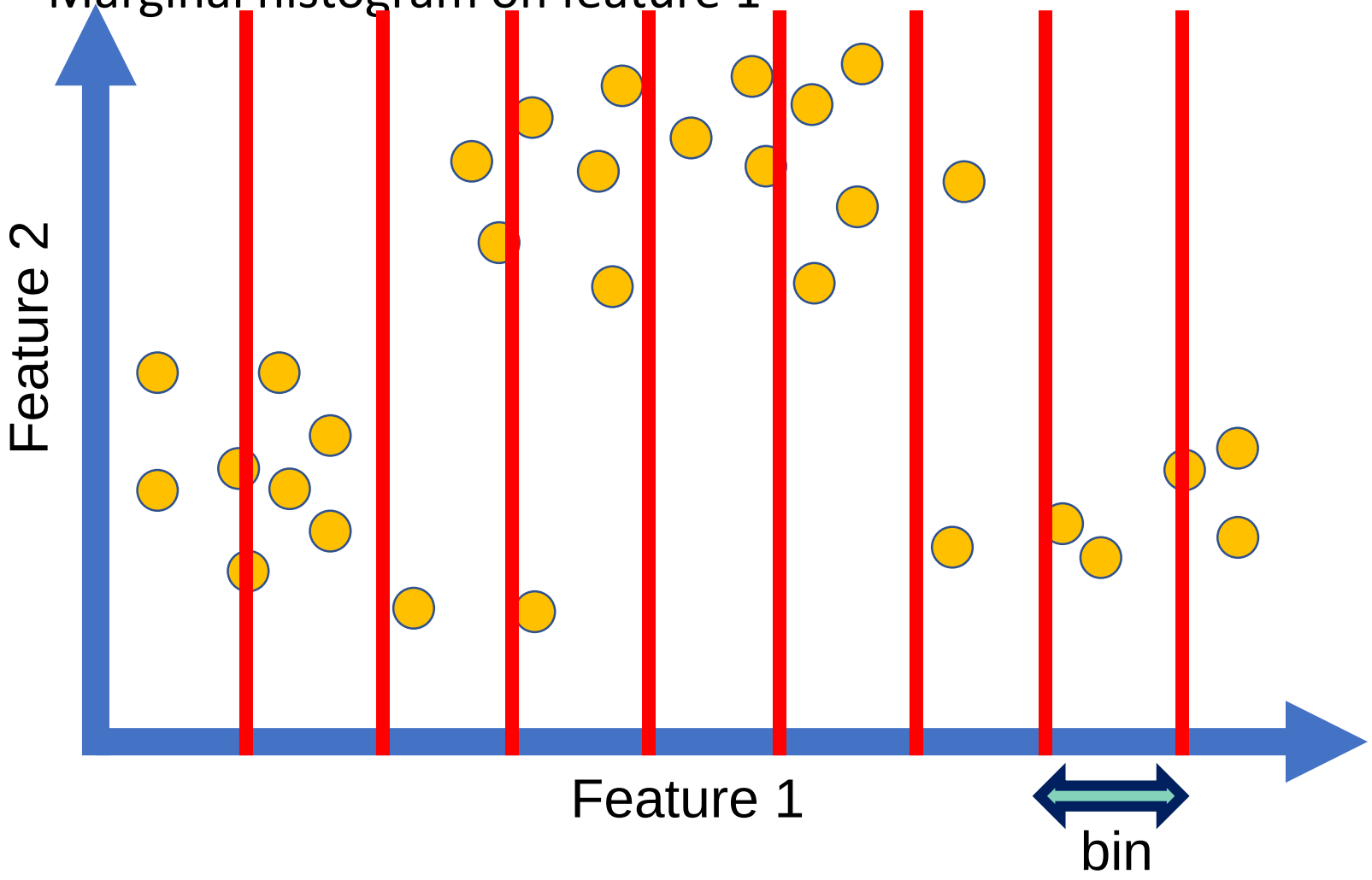


Image representations: histograms

- Marginal histogram on feature 2

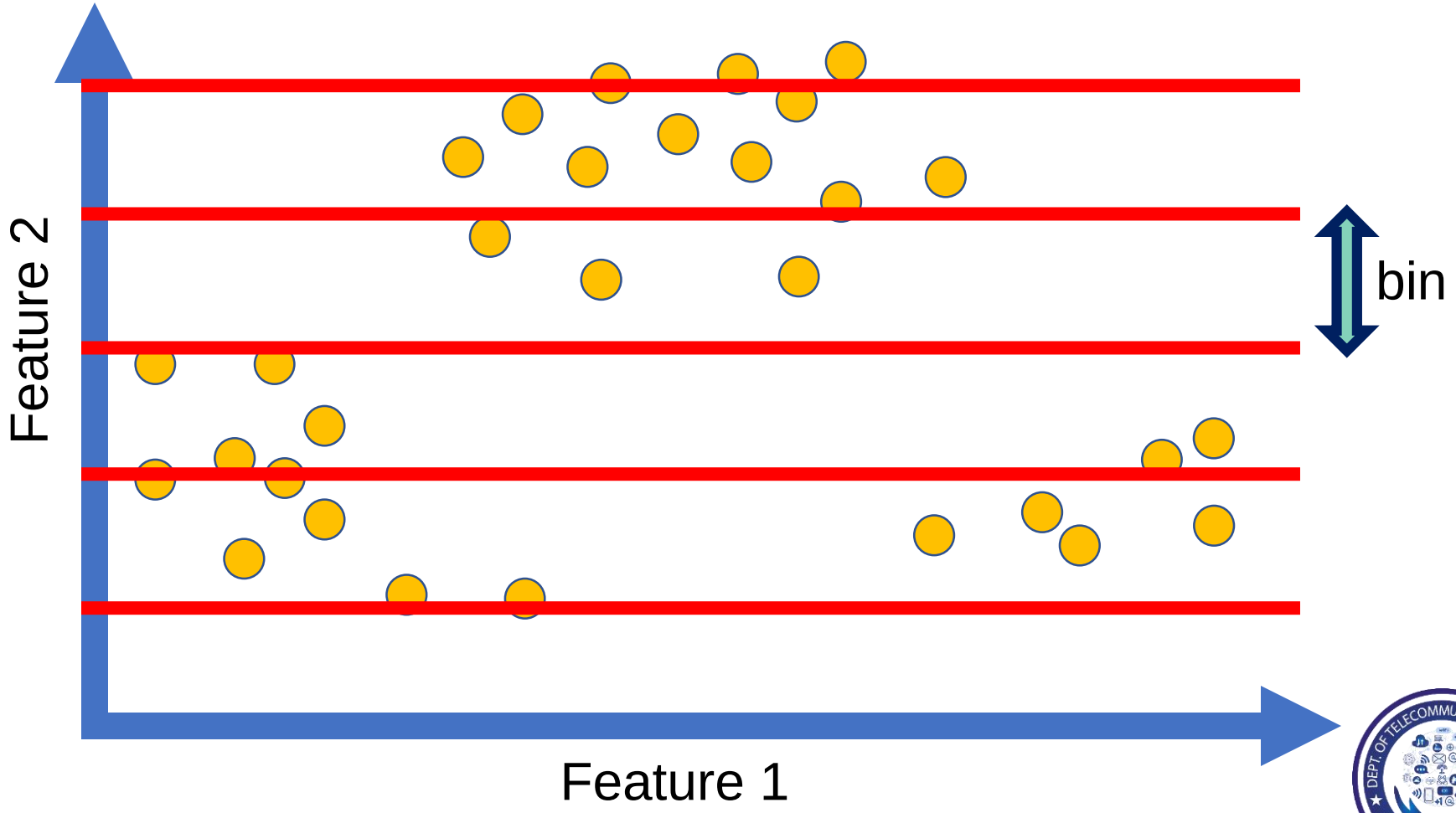
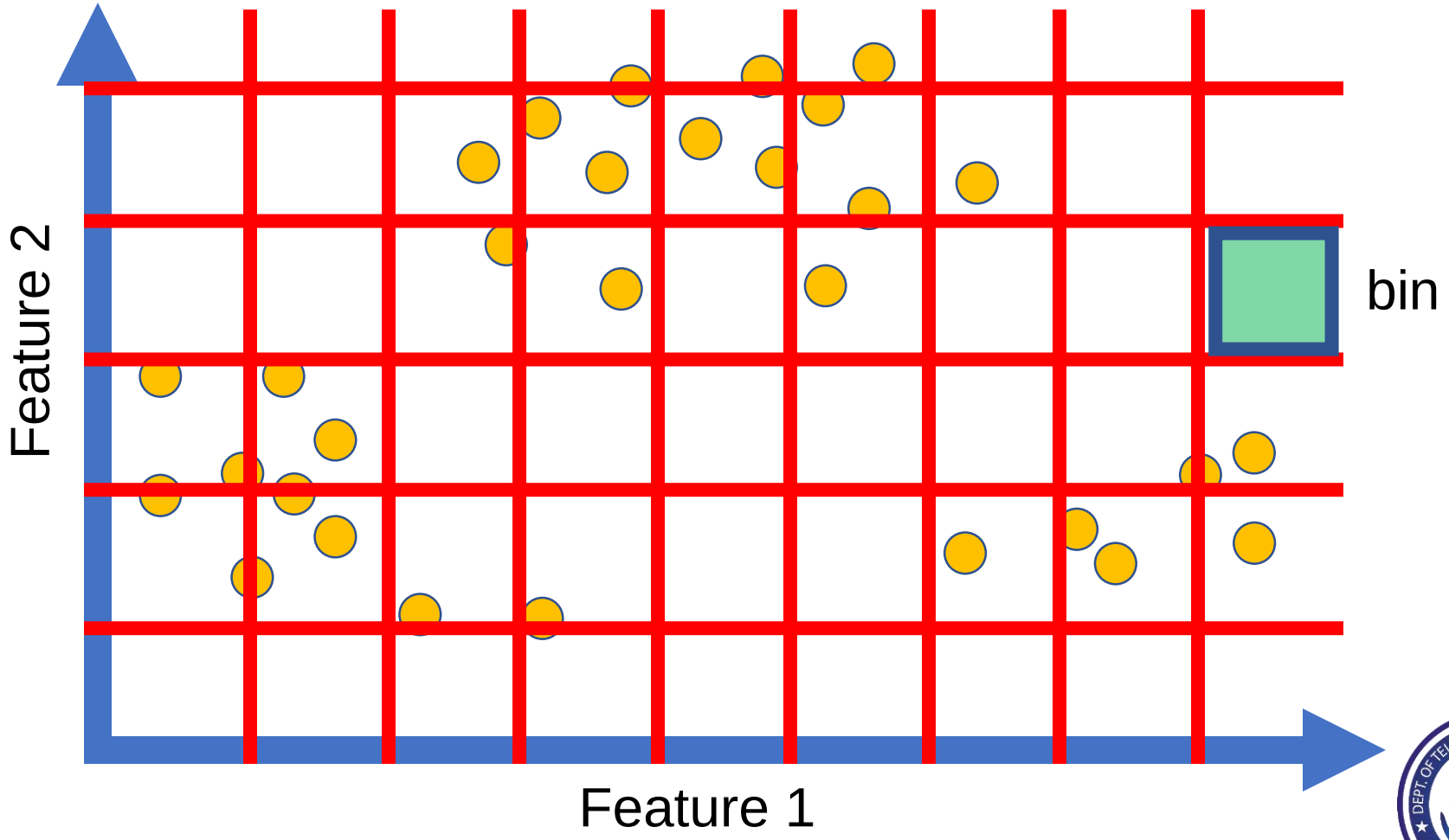
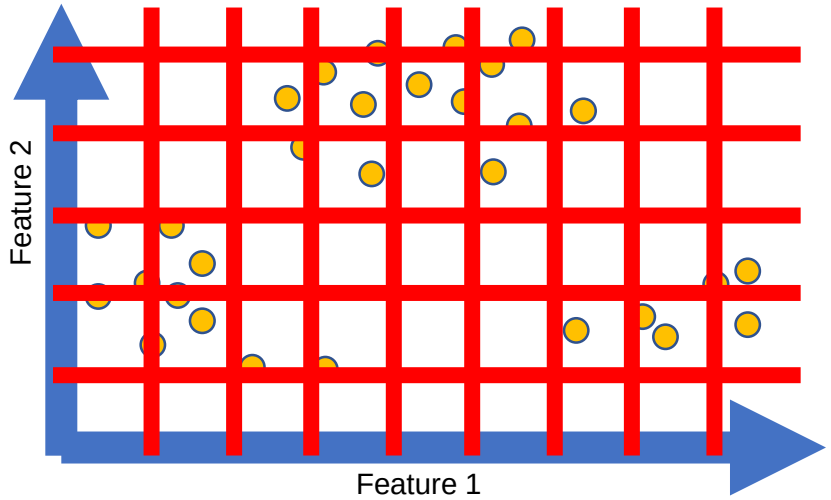


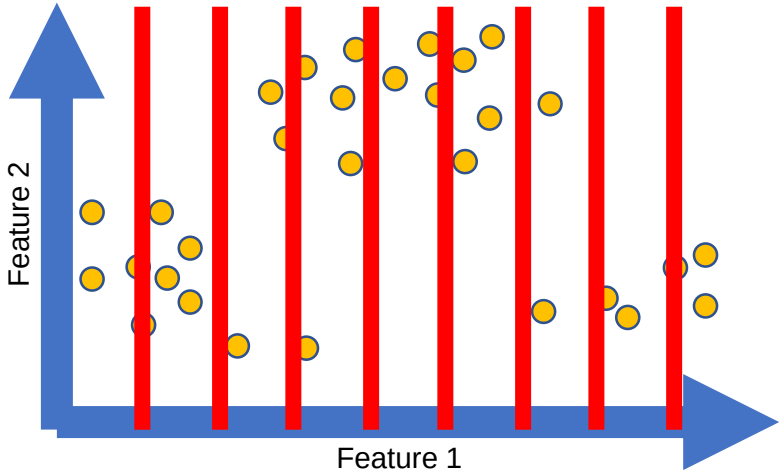
Image representations: histograms

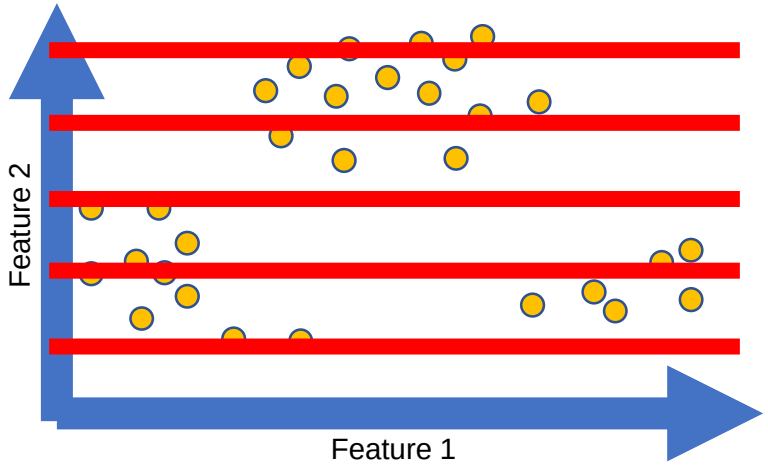
- Joint histogram



Modeling multi-dimensional data







Joint histogram

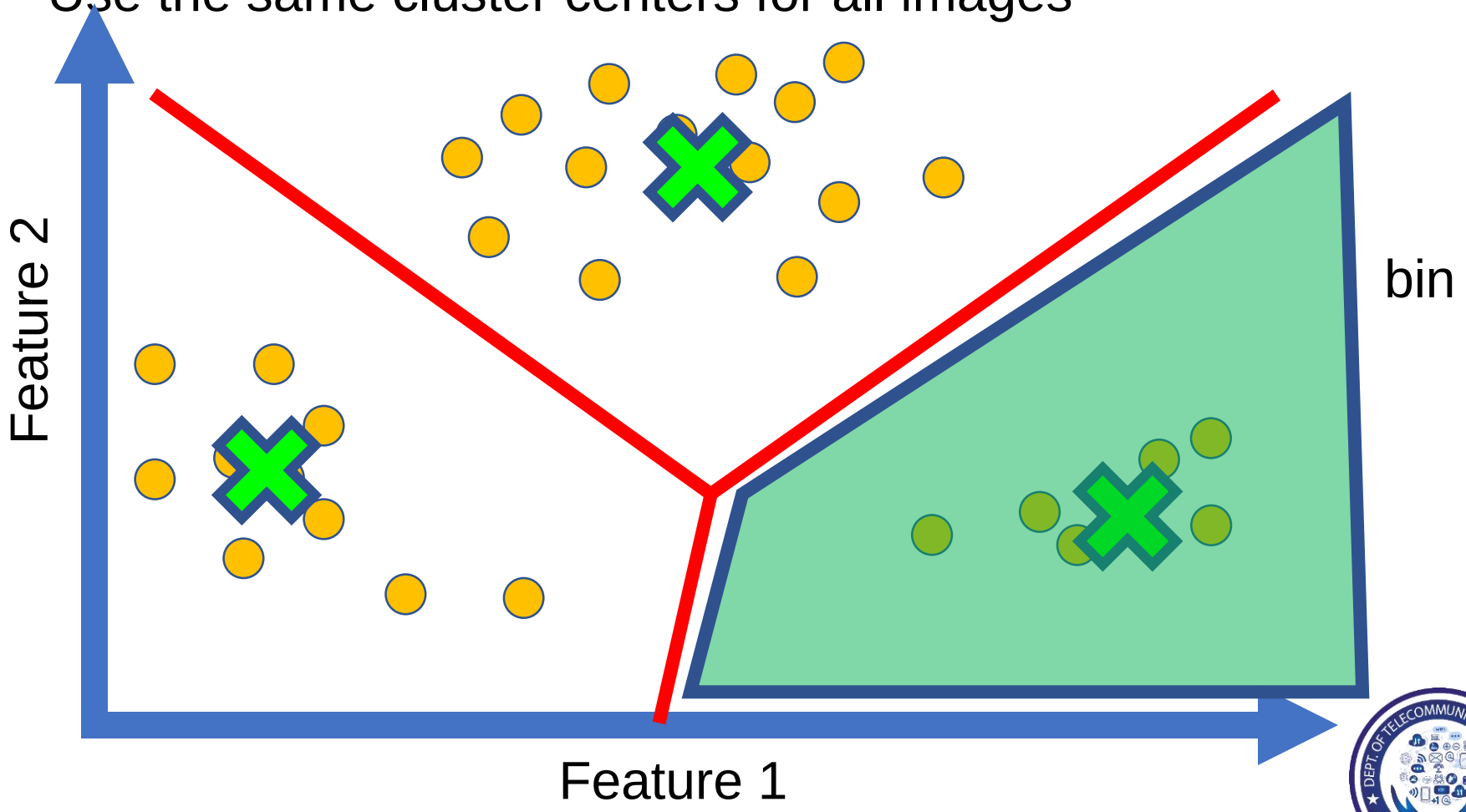
- Requires lots of data
- Loss of resolution to avoid empty bins

Marginal histogram

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

Modeling multi-dimensional data

- Clustering
- Use the same cluster centers for all images



Computing histogram distance

- 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

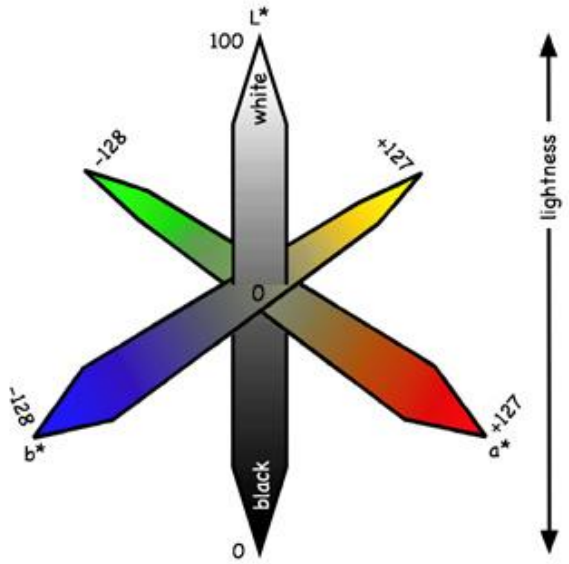
Many Bins
Need more data
Finer representation

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

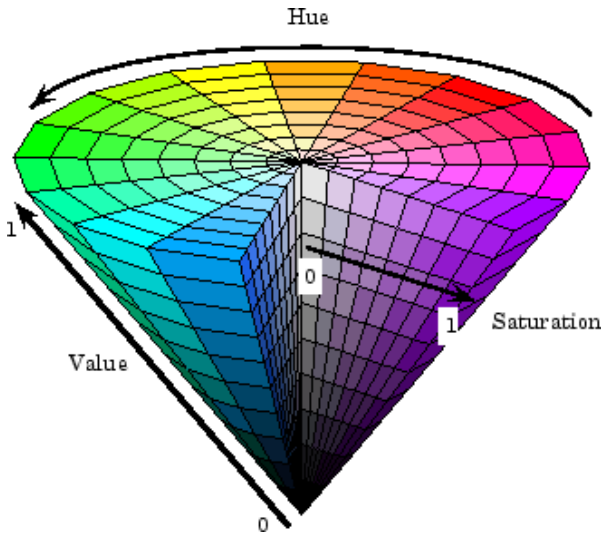
What kind of things do we compute histograms of?

- Color

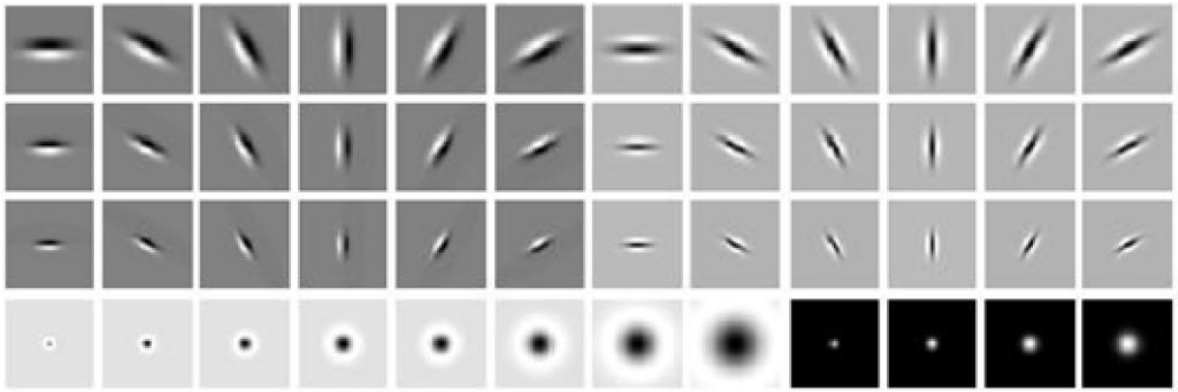
- Texture (filter banks



L*a*b* color space

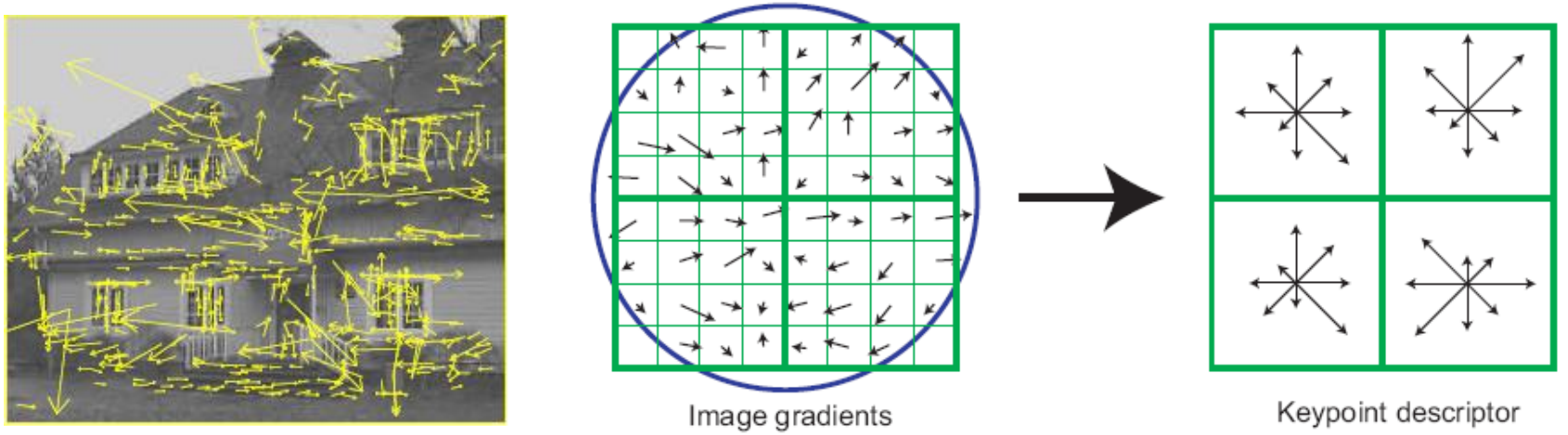


HSV color space



What kind of things do we compute histograms of?

- Histograms of descriptors



SIFT – [Lowe IJCV 2004]

- “Bag of visual words”

Analogy to documents

Of all the sensory impressions proceeding to the brain, the visual experiences are the dominant ones. Our perception of the world around us is based essentially on the messages that r… our eyes. For a long ti… retinal image wa… isual centers i… s a movie s… image … discove… know th… perceptio… more com… following the … ath to the various c… ortex, Hubel and Wiesel ha… demonstrate that the *message abo… image falling on the retina undergoes …-wise analysis in a system of nerve cell… stored in columns. In this system each c… has its specific function and is responsible… a specific detail in the pattern of the retinal image.*

sensory, brain, visual, perception, retinal, cerebral cortex, eye, cell, optical nerve, image Hubel, Wiesel

China is forecasting a trade surplus of $90bn (£51bn) to $100bn this year, a threefold increase on 2004's $32bn. The Commerce Ministry said the surplus would be created by a predicted 30% … $750bn, compared wi… $660bn. T… annoy th… China's… deliber… agrees … yuan is … governo… also need… demand so … country. China … e yuan against the dol… nd permitted it to trade within a narrow … but the US wants the yuan to be allowed … de freely. However, Beijing has made it cl… t it will take its time and tread carefully be… allowing the yuan to rise further in value.

China, trade, surplus, commerce, exports, imports, US, yuan, bank, domestic, foreign, increase, trade, value

Bag of *visual words*

- Image patches
- BoW histogram
- Codewords

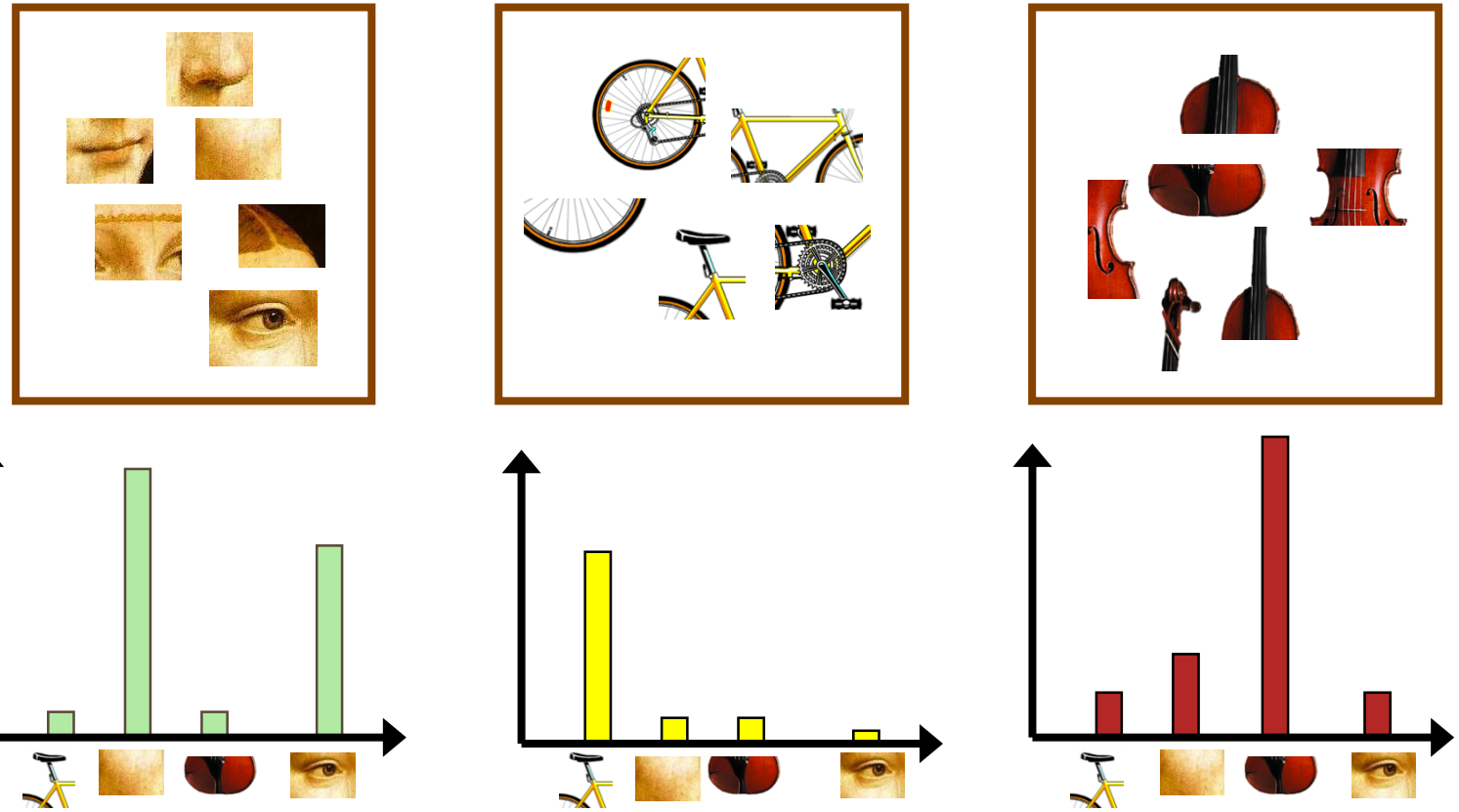


Image categorization with bag of words

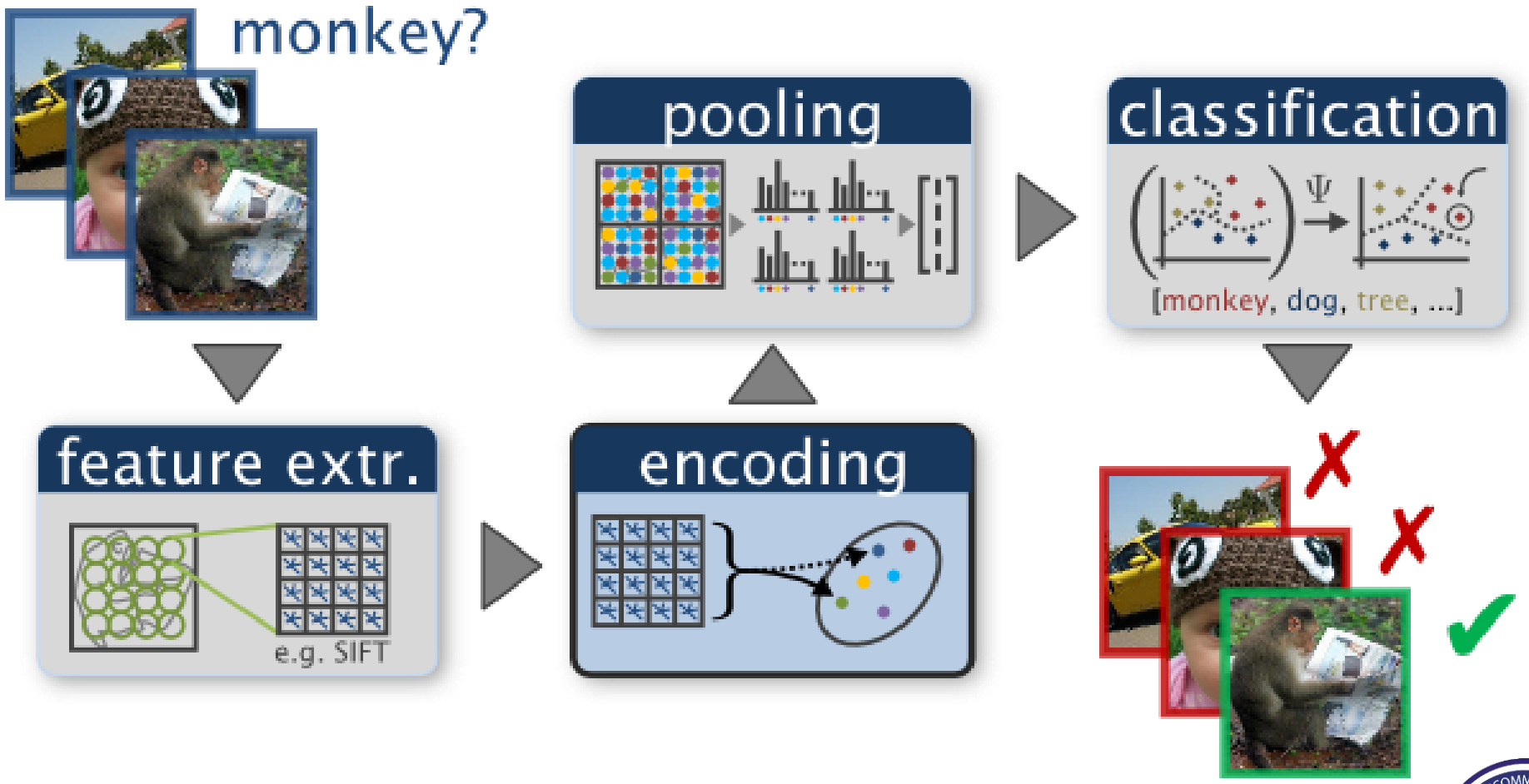
Training

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

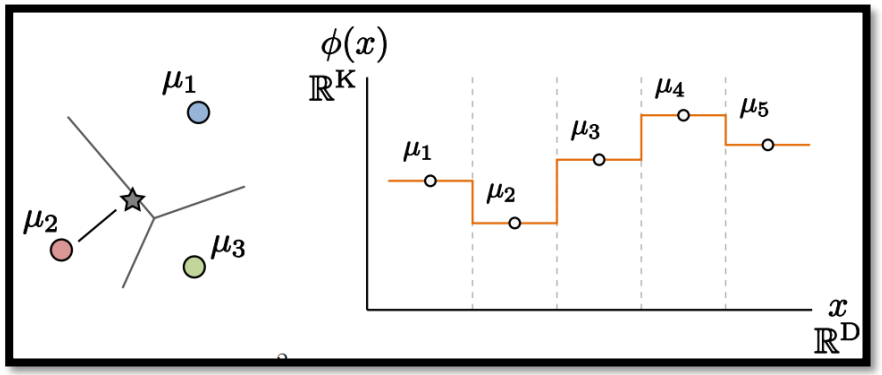
Bag of visual words image classification



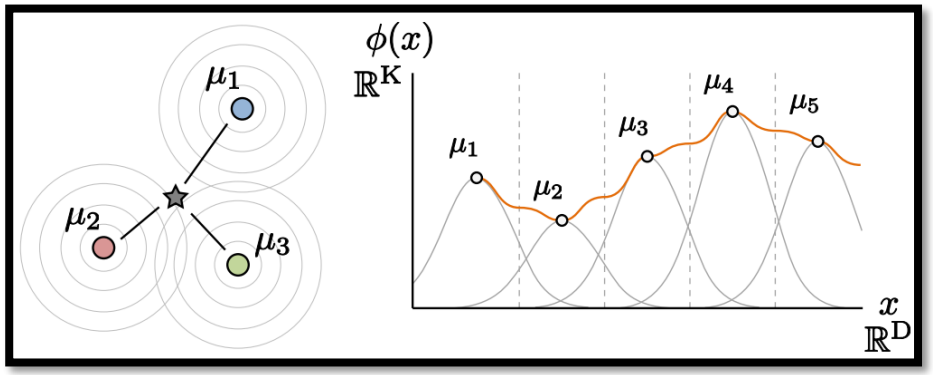
[Chatfieldet al. BMVC 2011]

Feature encoding

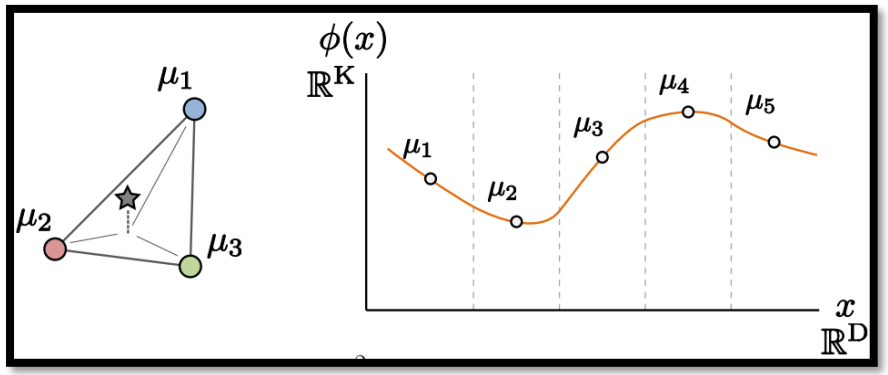
- Hard/soft assignment to clusters



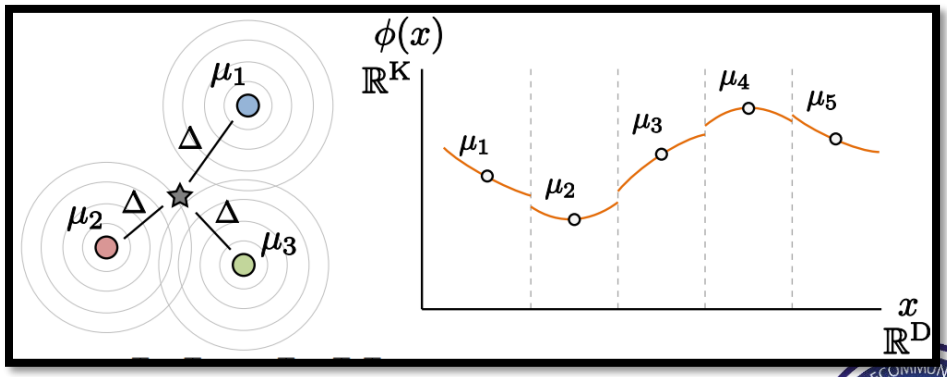
Histogram encoding



Kernel codebook encoding



Locality constrained encoding



Fisher encoding

[Chatfieldet al. BMVC 2011]

Fisher vector encoding

- Fit Gaussian Mixture Models

$$\Theta = (\mu_k, \Sigma_k, \pi_k : k = 1, \ldots, K)$$

- Posterior probability

$$q_{ik} = \frac{\exp\left[-\frac{1}{2}(\mathbf{x}_i - \mu_k)^T \Sigma_k^{-1} (\mathbf{x}_i - \mu_k)\right]}{\sum_{t=1}^{K} \exp\left[-\frac{1}{2}(\mathbf{x}_i - \mu_t)^T \Sigma_k^{-1} (\mathbf{x}_i - \mu_t)\right]}$$

- First and second order differences to cluster k

$$u_{jk} = \frac{1}{N\sqrt{\pi_k}} \sum_{i=1}^{N} q_{ik} \frac{x_{ji} - \mu_{jk}}{\sigma_{jk}},$$

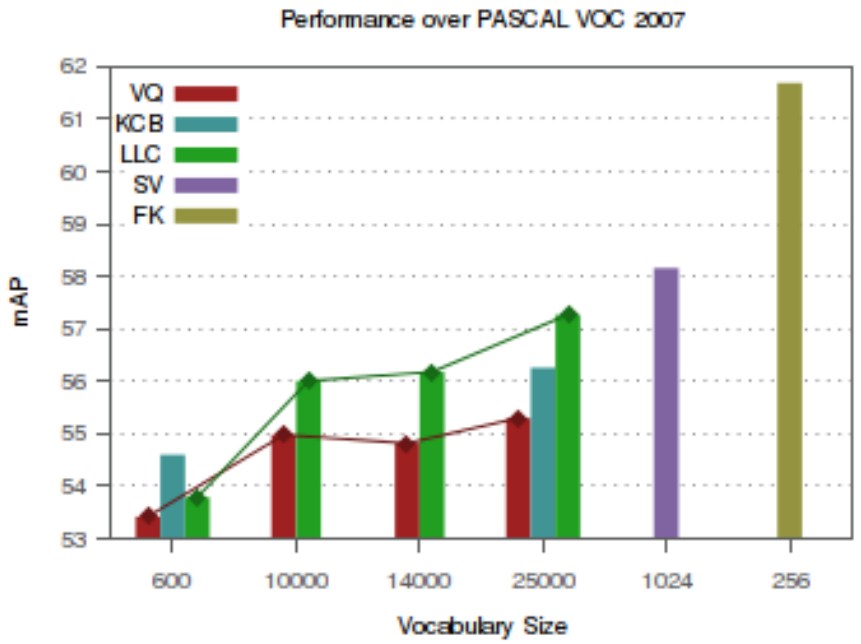
$$v_{jk} = \frac{1}{N\sqrt{2\pi_k}} \sum_{i=1}^{N} q_{ik} \left[\left(\frac{x_{ji} - \mu_{jk}}{\sigma_{jk}} \right)^2 - 1 \right]$$

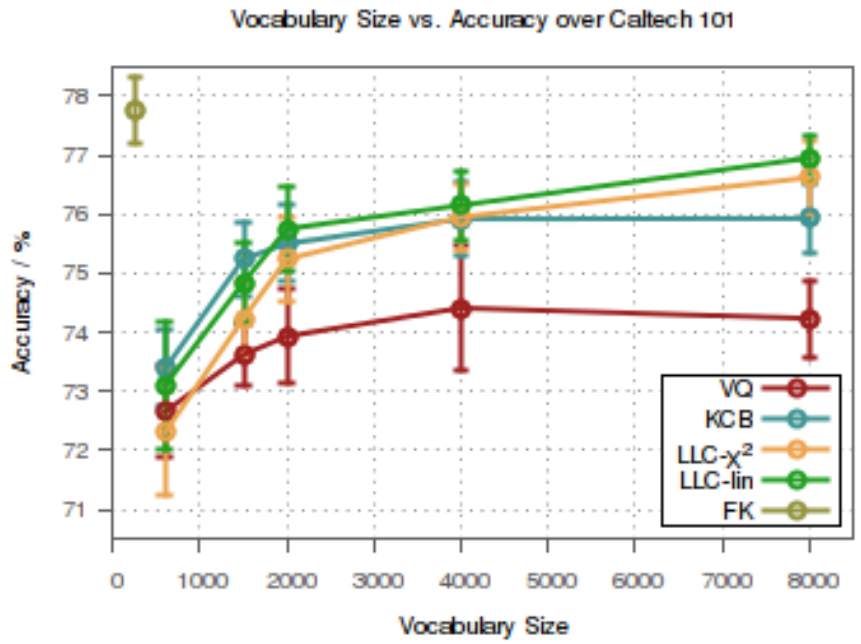
$$\Phi(I) = \begin{bmatrix} \vdots \\ \mathbf{u}_k \\ \vdots \\ \mathbf{v}_k \\ \vdots \end{bmatrix}$$

[Perronnin et al. ECCV 2010]

Performance comparisons

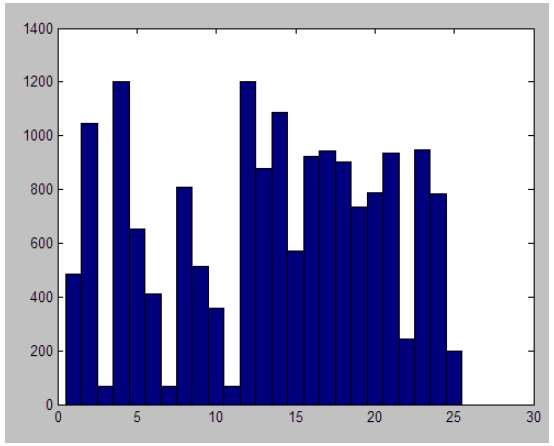
- Fisher vector encoding outperforms others
- Higher-order statistics helps

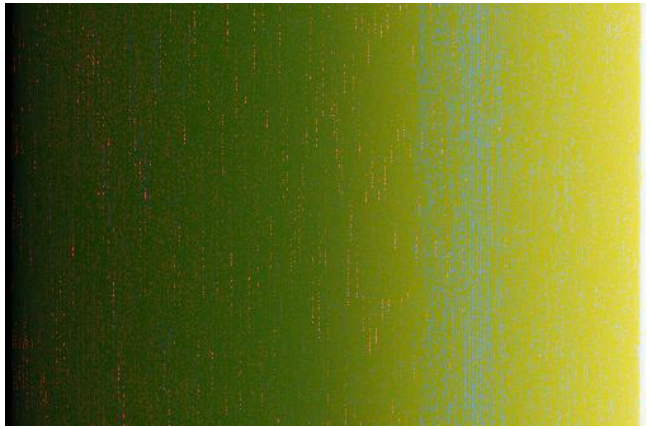




[Chatfieldet al. BMVC 2011]

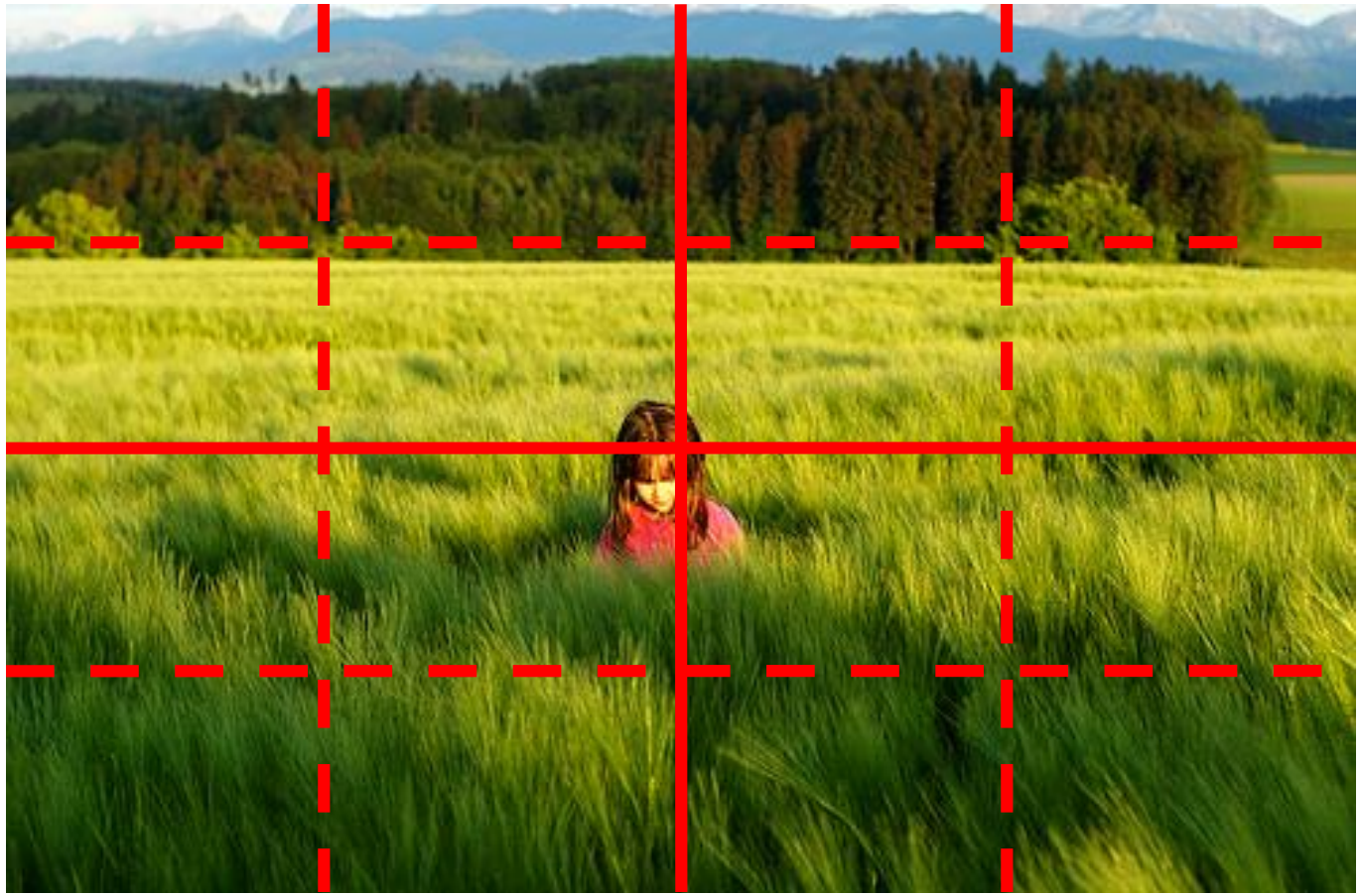
But what about spatial layout?





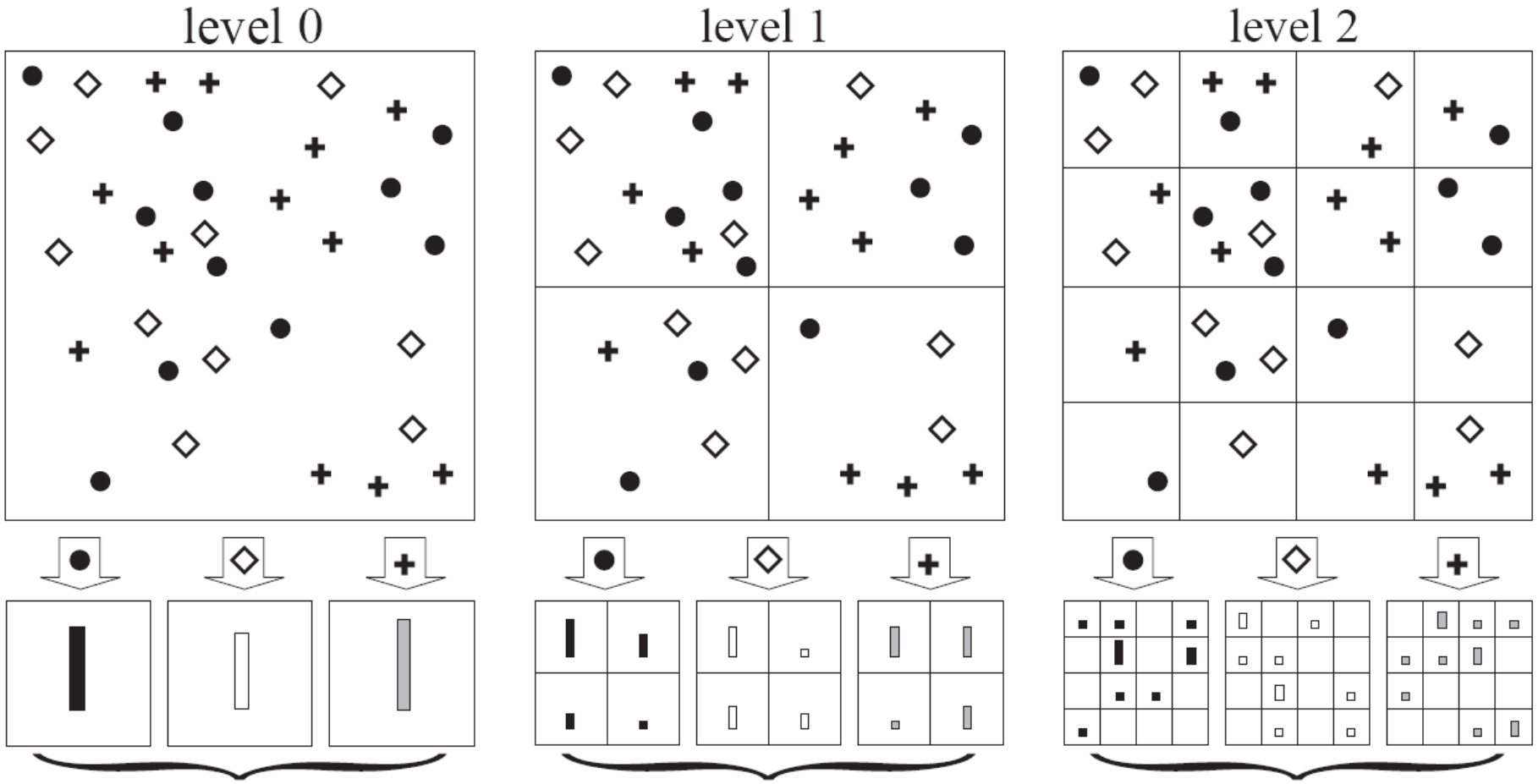
All of these images have the same color histogram

Spatial pyramid



Compute histogram in each spatial bin

Spatial pyramid

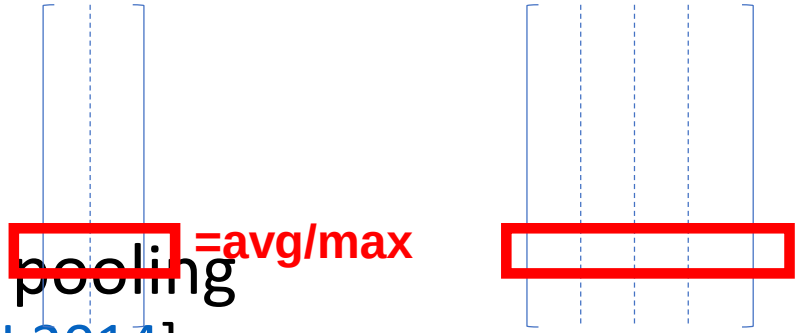


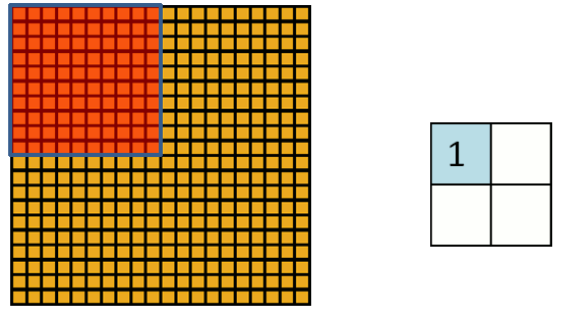
High number of features – PCA to reduce dimensionality

[Lazebnik et al. CVPR 2006]

Pooling

- Average/max pooling



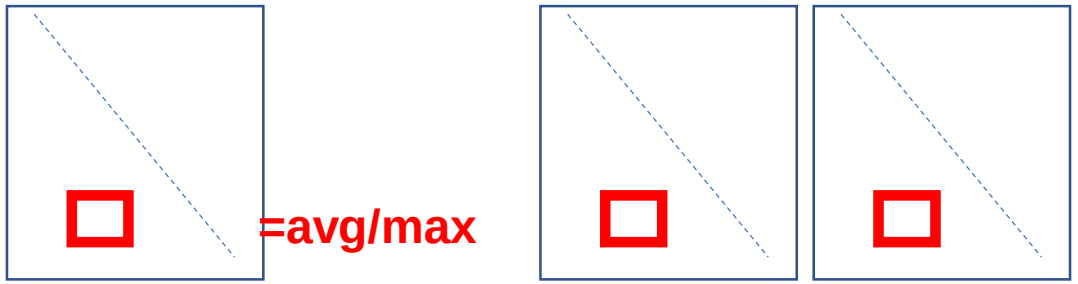


Convolved feature

Pooled feature

Source: Unsupervised Feature Learning and Deep Learning

- Second-order pooling [Joao et al. PAMI 2014]

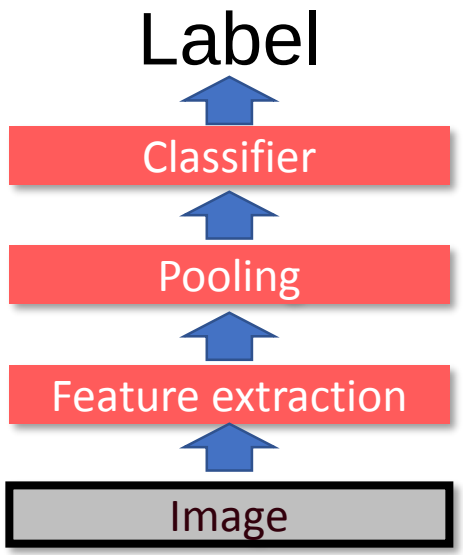


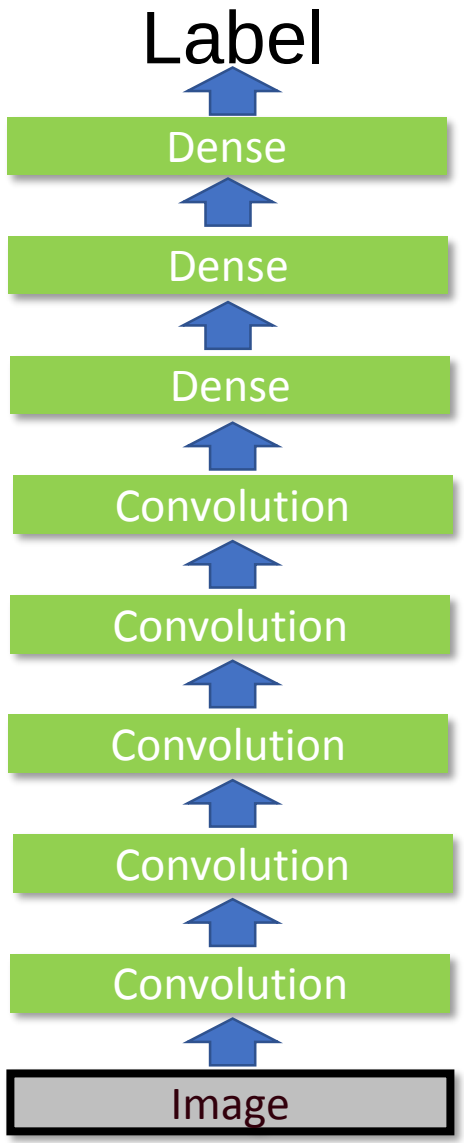
$$\mathbf{G}_{avg}(R_j) = \frac{1}{|F_{R_j}|} \sum_{i:(\mathbf{f}_i \in R_j)} \mathbf{x}_i \cdot \mathbf{x}_i^\top$$

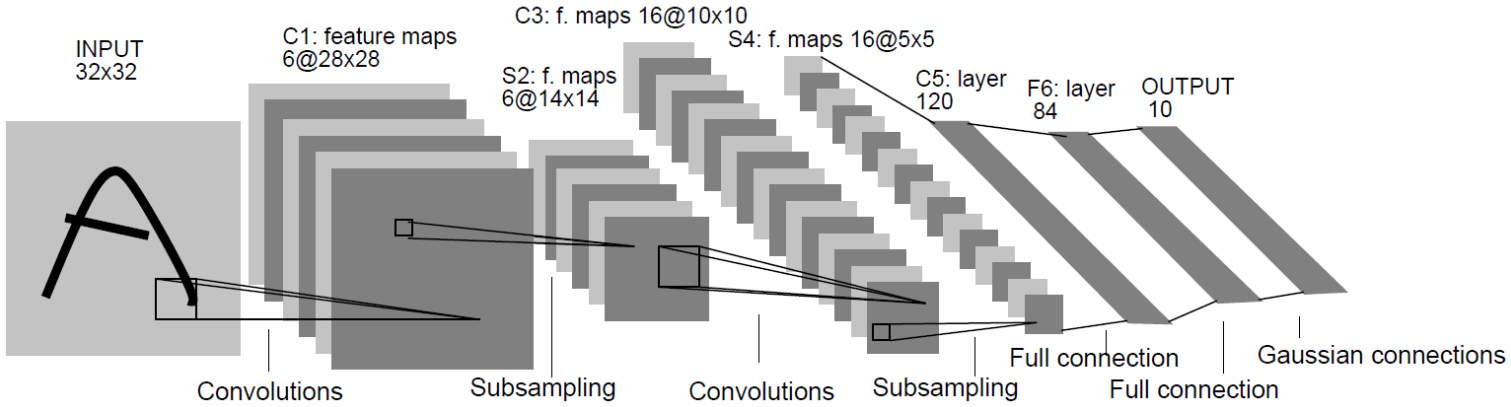
$$\mathbf{G}_{max}(R_j) = \max_{i:(\mathbf{f}_i \in R_j)} \mathbf{x}_i \cdot \mathbf{x}_i^\top$$

Shallow vs. deep learning

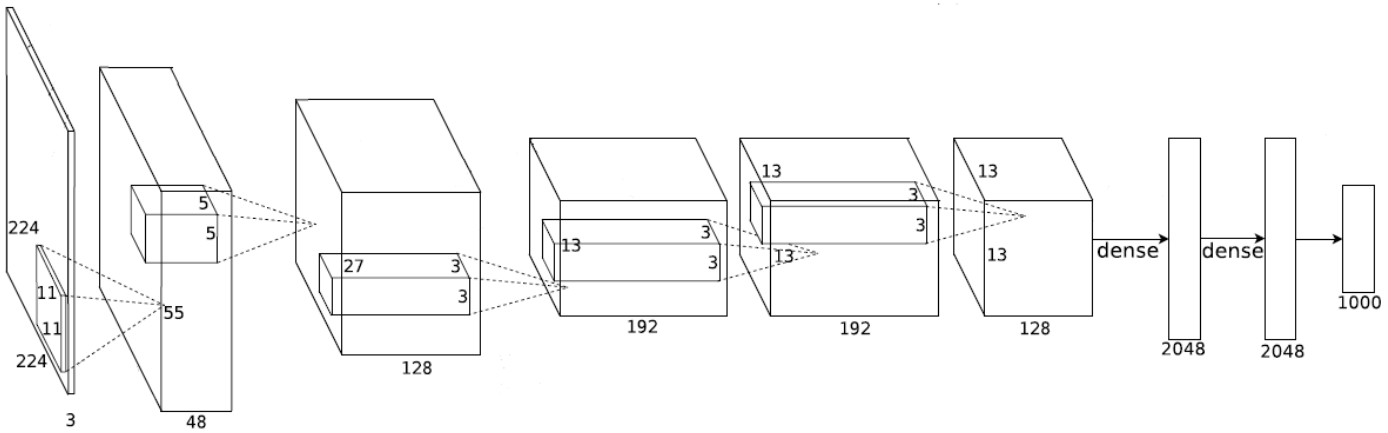
- Engineered vs. learned features





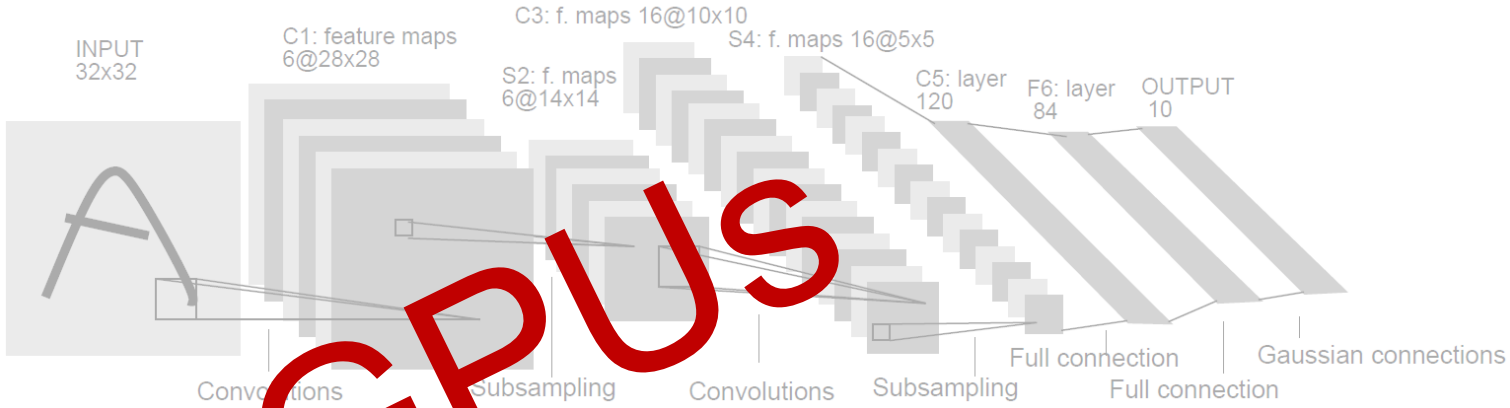


Gradient-Based Learning Applied to Document Recognition, LeCun, Bottou, Bengio and Haffner, Proc. of the IEEE, **1998**



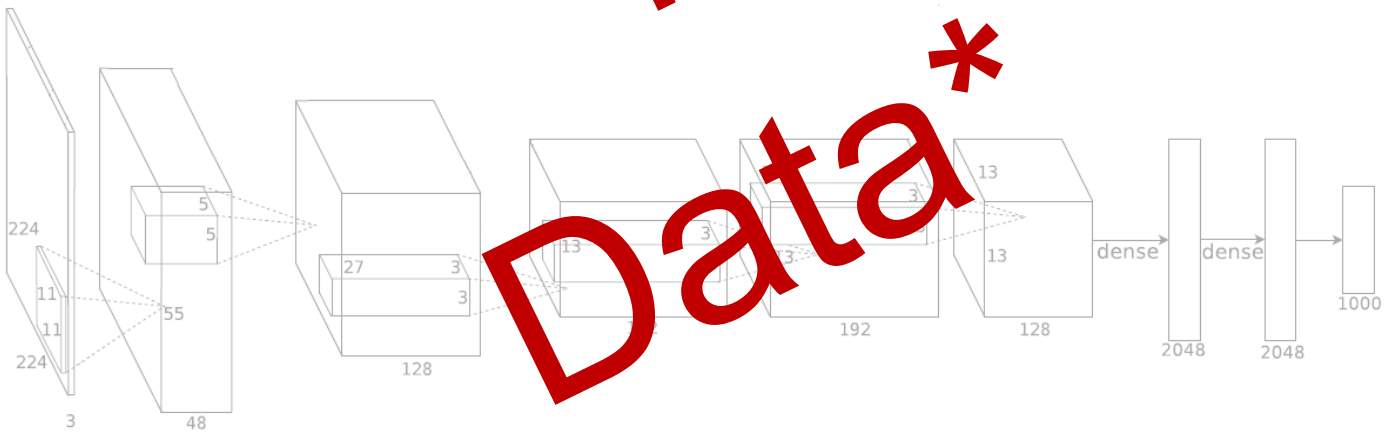
Imagenet Classification with Deep Convolutional Neural Networks, Krizhevsky, Sutskever, and Hinton, NIPS **2012**

Slide Credit: L. Zitnick



Gradient-Based Learning Applied to Document Recognition, LeCun, Bottou, Bengio and Haffner, Proc. of the IEEE, **1998**

GPUs + Data*

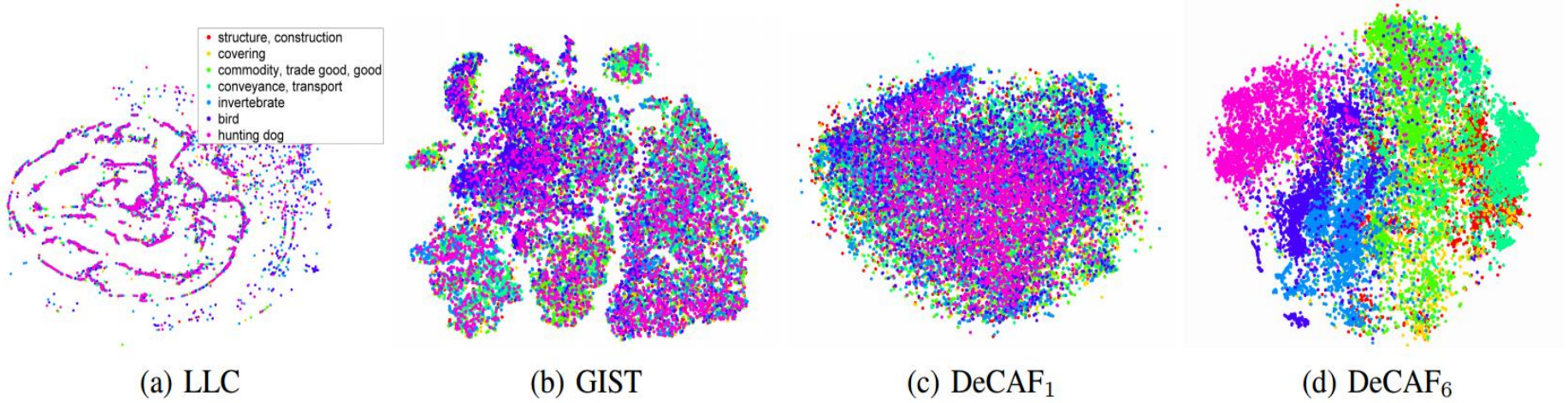


Imagenet Class Networks, Krizl

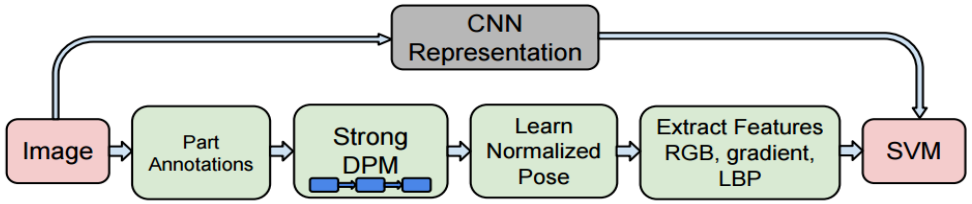
* Rectified activations and dropout

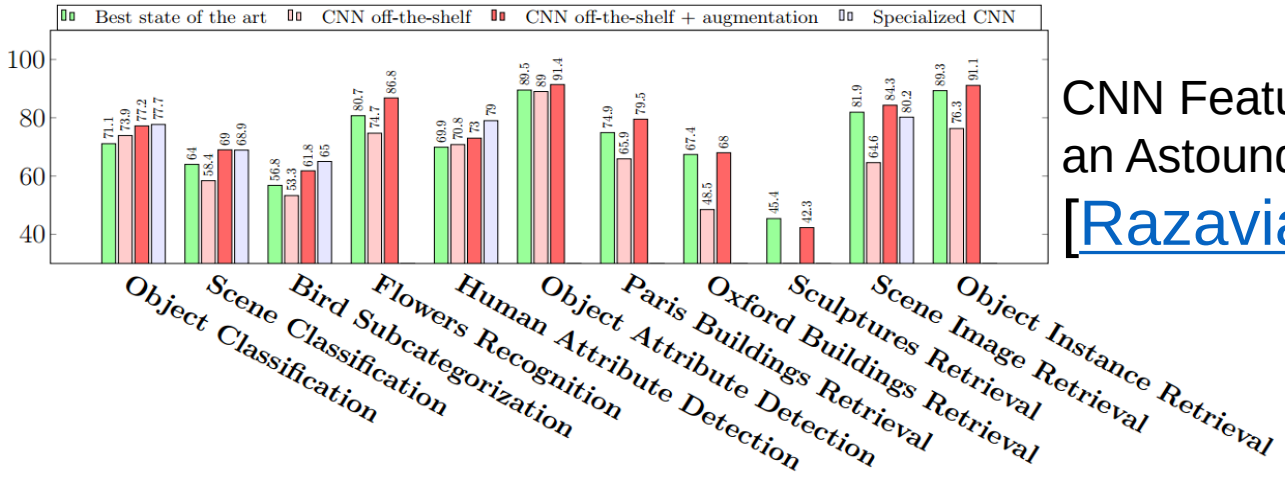
Slide Credit: L. Zitnick

Convolutional activation features



[Donahue et al. ICML 2013]

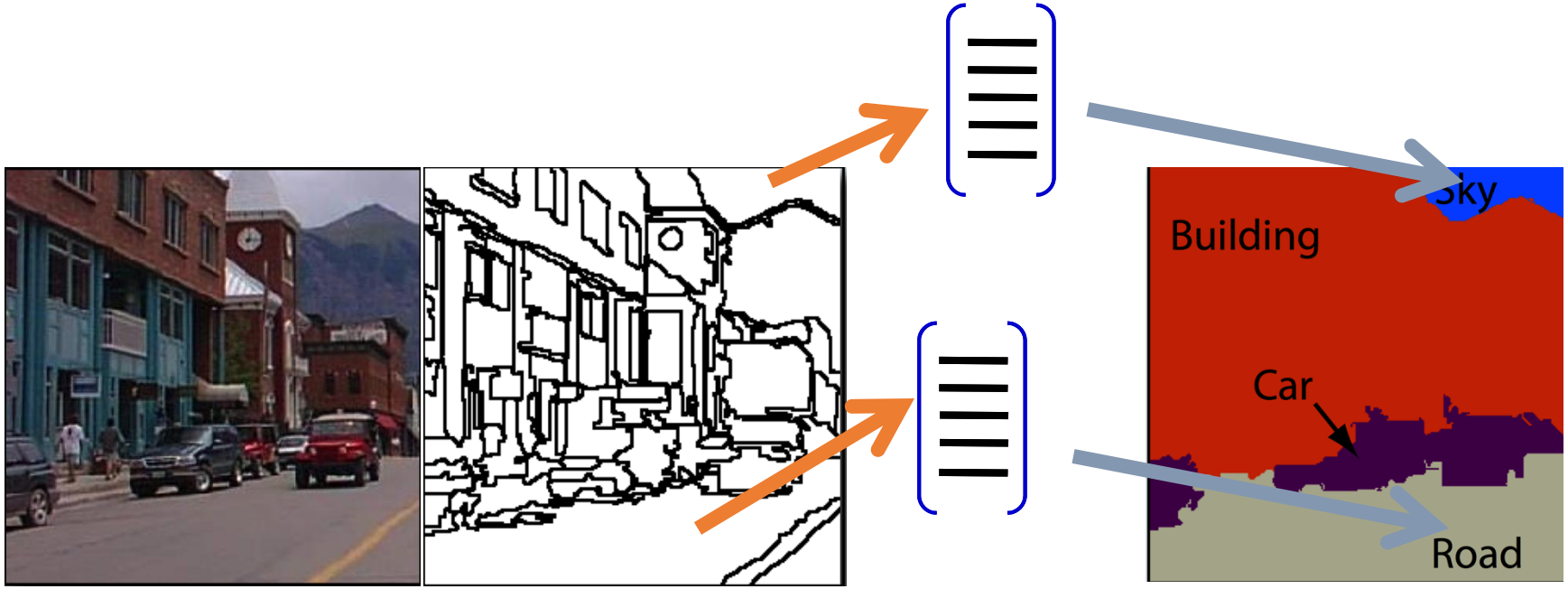




CNN Features off-the-shelf:
an Astounding Baseline for Recognition
[Razavian et al. 2014]

Region representation

- Segment the image into superpixels
- Use features to represent each image segment

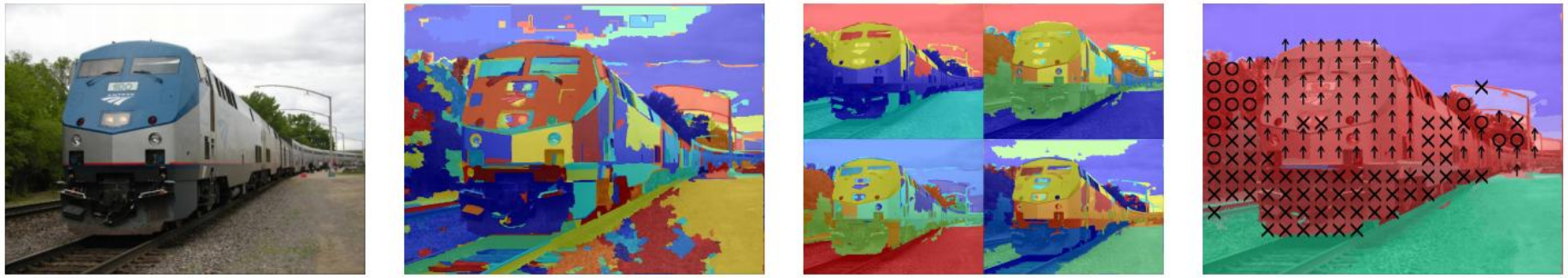


Region representation

- Color, texture, BoW
  - Only computed within the local region
- Shape of regions
- Position in the image

Working with regions

- Spatial support is important – multiple segmentation



(a) Input (b) Superpixels (c) Multiple Hypotheses (d) Geometric Labels

Geometric context [Hoiem et al. ICCV 2005]

- Spatial consistency – MRF smoothing

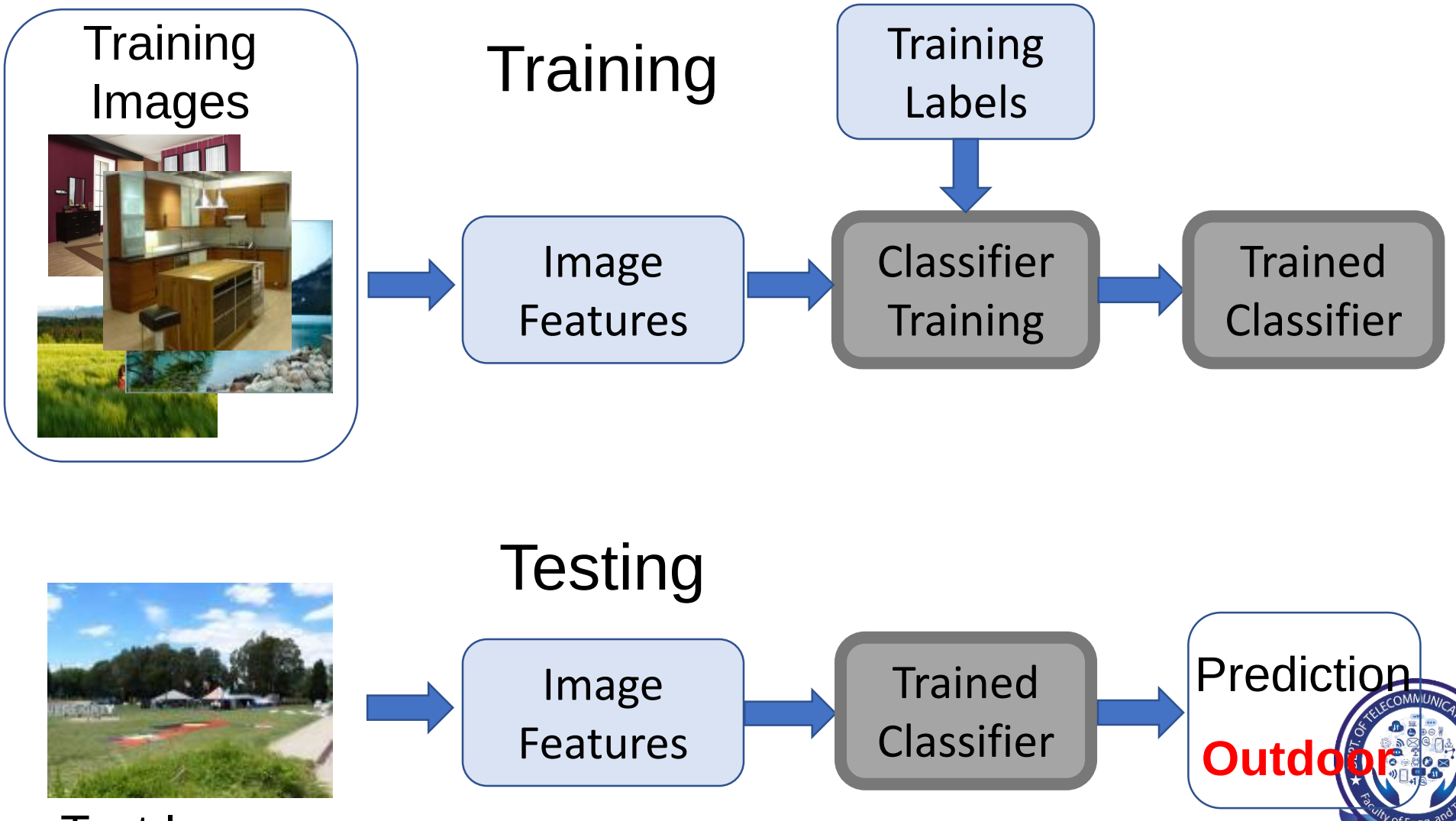
Things to remember

- Visual categorization help transfer knowledge

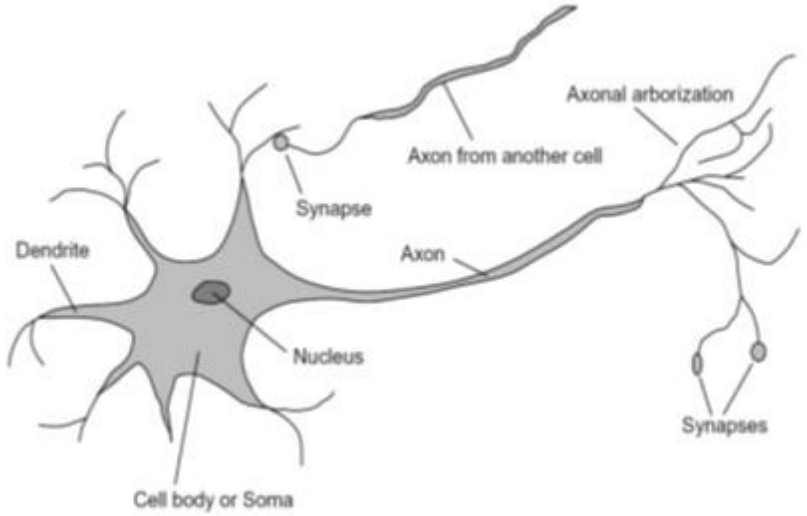
- Image features
  - Coverage, concision, directness
  - Color, gradients, textures, motion, descriptors
  - Histogram, feature encoding, and pooling
  - CNN as features

- Image/region categorization

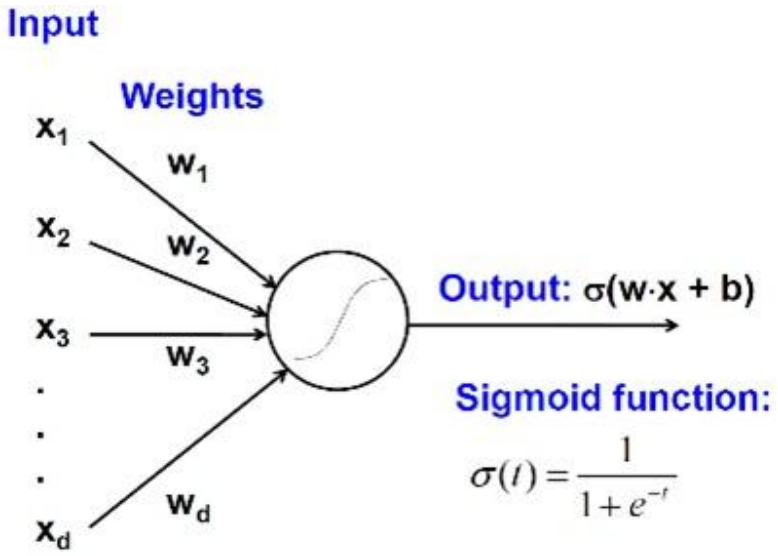
Next lecture - Classifiers



Biological neuron and Perceptrons



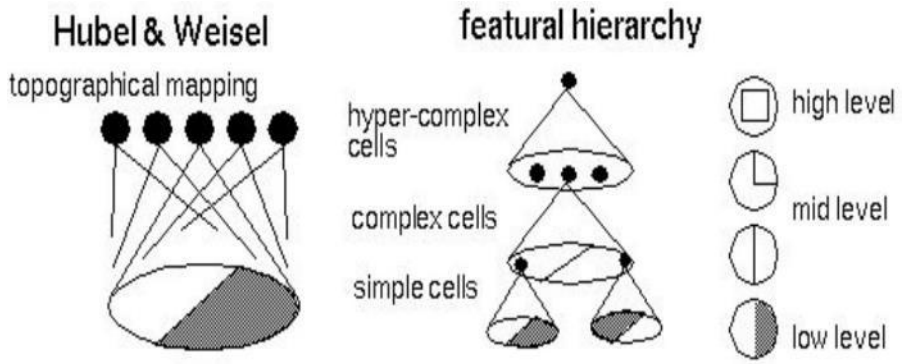
A biological neuron



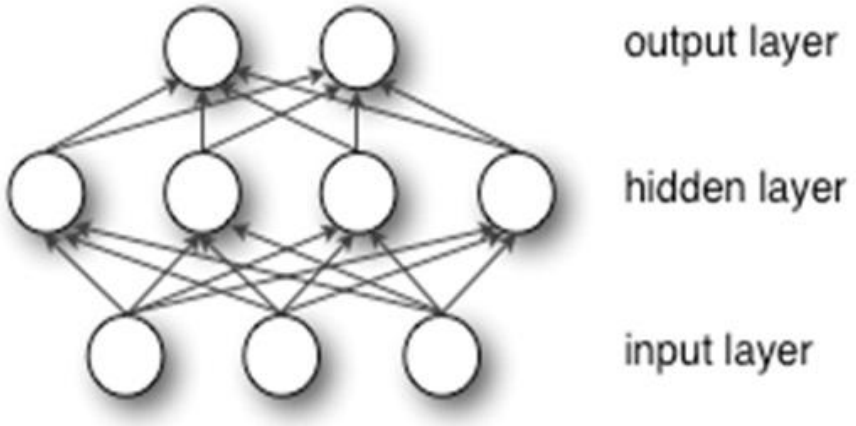
An artificial neuron (Perceptron)
- a linear classifier

owaja

Hubel/Wiesel Architecture and Multi-layer Neural Network



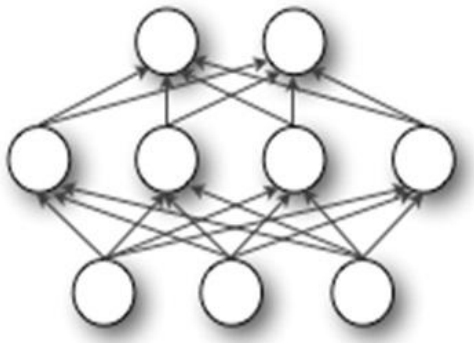
Hubel and Weisel's architecture



Multi-layer Neural Network
- A *non-linear* classifier

Multi-layer Neural Network

- A non-linear classifier
- **Training:** find network weights **w** to minimize the error between true training labels y_i and estimated labels $f_{\boldsymbol{w}}(\boldsymbol{x_i})$
- Minimization can be done $E(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - f_{\mathbf{w}}(\mathbf{x}_i)\right)^2$ ided f is differentiable
- This training method is called **back-propagation**

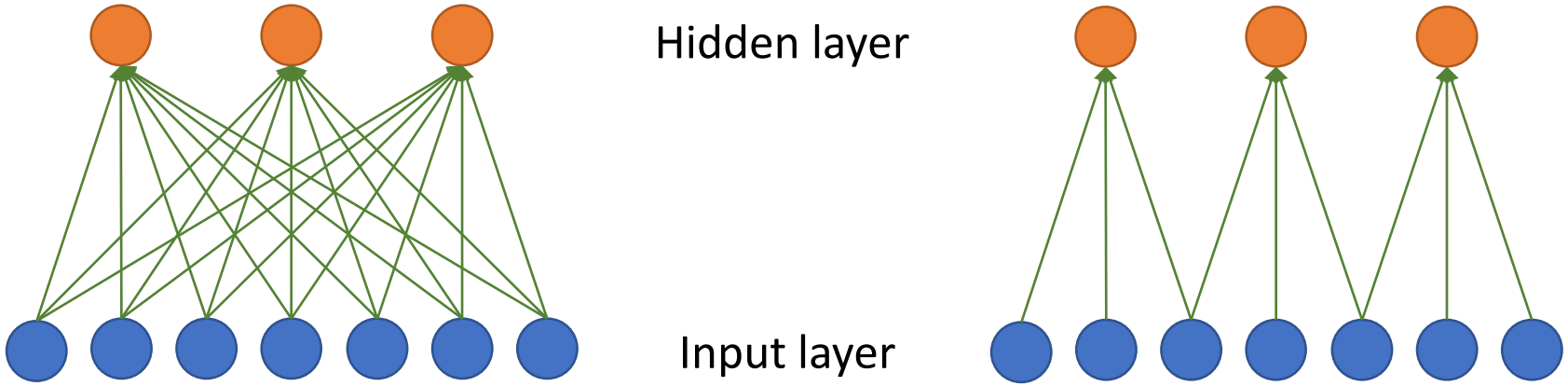


Convolutional Neural Networks

- Also known as CNN, ConvNet, DCN

- CNN = a multi-layer neural network with
  1. Local connectivity
  2. Weight sharing

CNN: Local Connectivity

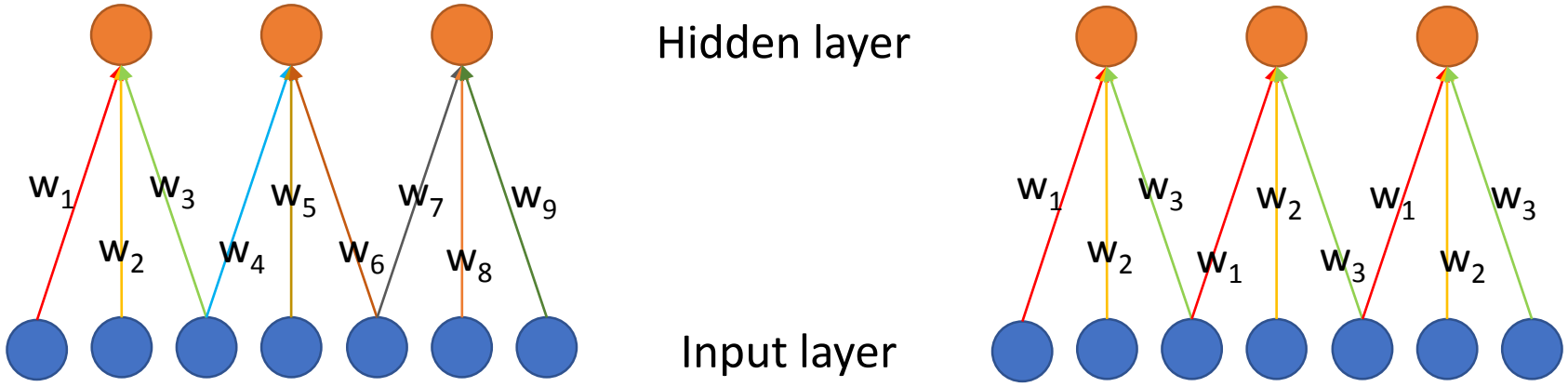


Global connectivity

Local connectivity

- # input units (neurons): 7
- # hidden units: 3
- Number of parameters
  - Global connectivity: 3 x 7 = 21
  - Local connectivity: 3 x 3 = 9

CNN: Weight Sharing

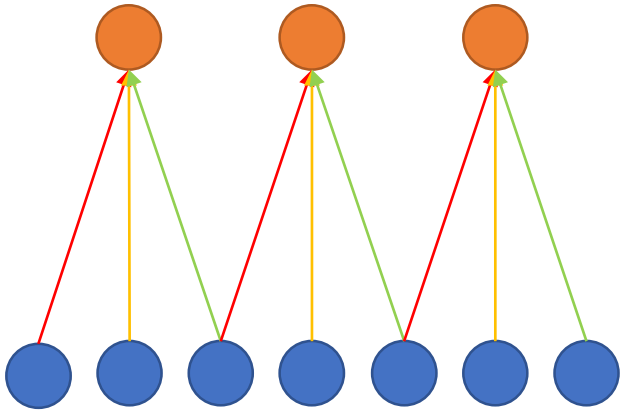


Without weight sharing

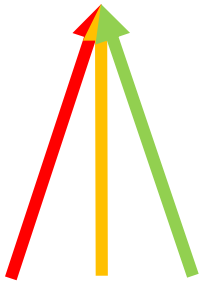
With weight sharing

- # input units (neurons): 7
- # hidden units: 3
- Number of parameters
  - Without weight sharing: 3 x 3 = 9
  - With weight sharing : 3 x 1 = 3

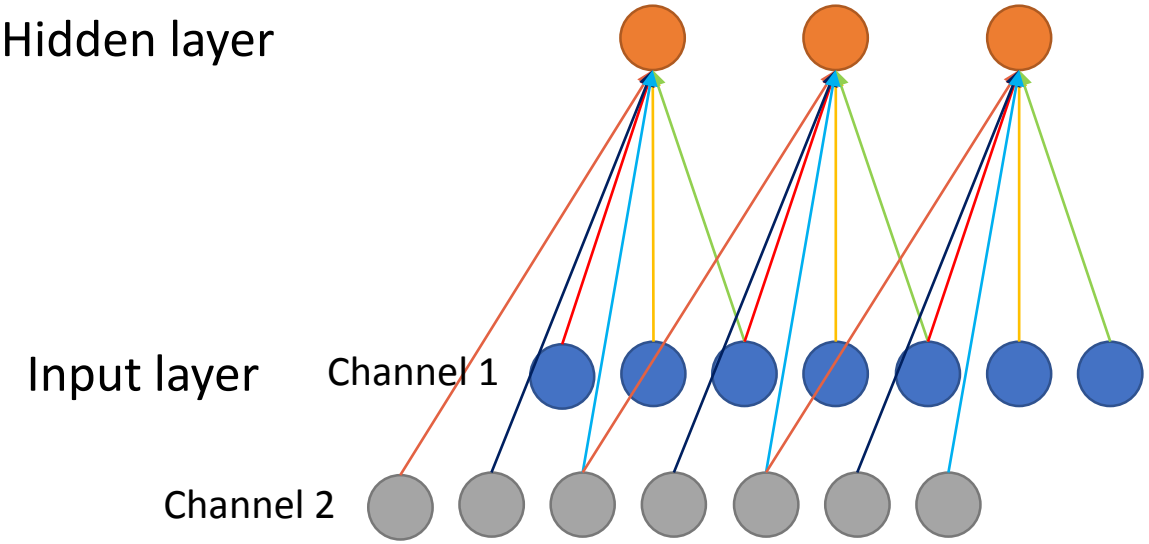
CNN with multiple input channels



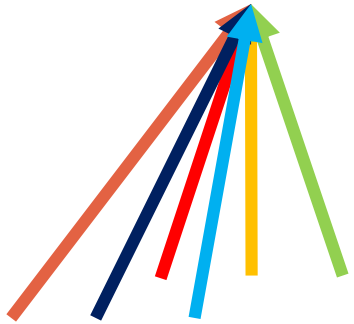
Single input channel



Filter weights

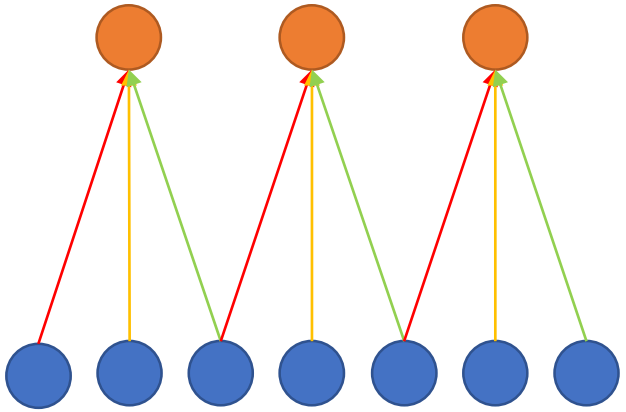


Multiple input channels



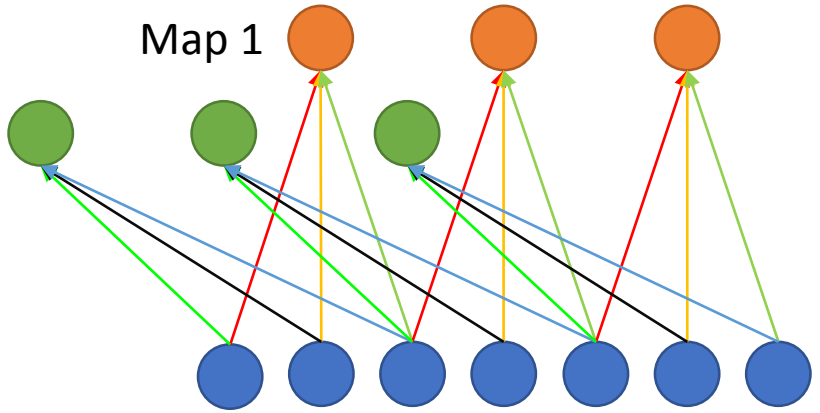
Filter weights

CNN with multiple output maps



Hidden layer

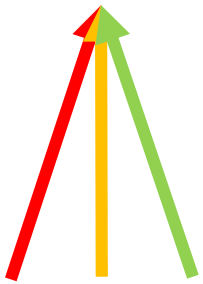
Input layer



Map 1

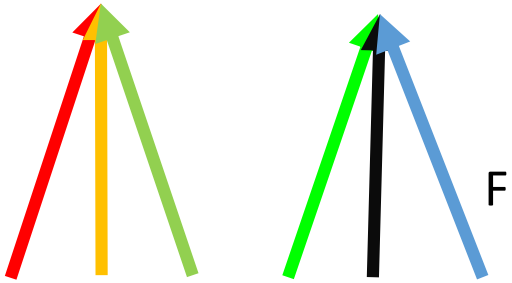
Map 2

Single output map



Filter weights

Multiple output maps



Filter 1

Filter 2

Filter weights

Putting them together

- Local connectivity
- Weight sharing
- Handling multiple input channels
- Handling multiple output maps

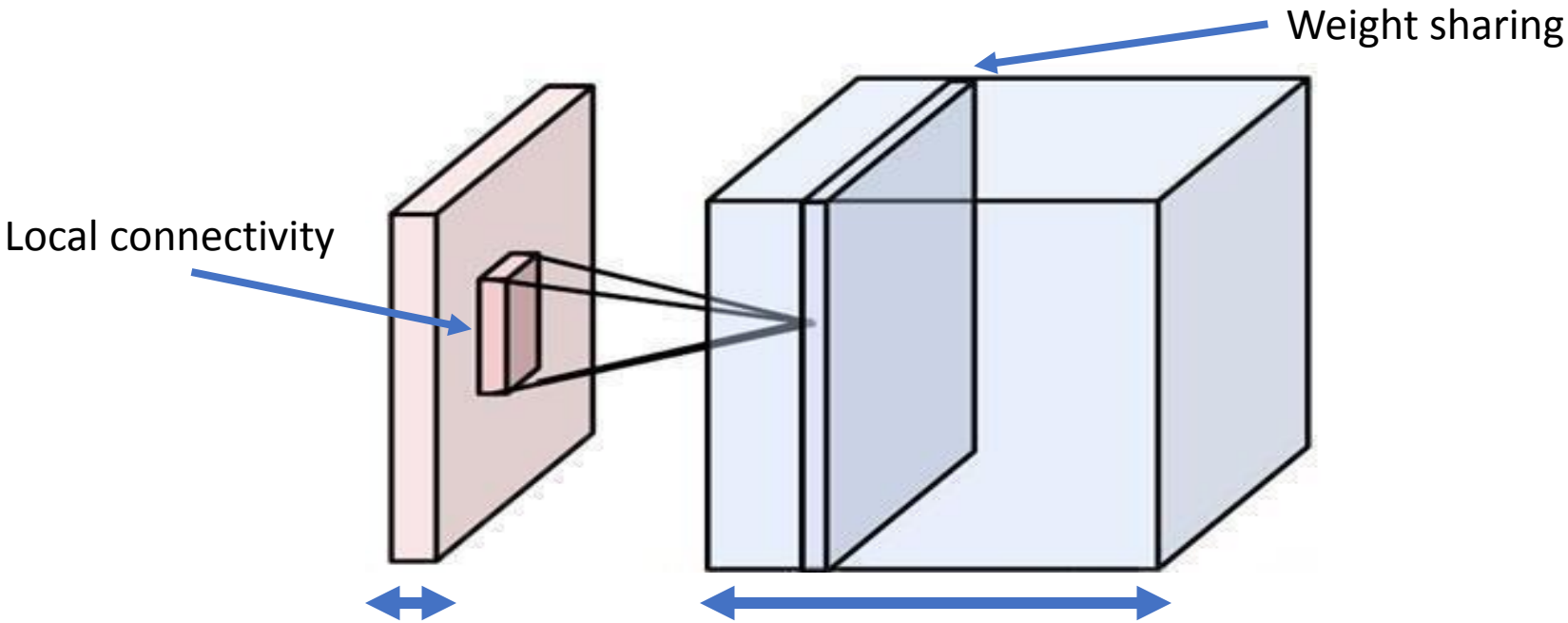
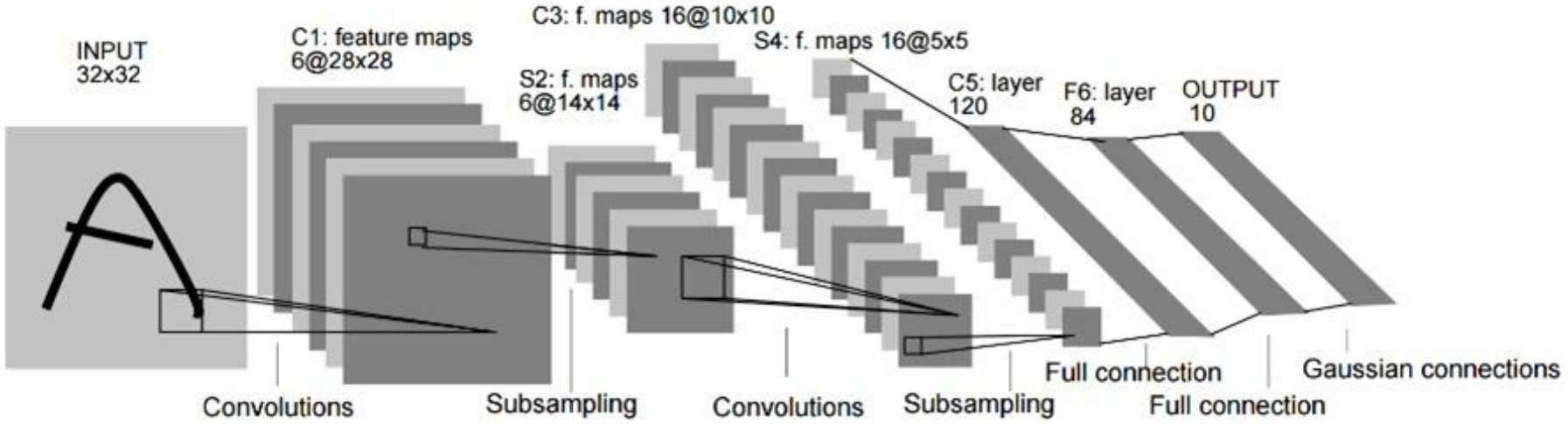


Image credit: A. Karpathy

LeNet [LeCun et al. 1998]

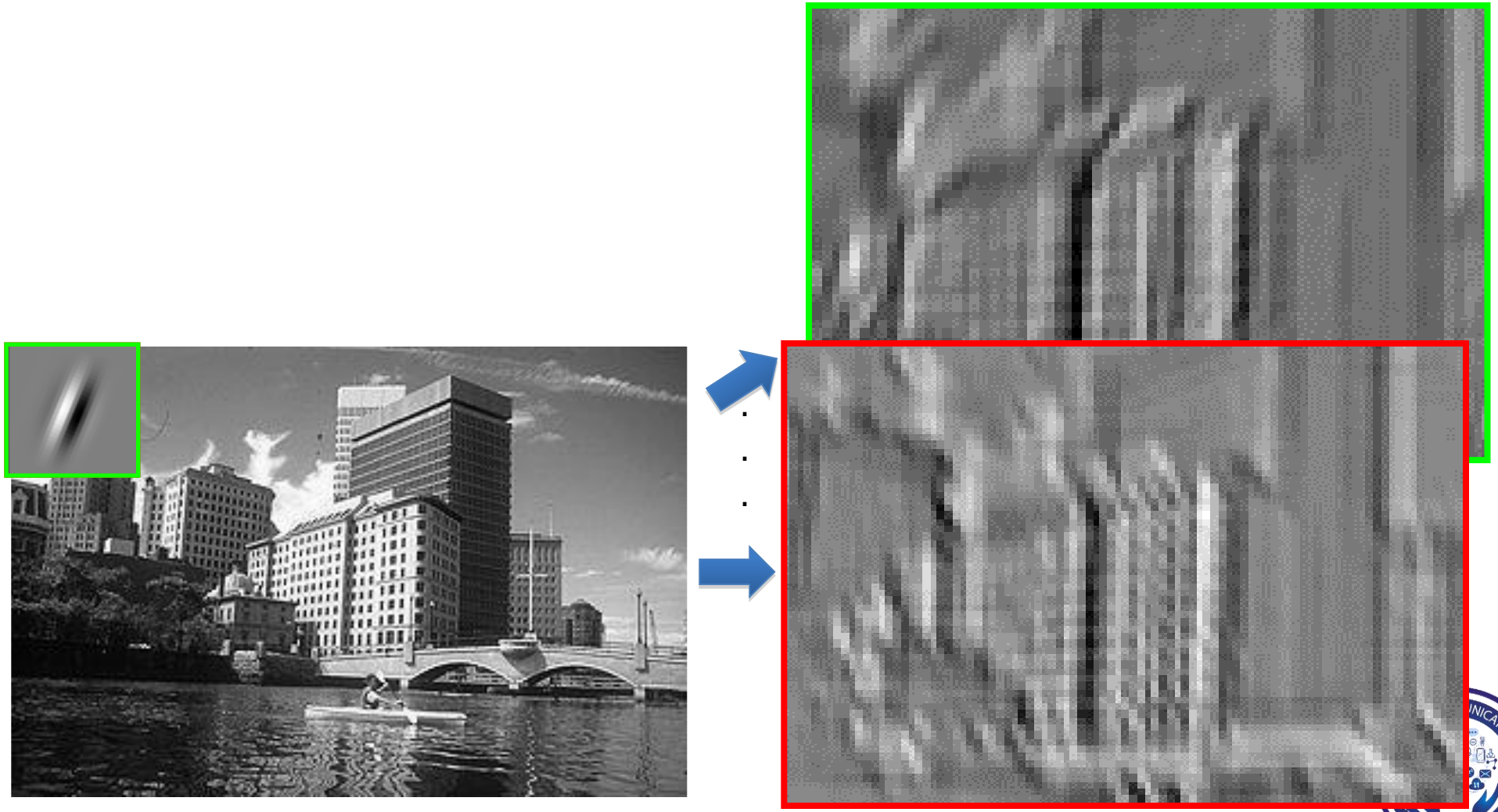


LeNet-1 from 1993

Gradient-based learning applied to document recognition [LeCun, Bottou, Bengio, Haffner 1998]

What is a Convolution?

- Weighted moving sum

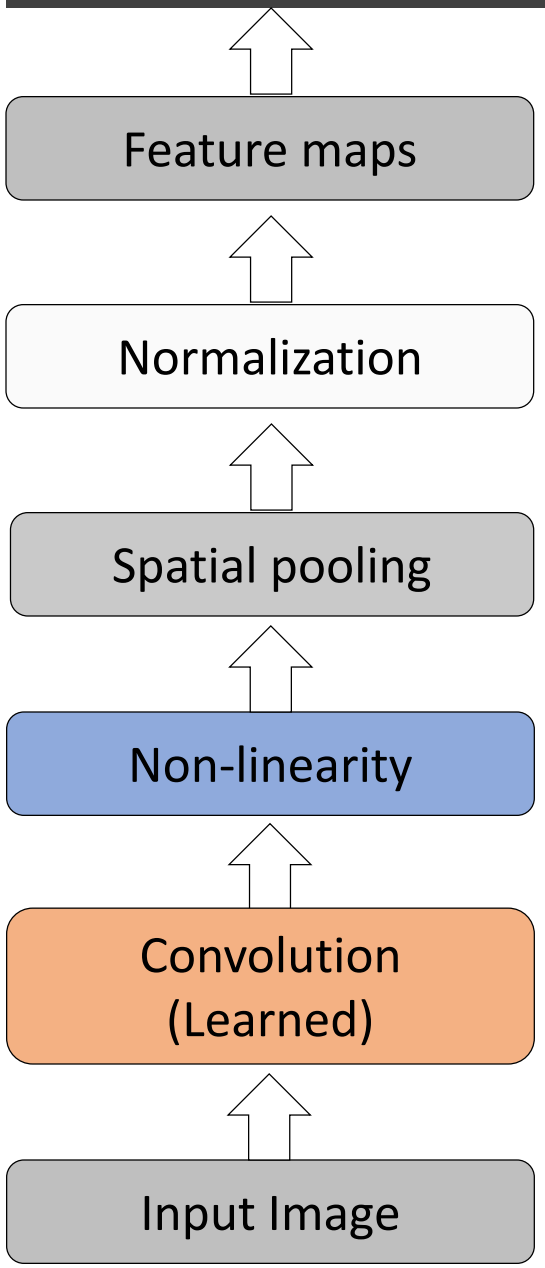


Input

Feature Activation Map

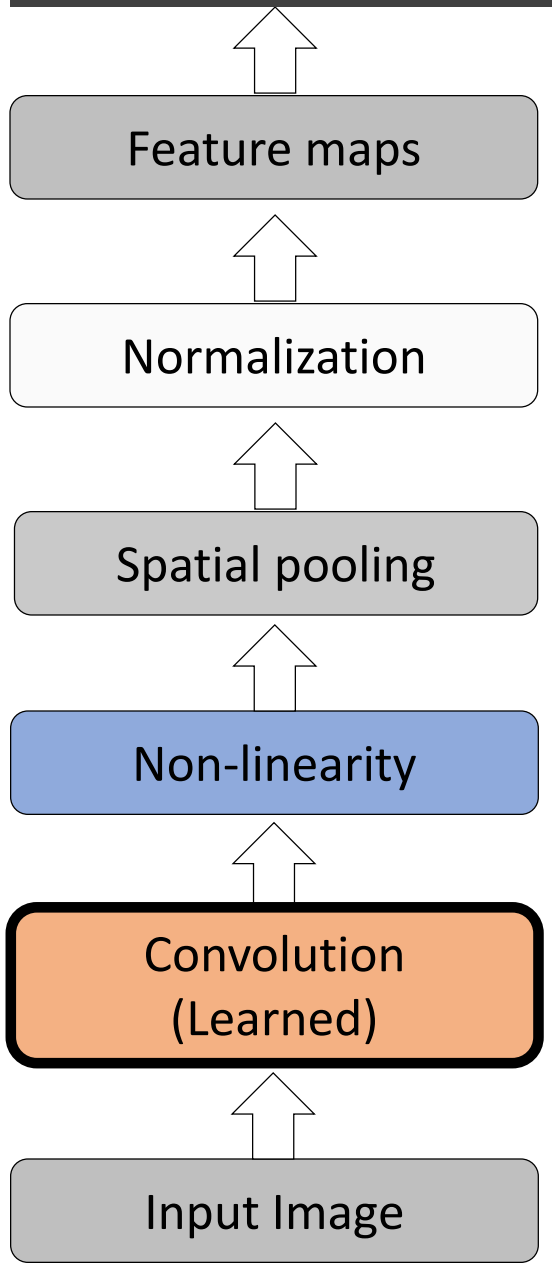
slide credit: S. Lazebnik

Convolutional Neural Networks

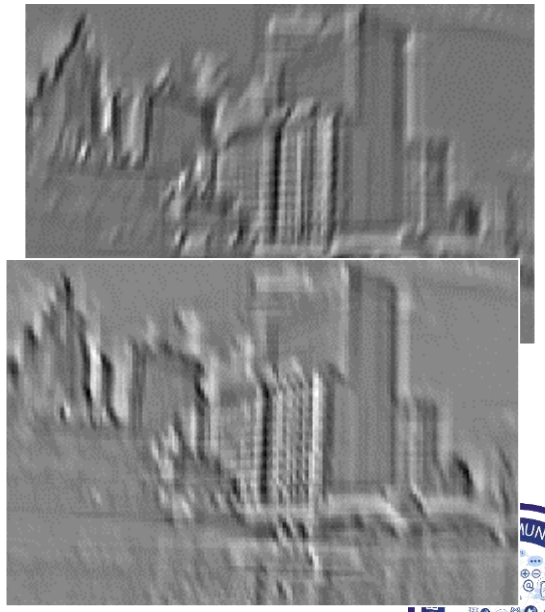


slide credit: S. Lazebnik

Convolutional Neural Networks



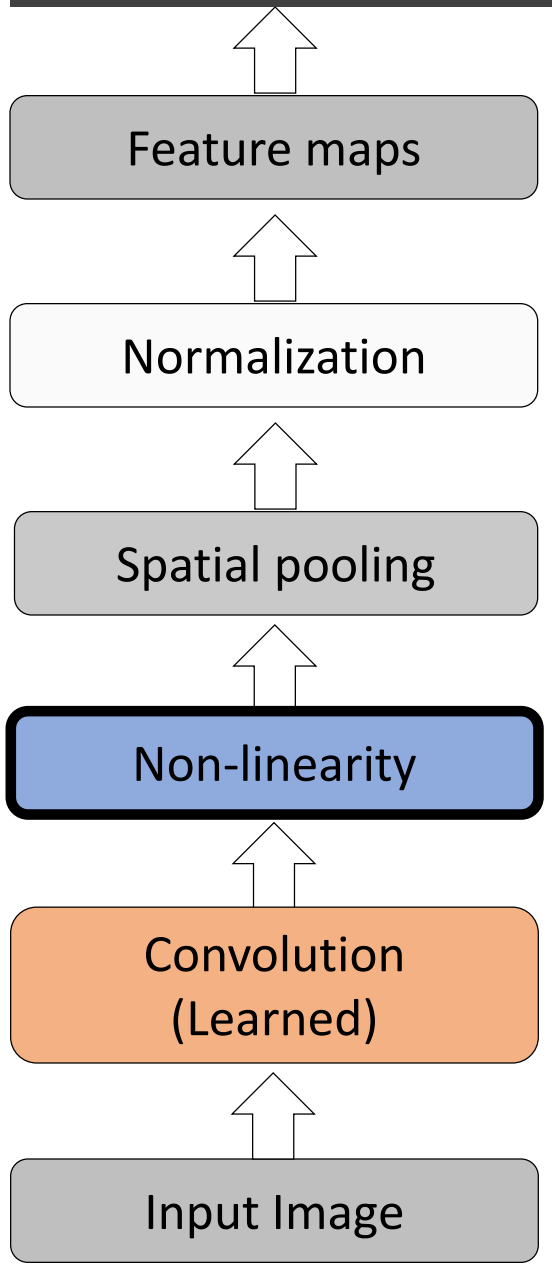
Input



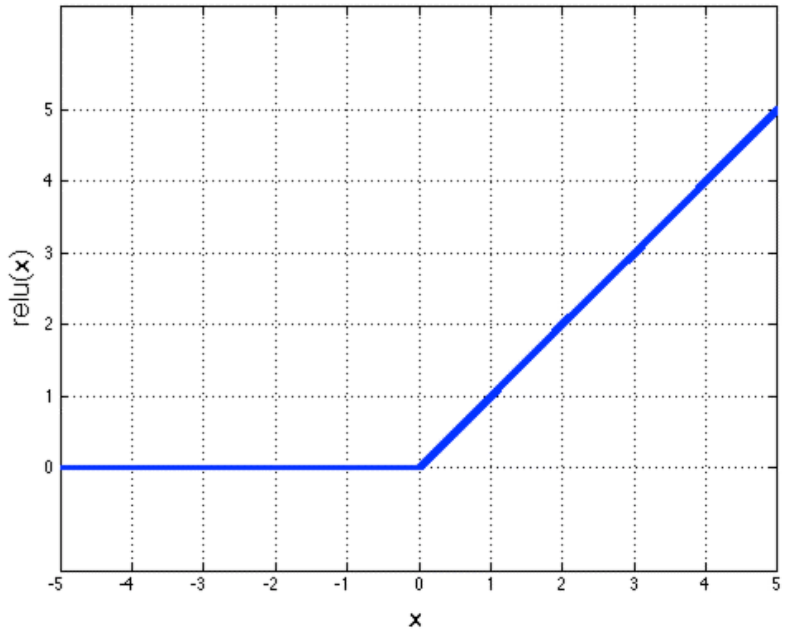
Feature Map

slide credit: S. Lazebnik

Convolutional Neural Networks

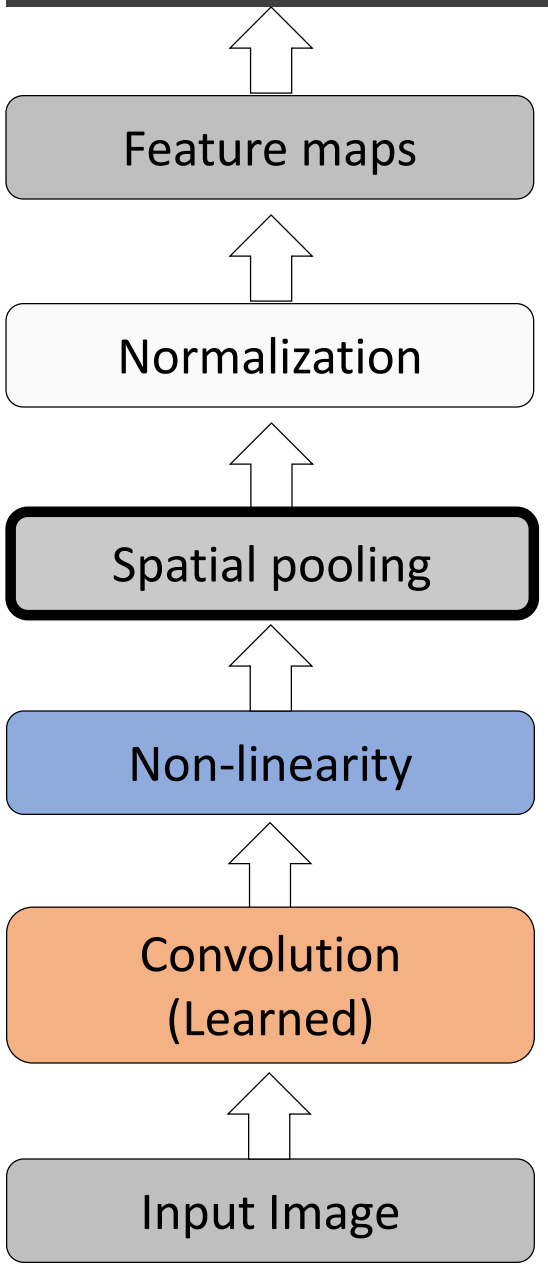


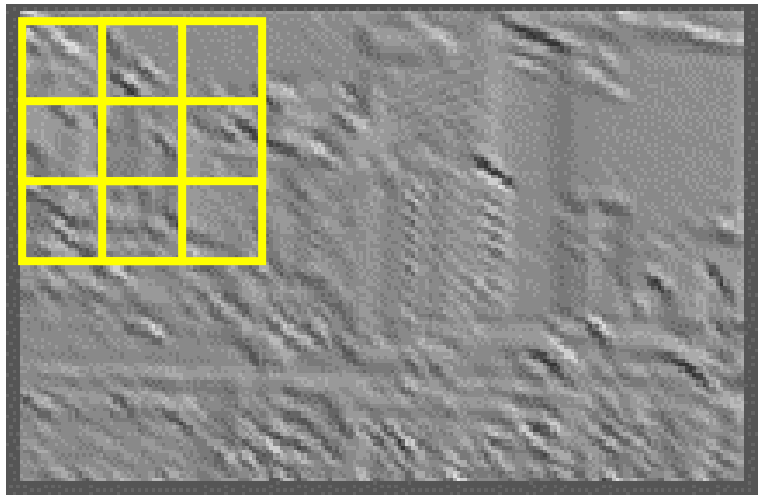
Rectified Linear Unit (ReLU)



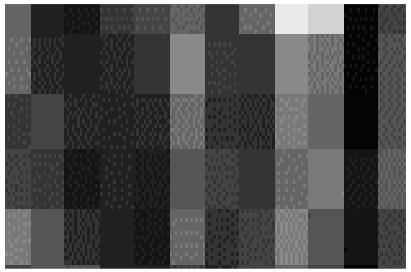
slide credit: S. Lazebnik

Convolutional Neural Networks





Max pooling

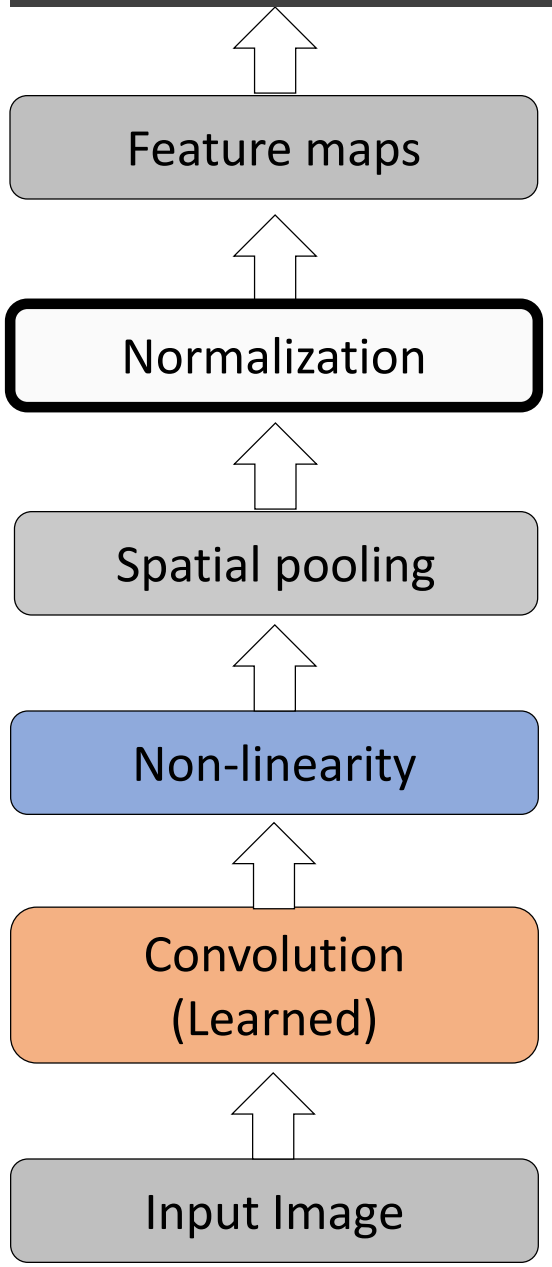


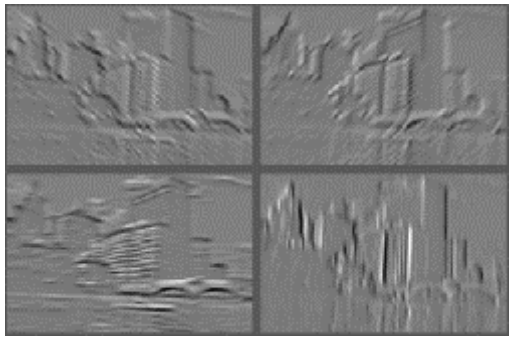
Max-pooling: a non-linear down-sampling

Provide *translation invariance*

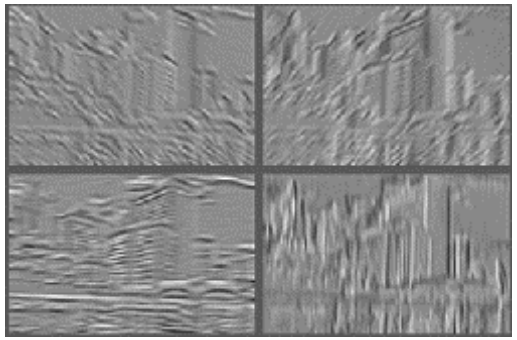
slide credit: S. Lazebnik

Convolutional Neural Networks





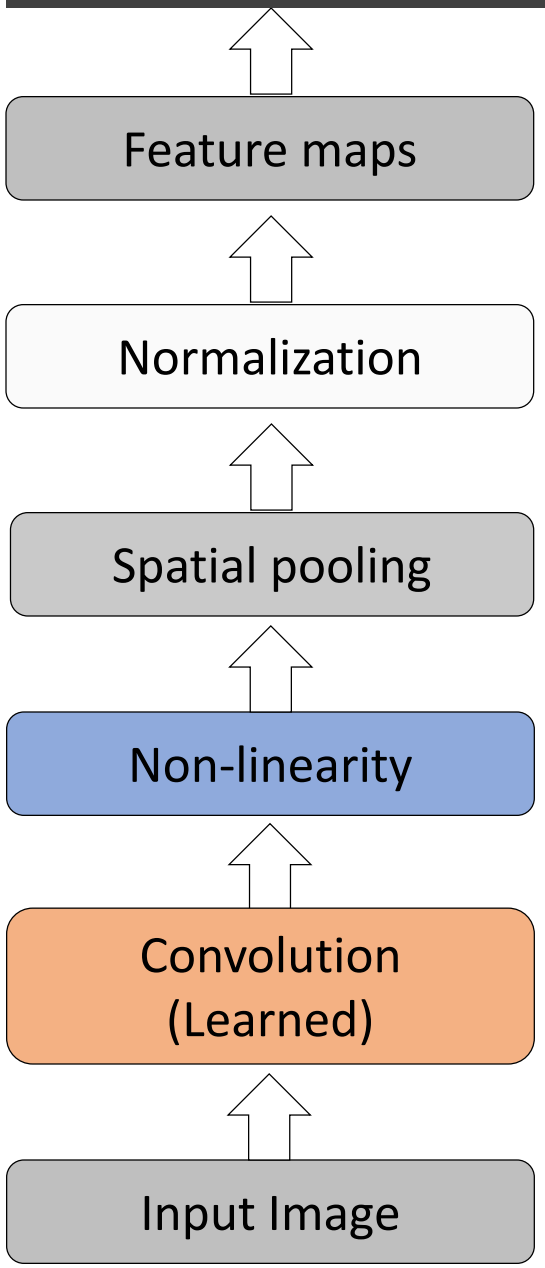
Feature Maps



Feature Maps
After Contrast
Normalization

slide credit: S. Lazebnik

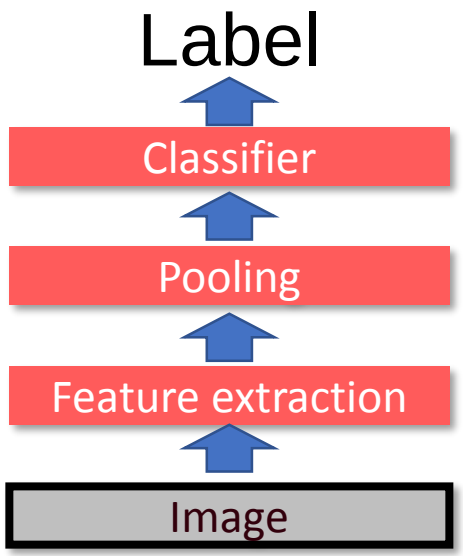
Convolutional Neural Networks

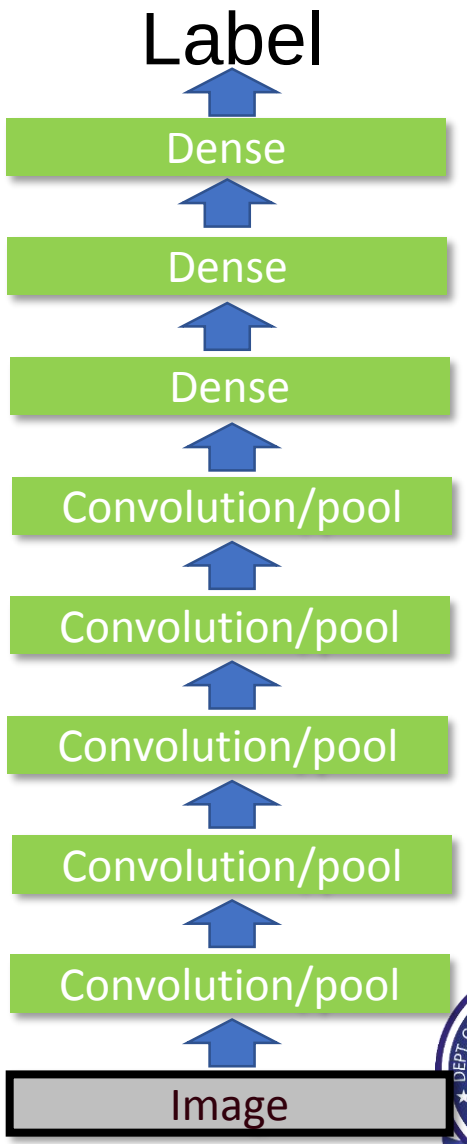


slide credit: S. Lazebnik

Engineered vs. learned features

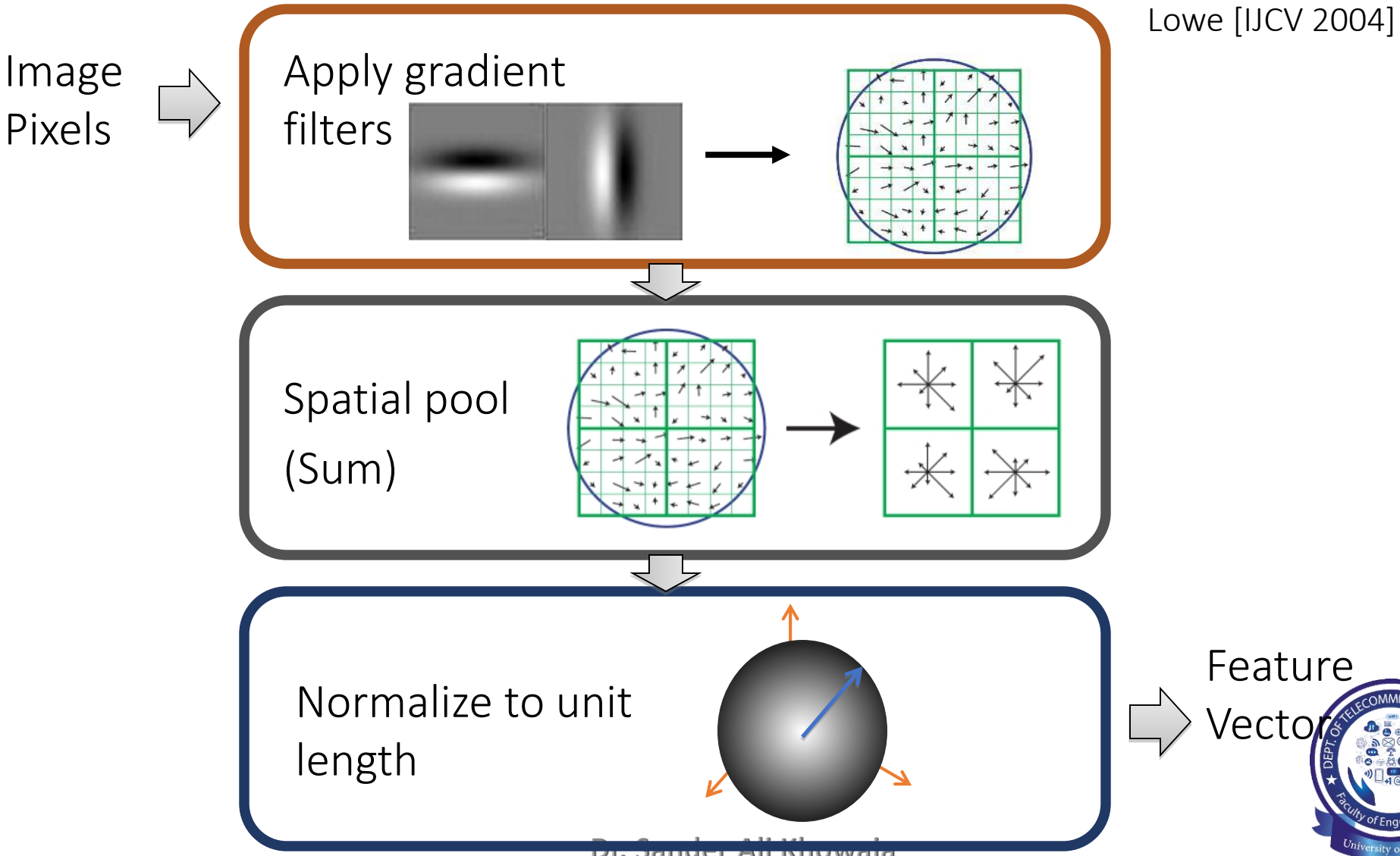
Convolutional filters are trained in a supervised manner by back-propagating classification error





SIFT Descriptor

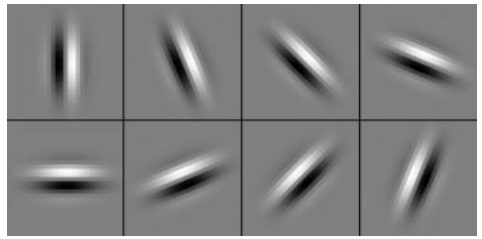
Lowe [IJCV 2004]



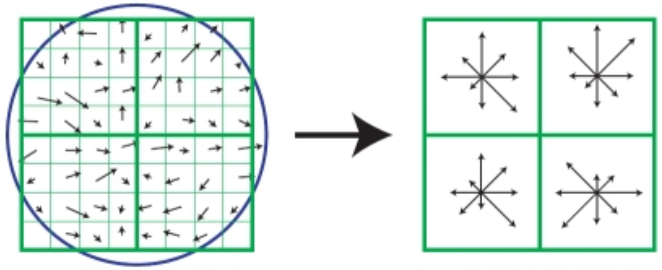
SIFT Descriptor

Image Pixels

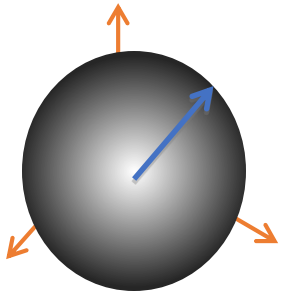
Apply oriented filters



Spatial pool (Sum)



Normalize to unit length



Feature Vector

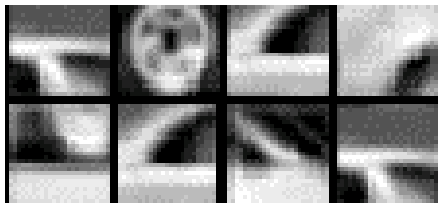
Lowe [IJCV 2004]

slide credit: R. Fergus

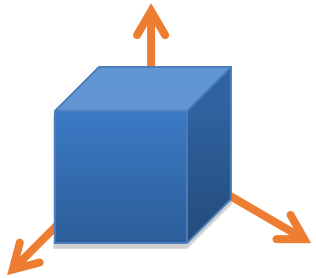
Spatial Pyramid Matching

SIFT Features

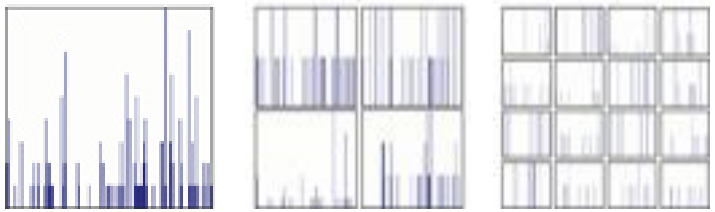
Filter with Visual Words



Max



Multi-scale spatial pool (Sum)

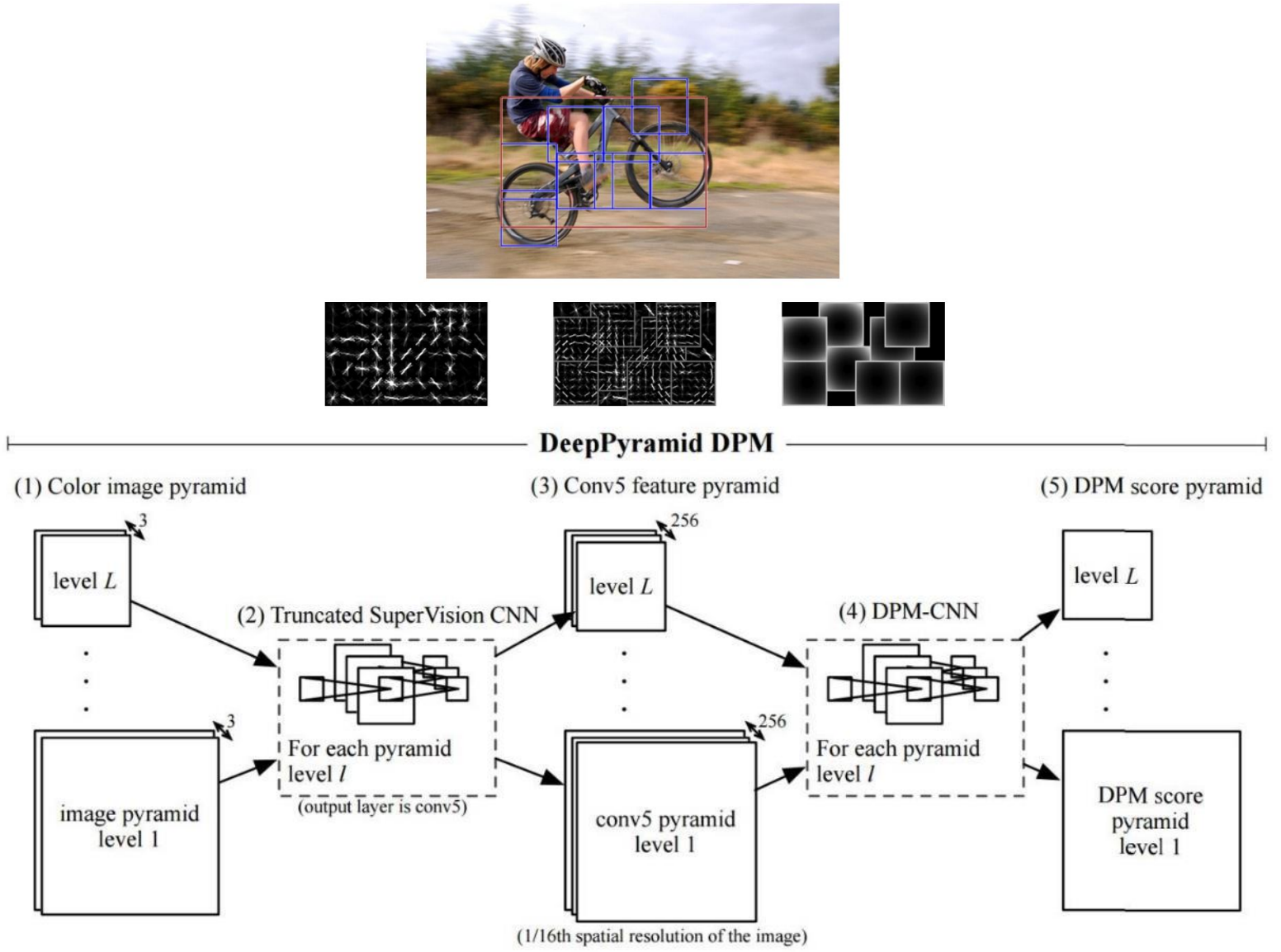


Classifier

Lazebnik, Schmid, Ponce [CVPR 2006]

slide credit: R. Fergus

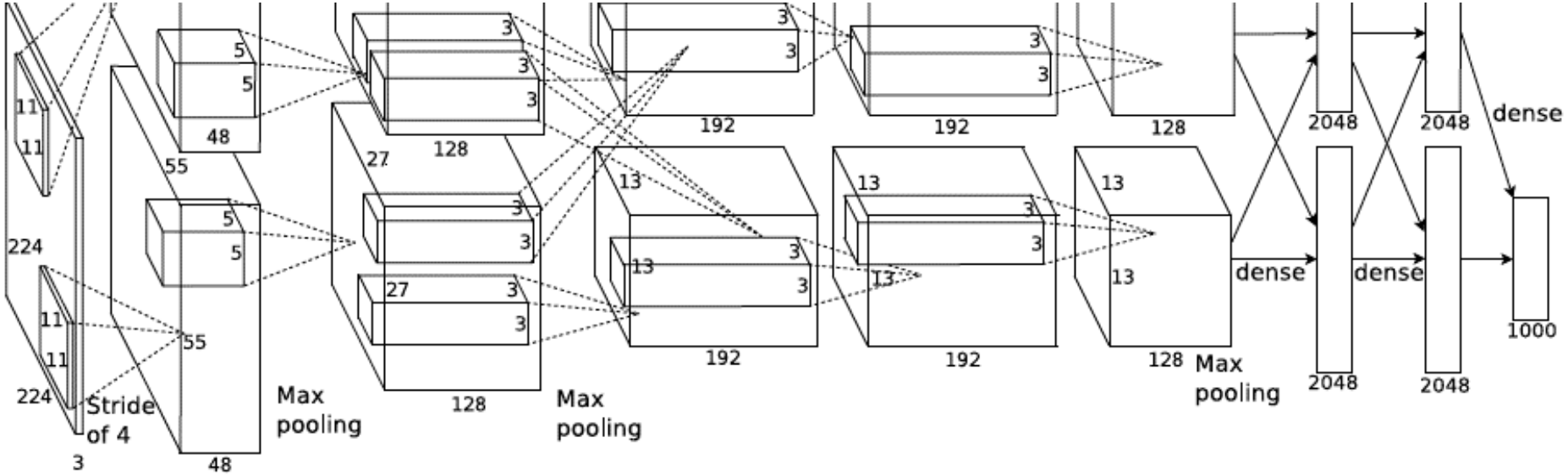
Deformable Part Model



Deformable Part Models are Convolutional Neural Networks [Girshick et al. CVPR 15]

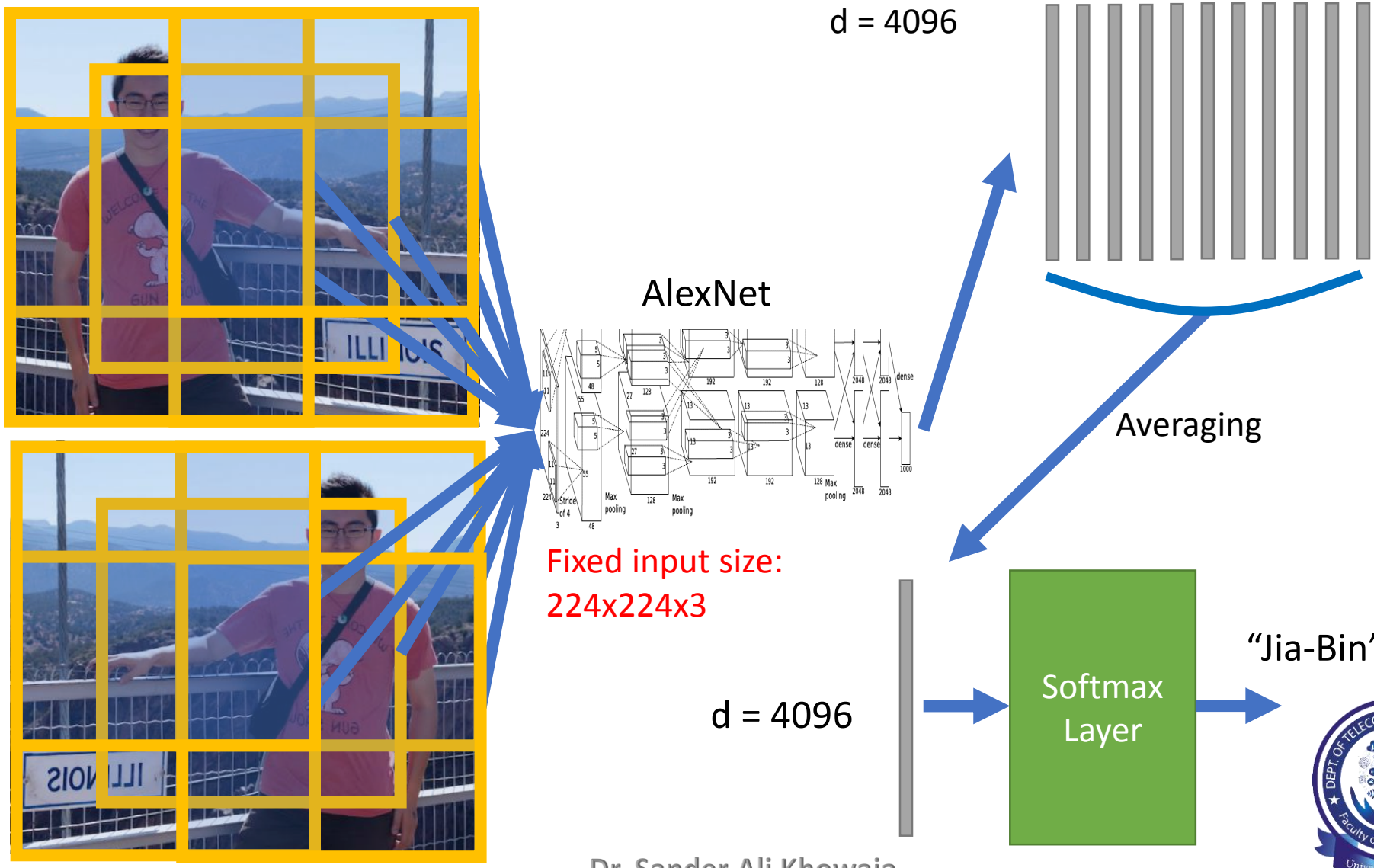
AlexNet

- Similar framework to LeCun'98 but:
  - Bigger model (7 hidden layers, 650,000 units, 60,000,000 params)
  - More data (10^6 vs. 10^3 images)
  - GPU implementation (50x speedup over CPU)
    - Trained on two GPUs for a week

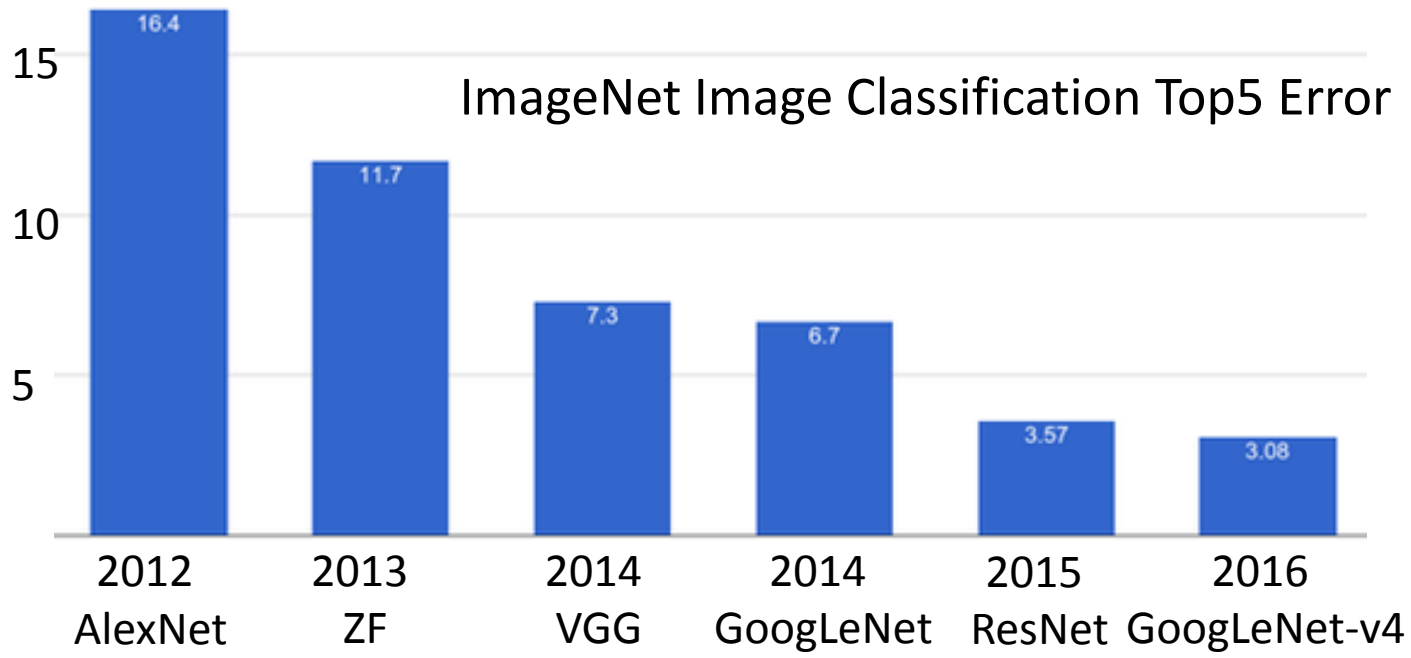


A. Krizhevsky, I. Sutskever, and G. Hinton,
ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012

Using CNN for Image Classification



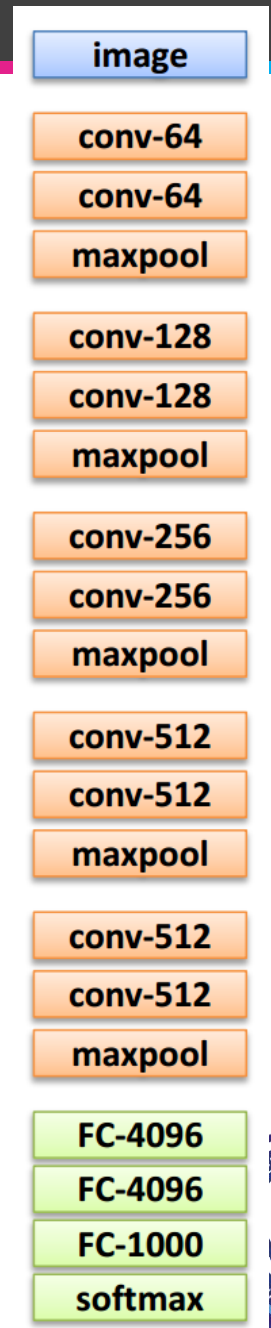
Progress on ImageNet



VGG-Net

- The deeper, the better

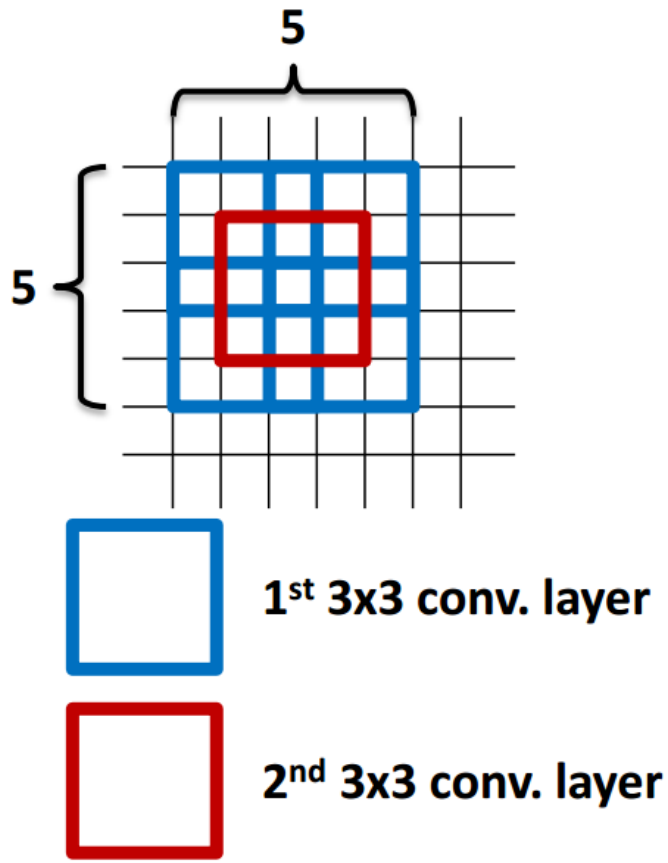
- Key design choices:
  - 3x3 conv. Kernels
    - very small
  - conv. stride 1
    - no loss of information
- Other details:
  - Rectification (ReLU) non-linearity
  - 5 max-pool layers (x2 reduction)
  - no normalization
  - 3 fully-connected (FC) layers



VGG-Net

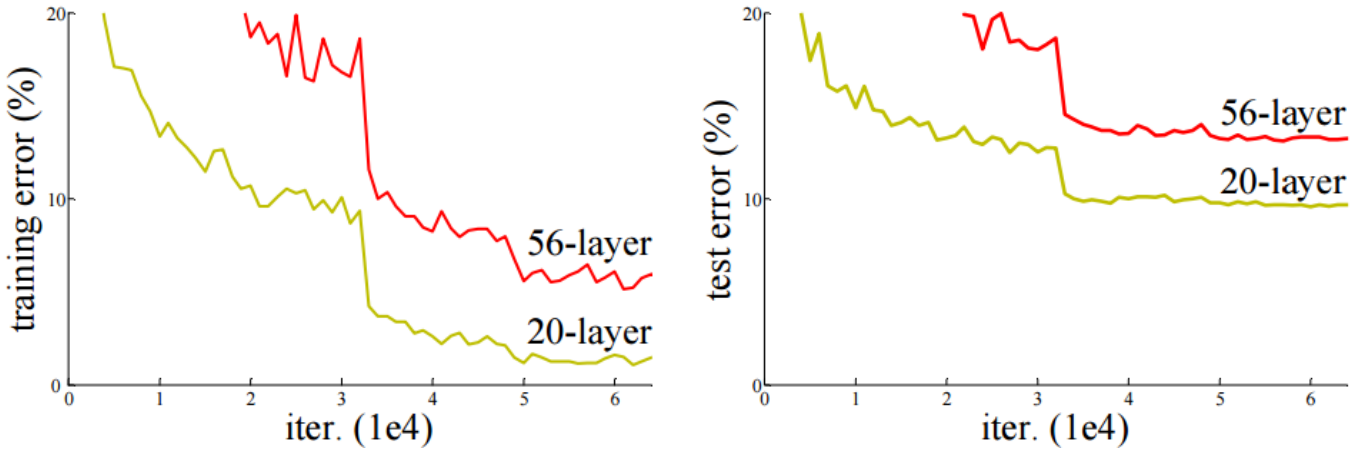
- Why 3x3 layers?
  - Stacked conv. layers have a large receptive field
  - two 3x3 layers – 5x5 receptive field
  - three 3x3 layers – 7x7 receptive field

- More non-linearity
  - Less parameters to learn
  - ~140M per net

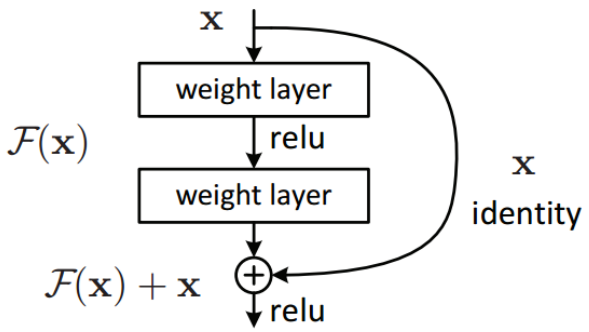


ResNet

- Can we just increase the #layer?



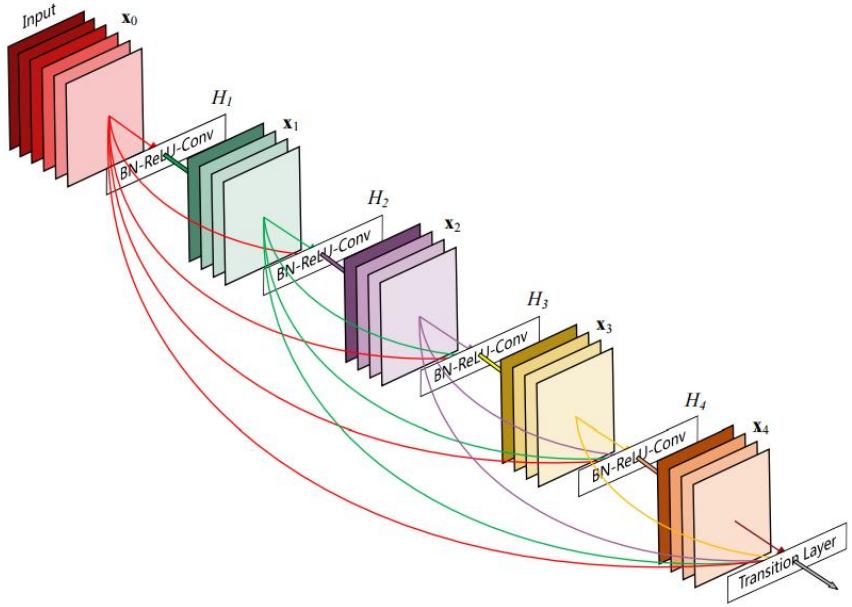
- How can we trai
  - Residual learning

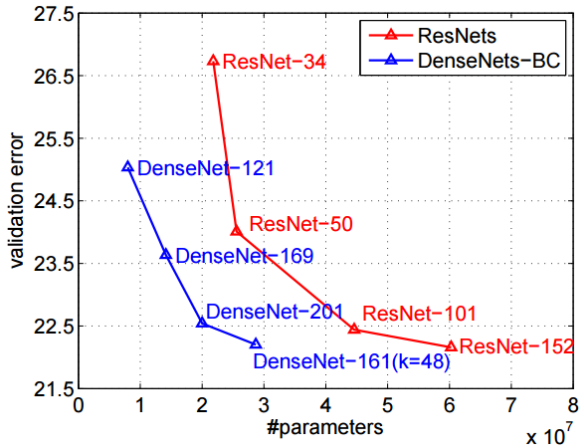


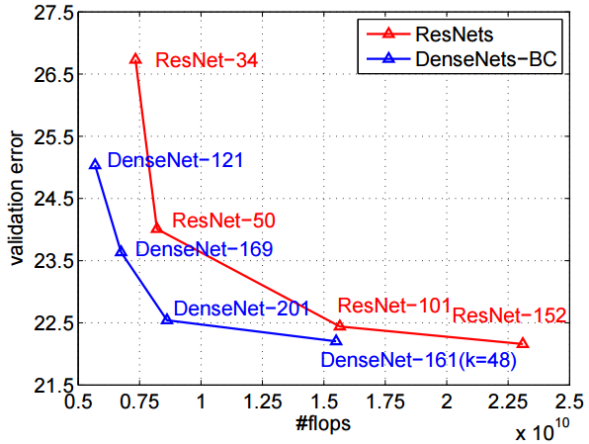
method	top-5 err. (**test**)
VGG [41] (ILSVRC'14)	7.32
GoogLeNet [44] (ILSVRC'14)	6.66
VGG [41] (v5)	6.8
PReLU-net [13]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

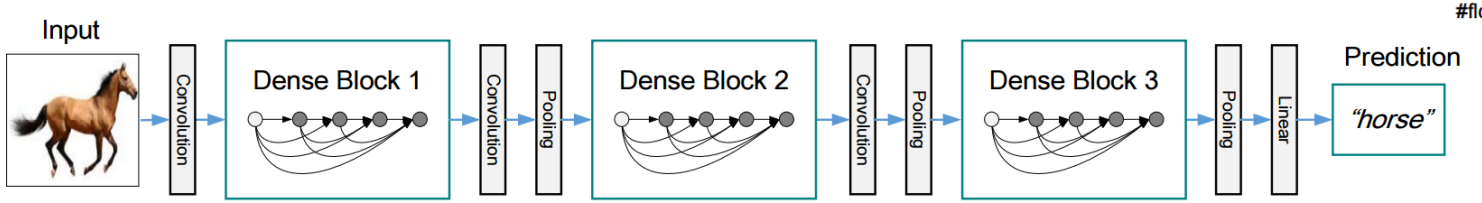
DenseNet

- Shorter connections (like ResNet) help
- Why not just connect them all?









Training Convolutional Neural Networks

- Backpropagation + stochastic gradient descent with momentum
  - Neural Networks: Tricks of the Trade
- Dropout
- Data augmentation
- Batch normalization
- Initialization
  - Transfer learning

Training CNN with gradient descent

- A CNN as composition of functions

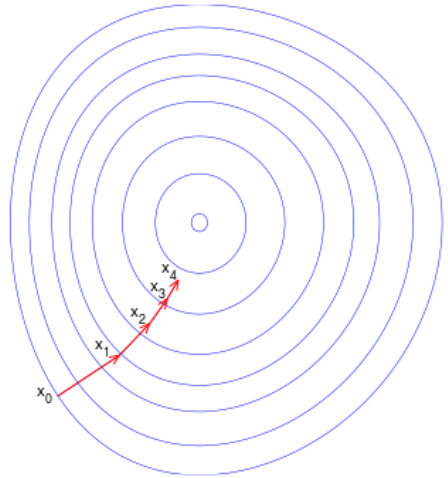
$$f_{\boldsymbol{w}}(\boldsymbol{x}) = f_L(\dots\ (f_2(f_1(\boldsymbol{x}; \boldsymbol{w}_1); \boldsymbol{w}_2)\ \dots; \boldsymbol{w}_L)$$

- Parameters

$$\boldsymbol{w} = (\boldsymbol{w_1}, \boldsymbol{w_2}, \dots \boldsymbol{w_L})$$

- Empirical loss function

$$L(\boldsymbol{w}) = \frac{1}{n}\sum_i l(z_i, f_{\boldsymbol{w}}(\boldsymbol{x_i}))$$



- Gradient descent

$$\boldsymbol{w^{t+1}} = \boldsymbol{w^t} - \eta_t \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{w}}(\boldsymbol{w^t})$$

New weight

Old weight

Learning rate

Gradient

An Illustrative example

$$f(x,y) = xy, \qquad \frac{\partial f}{\partial x} = y, \frac{\partial f}{\partial y} = x$$

Example: $x = 4, y = -3 \Rightarrow f(x,y) = -12$

Partial derivatives

$$\frac{\partial f}{\partial x} = -3, \qquad \frac{\partial f}{\partial y} = 4$$

Gradient

$$\nabla f = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}]$$

Example credit: Andrej Karpathy

$f(x, y, z) = (x + y)z = qz$

$$q = x + y$$

$$\frac{\partial q}{\partial x} = 1, \qquad \frac{\partial q}{\partial y} = 1$$

$$f = qz$$

$$\frac{\partial f}{\partial q} = z, \qquad \frac{\partial f}{\partial z} = q$$

Goal: compute the gradient

$$\nabla f = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}]$$

Example credit: Andrej Karpathy

$f(x, y, z) = (x + y)z = qz$

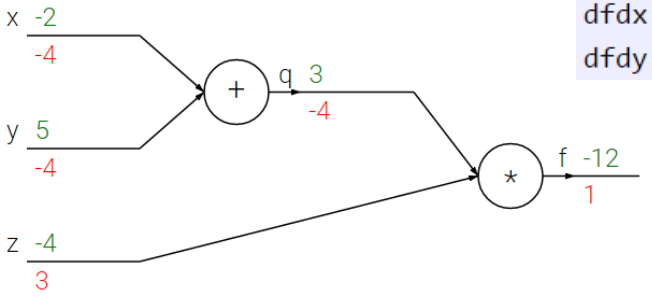
$$q = x + y$$
$$\frac{\partial q}{\partial x} = 1, \qquad \frac{\partial q}{\partial y} = 1$$

$$f = qz$$
$$\frac{\partial f}{\partial q} = z, \qquad \frac{\partial f}{\partial z} = q$$

Chain rule:

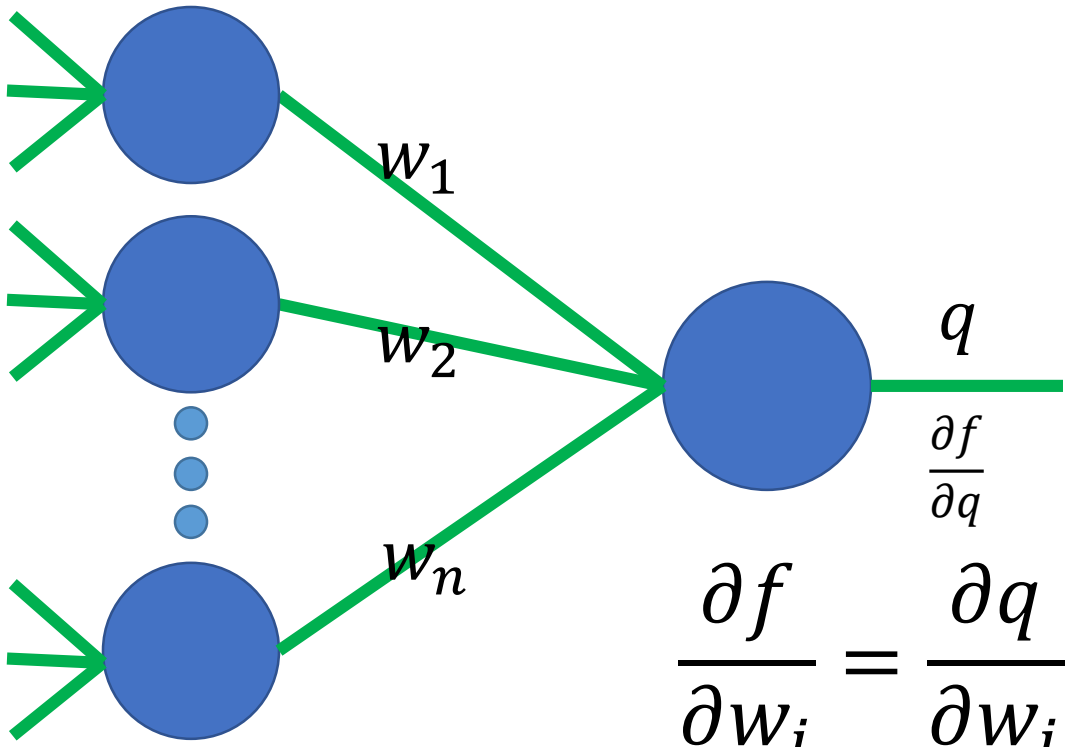
$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial x}$$

```
# set some inputs
x = -2; y = 5; z = -4

# perform the forward pass
q = x + y # q becomes 3
f = q * z # f becomes -12

# perform the backward pass (backpropagation) in reverse order:
# first backprop through f = q * z
dfdz = q # df/dz = q, so gradient on z becomes 3
dfdq = z # df/dq = z, so gradient on q becomes -4
# now backprop through q = x + y
dfdx = 1.0 * dfdq # dq/dx = 1. And the multiplication here is the chain rule!
dfdy = 1.0 * dfdq # dq/dy = 1
```



Example credit: Andrej Karpathy

Backpropagation (recursive chain rule)



$$\frac{\partial f}{\partial w_i} = \frac{\partial q}{\partial w_i}\frac{\partial f}{\partial q}$$

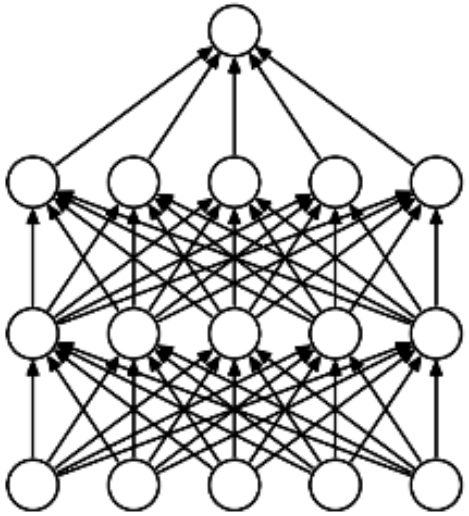
Local gradient

Gate gradient

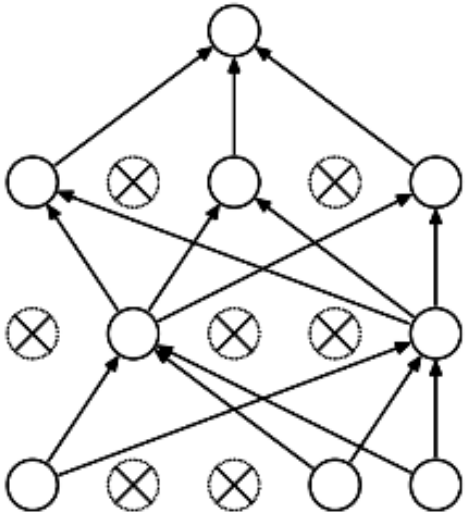
Can be computed during forward pass

The gate receives this during backprop

Dropout



(a) Standard Neural Net



(b) After applying dropout.

Intuition: successful conspiracies

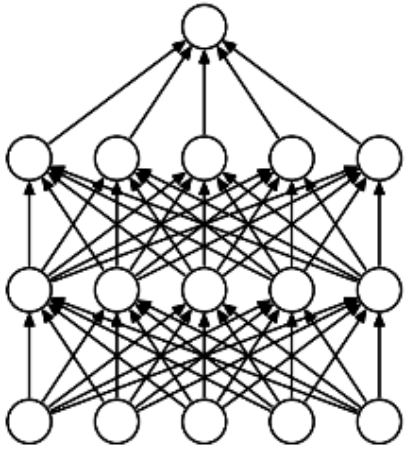
- 50 people planning a conspiracy

- Strategy A: plan a big conspiracy involving 50 people
  - Likely to fail. 50 people need to play their parts correctly.

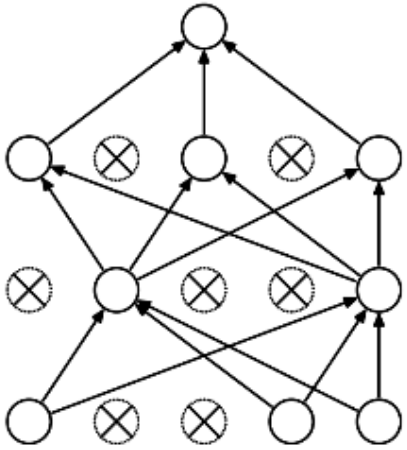
- Strategy B: plan 10 conspiracies each involving 5 people
  - Likely to succeed!

Dropout: A simple way to prevent neural networks from overfitting [Srivastava JMLR 2014]

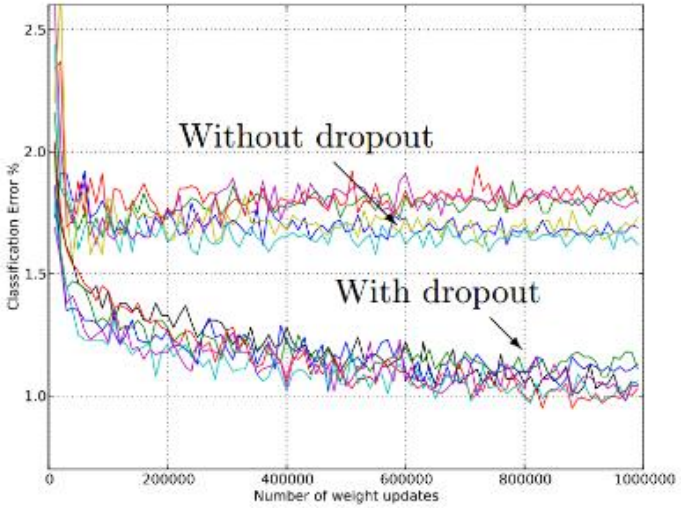
Dropout

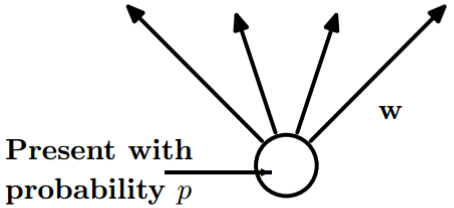


(a) Standard Neural Net

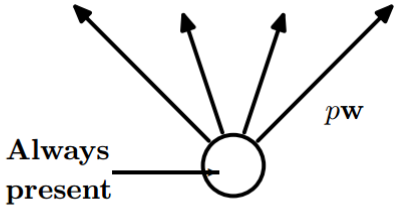


(b) After applying dropout.





(a) At training time



(b) At test time

Main Idea: approximately combining exponentially many different neural network architectures efficiently

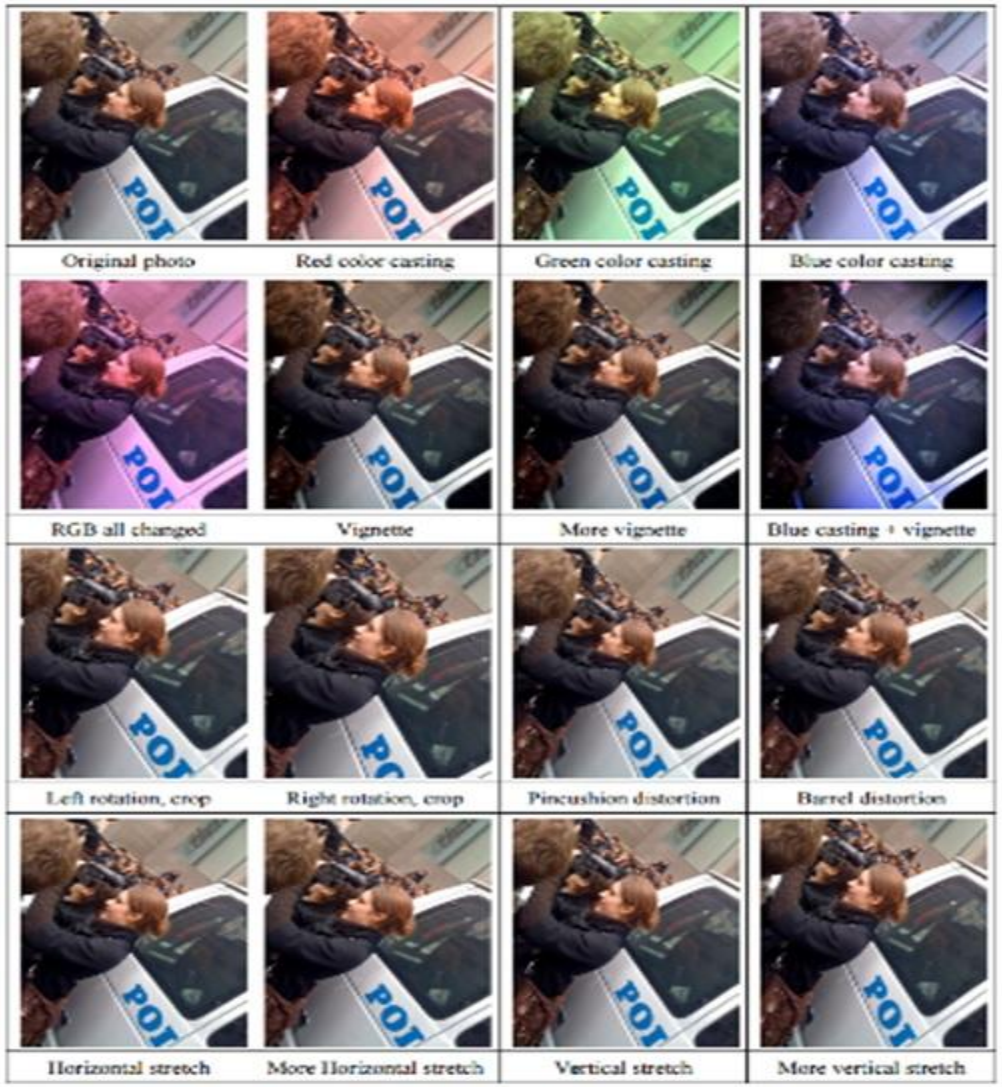
Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SVM on Fisher Vectors of Dense SIFT and Color Statistics	-	-	27.3
Avg of classifiers over FVs of SIFT, LBP, GIST and CSIFT	-	-	26.2
Conv Net + dropout (Krizhevsky et al., 2012)	40.7	18.2	-
Avg of 5 Conv Nets + dropout (Krizhevsky et al., 2012)	38.1	16.4	16.4

Table 6: Results on the ILSVRC-2012 validation/test set.

Dropout: A simple way to prevent neural networks from overfitting [Srivastava JMLR 2014]

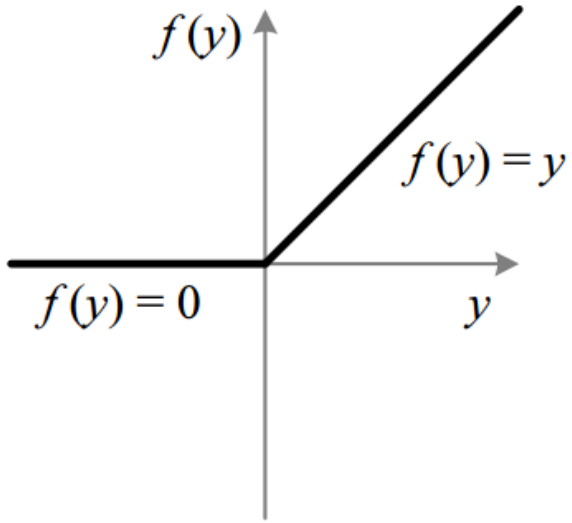
Data Augmentation (Jittering)

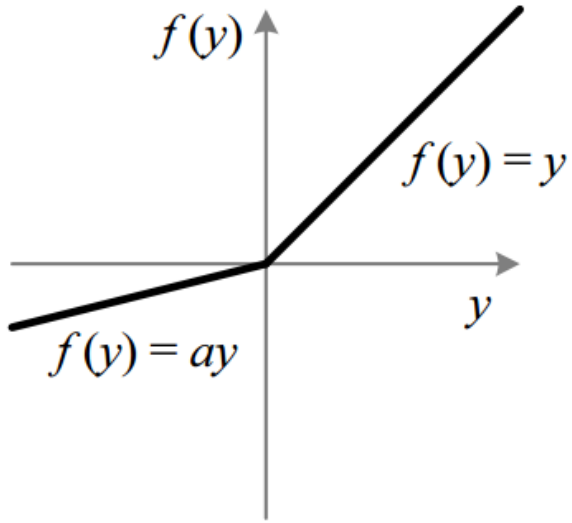
- Create *virtual* training samples
  - Horizontal flip
  - Random crop
  - Color casting
  - Geometric distortion



Deep Image [Wu et al. 2015]

Parametric Rectified Linear Unit

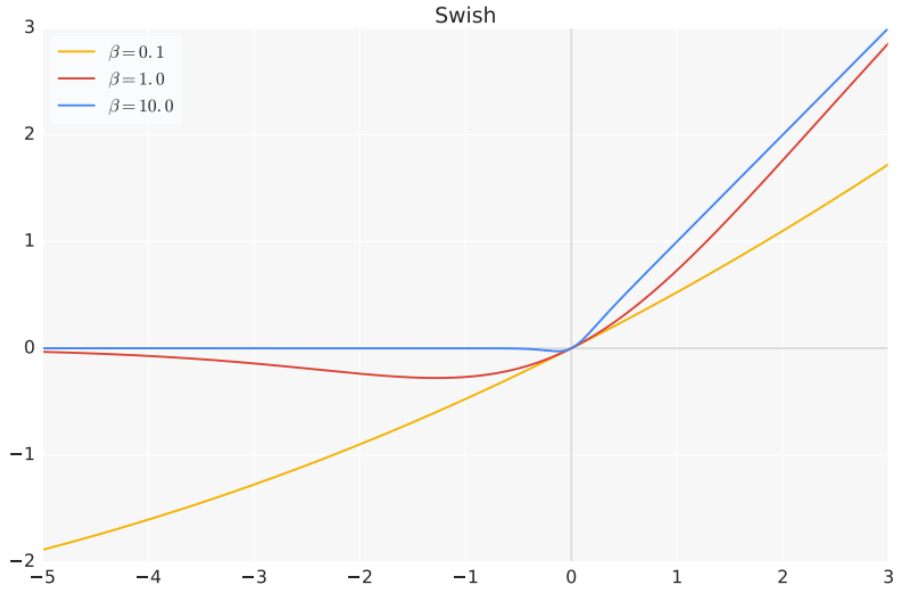




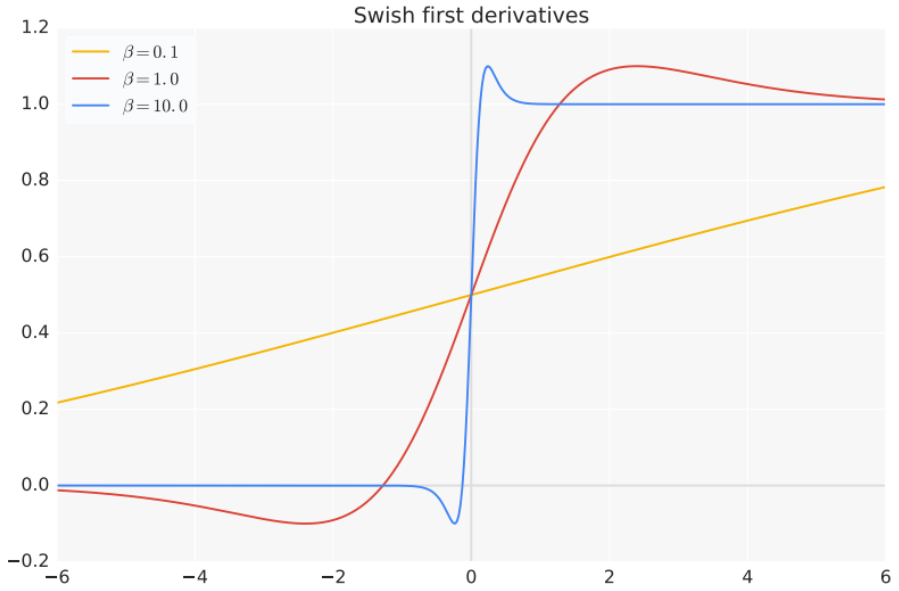
	team	top-5 (**test**)
in competition ILSVRC 14	MSRA, SPP-nets [11]	8.06
	VGG [25]	7.32
	GoogLeNet [29]	6.66
post-competition	VGG [25] (arXiv v5)	6.8
	Baidu [32]	5.98
	MSRA, PReLU-nets	**4.94**

Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification [He et al., 2015]

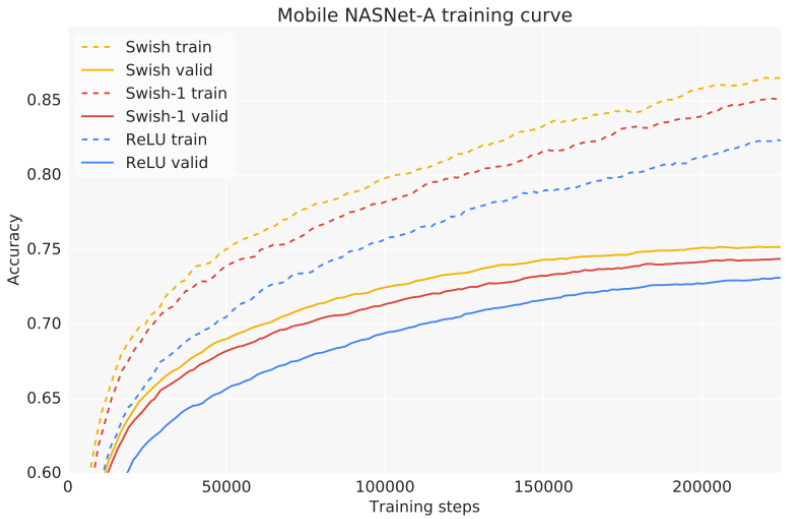
Swish



The Swish activation function



First derivatives of Swish



Model	Top-1 Acc. (%)			Top-5 Acc. (%)		
LReLU	73.8	73.9	74.2	91.6	91.9	91.9
PReLU	74.6	74.7	74.7	92.4	92.3	92.3
Softplus	74.0	74.2	74.2	91.6	91.8	91.9
ELU	74.1	74.2	74.2	91.8	91.8	91.8
SELU	73.6	73.7	73.7	91.6	91.7	91.7
GELU	74.6	-	-	92.0	-	-
ReLU	73.5	73.6	73.8	91.4	91.5	91.6
Swish-1	74.6	74.7	74.7	92.1	92.0	92.0
Swish	74.9	74.9	75.2	92.3	92.4	92.4

https://arxiv.org/abs/1710.05941

Batch Normalization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

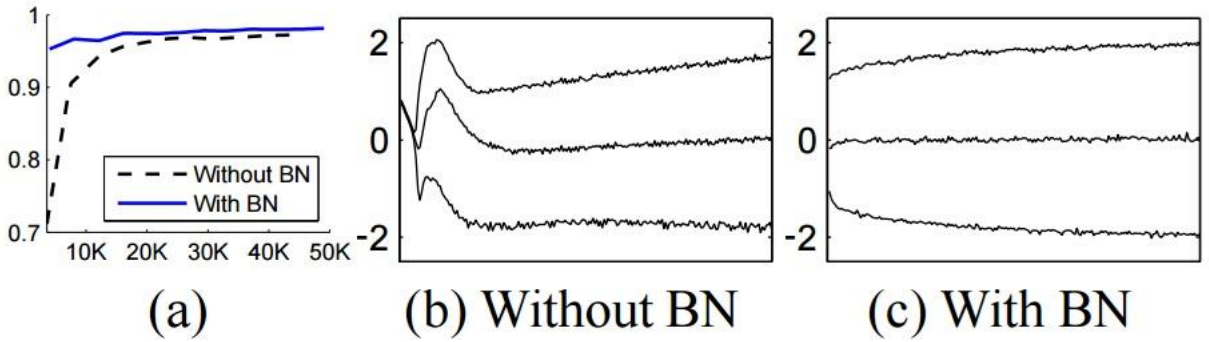
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$



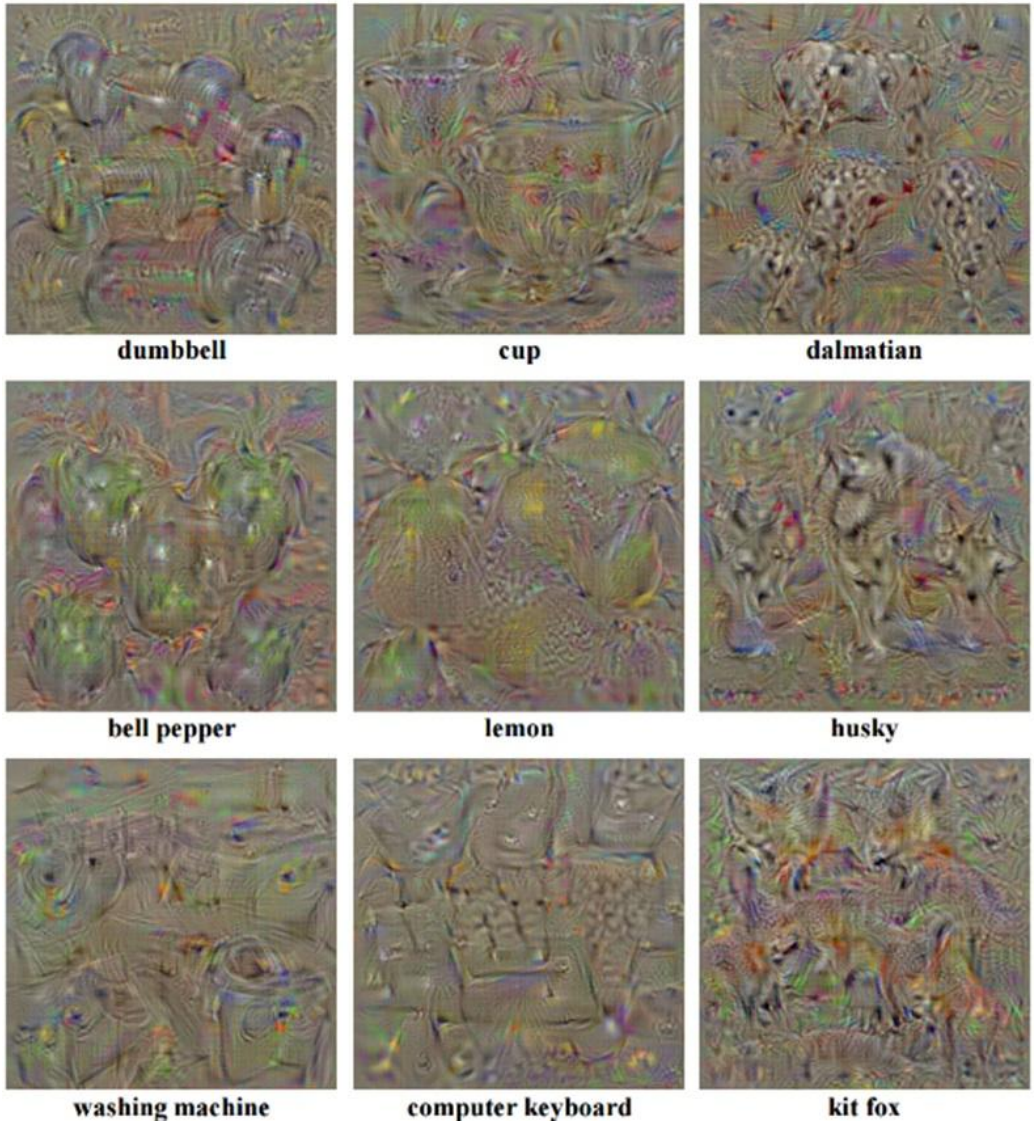
(a) (b) Without BN (c) With BN

Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [Ioffe and Szegedy 2015]

Understanding and Visualizing CNN

- Find images that maximize some class scores
- Individual neuron activation
- Breaking CNNs

Find images that maximize some class scores



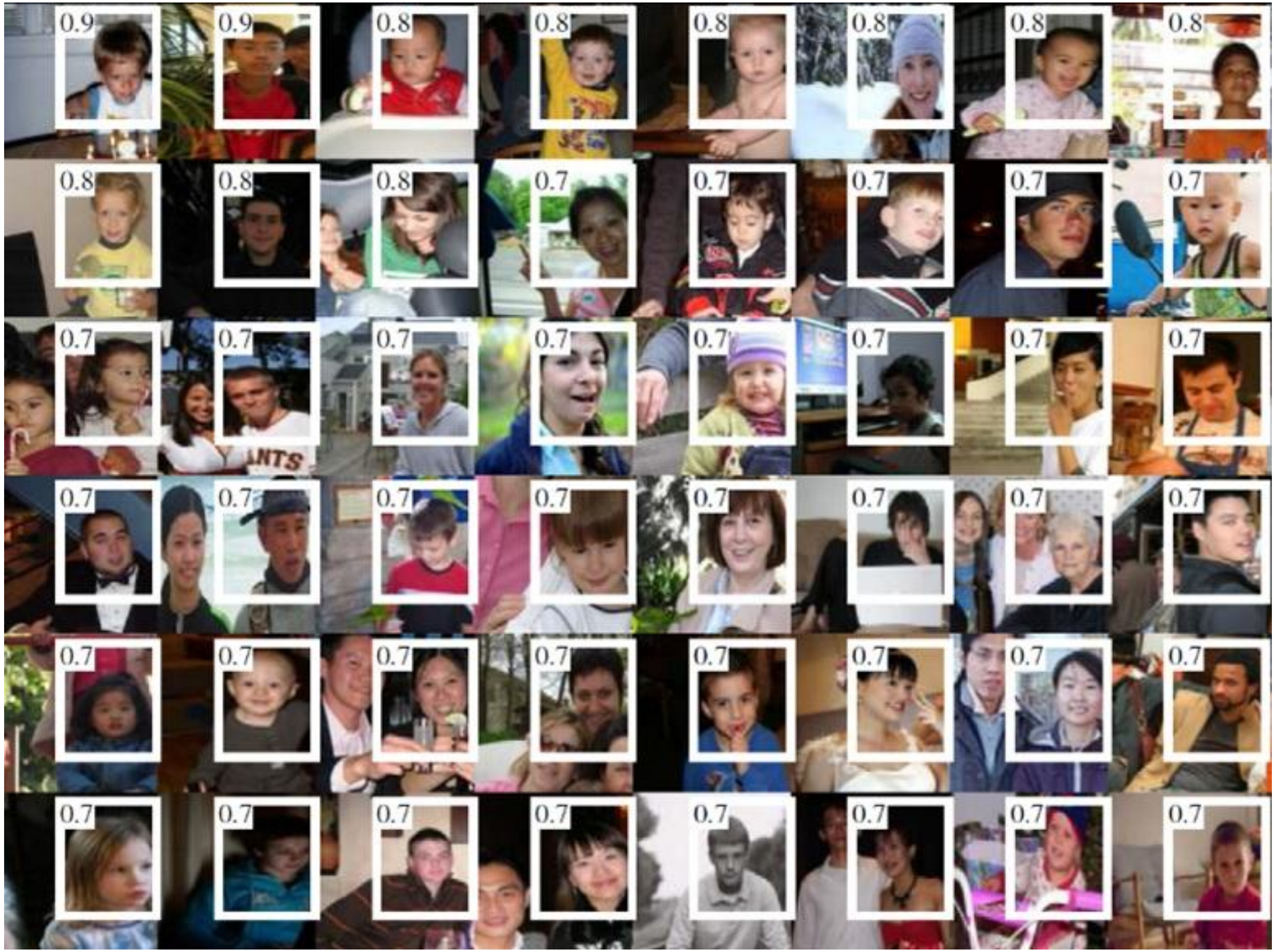
person: HOG template

Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps
[Simonyan et al. ICLR Workshop 2014]

Individual Neuron Activation

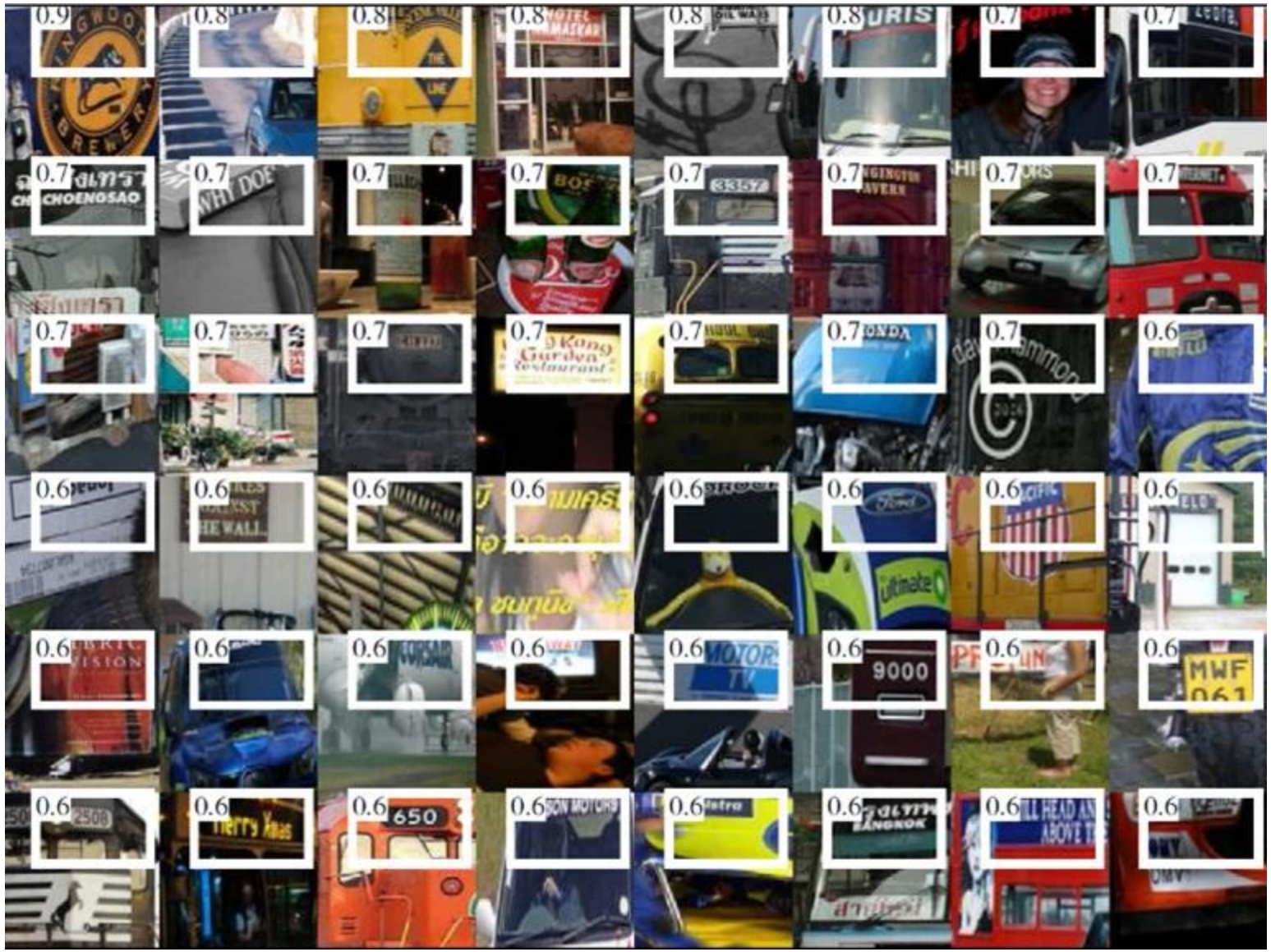
RCNN [Girshick et al. CVPR 2014]

Individual Neuron Activation



RCNN [Girshick et al. CVPR 2014]

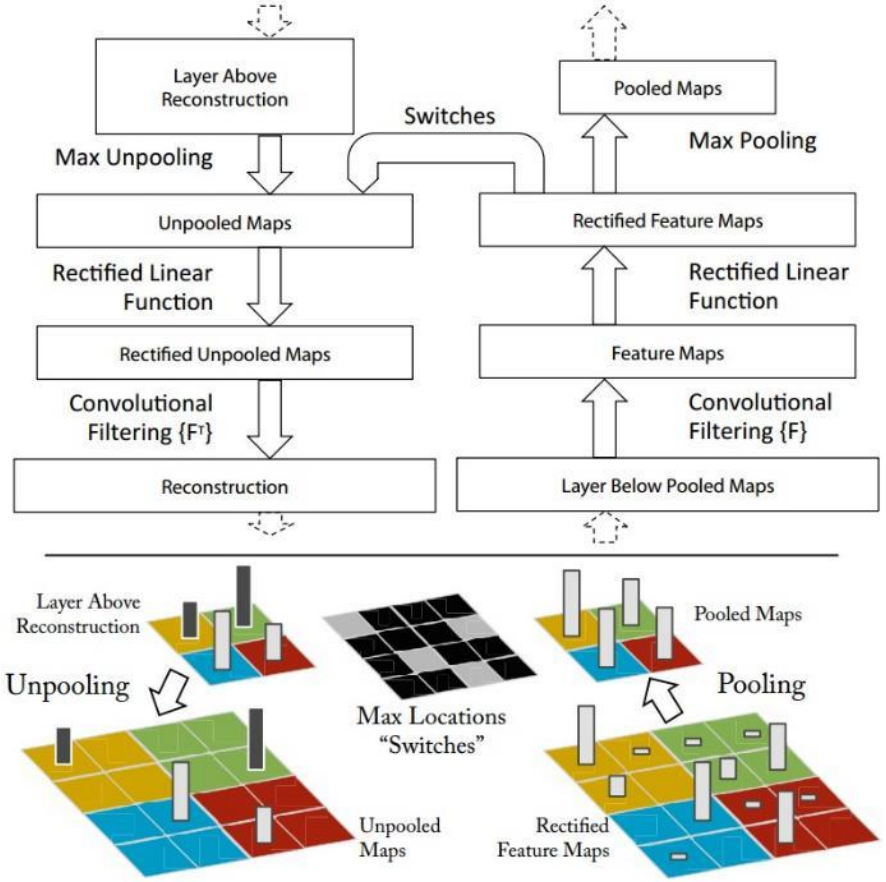
Individual Neuron Activation



RCNN [Girshick et al. CVPR 2014]

Map activation back to the input pixel space

- What input pattern originally caused a given activation in the feature maps?



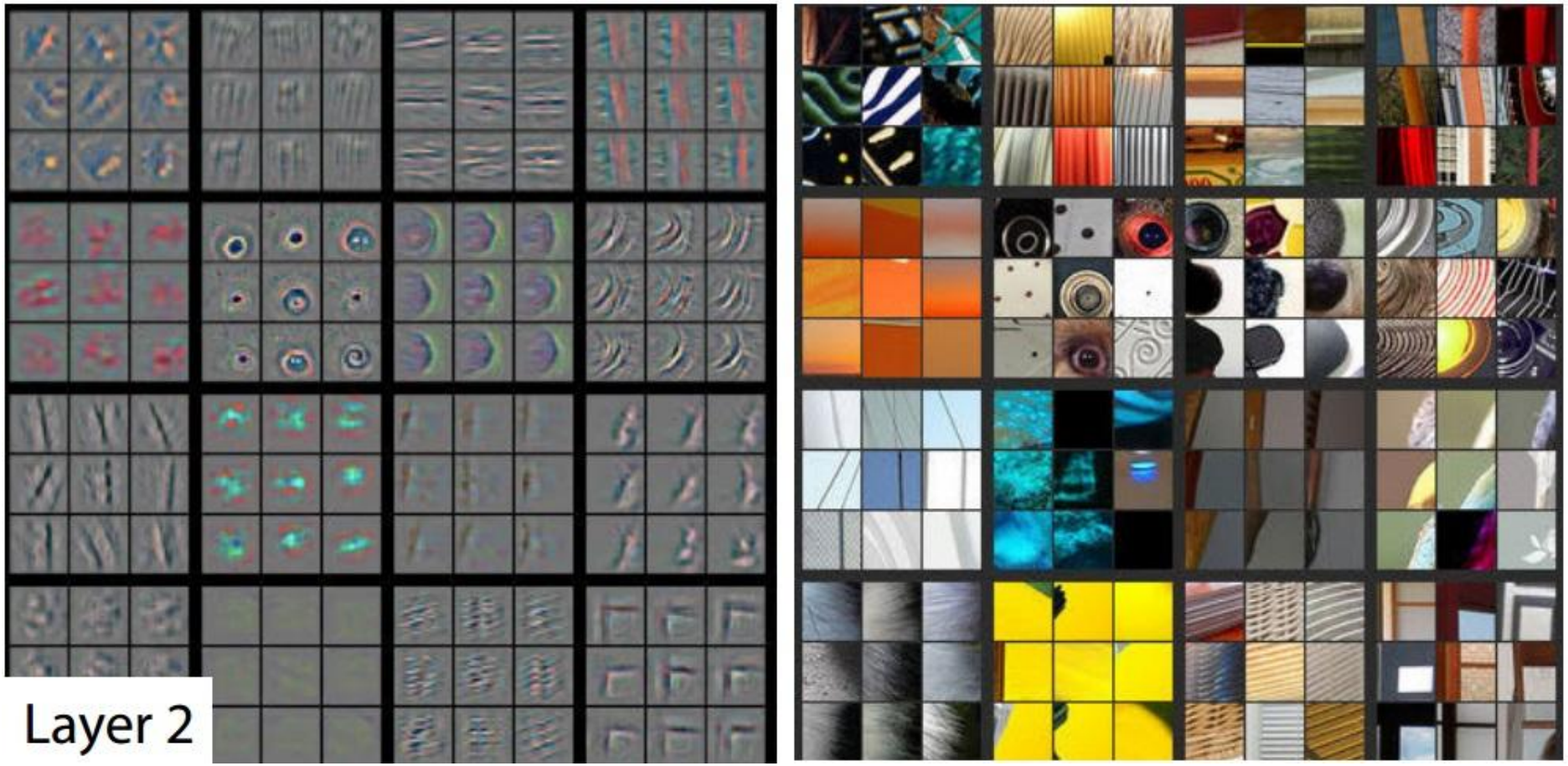
Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]

Layer 1

Layer 1

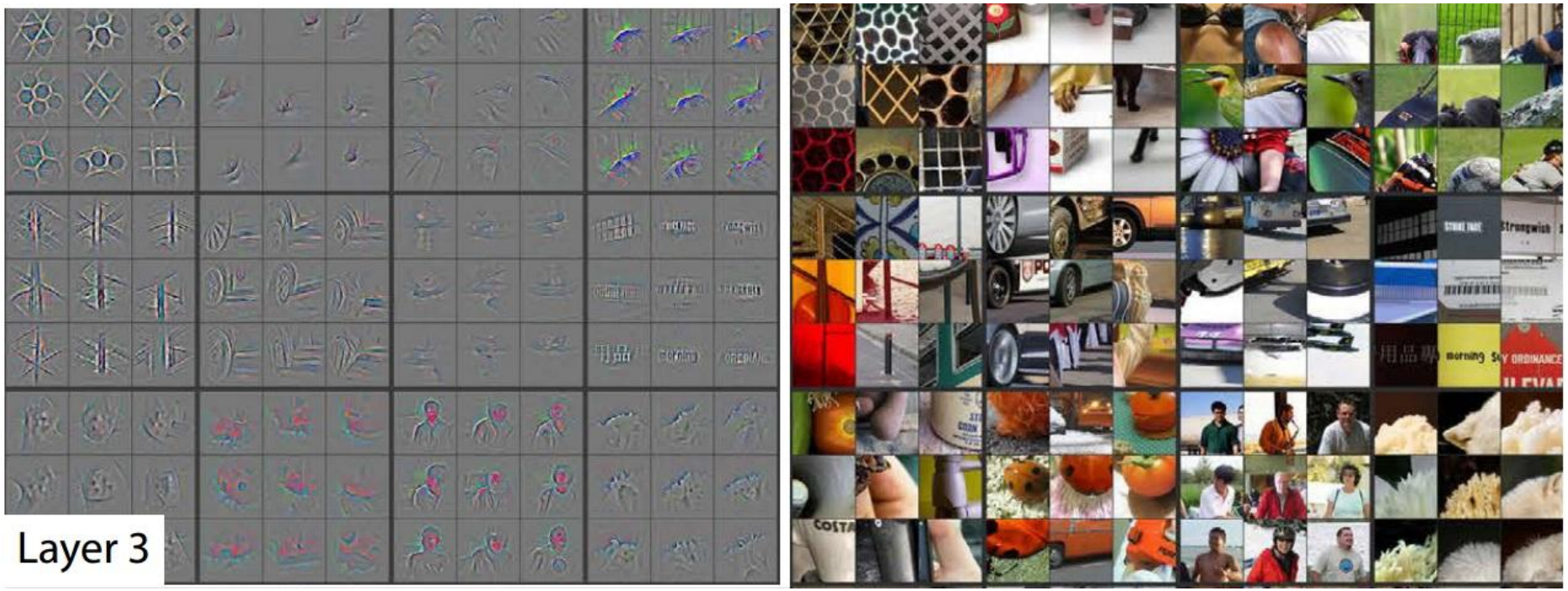
Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]

Layer 2



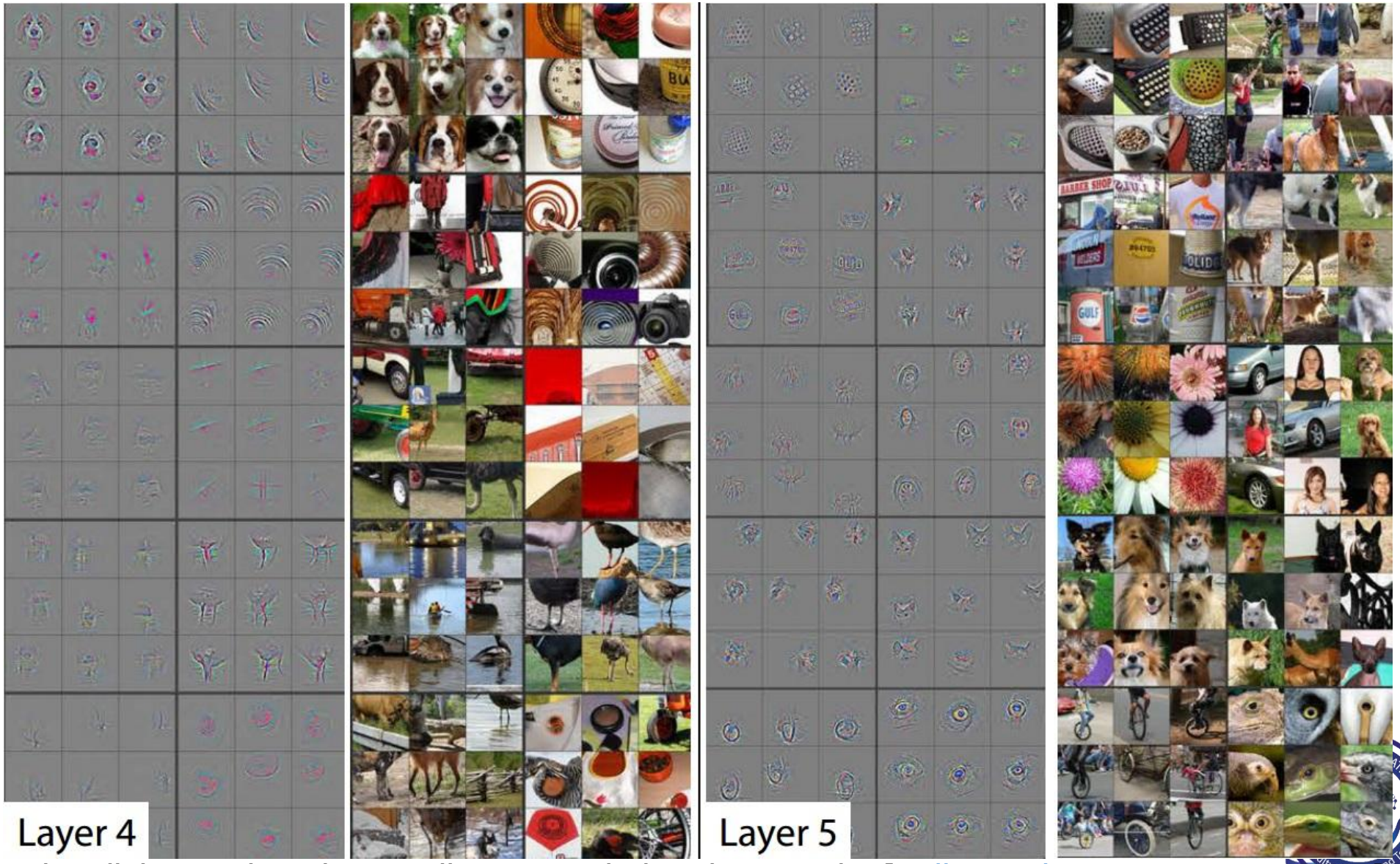
Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]

Layer 3



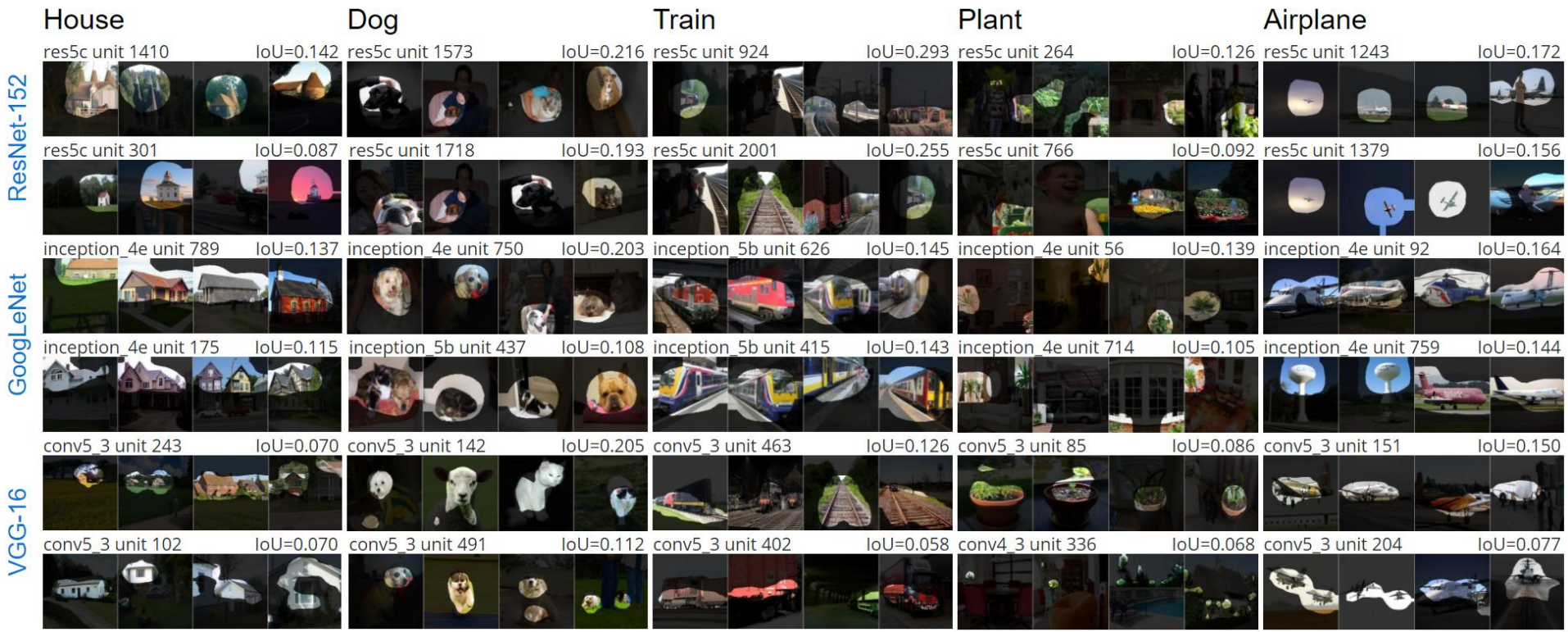
Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]

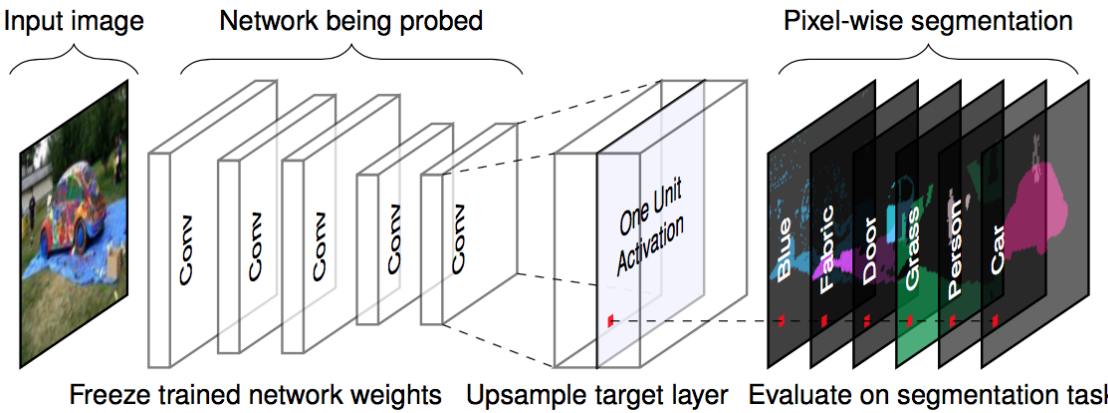
Layer 4 and 5



Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]

Network Dissection





http://netdissect.csail.mit.edu/

Deep learning library

- TensorFlow
  - Research + Production
- PyTorch
  - Research
- Caffe2
  - Production

Things to remember

- Convolutional neural networks
  - A cascade of conv + ReLU + pool
  - Representation learning
  - Advanced architectures
  - Tricks for training CNN
- Visualizing CNN
  - Activation
  - Dissection

Resources

- http://deeplearning.net/
  - Hub to many other deep learning resources
- https://github.com/ChristosChristofidis/awesome-deep-learning
  - A resource collection deep learning
- https://github.com/kjw0612/awesome-deep-vision
  - A resource collection deep learning for computer vision
- http://cs231n.stanford.edu/syllabus.html
  - Nice course on CNN for visual recognition

Things to remember

- Overview
  - Neuroscience, Perceptron, multi-layer neural networks
- Convolutional neural network (CNN)
  - Convolution, nonlinearity, max pooling
  - CNN for classification and beyond
- Understanding and visualizing CNN
  - Find images that maximize some class scores; visualize individual neuron activation, input pattern and images; breaking CNNs
- Training CNN
  - Dropout; data augmentation; batch normalization; transfer learning

Geoffery Hinton is a legend (its just fo

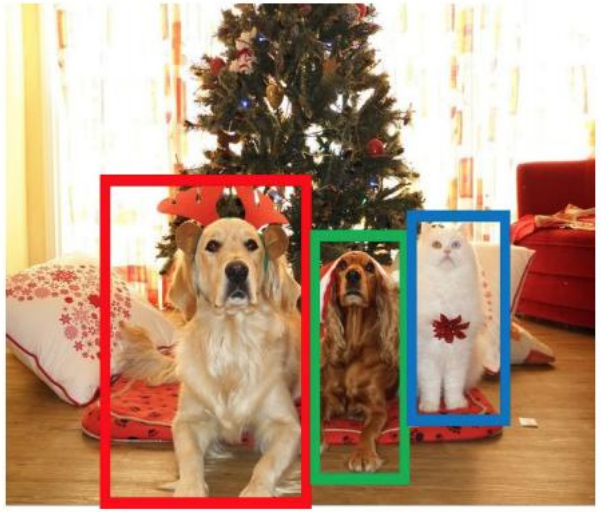
‘Godfather of AI’ Geoffrey Hinton quits Google and warns over dangers of misinformation

The neural network pioneer says dangers of chatbots were ‘quite scary’ and warns they could be exploited by ‘bad actors’

Dr Geoffrey Hinton, the ‘godfather of AI’, has left Google. Photograph: Linda Nylind/The Guardian

The man often touted as the godfather of AI has quit Google, citing concerns over the flood of misinformation, the possibility for AI to upend the job market, and the “existential risk” posed by the creation of a true digital intelligence.

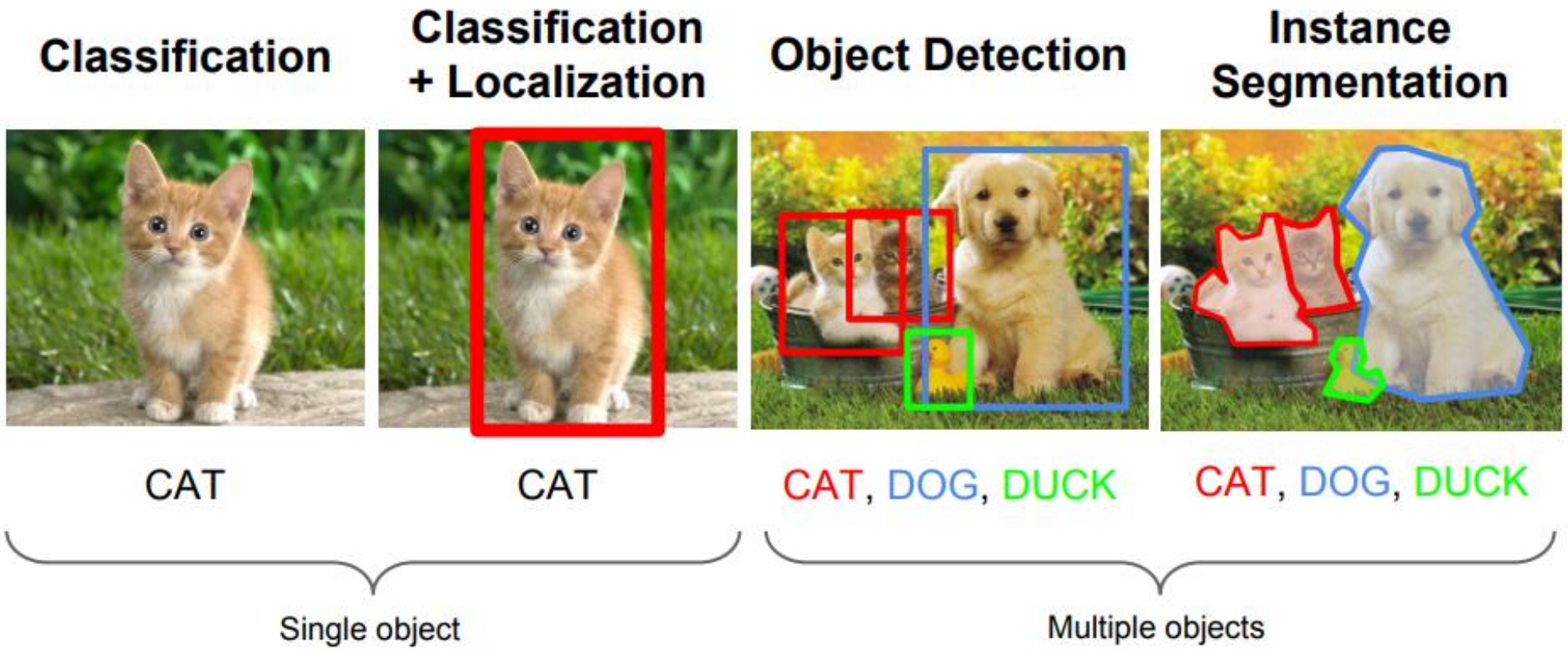
Object Detection



Many slides from D. Hoiem, J. Hays, J. Johnson, R. Girshick

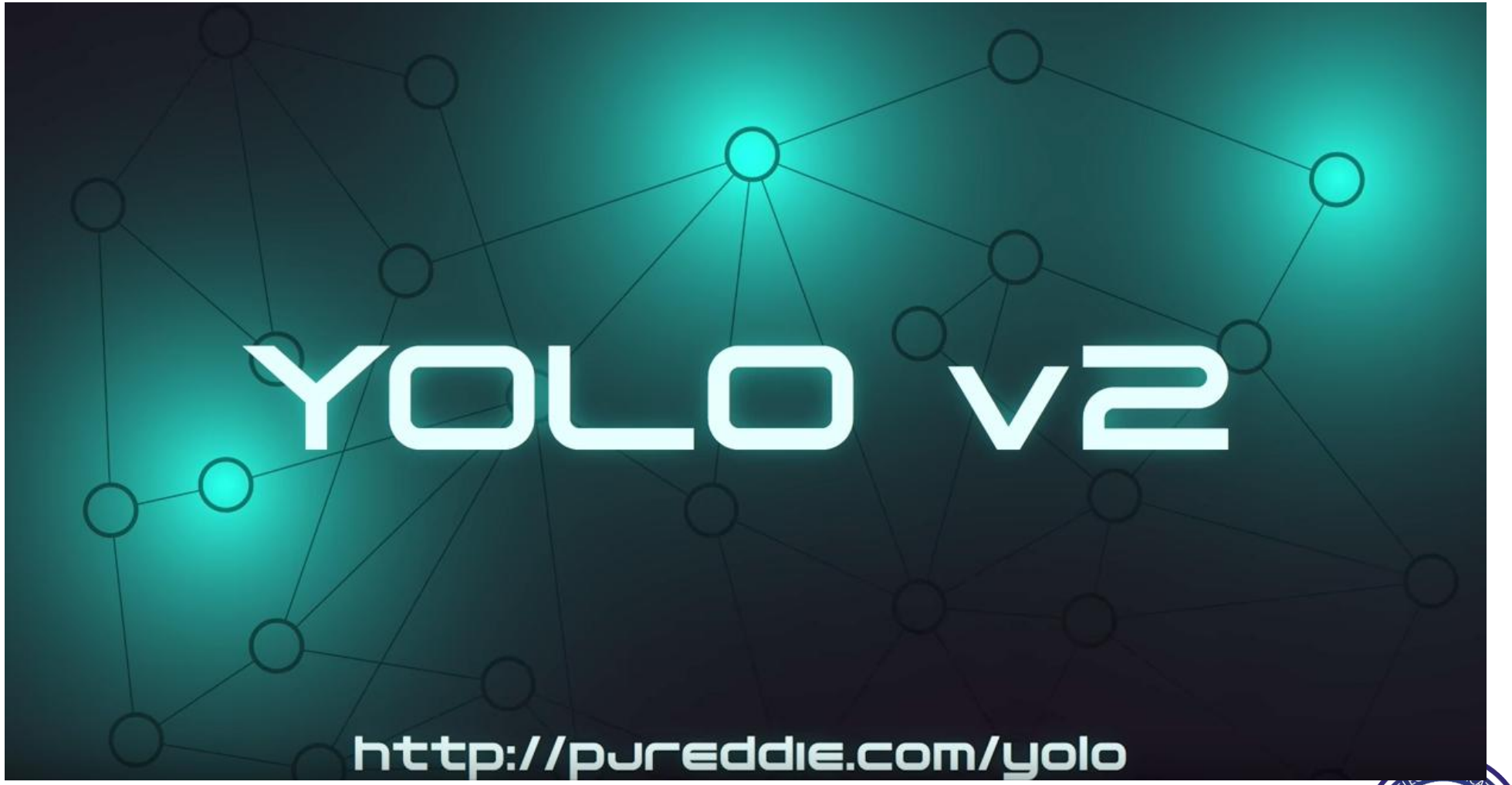
Dr. Sander Ali Khowaja

Roadmap



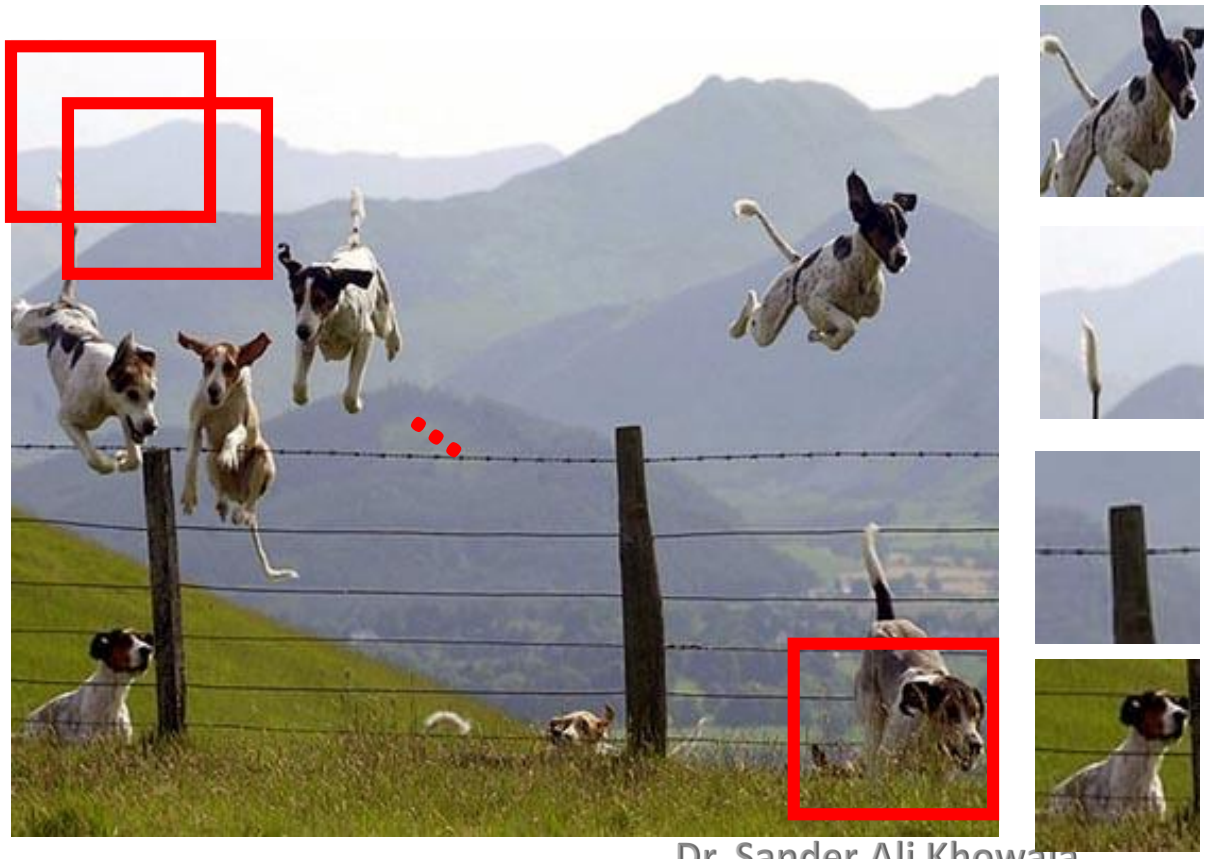
Slide from A. Karpathy

Demo



Object Category Detection

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



Dog Model

Object or
Non-Object?

Challenges in modeling the object class

Illumination

Object pose

Clutter

Occlusions

Intra-class appearance

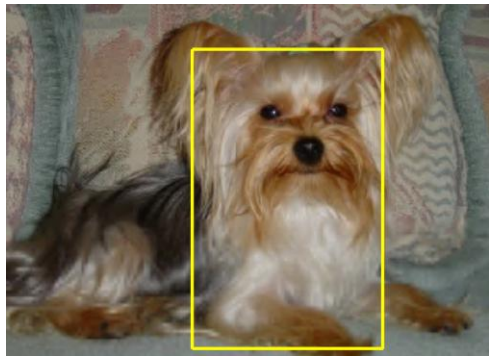
Viewpoint

Slide from K. Grauman, B. Leibe

Challenges in modeling the non-object class

True Detections

Bad Localization

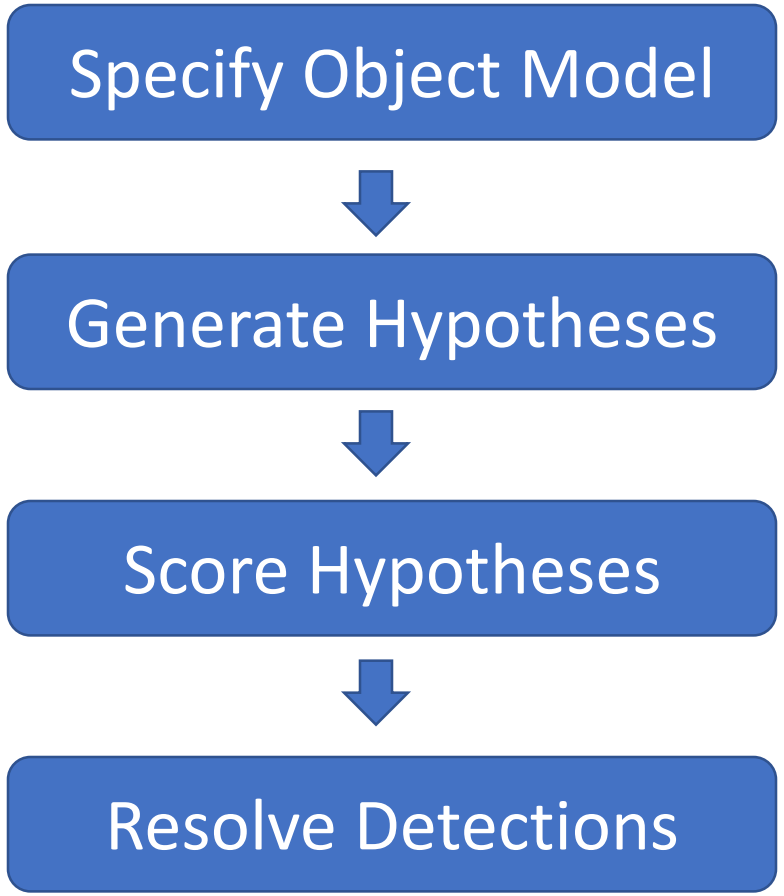


Confused with Similar Object

Misc. Background

Confused with Dissimilar Objects

General Process of Object Recognition

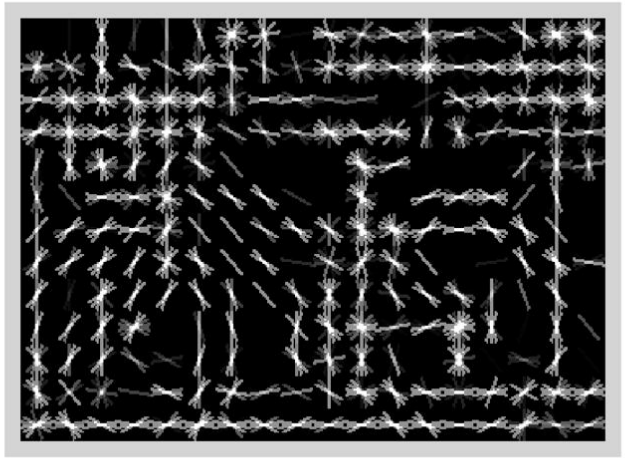


What are the object parameters?

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

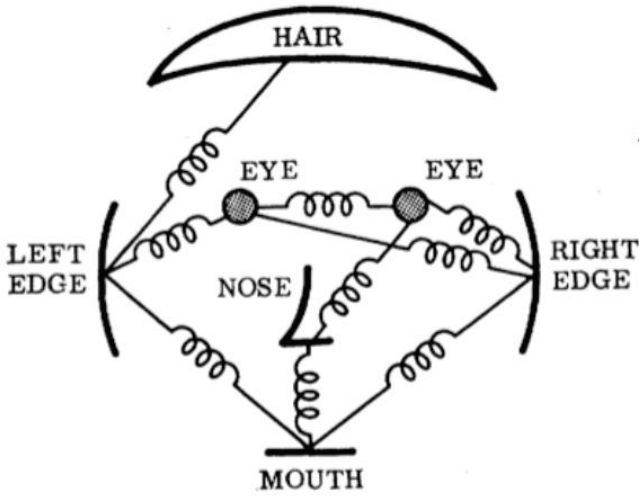


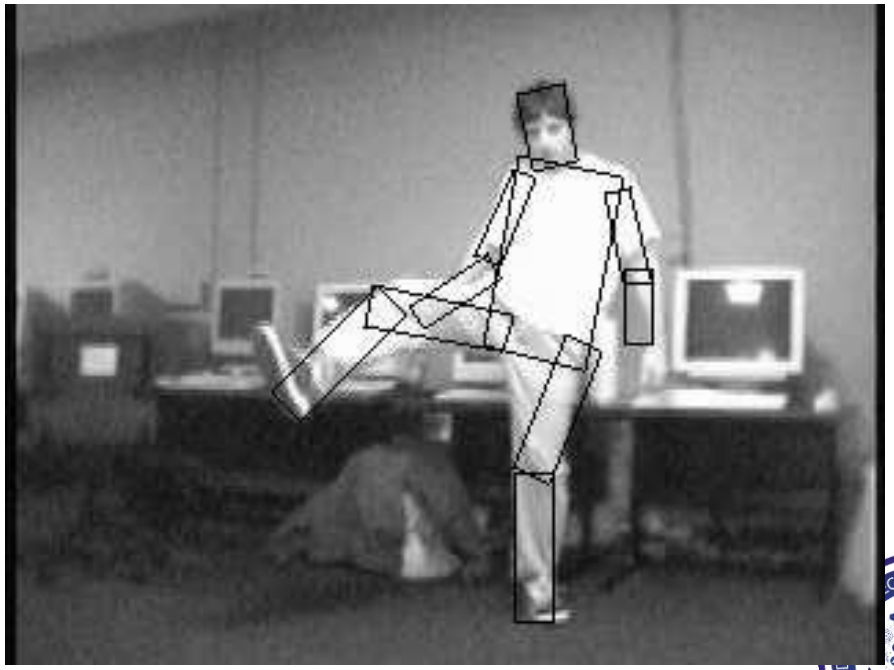
Template Visualization

Images from Felzenszwalb

Specifying an object model

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



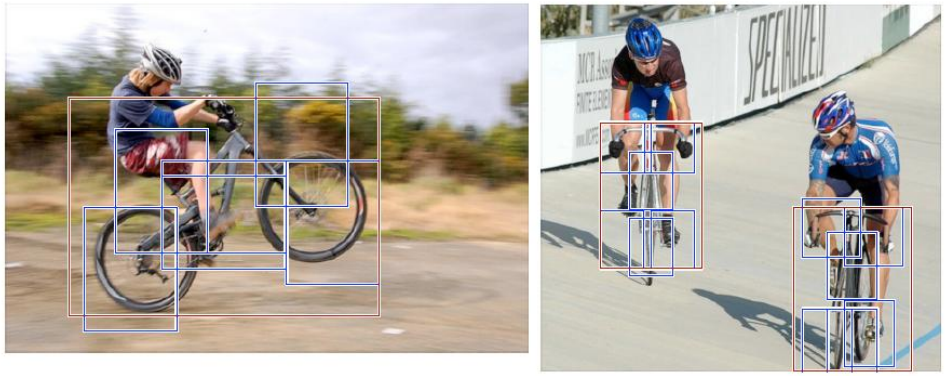


Images from Felzenszwalb

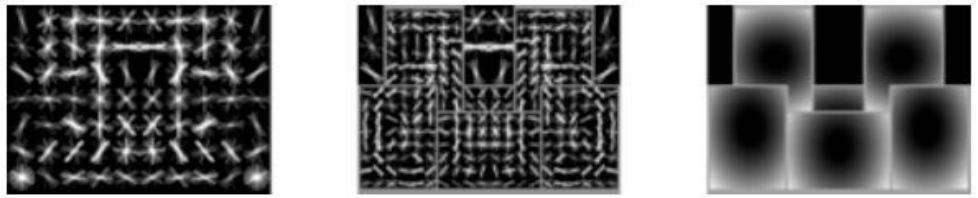
Specifying an object model

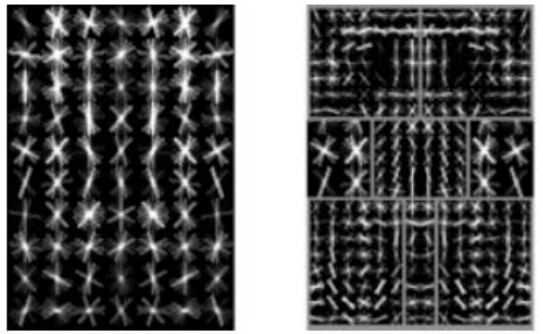
3. Hybrid template/parts model

Detections



Template Visualization





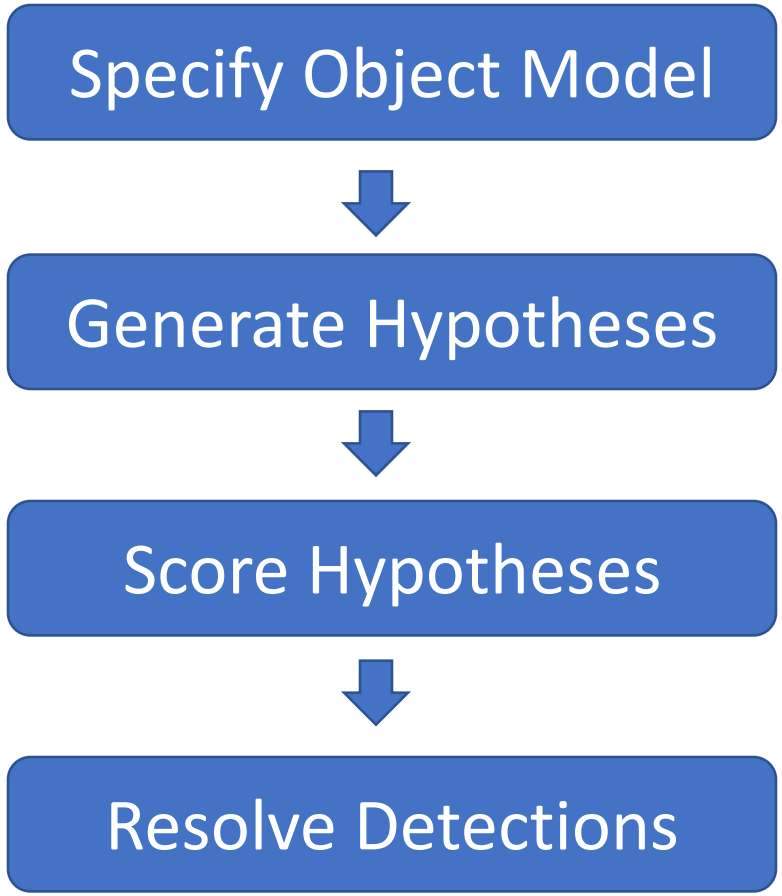
root filters
coarse resolution

part filters
finer resolution

deformation
models

Felzenszwalb et al. 2008

General Process of Object Recognition



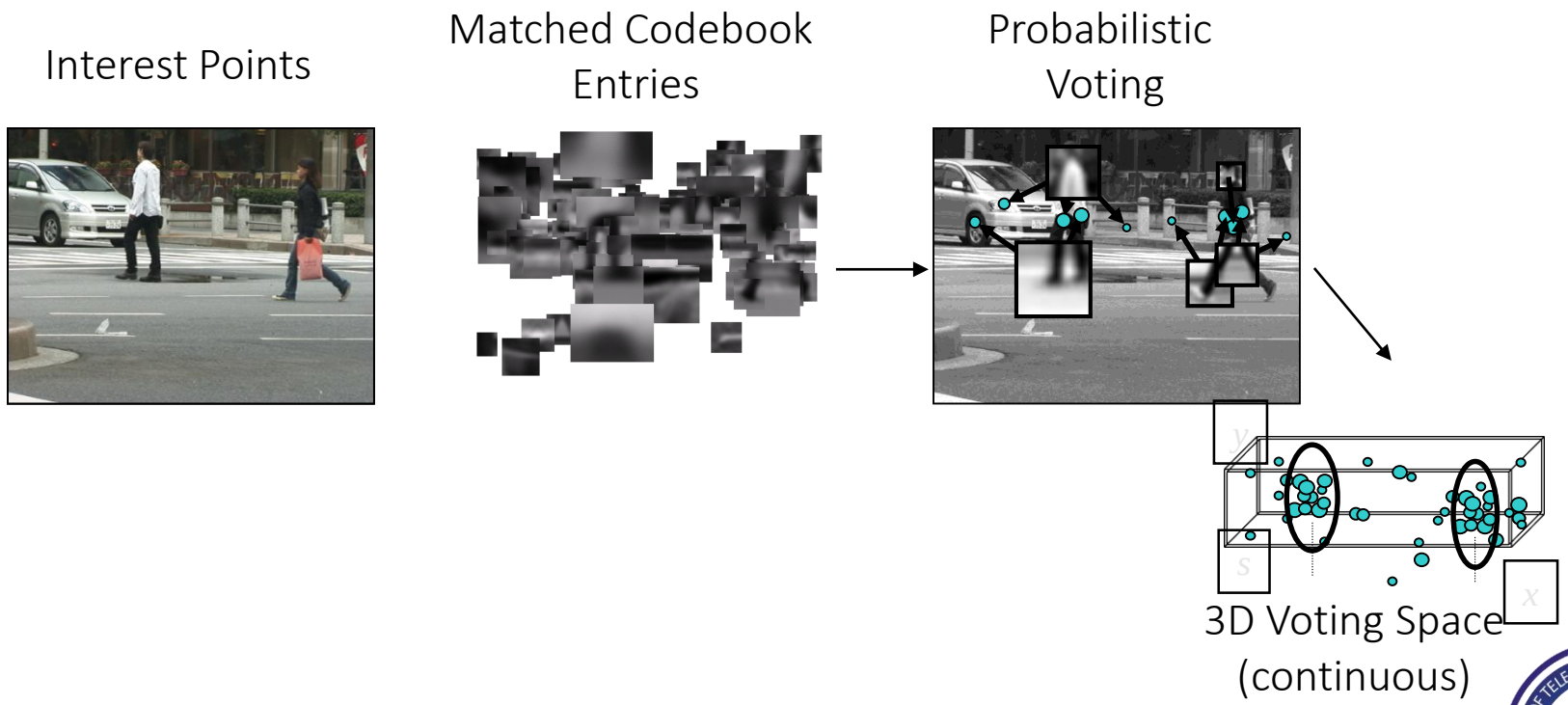
Propose an alignment of the model to the image

Generating hypotheses

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

Generating hypotheses

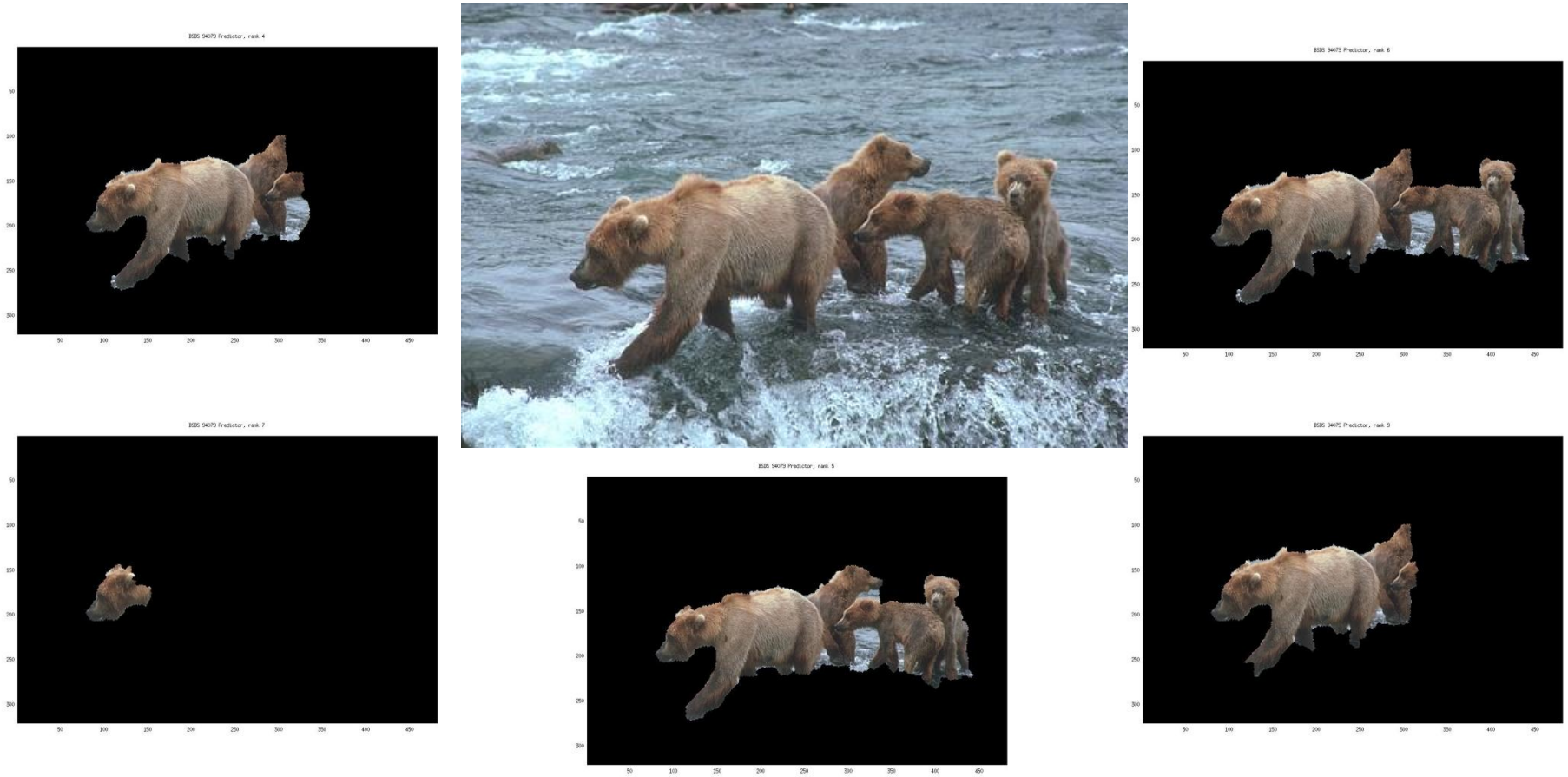
2. Voting from patches/keypoints



ISM model by Leibe et al.

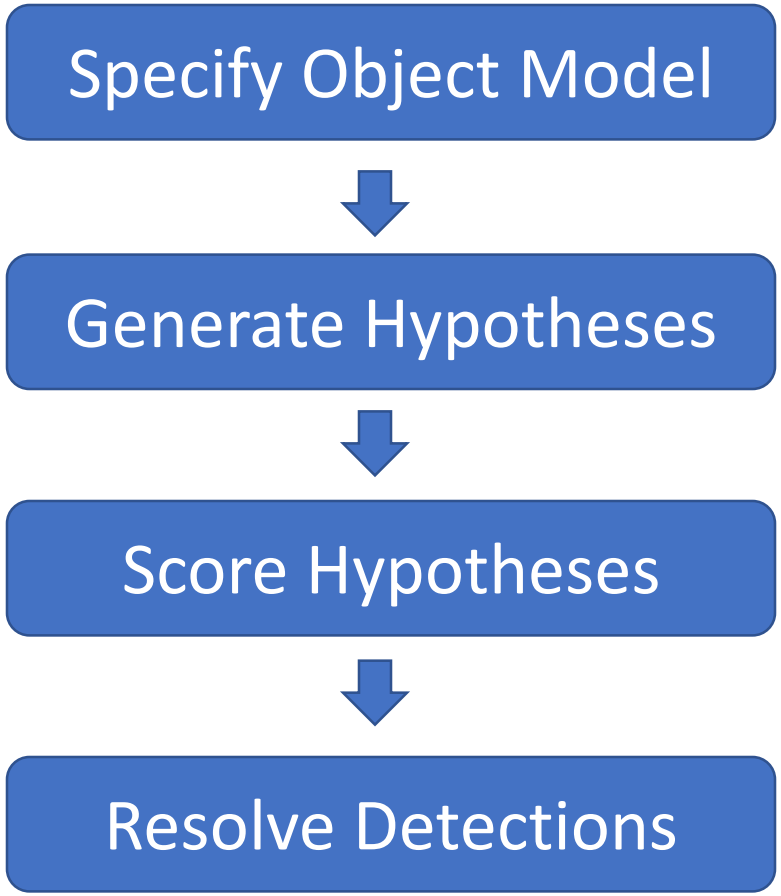
Generating hypotheses

3. Region-based proposal



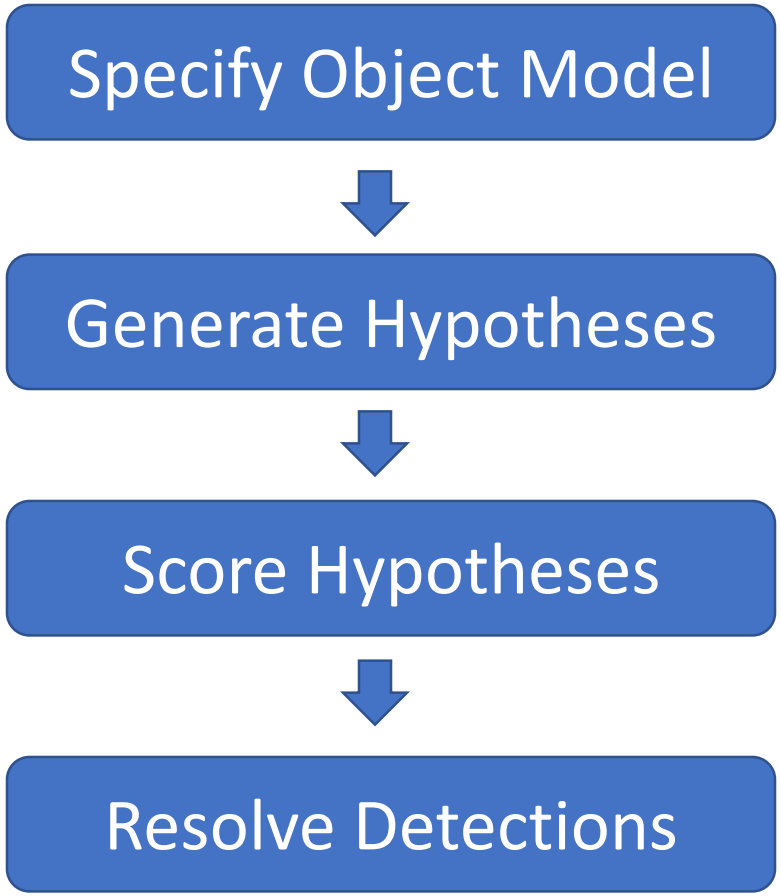
Endres Hoiem 2010

General Process of Object Recognition



Mainly-gradient based or CNN features, usually based on summary representation, many classifiers

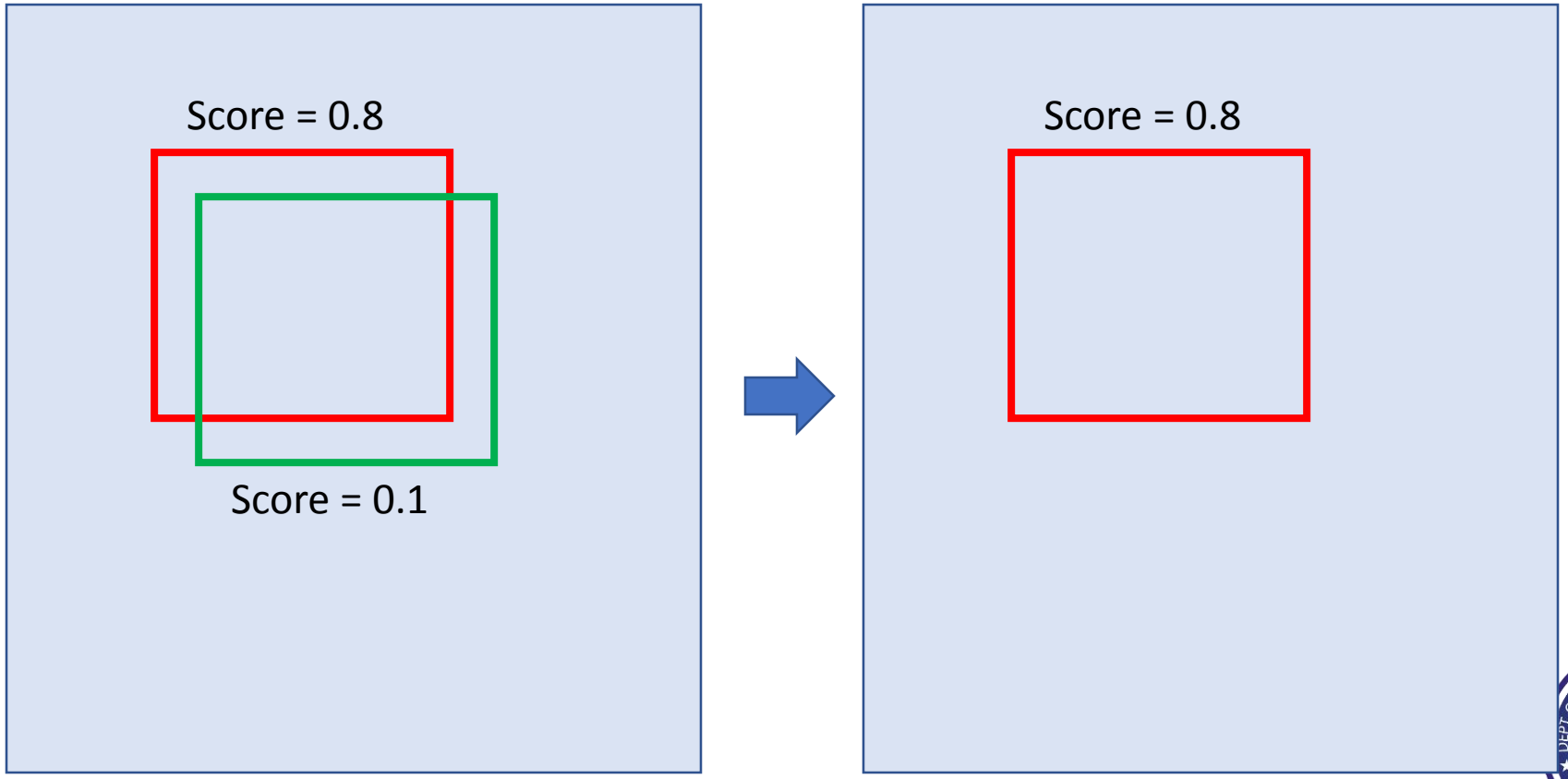
General Process of Object Recognition



Rescore each proposed object based on whole set

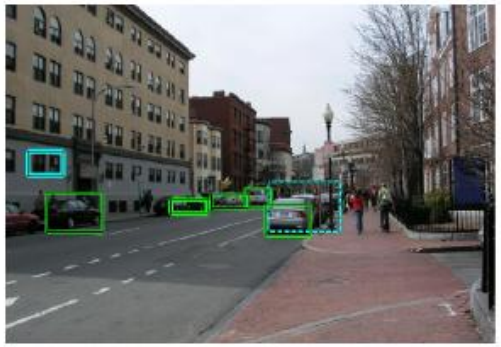
Resolving detection scores

1. Non-max suppression



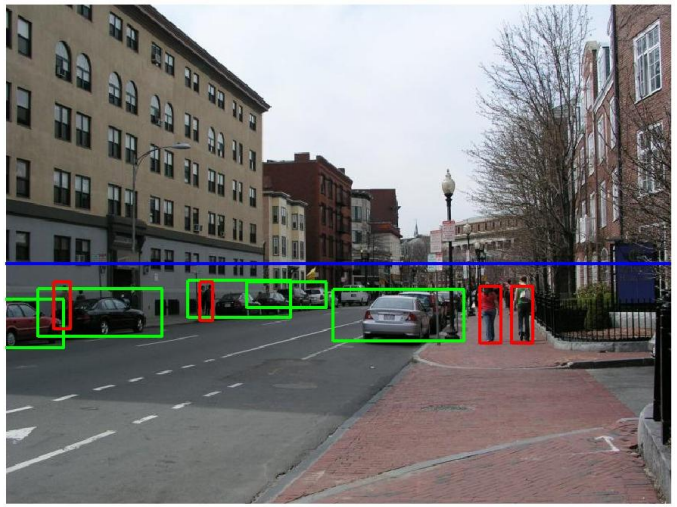
Resolving detection scores

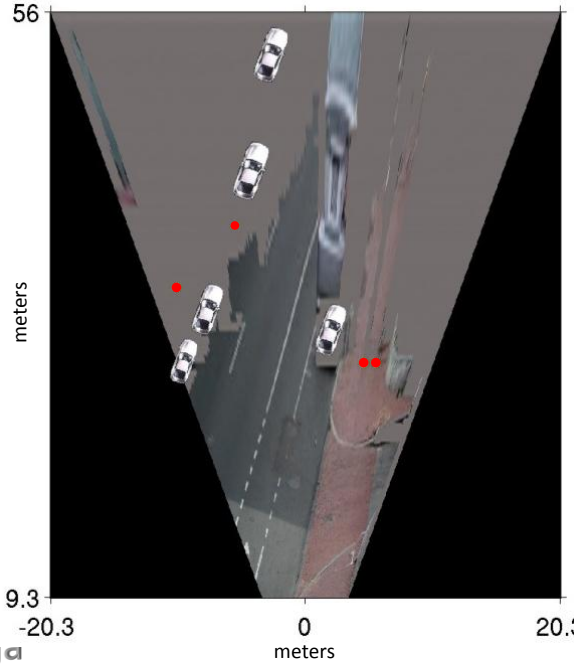
2. Context/reasoning



(g) Car Detections: Local

(h) Ped Detections: Local





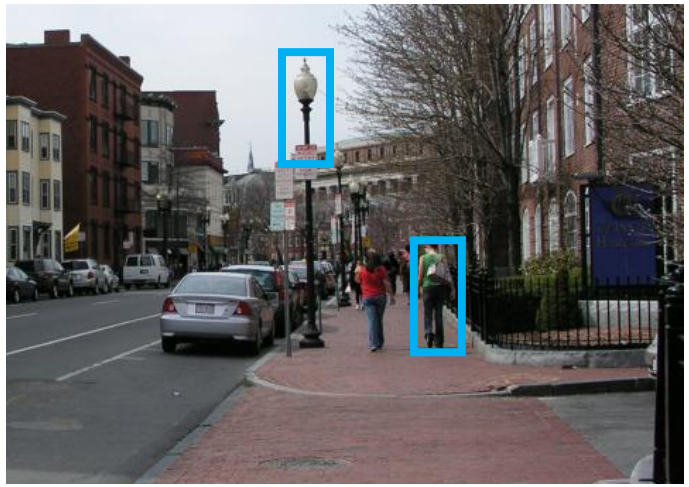
Hoiem et al. 2006

Object category detection in computer vision

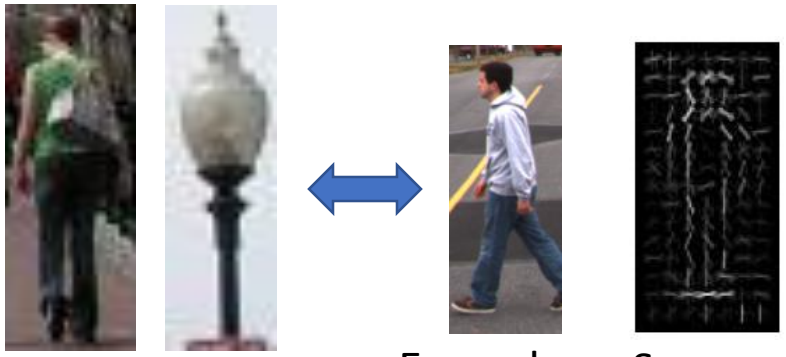
Goal: detect all pedestrians, cars, monkeys, etc in image

Basic Steps of Category Detection

1. Align
   - E.g., choose position, scale orientation
   - How to make this tractable?



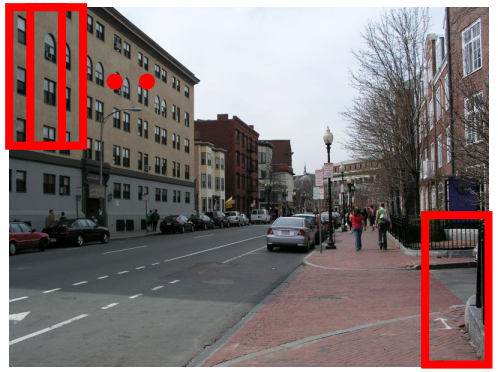
2. Compare
   - Compute similarity to an example object or to a summary representation
   - Which differences in appearance are important?



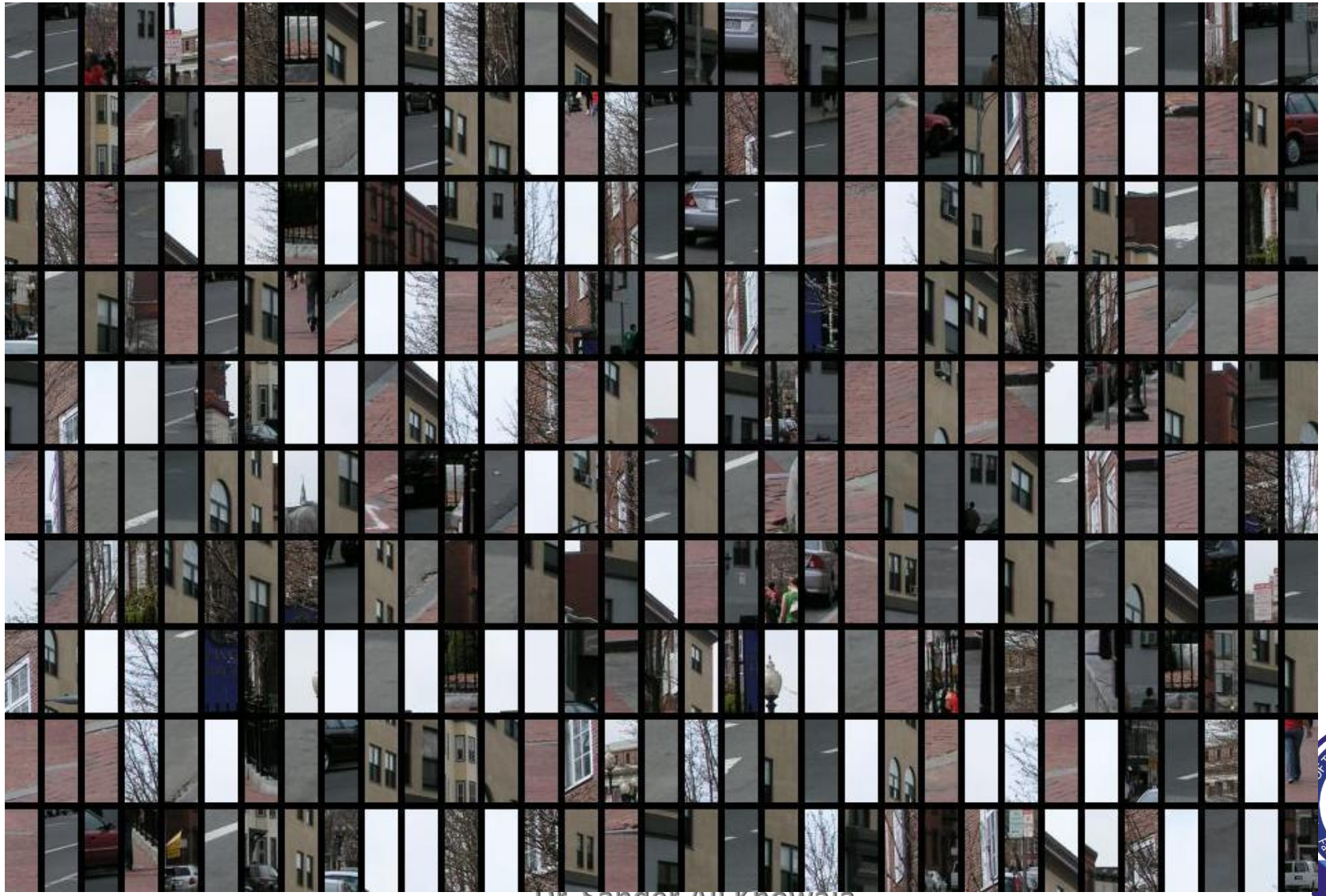
Aligned Possible Objects

Exemplar

Summary



Each window is separately classified

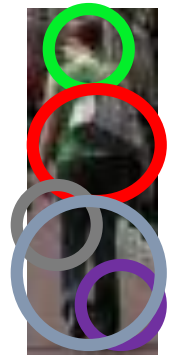


Statistical Template

- Object model = sum of scores of features at fixed positions

+3 +2 -2 -1 -2.5 = -0.5 ? > 7.5

Non-object



+4 +1 +0.5 +3 +0.5 = 10.5 ? > 7.5

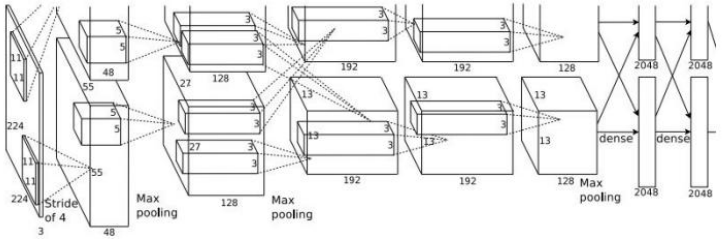
Object

CNN as feature extractor

Image credit: Justin Johnson

CNN as feature extractor

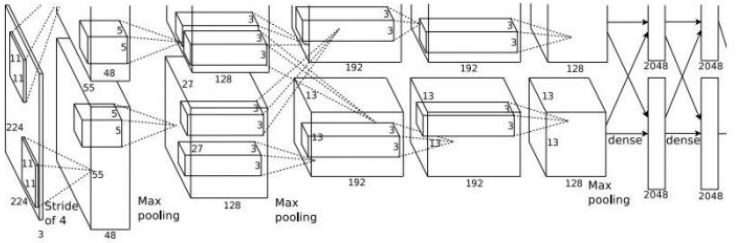
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? NO
Background? YES

CNN as feature extractor

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

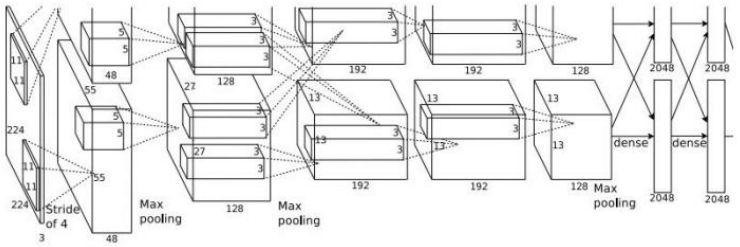


Dog? YES
Cat? NO
Background? NO

Slides by Justin Johnson

CNN as feature extractor

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

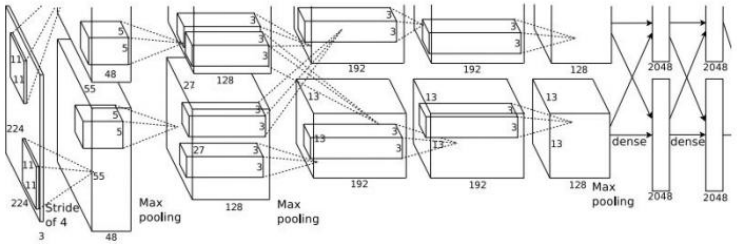


Dog? YES
Cat? NO
Background? NO

Slides by Justin Johnson

CNN as feature extractor

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? YES
Background? NO

Slides by Justin Johnson

CNN as feature extractor

- What could be the problems?

CNN as feature extractor

- What could be the problems?
  - Suppose we have a 600 x 600 image, if sliding window size is 20 x 20, then have (600-20+1) x (600-20+1) = ~330,000 windows

CNN as feature extractor

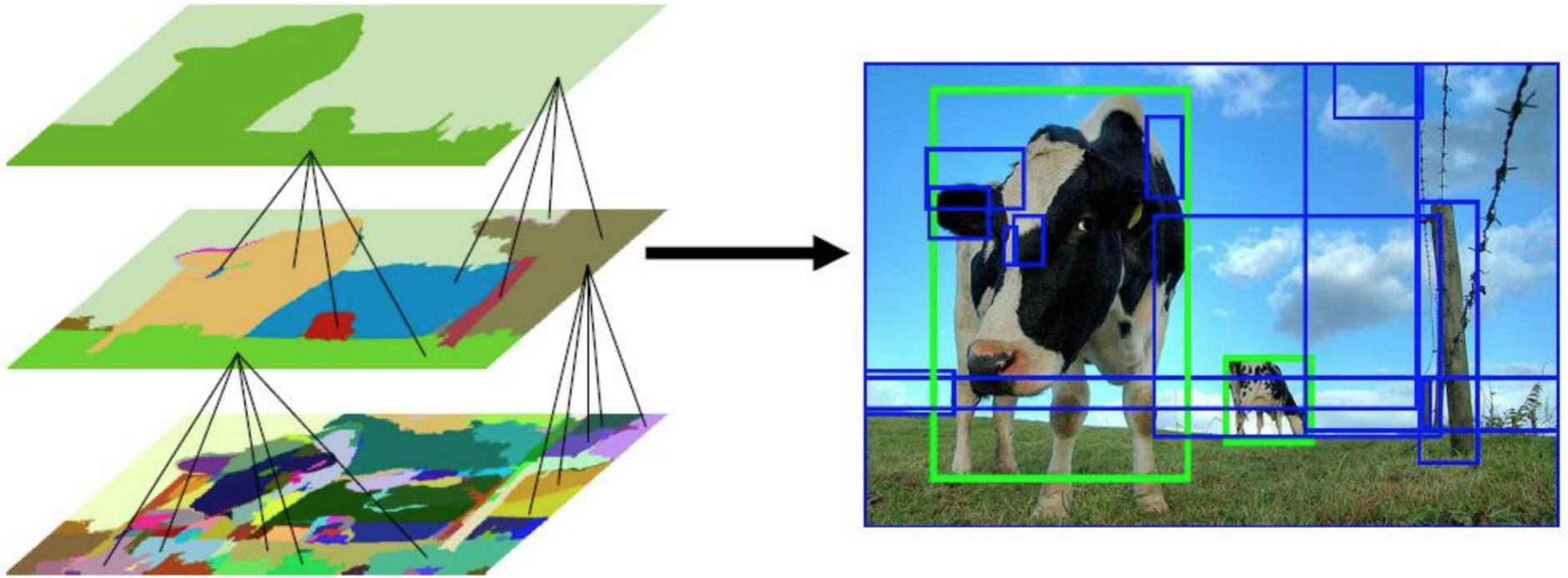
- What could be the problems?
  - Suppose we have a 600 x 600 image, if sliding window size is 20 x 20, then have (600-20+1) x (600-20+1) = ~330,000 windows
  - Sometimes we want to have more accurate results -> multi-scale detection
    - Resize image
    - Multi-scale sliding window

CNN as feature extractor

- What could be the problems?
  - Suppose we have a 600 x 600 image, if sliding window size is 20 x 20, then have (600-20+1) x (600-20+1) = ~330,000 windows
  - Sometimes we want to have more accurate results -> multi-scale detection
    - Resize image
    - Multi-scale sliding window
  - For each image, we need to do the forward pass in the CNN for ~330,000 times. -> Slow!!!

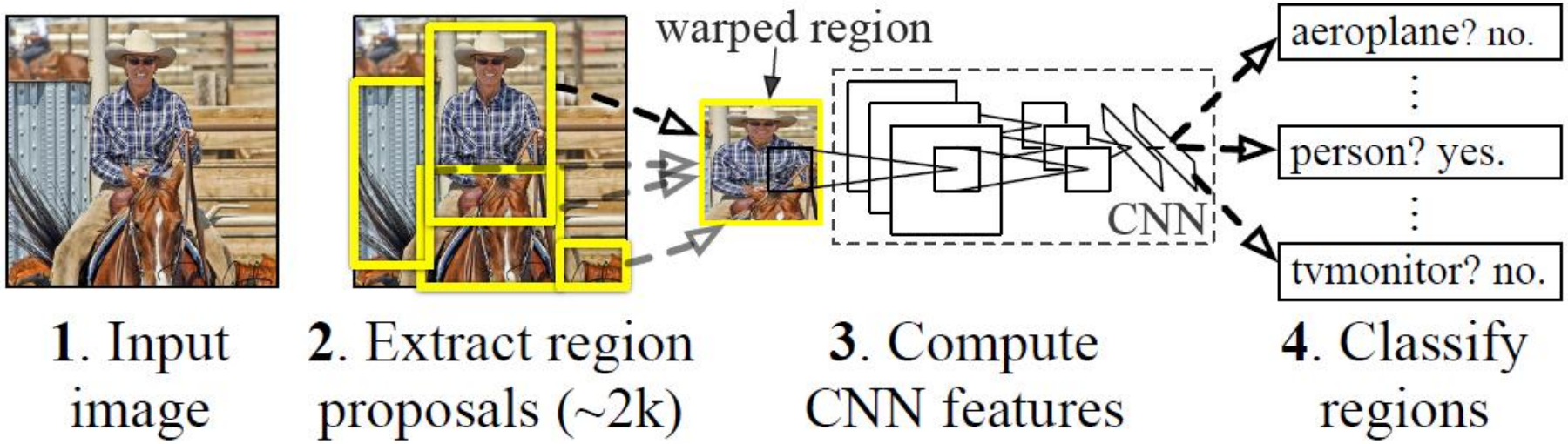
Region Proposal

- Solution
  - Use some fast algorithms to filter out some regions first, only feed the potential region (region proposals) into CNN
  - E.g. selective search



Uijilings et al. IJCV 2013

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, refine bounding box localization (bbox regression) simultaneously

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

Bounding Box Regression

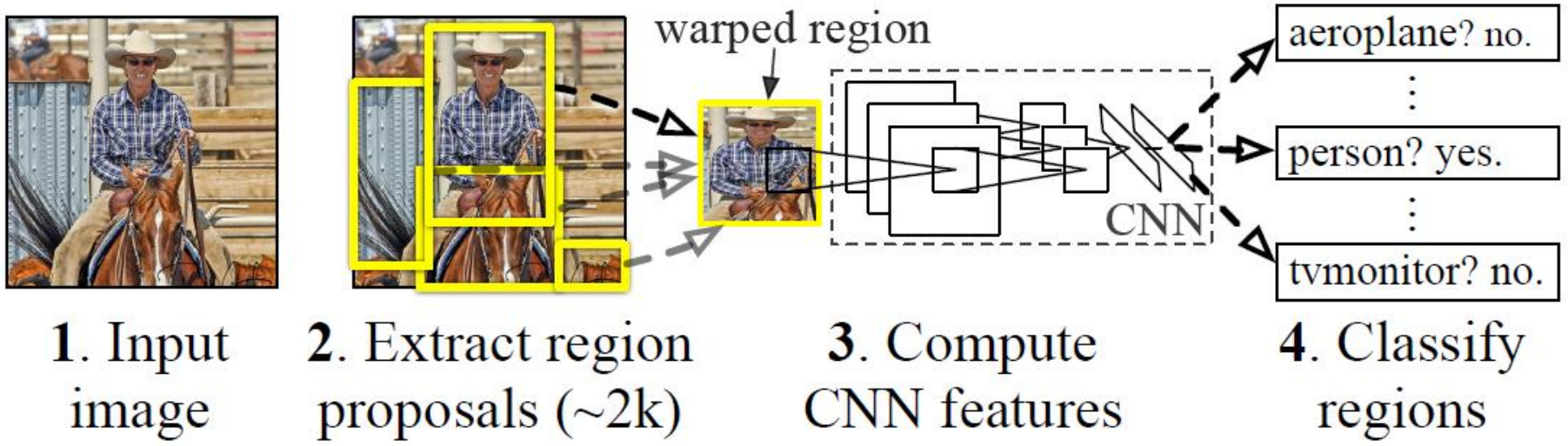
- Intuition
  - If you observe part of the object, according to the seen examples, you should be able to refine the localization
  - E.g. given the red box below, since you've seen many airplanes, you know this is not a good localization, you will adjust it to the green one

Bounding Box Regression

- Intuition
  - If you observe part of the object, according to the seen examples, you should be able to refine the localization
  - E.g. given the red box below, since you've seen many airplanes, you know this is not a good localization, you will adjust it to the green one

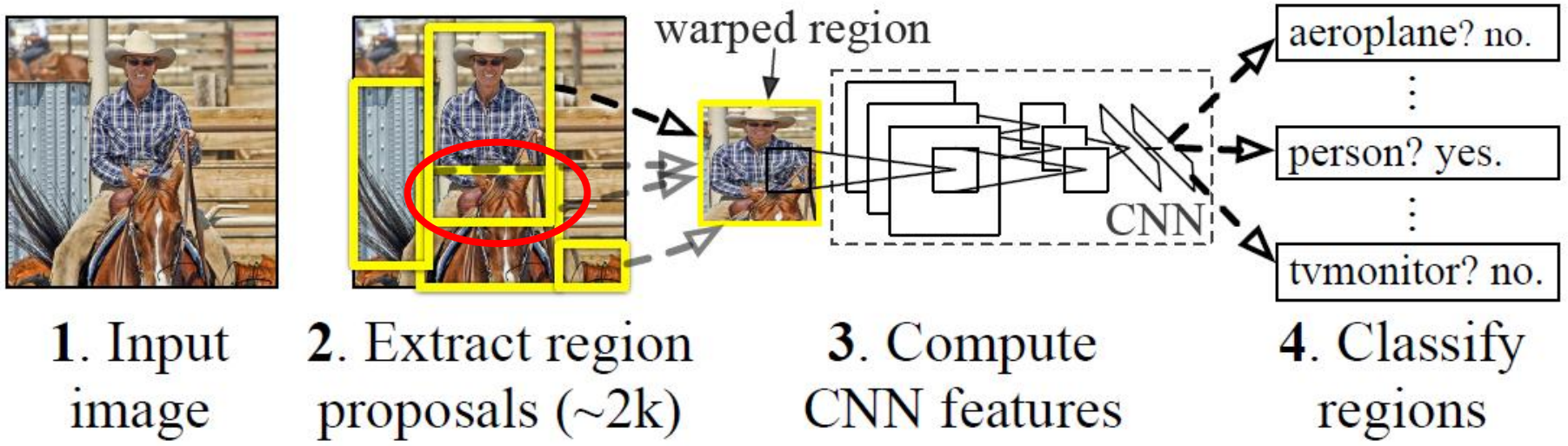
R-CNN (Girshick et al. CVPR 2014)

- What could be the problems?



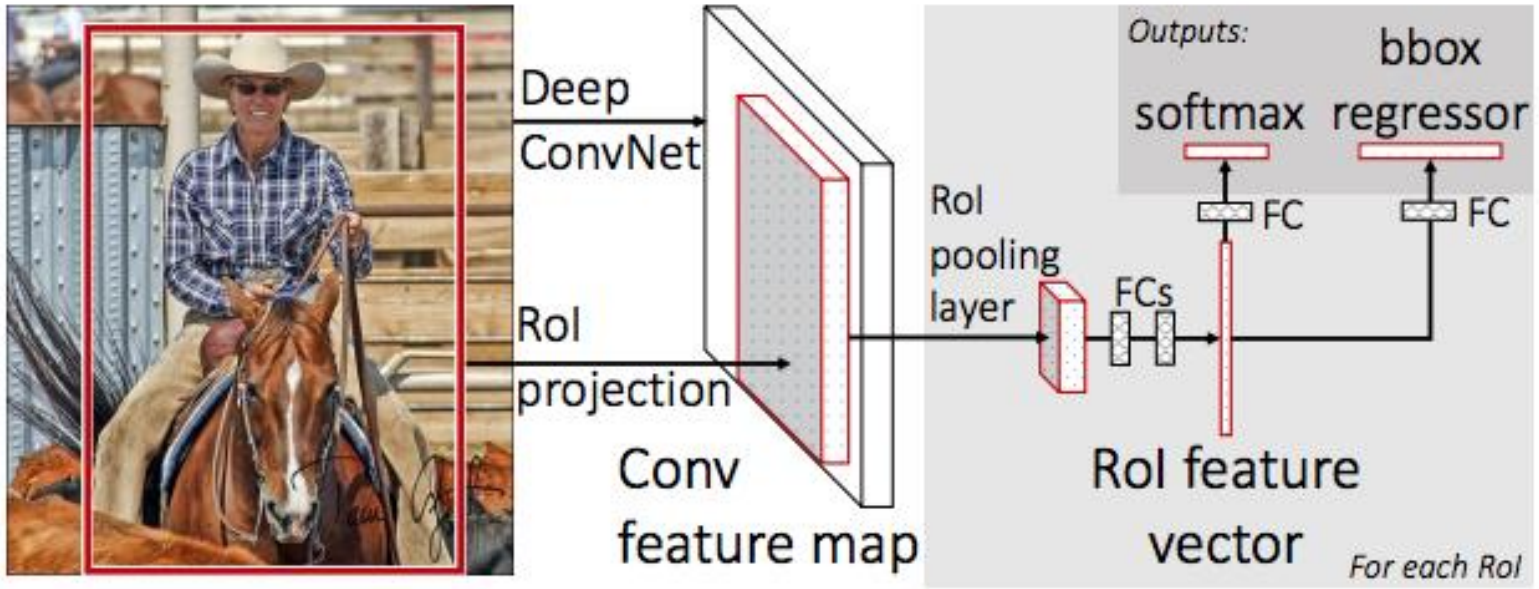
R-CNN (Girshick et al. CVPR 2014)

- What could be the problems?
  - Repetitive computation! For overlapping regions, we feed it multiple times into CNN



Fast R-CNN (Girshick ICCV 2015)

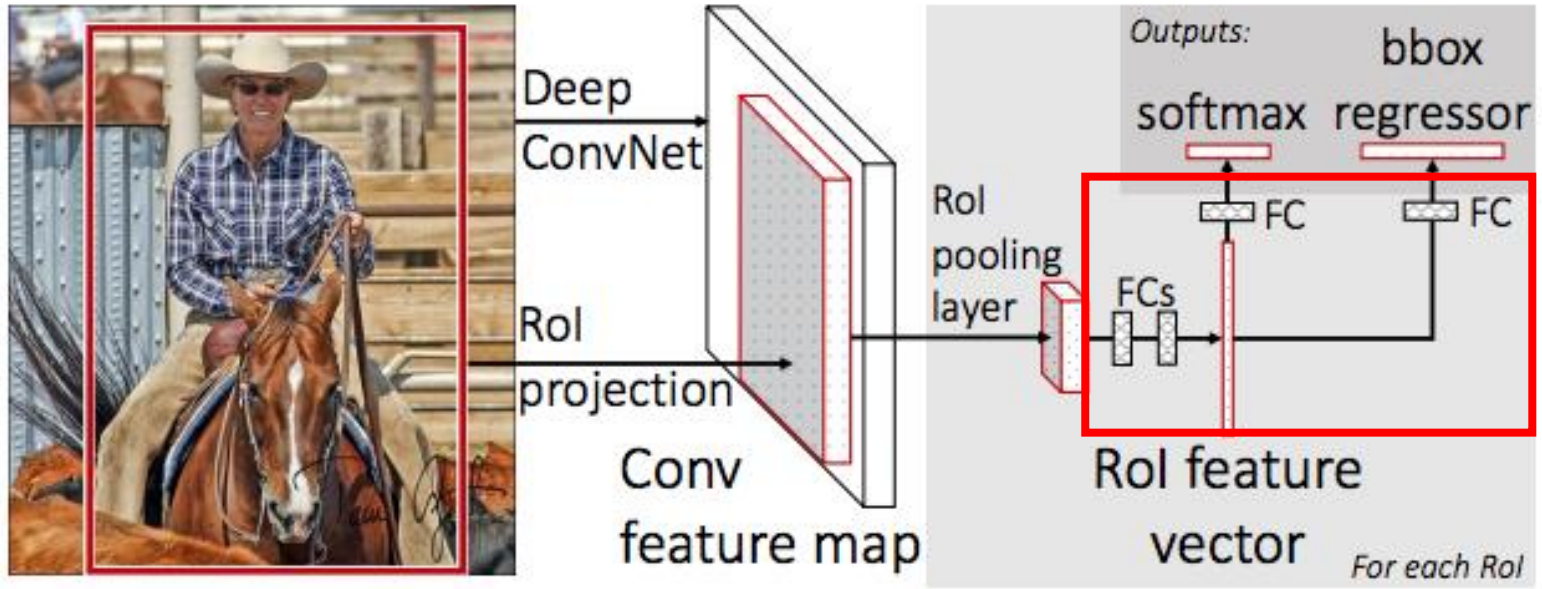
- Solution
  - Why not feed the whole image into CNN only once! Then crop features instead of image itself



https://arxiv.org/pdf/1504.08083.pdf

Fast R-CNN (Girshick ICCV 2015)

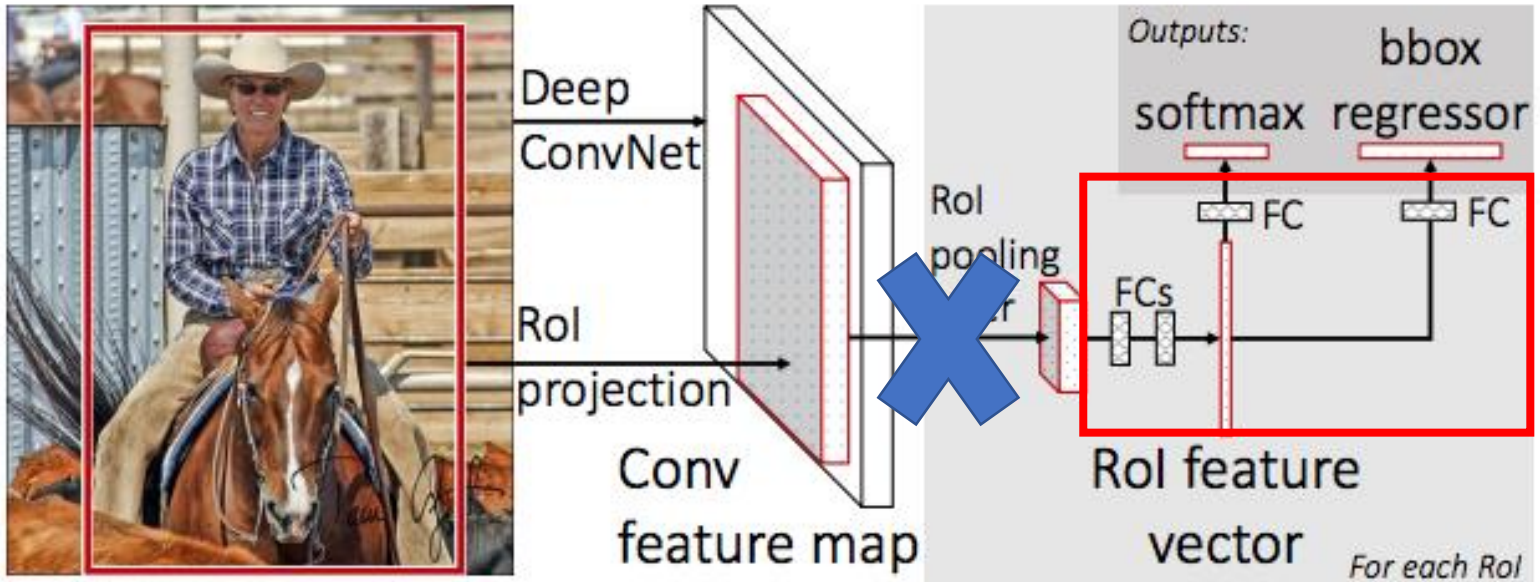
- How to crop features?
  - Since we have fully-connected layers, the size of feature map for each bounding box should be a fixed number



https://arxiv.org/pdf/1504.08083.pdf

Fast R-CNN (Girshick et al. ICCV 2015)

- How to crop features?
  - Since we have fully-connected layers, the size of feature map for each bounding box should be a fixed number
  - Resize/Interpolate the feature map as fixed size?
    - Not optimal. This operation is hard to backprop -> we cannot train the conv layers for this problem



https://arxiv.org/pdf/1504.08083.pdf

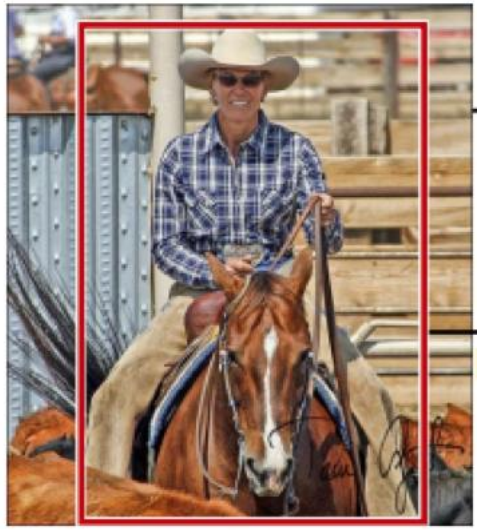
Fast R-CNN (Girshick et al. ICCV 2015)

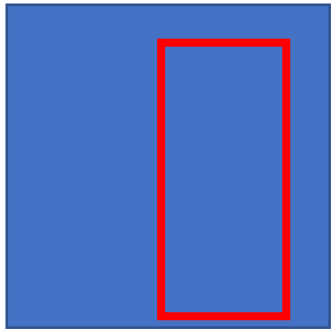
- How to crop features?
  - Since we have fully-connected layers, the size of feature map for each bounding box should be a fixed number
  - Resize/Interpolate the feature map as fixed size?
    - Not optimal. This operation is hard to backprop -> we cannot train the conv layers for this problem
  - RoI (Region of Interest) Pooling

https://arxiv.org/pdf/1504.08083.pdf

RoI Pooling

- Step 1: Get bounding box for feature map from bounding box for image
  - Due the (down)convolution / pooling operations, feature map would have a smaller size than the original image





Feature map

https://arxiv.org/pdf/1504.08083.pdf

RoI Pooling

- Step 2: Divide cropped feature map into fixed number of sub-regions
  - The last column and last row might be smaller

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Feature map
4 x 4 x 1

Make it as 2x2

0	1	2
4	5	6
8	9	10

1	2	3
5	6	7
9	10	11
13	14	15

https://arxiv.org/pdf/1504.08083.pdf

RoI Pooling

- Step 3: For each sub-region, perform max pooling (pick the max one)

0	1	2
4	5	6
8	9	10

Max pooling

5	6
9	10

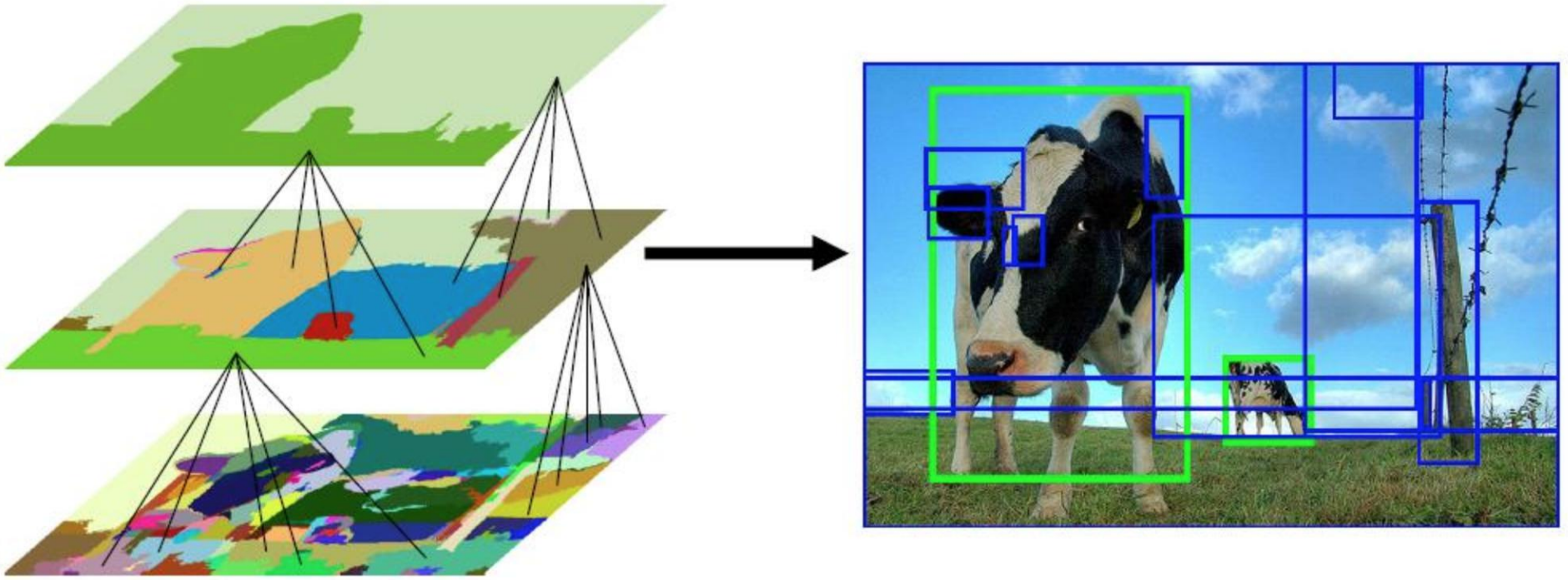
https://arxiv.org/pdf/1504.08083.pdf

Fast R-CNN (Girshick et al. ICCV 2015)

- What could be the problems?

Fast R-CNN (Girshick et al. ICCV 2015)

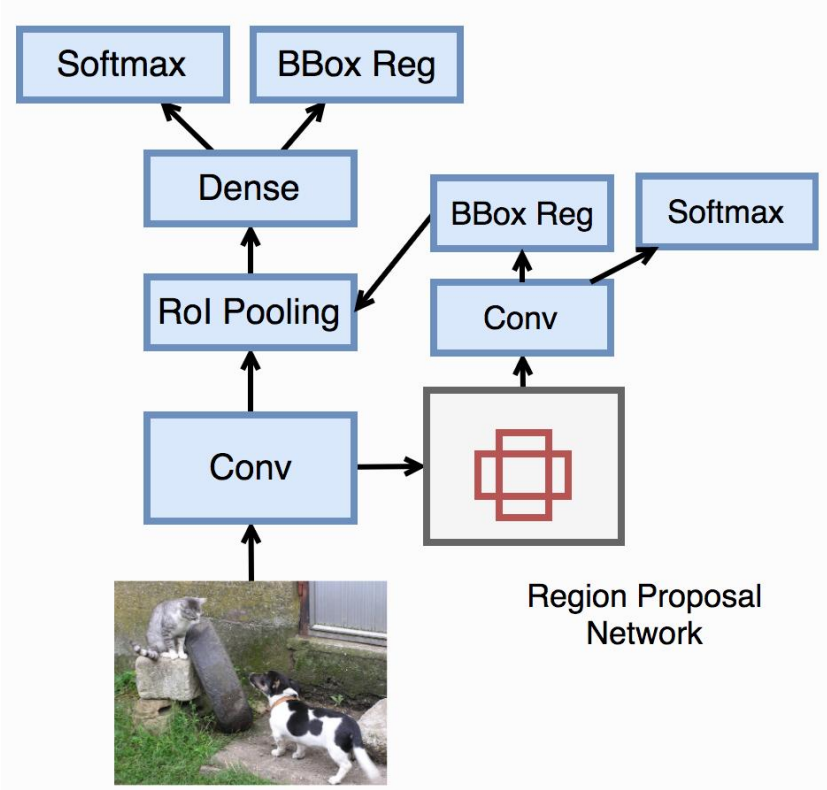
- What could be the problems?
  - Why we need the region proposal pre-processing step? That's not "deep learning" at all. Not cool!



Uijilings et al. IJCV 2013

Faster R-CNN (Ren et al. NIPS 2015)

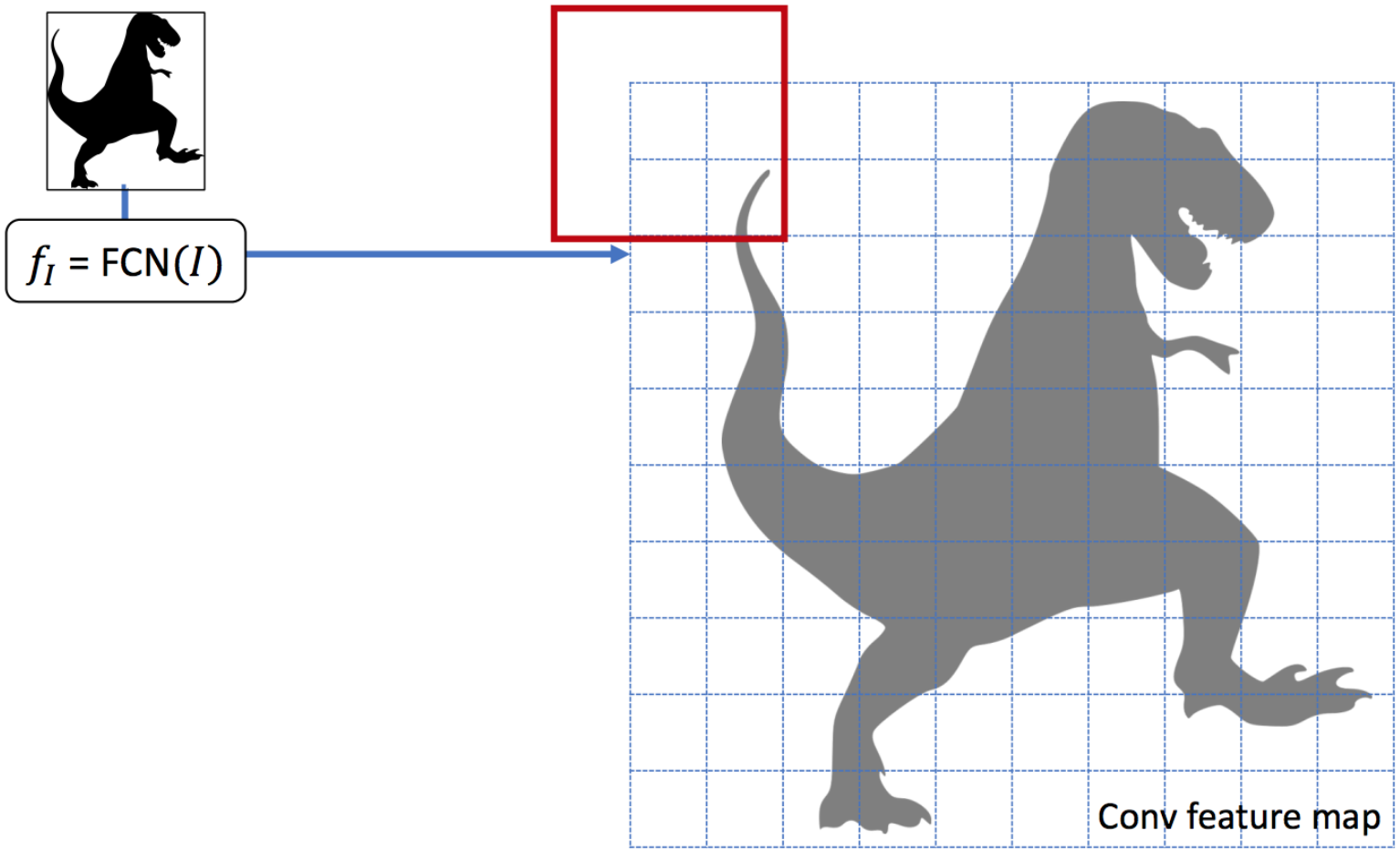
- Solution
  - Why not generate region proposals using CNN??! -> RPN



https://arxiv.org/pdf/1506.01497.pdf

Image credit: http://zh.gluon.ai/chapter_computer-vision/object-detection.html

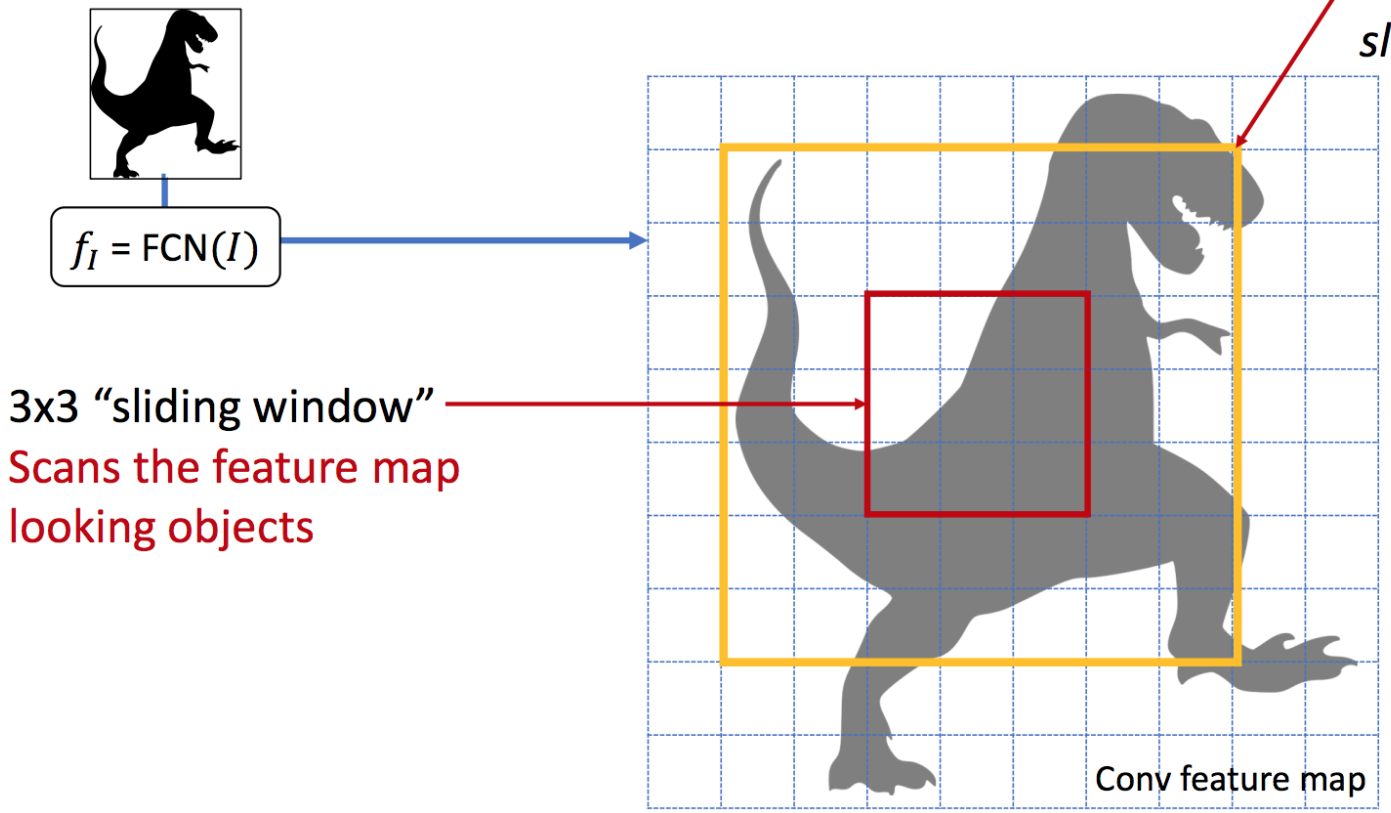
RPN: Region Proposal Network



https://arxiv.org/pdf/1506.01497.pdf

Slides by Ross Girshick

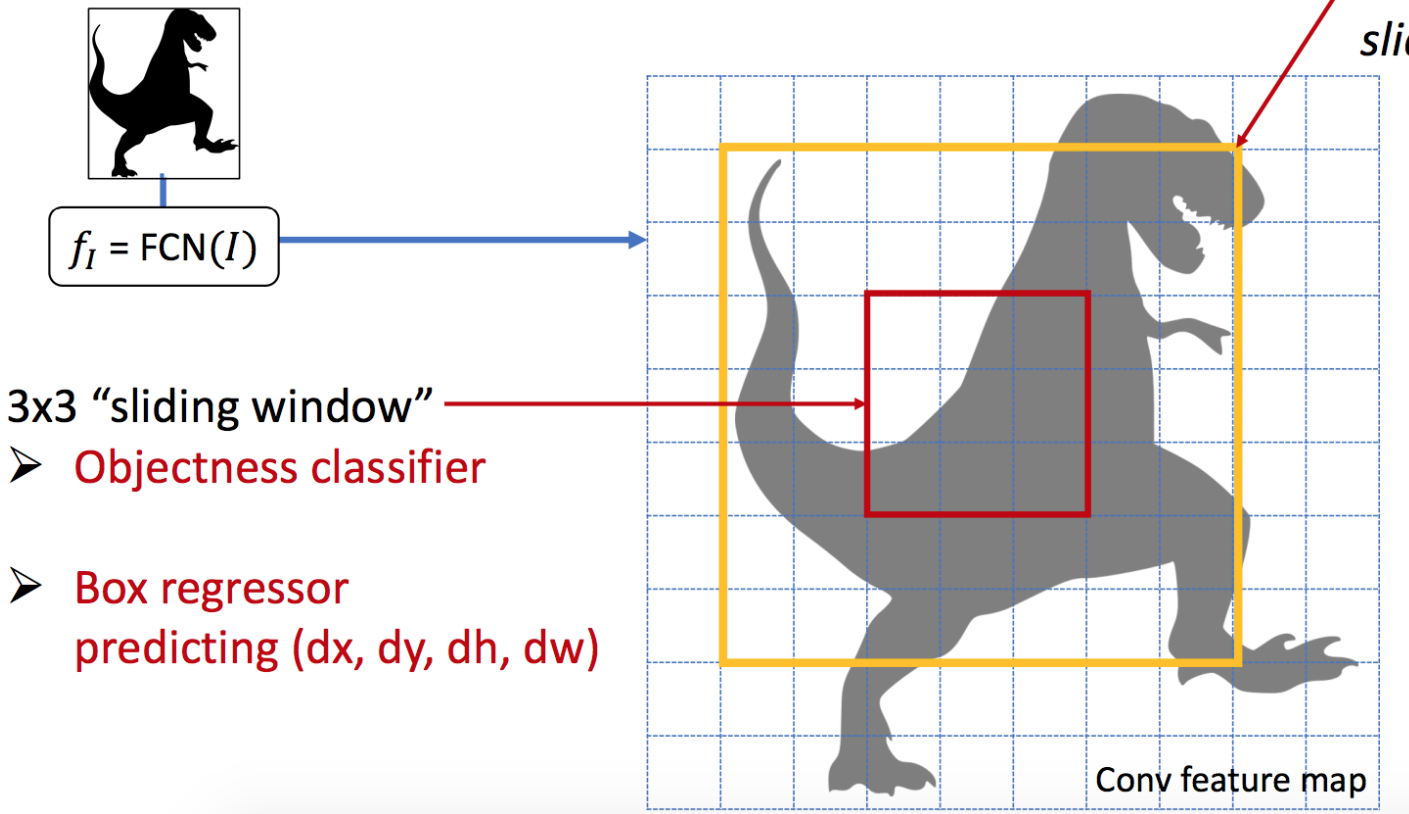
RPN: Anchor Box



https://arxiv.org/pdf/1506.01497.pdf

Slides by Ross Girshick

RPN: Anchor Box

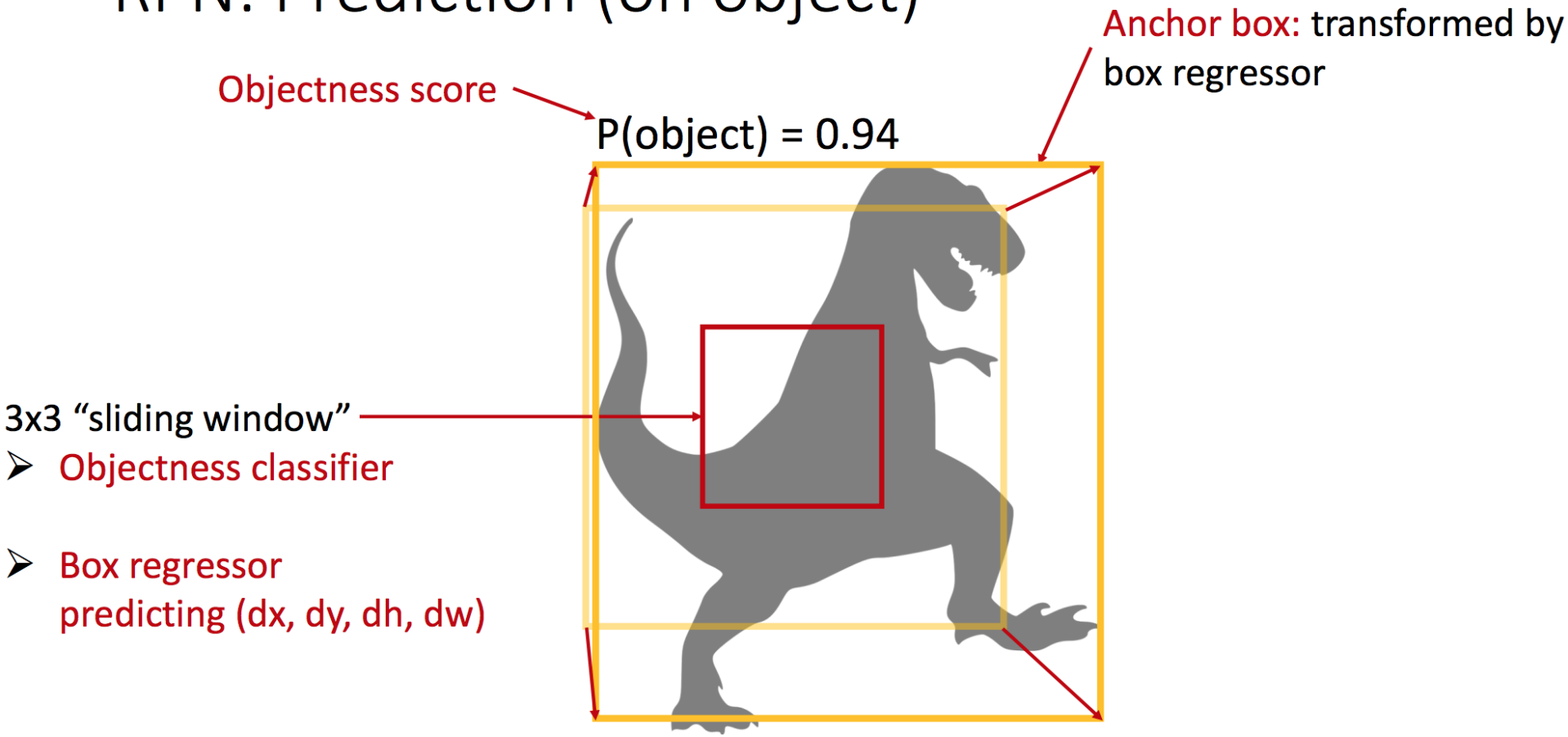


- Objectness classifier
- Box regressor predicting (dx, dy, dh, dw)

https://arxiv.org/pdf/1506.01497.pdf

Slides by Ross Girshick

RPN: Prediction (on object)

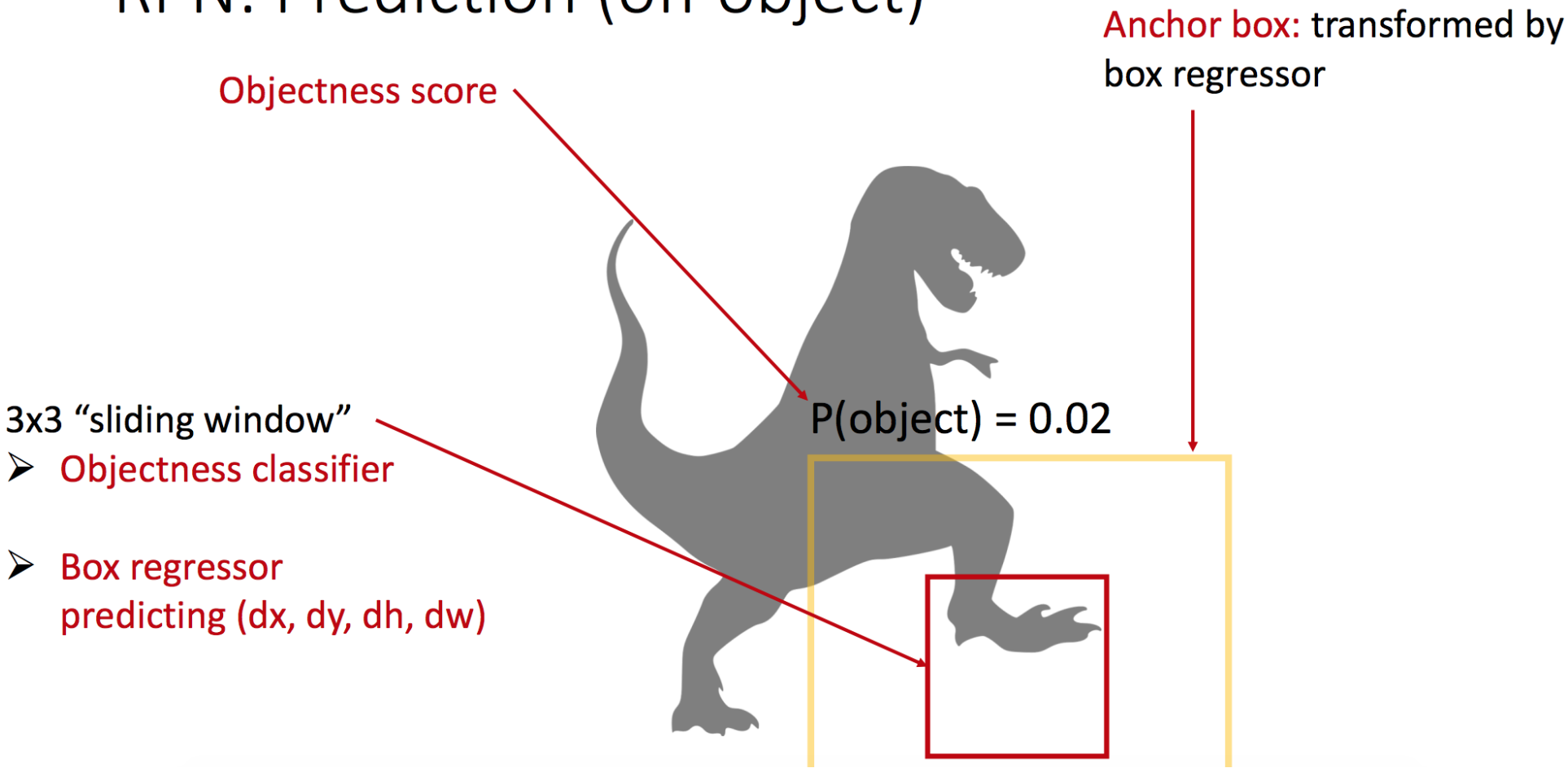


https://arxiv.org/pdf/1506.01497.pdf

Slides by Ross Girshick

Faster R-CNN (Ren et al. NIPS 2015)

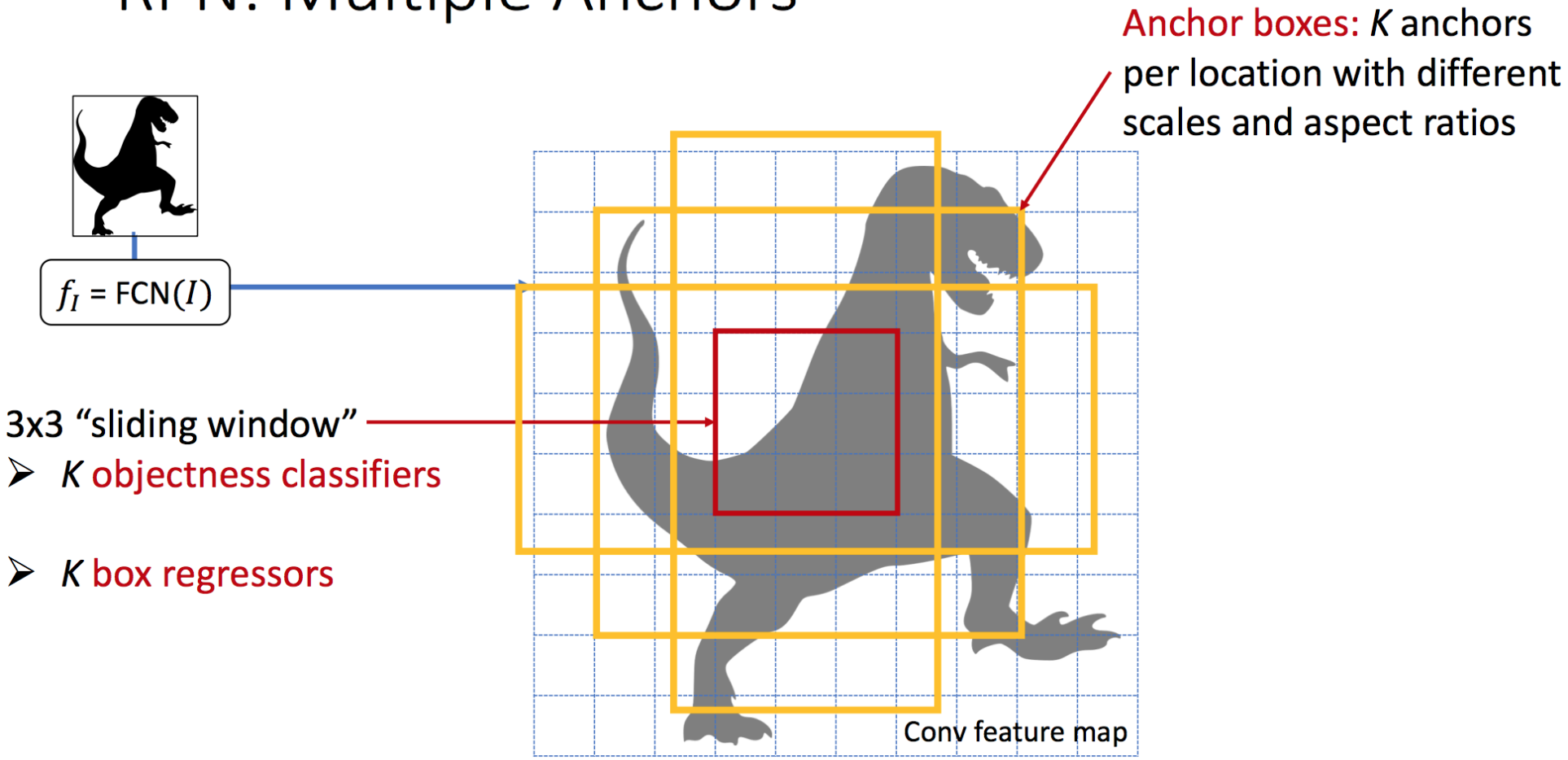
RPN: Prediction (off object)



https://arxiv.org/pdf/1506.01497.pdf

Slides by Ross Girshick

RPN: Multiple Anchors

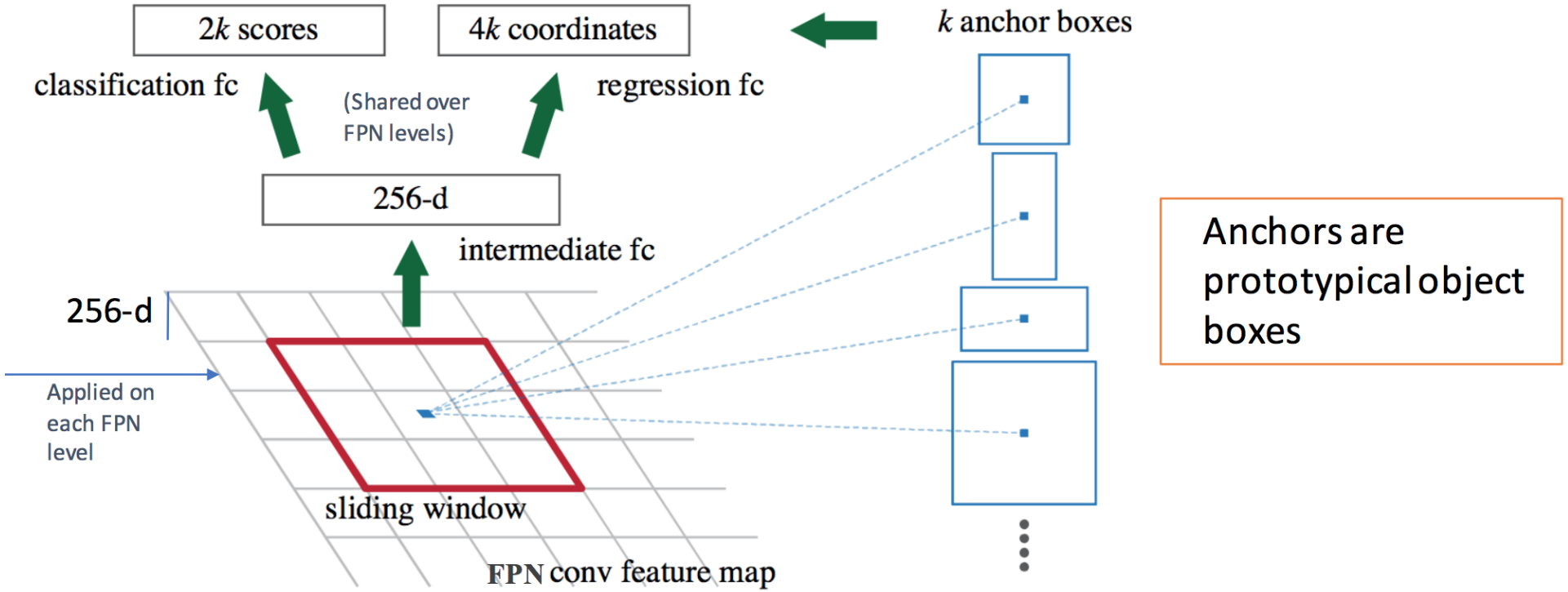


https://arxiv.org/pdf/1506.01497.pdf

Slides by Ross Girshick

Faster R-CNN (Ren et al. NIPS 2015)

- Solution
  - Why not generate region proposals using CNN??!



Anchors are prototypical object boxes

https://arxiv.org/pdf/1506.01497.pdf

Slides by Ross Girshick

Faster R-CNN (Ren et al. NIPS 2015)

- What could be the problems

https://arxiv.org/pdf/1506.01497.pdf

Faster R-CNN (Ren et al. NIPS 2015)

- What could be the problems
  - Two-stage detection pipeline is still too slow to apply on real-time videos

https://arxiv.org/pdf/1506.01497.pdf

One-stage detection

- Solution
  - Don’t generate object proposals!
  - Consider a tiny subset of the output space by design; directly classify this small set of boxes

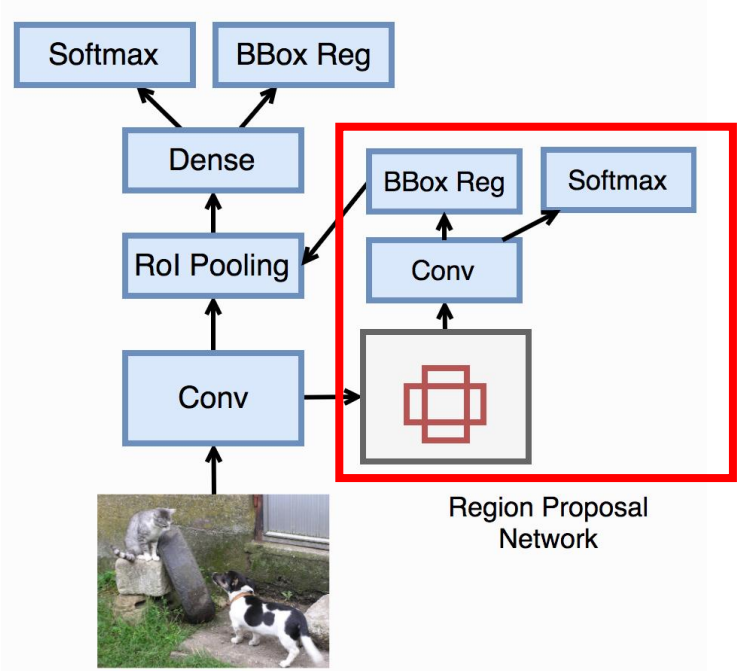


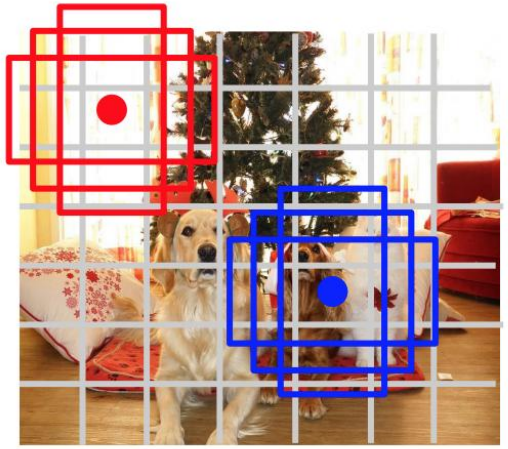
Image credit: http://zh.gluon.ai/chapter_computer-vision/object-detection.html

One-stage detection

- Solution

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

Input image
3 x H x W



Divide image into grid
7 x 7

Image a set of **base boxes** centered at each grid cell
Here B = 3

Within each grid cell:

- Regress from each of the B base boxes to a final box with 5 numbers: (dx, dy, dh, dw, confidence)
- Predict scores for each of C classes (including background as a class)

Output:
7 x 7 x (5 * B + C)

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

One-stage detection

- What could be the problems?

One-stage detection

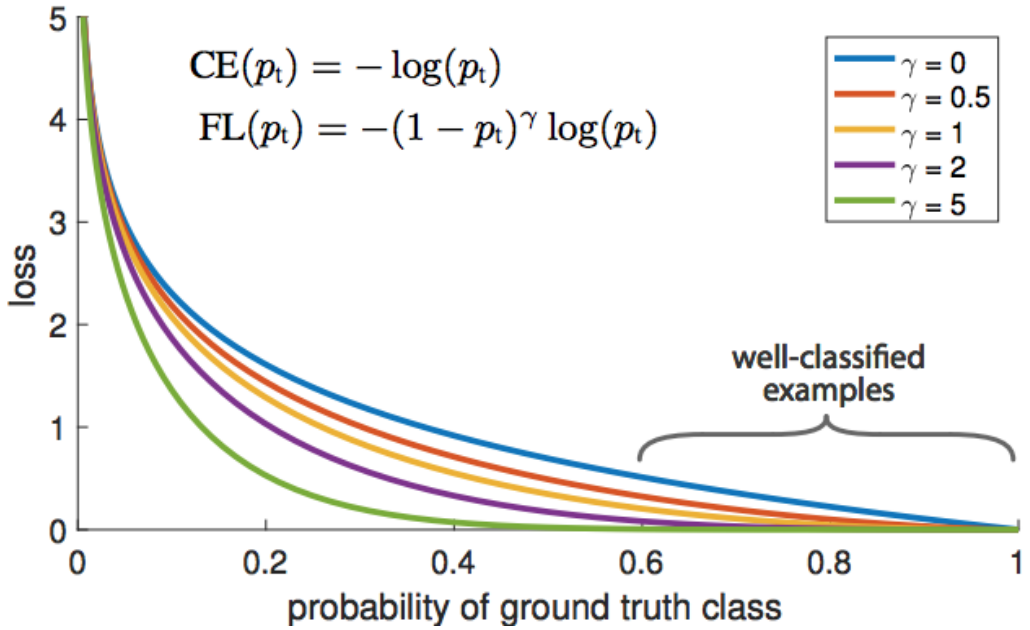
- What could be the problems?
  - The extreme foreground-background class imbalance -> we have a lot more negative examples.

One-stage detection

- What could be the problems?
  - The extreme foreground-background class imbalance -> we have a lot more negative examples.
  - Even though they have small loss values, the gradients overwhelm the model

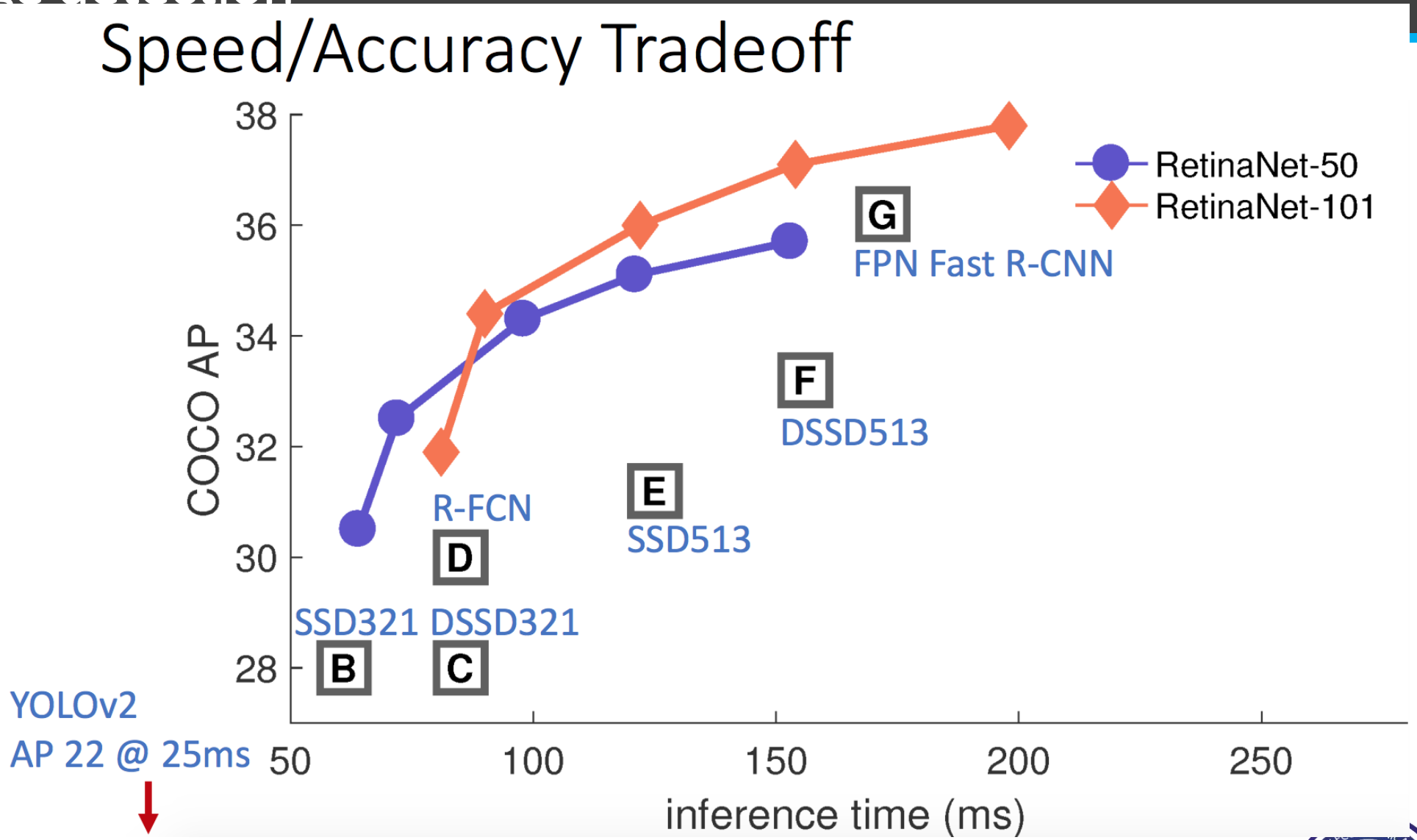
Focal Loss for Dense Object Detection (Lin et al. ICCV 2017)

- Solution
  - For easy examples, we down-weight it loss, so that the gradients from these example have smaller impact to the model

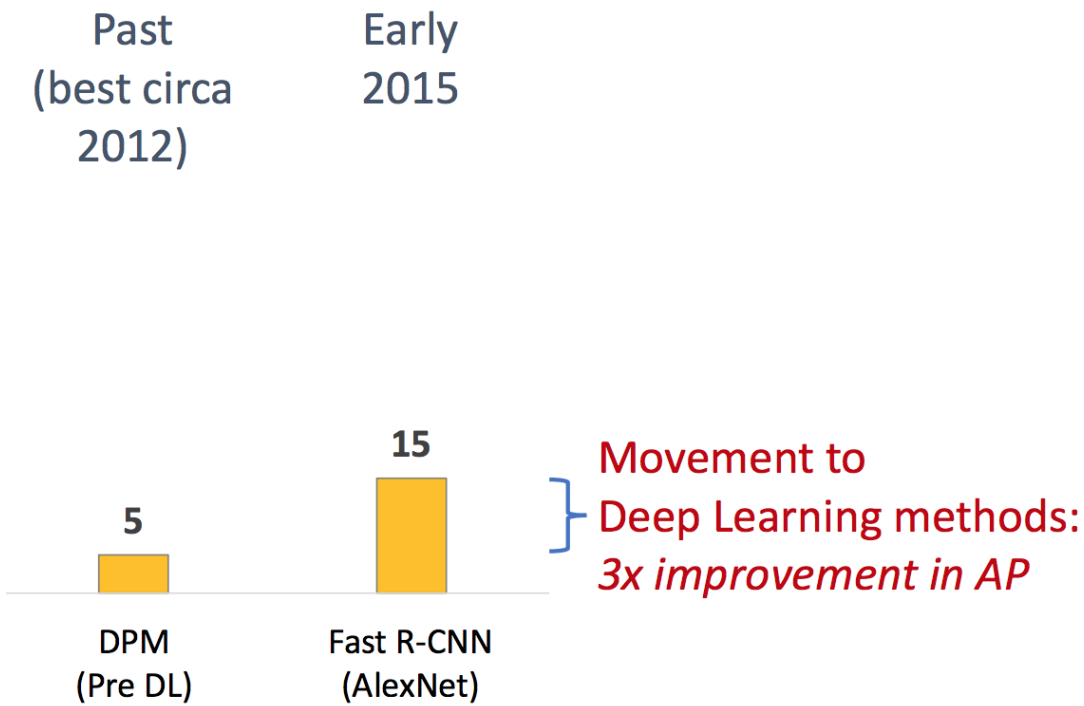


https://arxiv.org/pdf/1708.02002.pdf

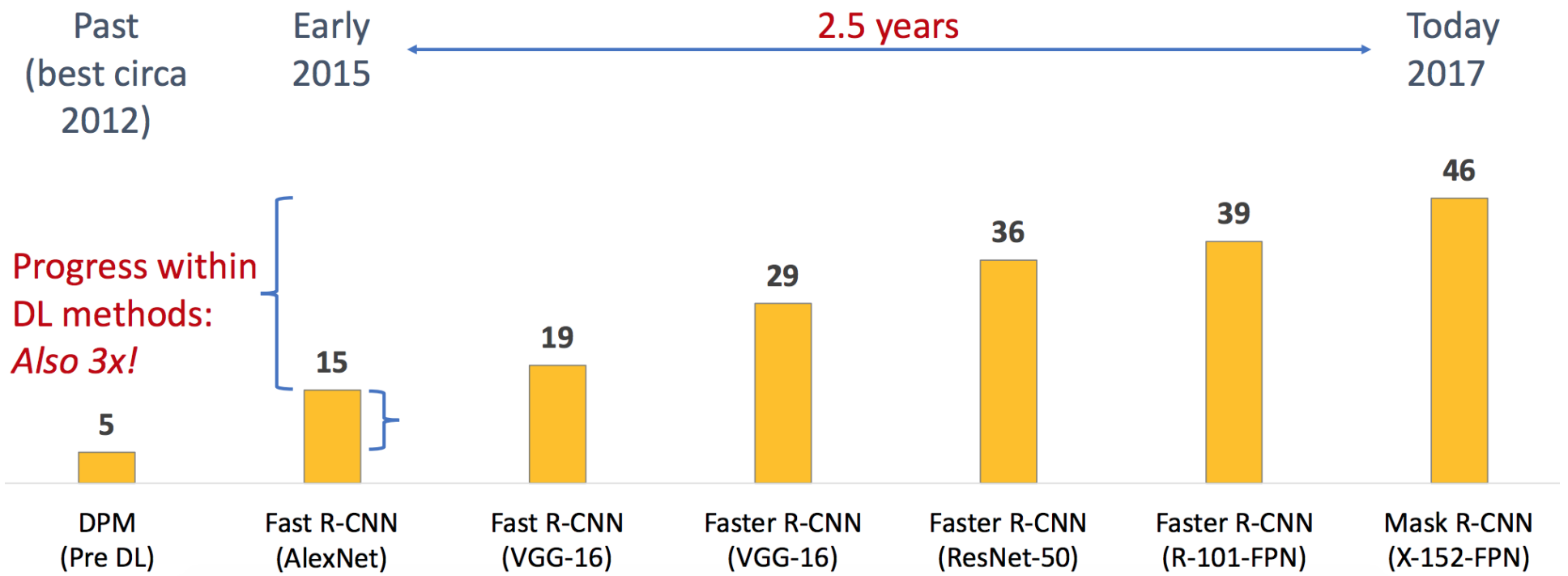
One-stage detection



COCO Object Detection Average Precision (%)

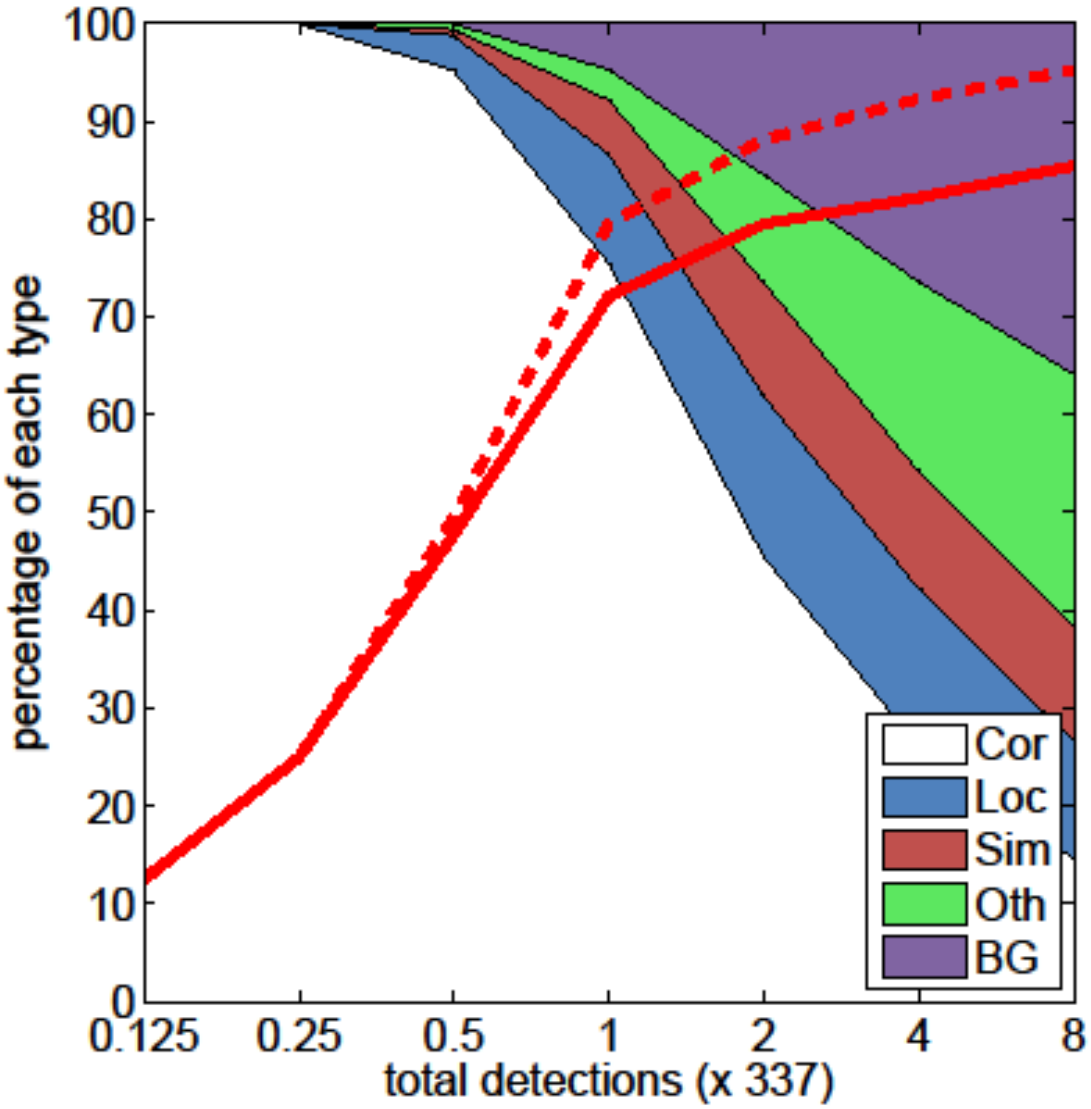


COCO Object Detection Average Precision (%)



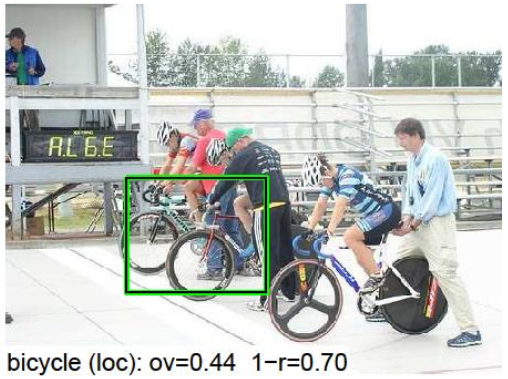
Mistakes are often reasonable

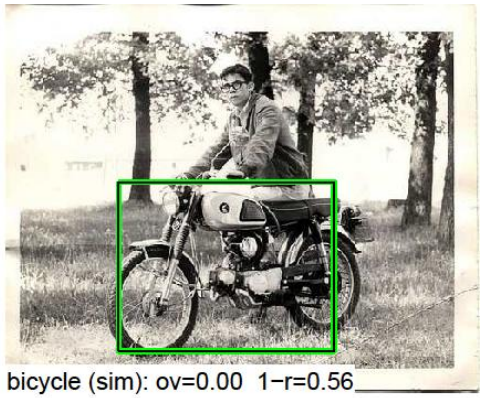
Bicycle: AP = 0.73



R-CNN results

Confident Mistakes



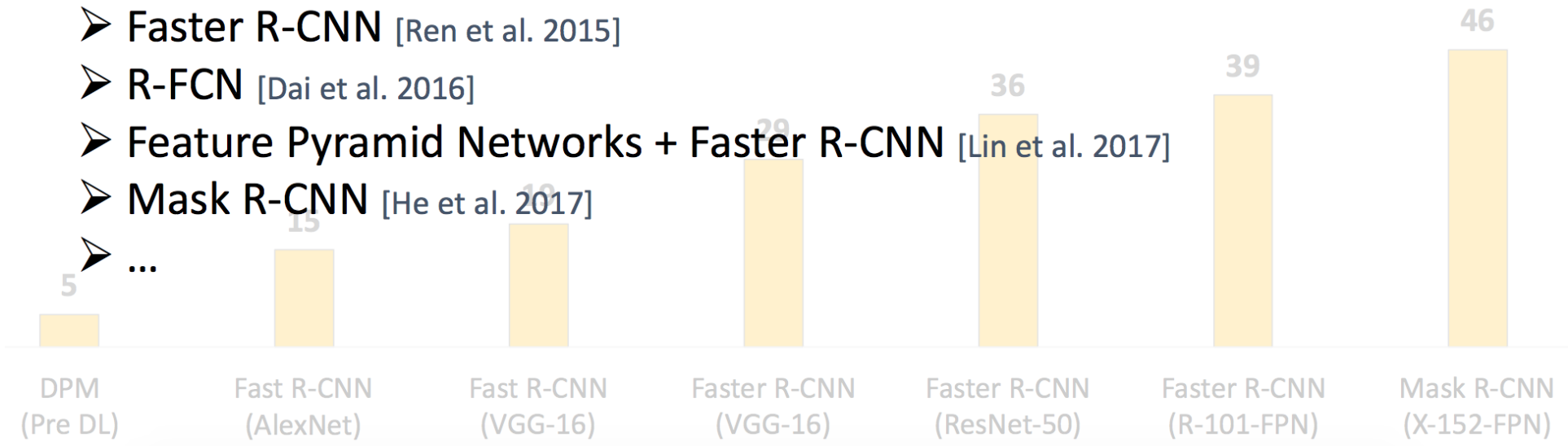


Influential Works in Detection

- Sung-Poggio (1994, 1998) : ~2100 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) : ~4200
  - "Parts" at fixed position, non-maxima suppression, simple cascade, rotation, pretty good accuracy, fast
- Schneiderman-Kanade (1998-2000,2004) : ~2250
  - Careful feature/classifier engineering, excellent results, cascade
- Viola-Jones (2001, 2004) : ~20,000
  - Haar-like features, Adaboost as feature selection, hyper-cascade, very fast, easy to implement
- Dalal-Triggs (2005) : ~11000
  - Careful feature engineering, excellent results, HOG feature, online code
- Felzenszwalb-Huttenlocher (2000): ~1600
  - Efficient way to solve part-based detectors
- Felzenszwalb-McAllester-Ramanan (2008,2010)? ~4000
  - Excellent template/parts-based blend

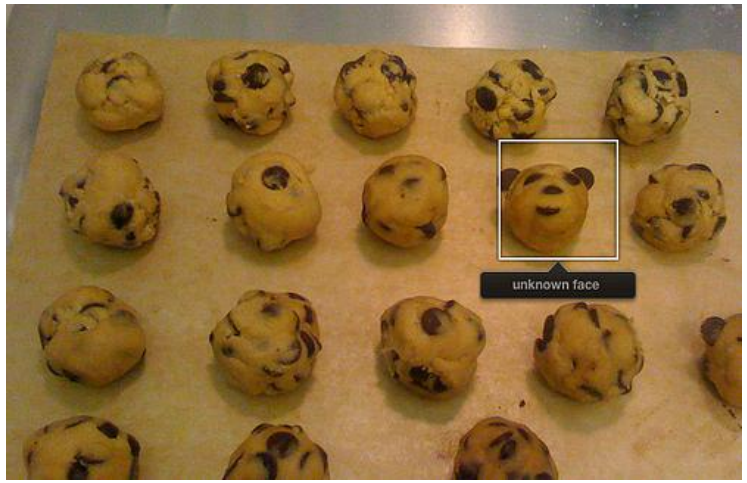
Influential Works in Detection

- R-CNN [Girshick et al. 2014]
- SPP-net [He et al. 2014]
- Fast R-CNN [Girshick. 2015]
- Faster R-CNN [Ren et al. 2015]
- R-FCN [Dai et al. 2016]
- Feature Pyramid Networks + Faster R-CNN [Lin et al. 2017]
- Mask R-CNN [He et al. 2017]
- …



Fails in commercial face detection

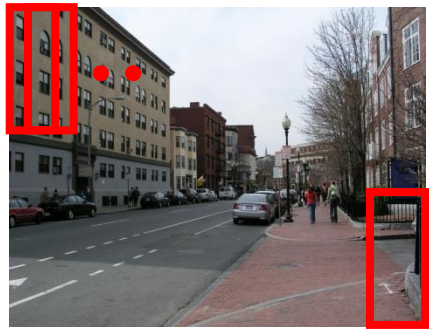
- Things iPhoto thinks are faces



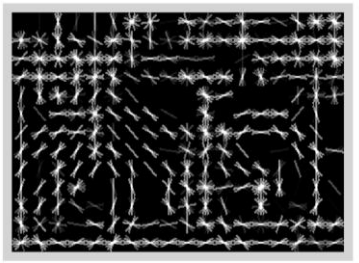
http://www.oddee.com/item_98248.aspx

Summary: statistical templates

Propose Window → Extract Features → Classify → Post-process



Sliding window: scan image pyramid



HOG

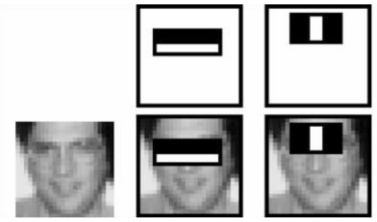
SVM

Boosted stubs

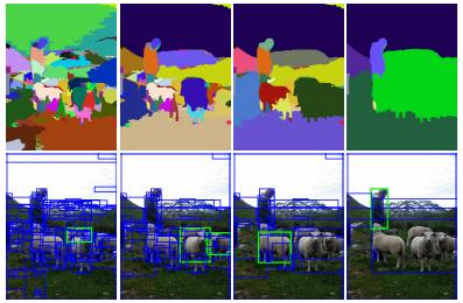
Neural network

Non-max suppression

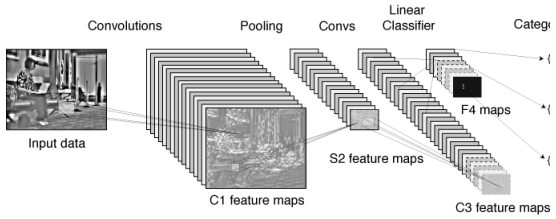
Segment or refine localization



Fast randomized features

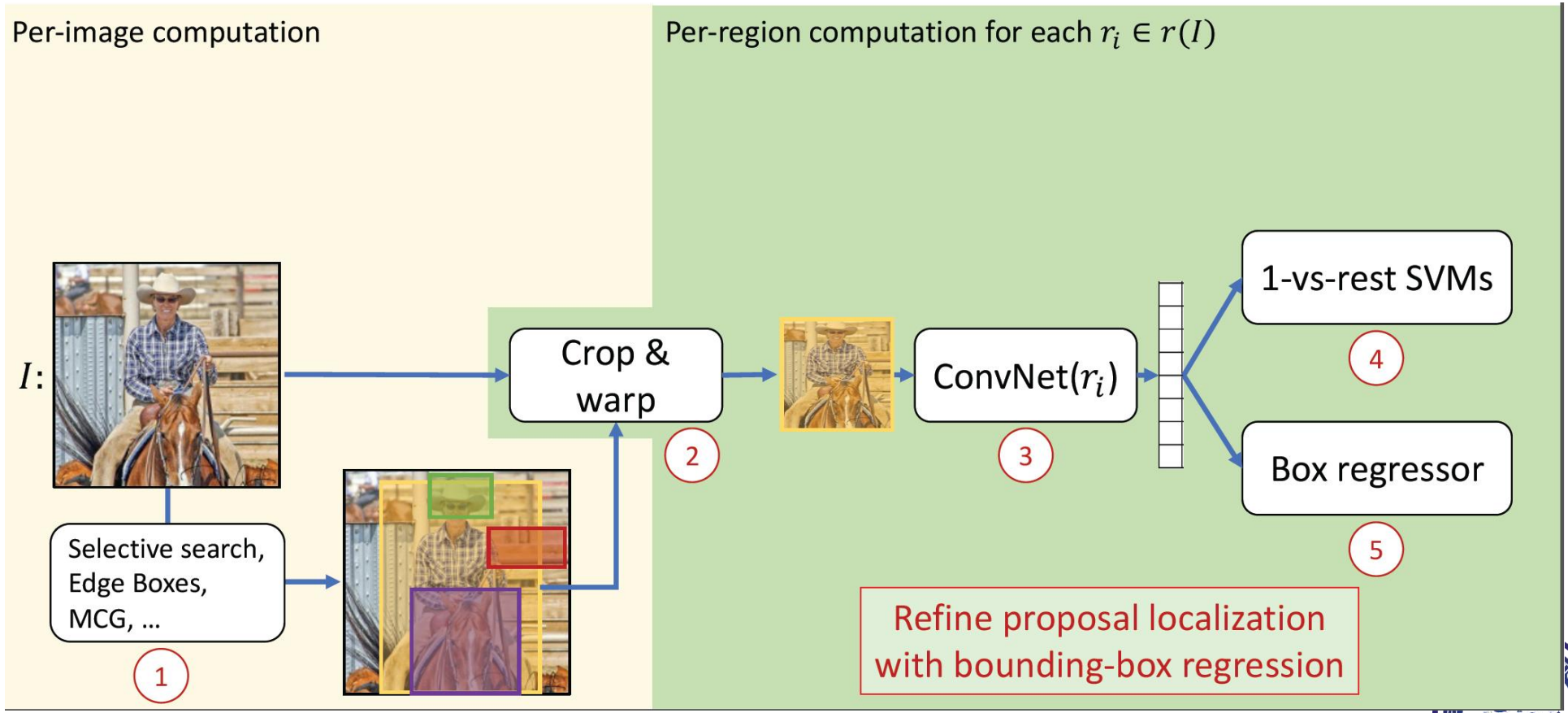


Region proposals: edge/region-based, resize to fixed window



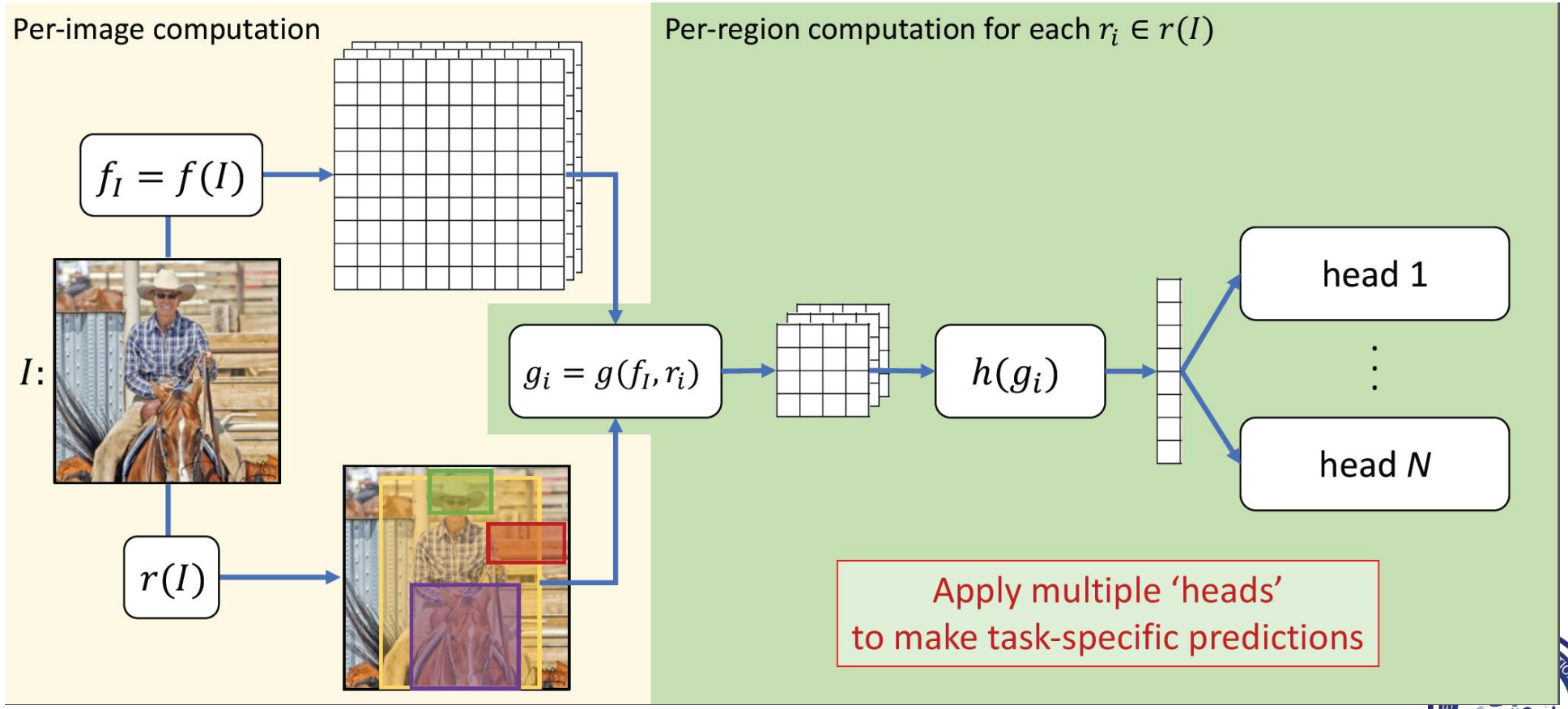
CNN features

Region-based CNN



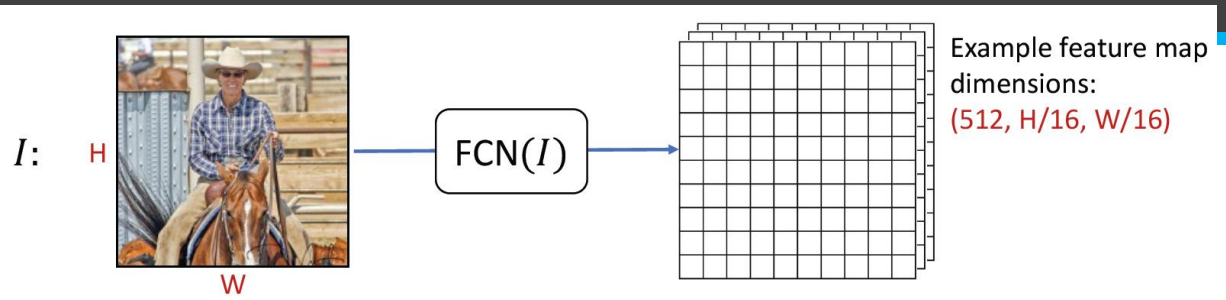
Slides: Ross Girshick

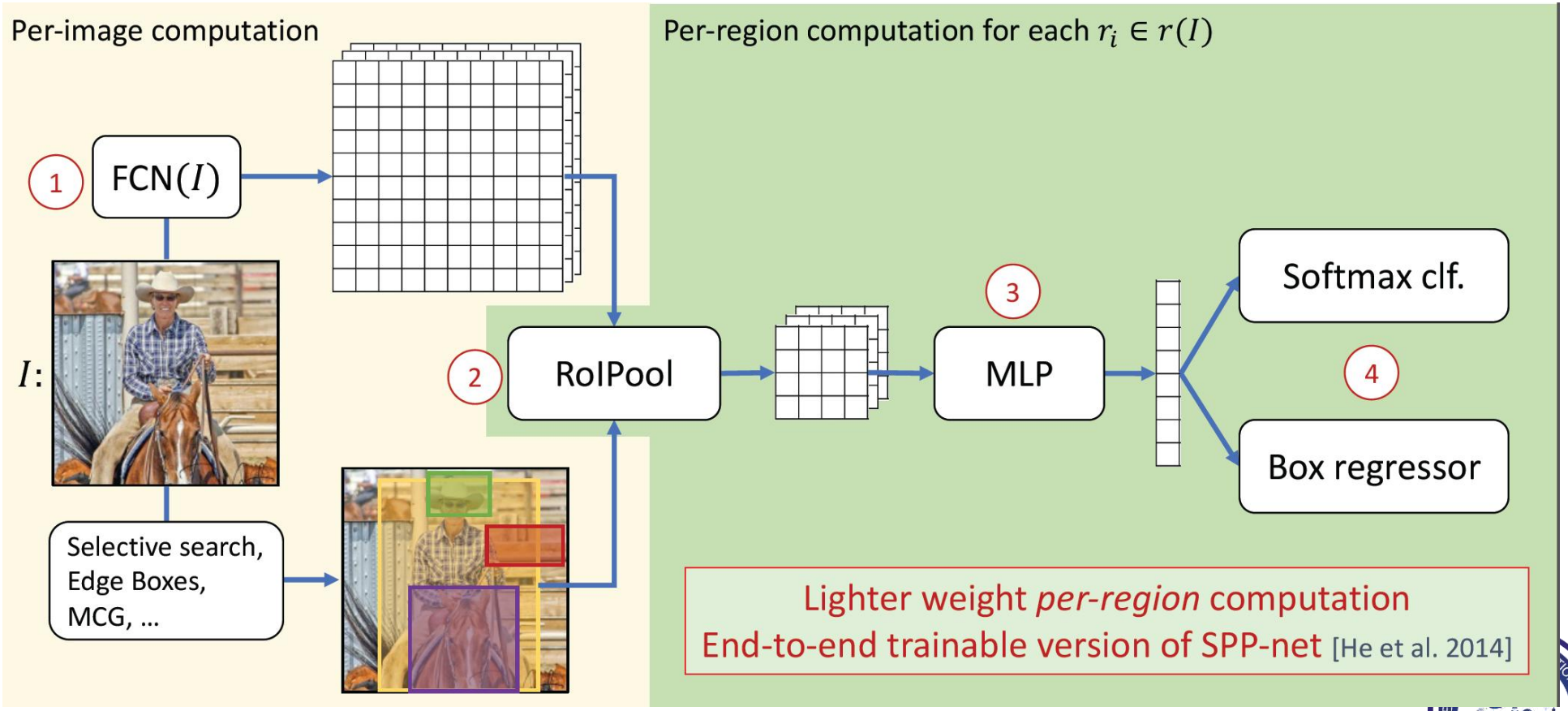
Generalized R-CNN



Slides: Ross Girshick

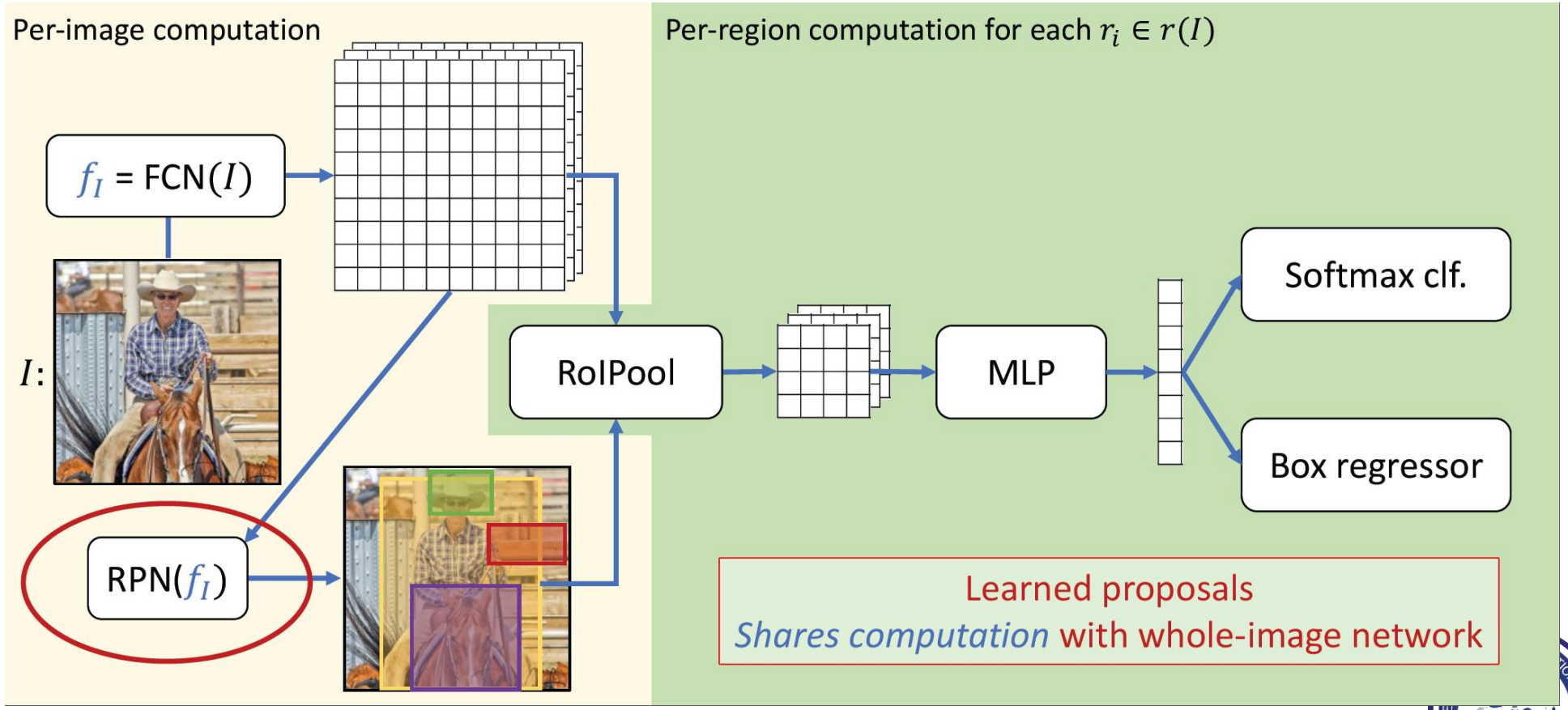
Fast R-CNN





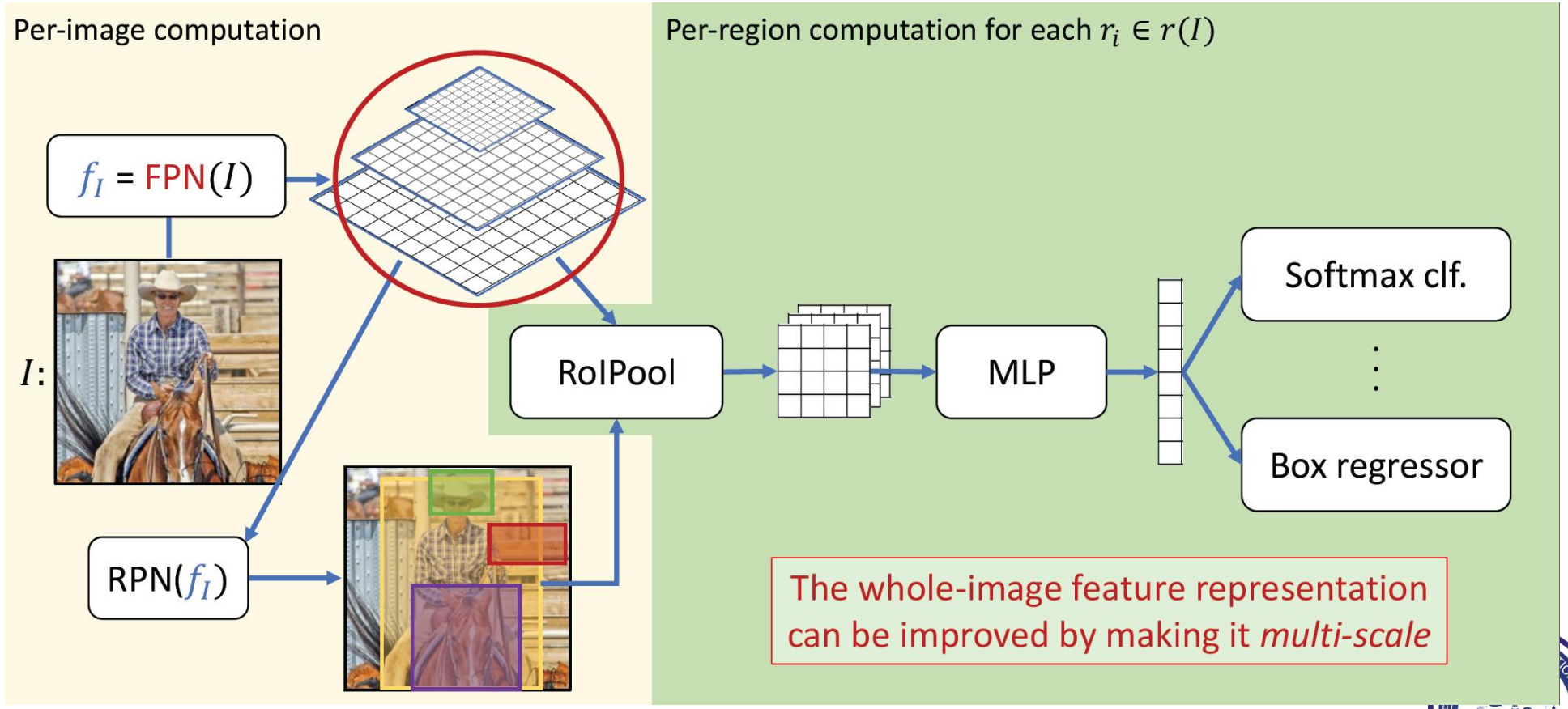
Slides: Ross Girshick

Faster RCNN



Slides: Ross Girshick

Faster RCNN + Feature Pyramid Network

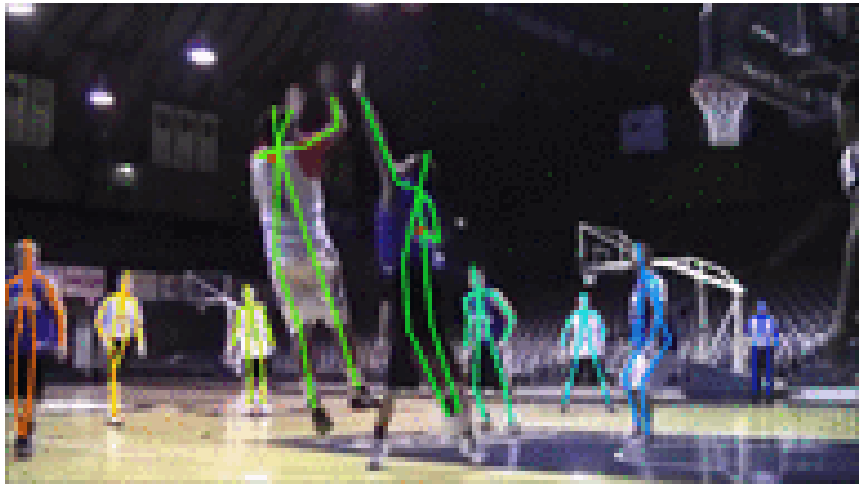


Slides: Ross Girshick

Part/keypoint Prediction

http://mscoco.org/dataset/#keypoints-challenge2016

PoseTrack



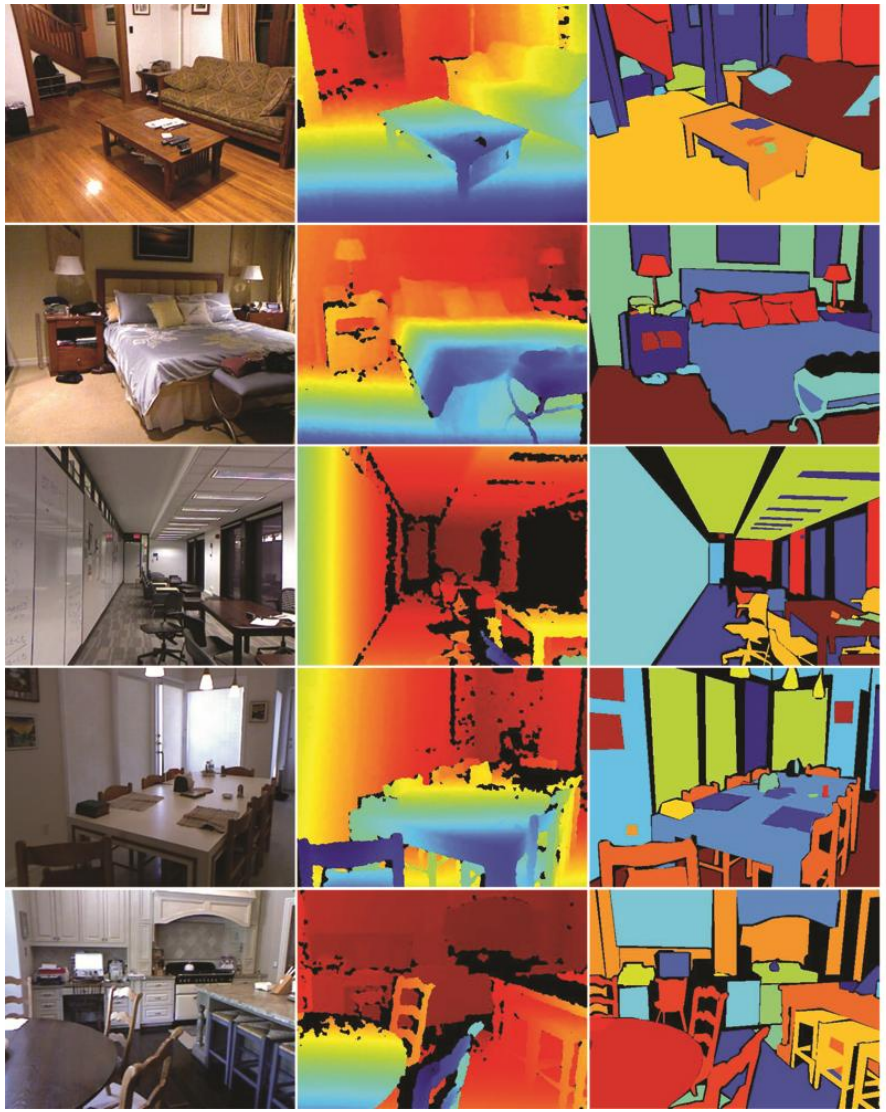
https://posetrack.net/

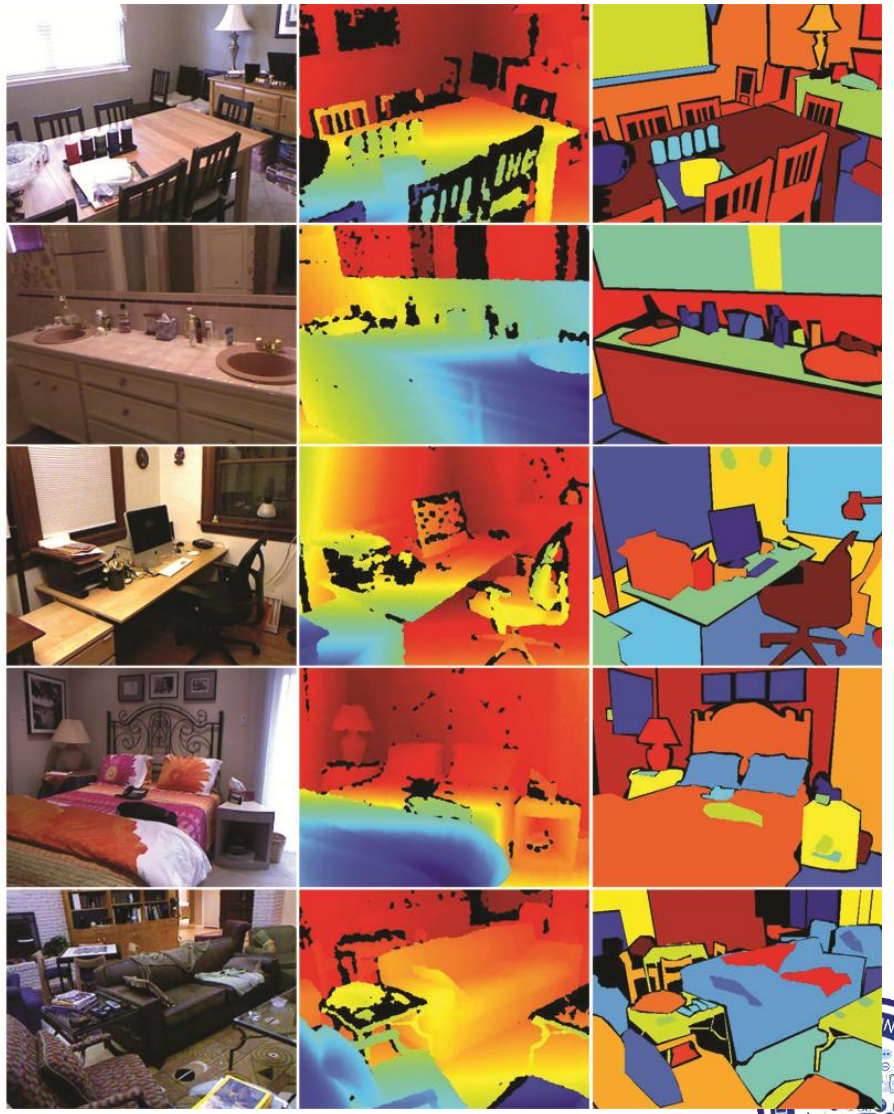
Semantic Segmentation

http://mscoco.org/dataset/#detections-challenge2016

https://www.cityscapes-dataset.com/examples/#fine-annotations

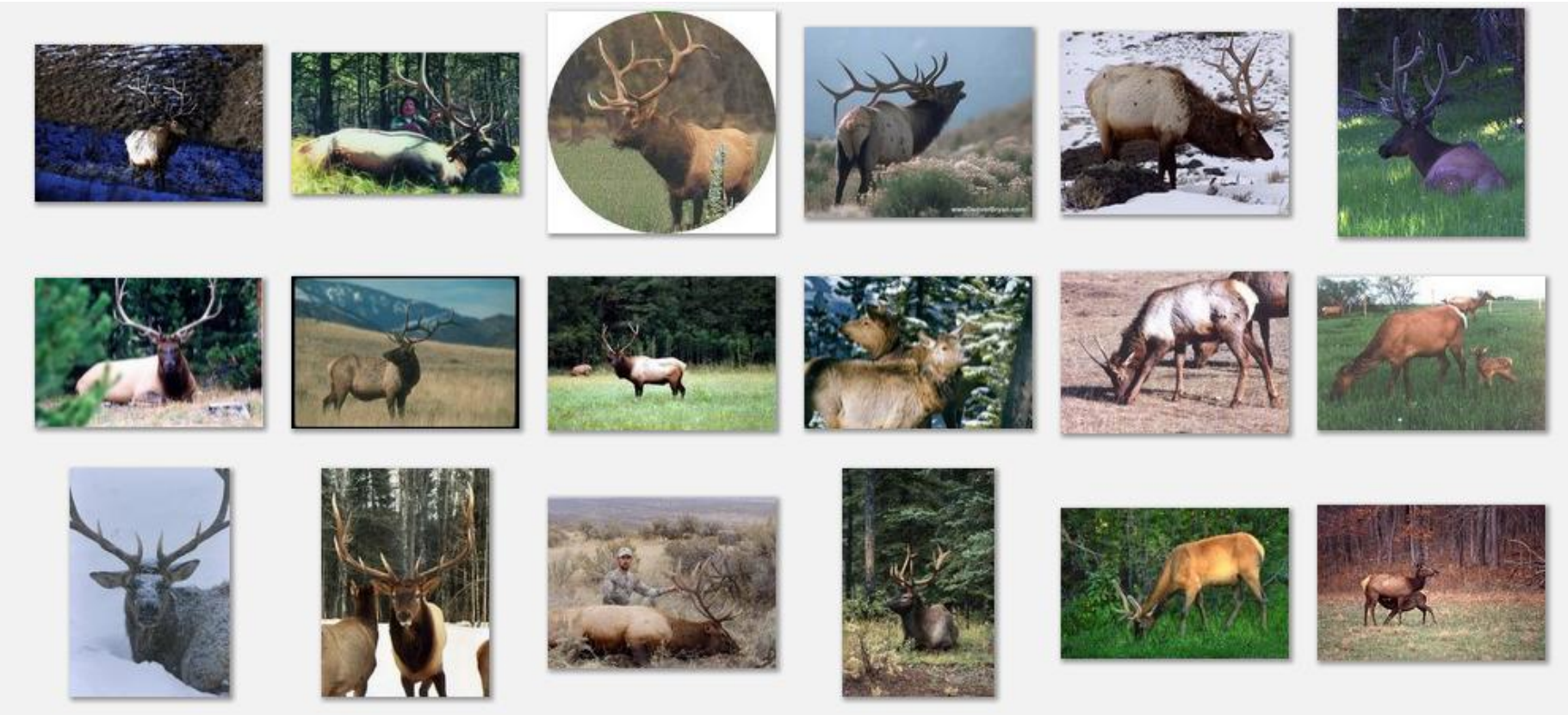
Semantic Segmentation





http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html

Deformable objects



Images from Caltech-256

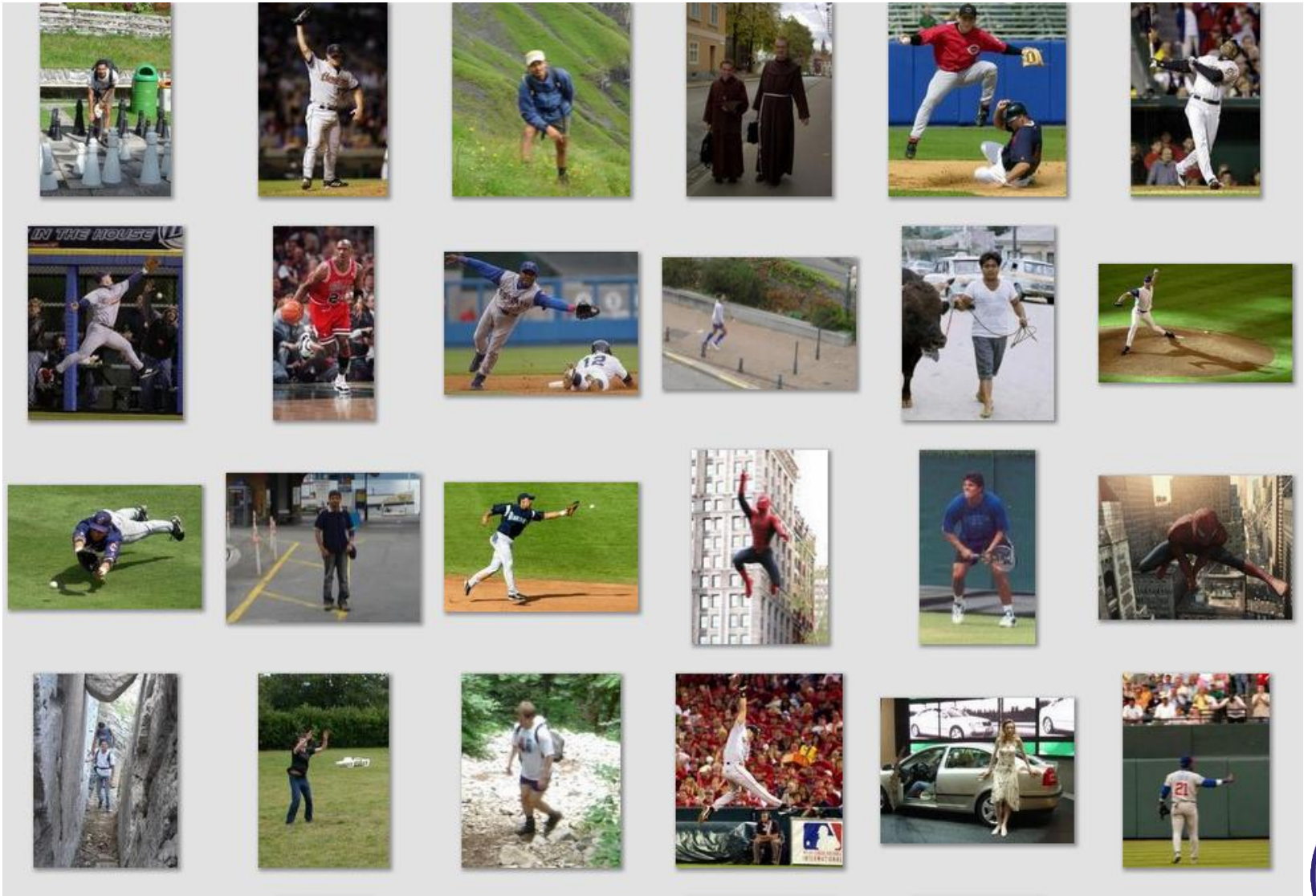
Slide Credit: Duan Tran

Graphical models vs. sequential/parallel prediction

- Advantages of BP/graphcut/etc
  - Elegant
  - Relations are explicitly modeled
  - Exact inference in some cases

- Advantages of sequential/parallel prediction
  - Simple procedures for training and inference
  - Learns how much to rely on each prediction
  - Can model very complex relations

Deformable objects



Images from D. Ramanan's dataset

Slide Credit: Duan Tran

Compositional objects

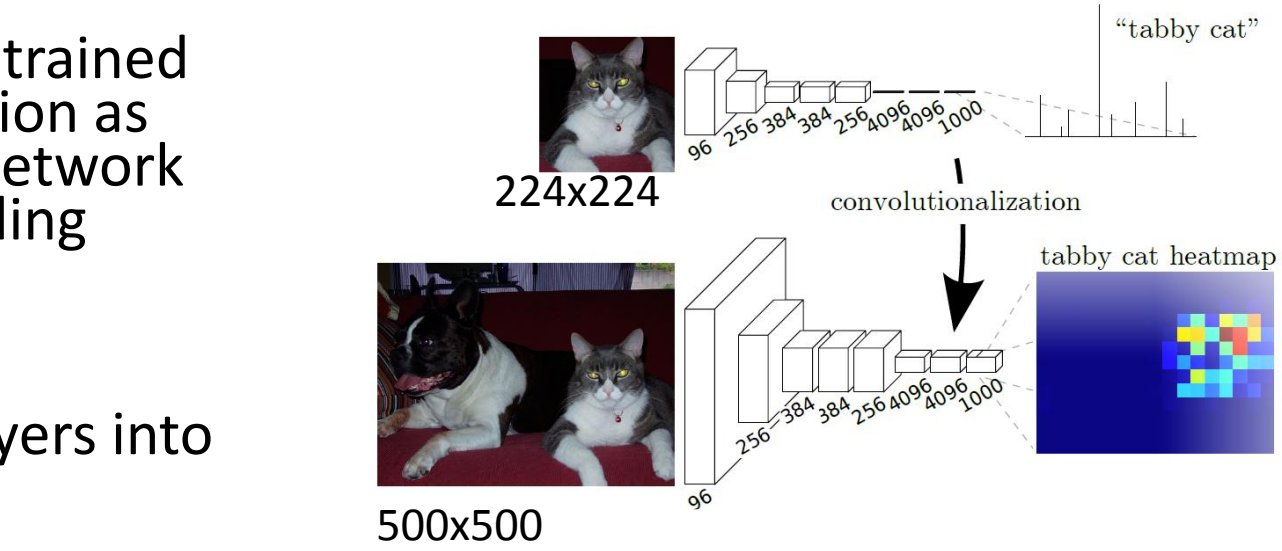
What if you want to label every pixel?

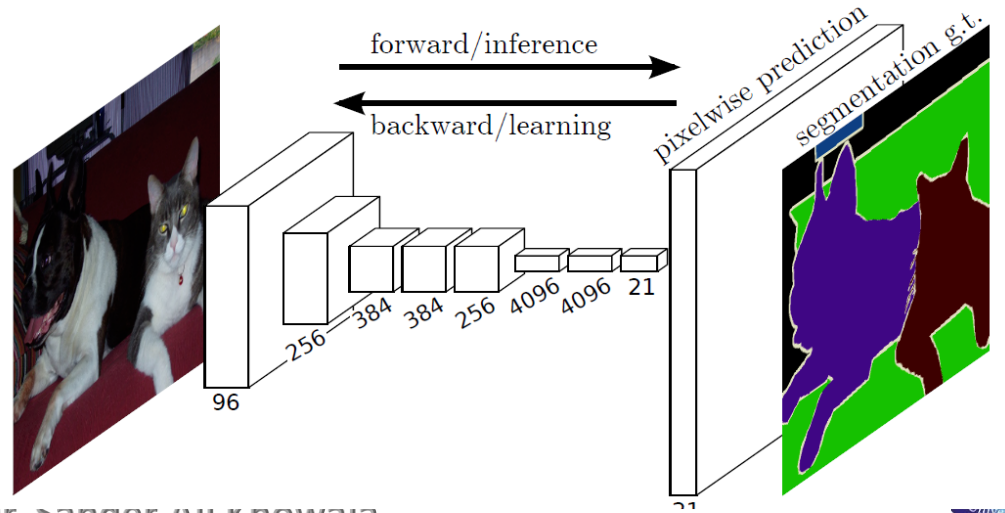
"Stuff" can be hard to capture with bounding boxes

https://www.cityscapes-dataset.com/examples/#fine-annotations

Fully convolutional networks for semantic segmentation – Long Shelhamer Darrel 2015

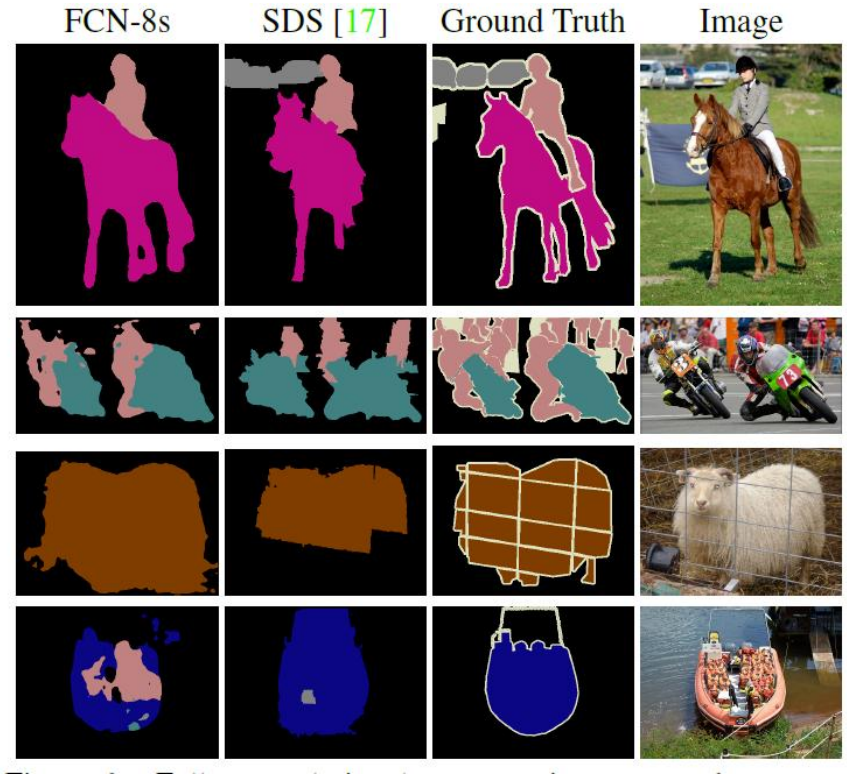
- Use network trained for classification as pre-trained network for pixel labeling
- Convert fully connected layers into convolutions
- Add features from earlier conv layers to improve resolution
- Fine-tune for pixel labeling task





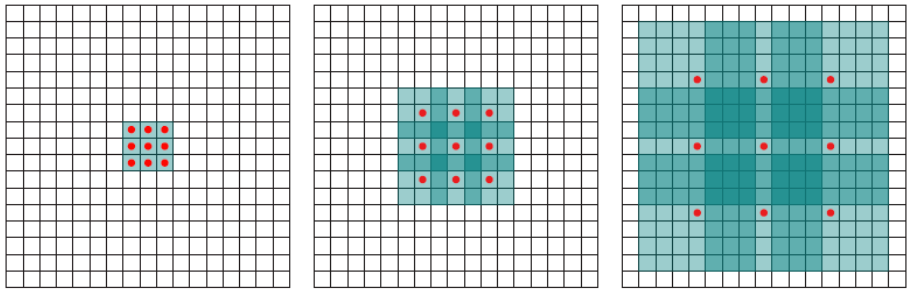
“Fully convolutional” results

- Takes advantage of pre-training from classification
- Applied to objects and scenes (NYUd v2)
- But feature pooling reduces spatial sensitivity and resolution



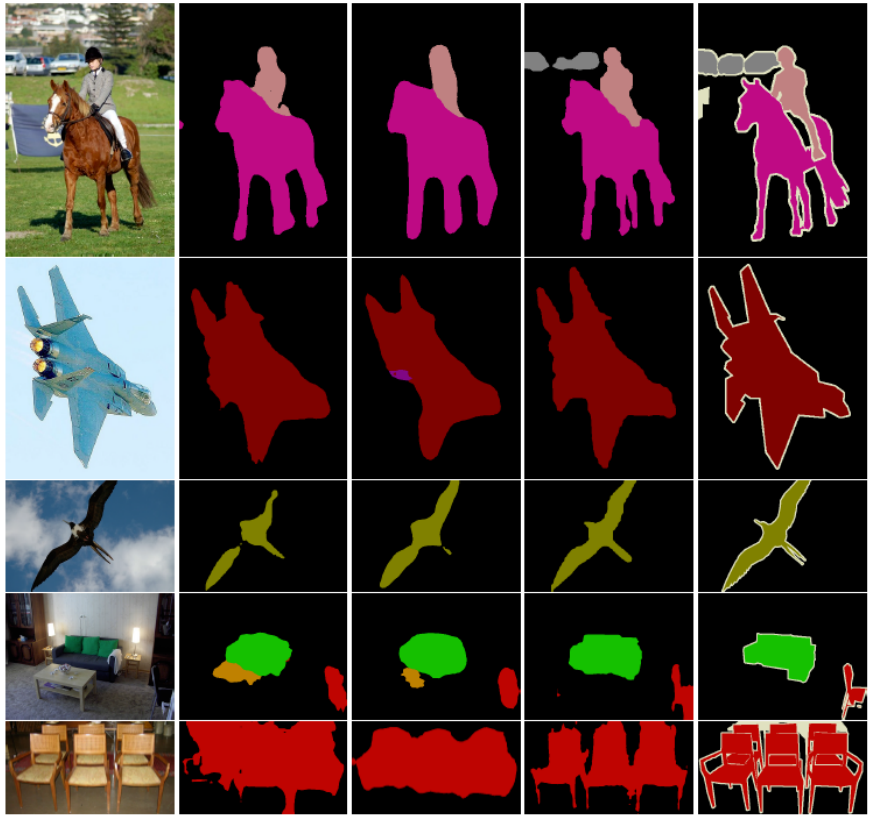
Dilated Convolutions – Yu Kolton 2016

- Replacing last two pooling layers with “dilated convolution” that filters a sparse 3x3 grid of pixels



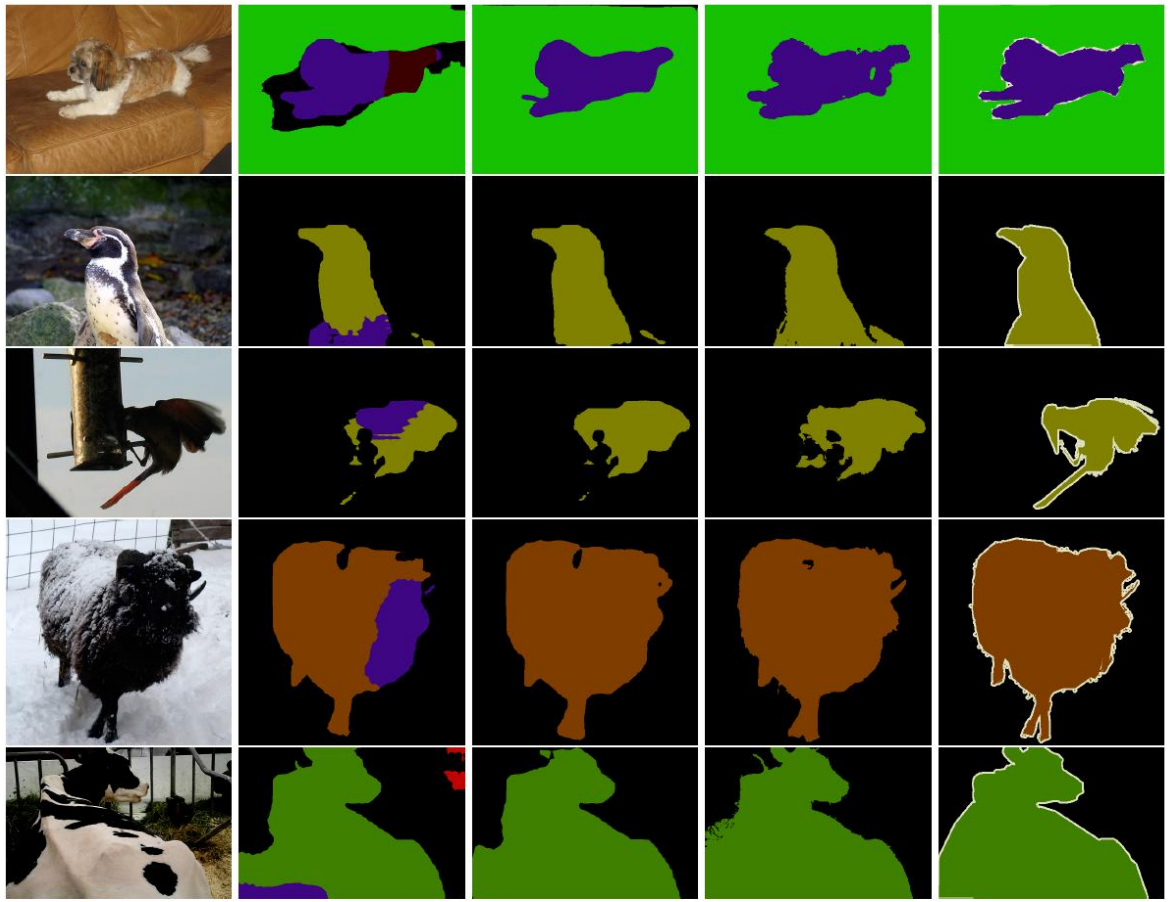
- Enables large receptive field with few parameters

- Improves resolution



(a) Image (b) FCN-8s (c) DeepLab (d) Our front end (e) Ground truth

Dilated Convolutions results



(a) Image (b) Front end (c) +Context (d) +CRF-RNN (e) Ground truth

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean IoU
DeepLab++	89.1	38.3	88.1	63.3	69.7	87.1	**83.1**	85	29.3	76.5	56.5	79.8	77.9	85.8	82.4	57.4	84.3	54.9	80.5	64.1	72.7
DeepLab-MSc++	89.2	46.7	88.5	63.5	68.4	87.0	81.2	86.3	32.6	80.7	62.4	81.0	81.3	84.3	82.1	56.2	84.6	58.3	76.2	67.2	73.9
CRF-RNN	90.4	**55.3**	88.7	**68.4**	69.8	88.3	82.4	85.1	32.6	78.5	**64.4**	79.6	81.9	**86.4**	81.8	**58.6**	82.4	53.5	77.4	**70.1**	74.7
Front end	86.6	37.3	84.9	62.4	67.3	86.2	81.2	82.1	32.6	77.4	58.3	75.9	81	83.6	82.3	54.2	81.5	50.1	77.5	63	71.3
Context	89.1	39.1	86.8	62.6	68.9	88.2	82.6	87.7	33.8	81.2	59.2	81.8	87.2	83.3	83.6	53.6	84.9	53.7	80.5	62.9	73.5
Context + CRF	91.3	39.9	**88.9**	64.3	69.8	88.9	82.6	89.7	34.7	82.7	59.5	83	88.4	84.2	85	55.3	86.7	54.4	**81.9**	63.6	74.7
Context + CRF-RNN	**91.7**	39.6	87.8	63.1	**71.8**	**89.7**	82.9	**89.8**	**37.2**	**84**	63	**83.3**	**89**	83.8	**85.1**	56.8	**87.6**	**56**	80.2	64.7	**75.3**