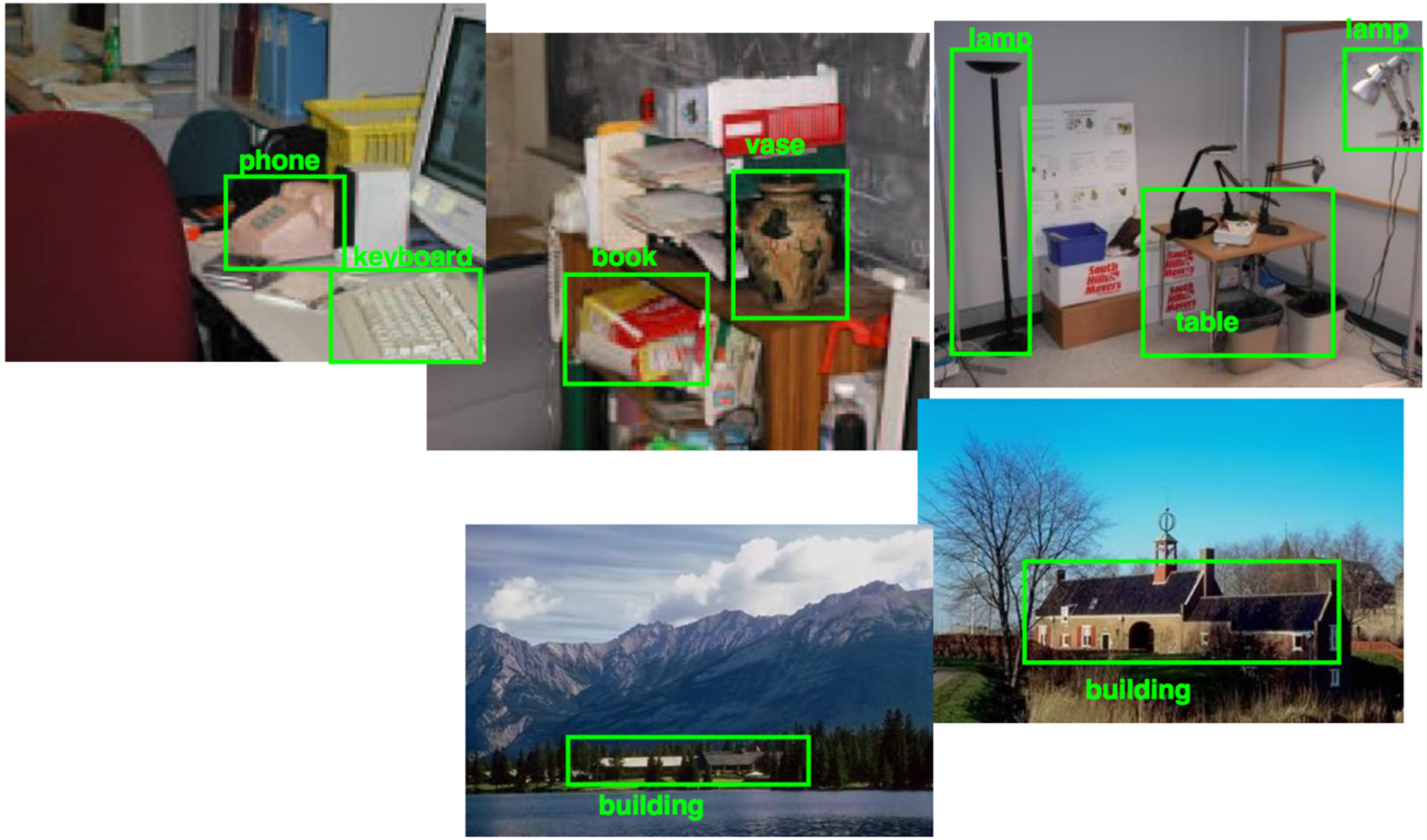
Recognition

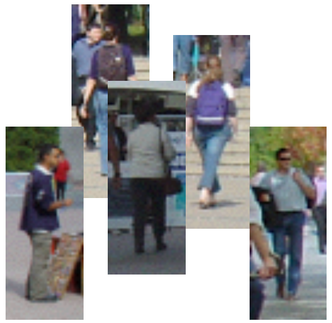
Gary Overett (Slides adapted from CMU 16-720 2014)



Some of the material from: Fei-Fei Li, Antonio Torralba, Szeliski&Seitz, Rob Fergus

Training and Test

Training Examples



Learn an internal model of the examples.

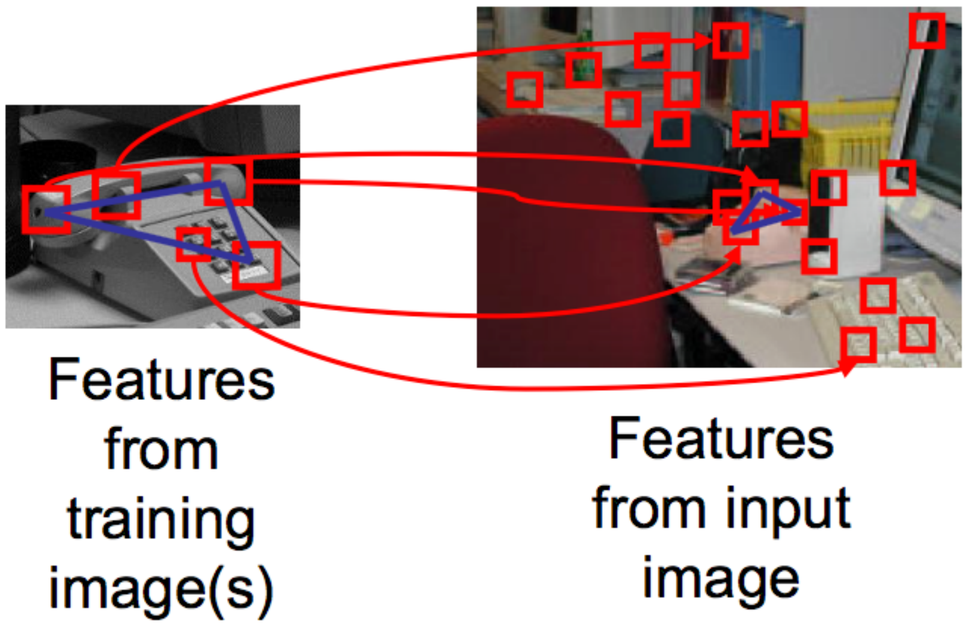
Query Image

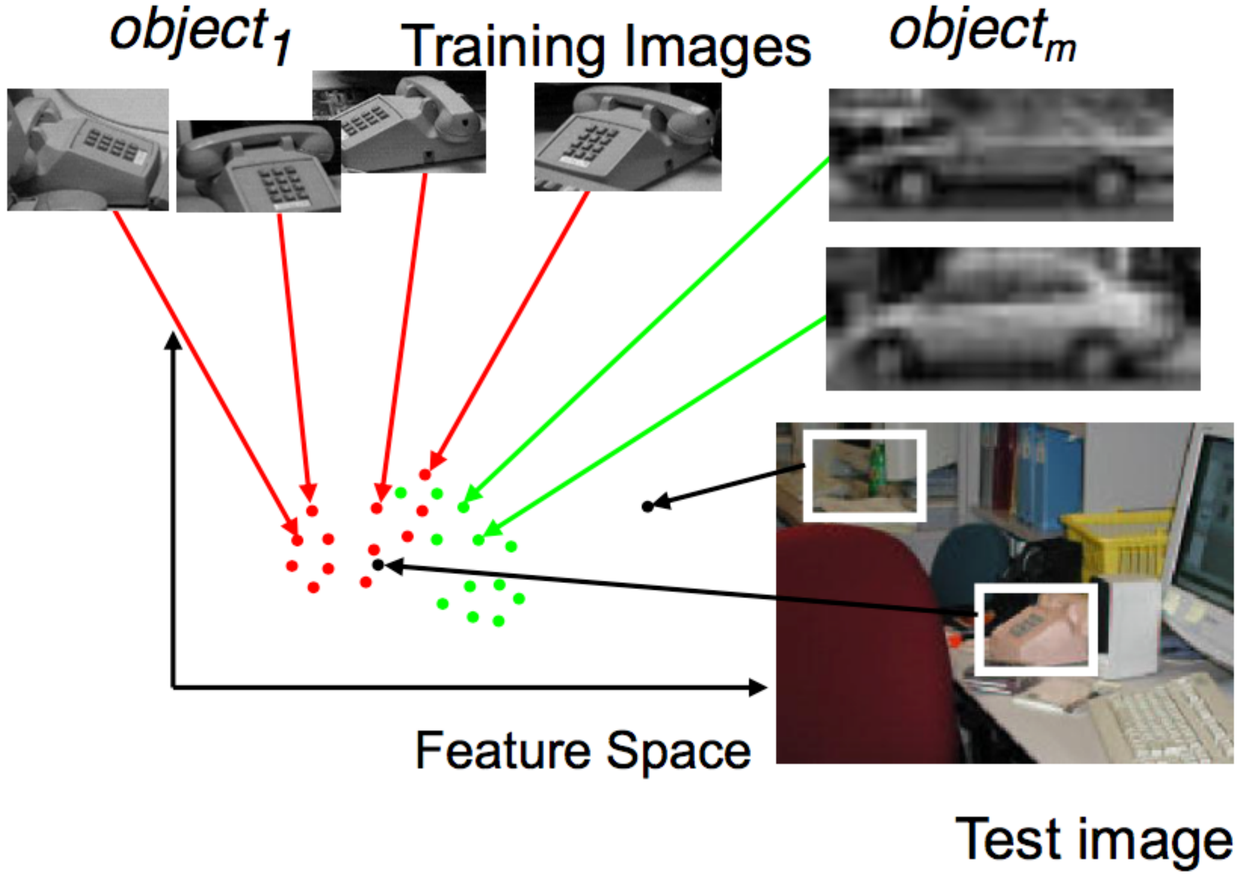
Output

Negative Class

Which of these IS NOT a Person/Pedestrian?

2 Common Approaches





Approaches based on using feature matches and geometric relations

Approaches based on classifying/matching image patches (windows)

The PASCAL Visual Object Classes Challenge 2007 (20 Classes)

Person: person
Animal: bird, cat, cow, dog, horse, sheep
Vehicle: aeroplane, bicycle, boat, bus, car, motorbike, train
Indoor: bottle, chair, dining table, potted plant, sofa, tv/monitor

M. Everingham, Luc van Gool , C. Williams, J. Winn, A. Zisserman 2007

ImageNet (http://www.image-net.org/)

- Total number of non-empty synsets: 15589
- Total number of images: 11,231,732
- Number of images with bounding box annotations: 195,331
- Number of synsets with SIFT features: 1000
- Number of images with SIFT features: 1.2 million

Lotus Hill Research Institute image corpus

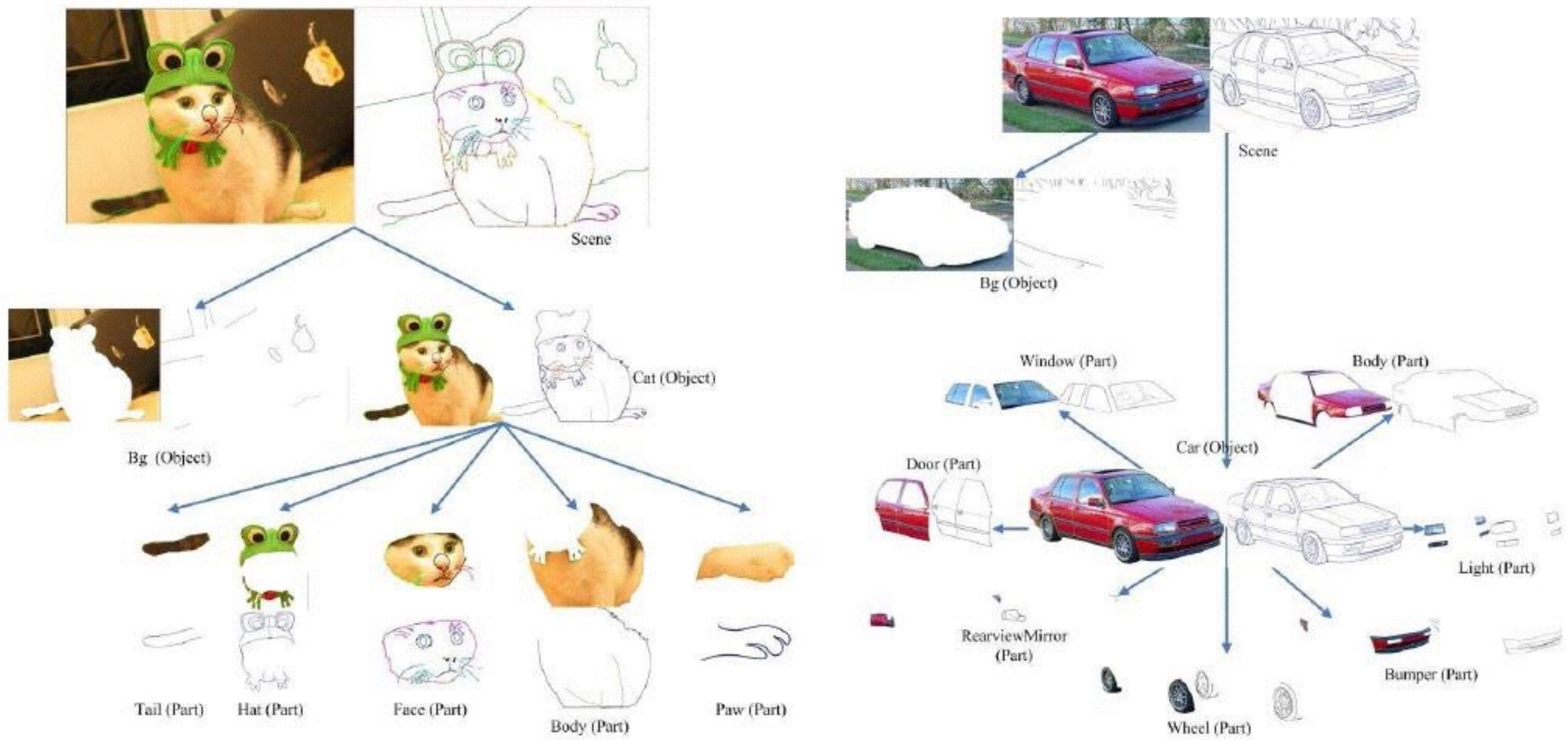


Figure 5: Two examples of the parse trees (cat and car) in the Lotus Hill Research Institute image corpus. From [87].

Z.Y. Yao, X. Yang, and S.C. Zhu, 2007

Terminology

- Classification: Is the object in the image?
  - One-vs.-all
  - Forced choice among N
- Detection: Is the object in the image and where?

Measuring Performance

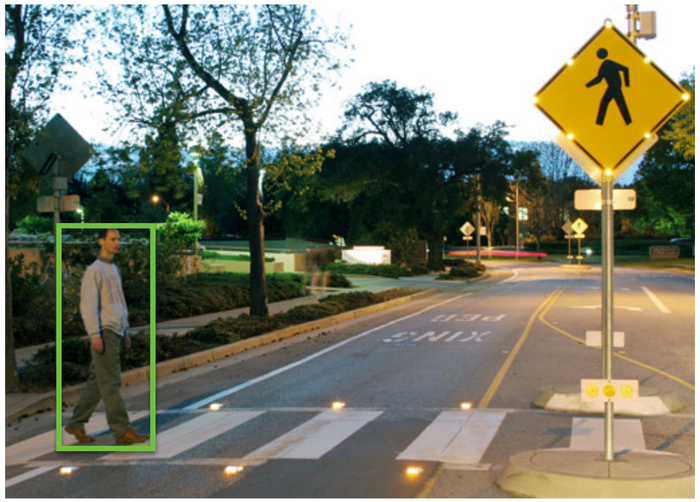
Intersection over Union

- Correct Detection if…

$$\frac{\text{GT} \cap \text{D}}{\text{GT} \cup \text{D}} = \frac{|\text{GT} \cap \text{D}|}{|\text{GT} \cup \text{D}|} > T$$

Terminology

True Positive (TP)
# of correct detections



False Positive (FP)
# of incorrect detections

False Negative (FN)
of **missed** detections

True Negative (TN)
# of correct non-detections

Green = Detector->Pedestrian

Green = Detector->No Pedestrian

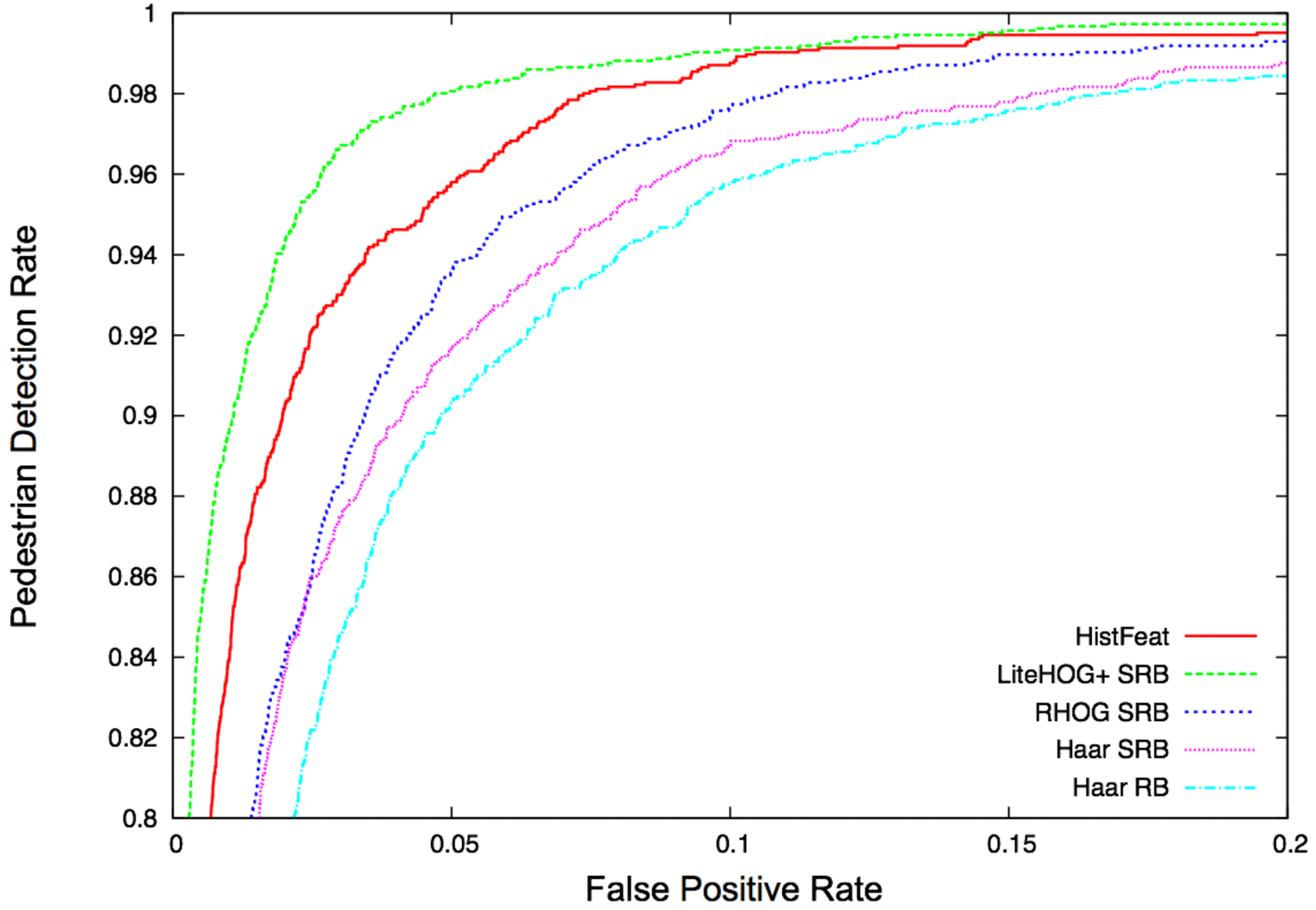
ROC Curve

Detection Rate

- TP/(TP-FN)

Prefer a Single Measure?

- AUC = Area Under the Curve
- EER = Equal error rate
- Others

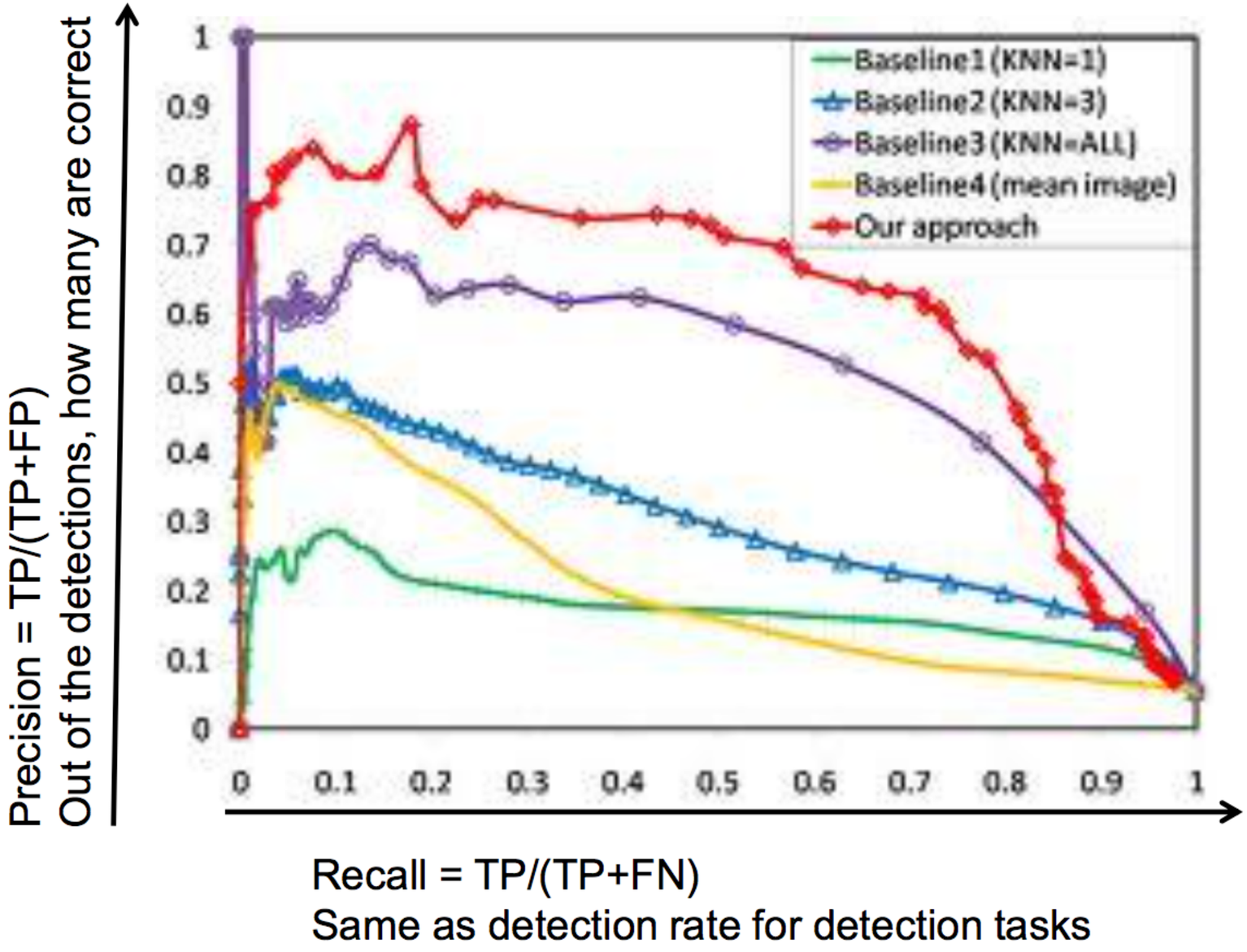


Precision-Recall Curve

- Invented for retrieval tasks
- Don't need to know the TN

Prefer a Single Measure?

- AUPRC = Area Under the Curve
- AP = Average Precision
- Others

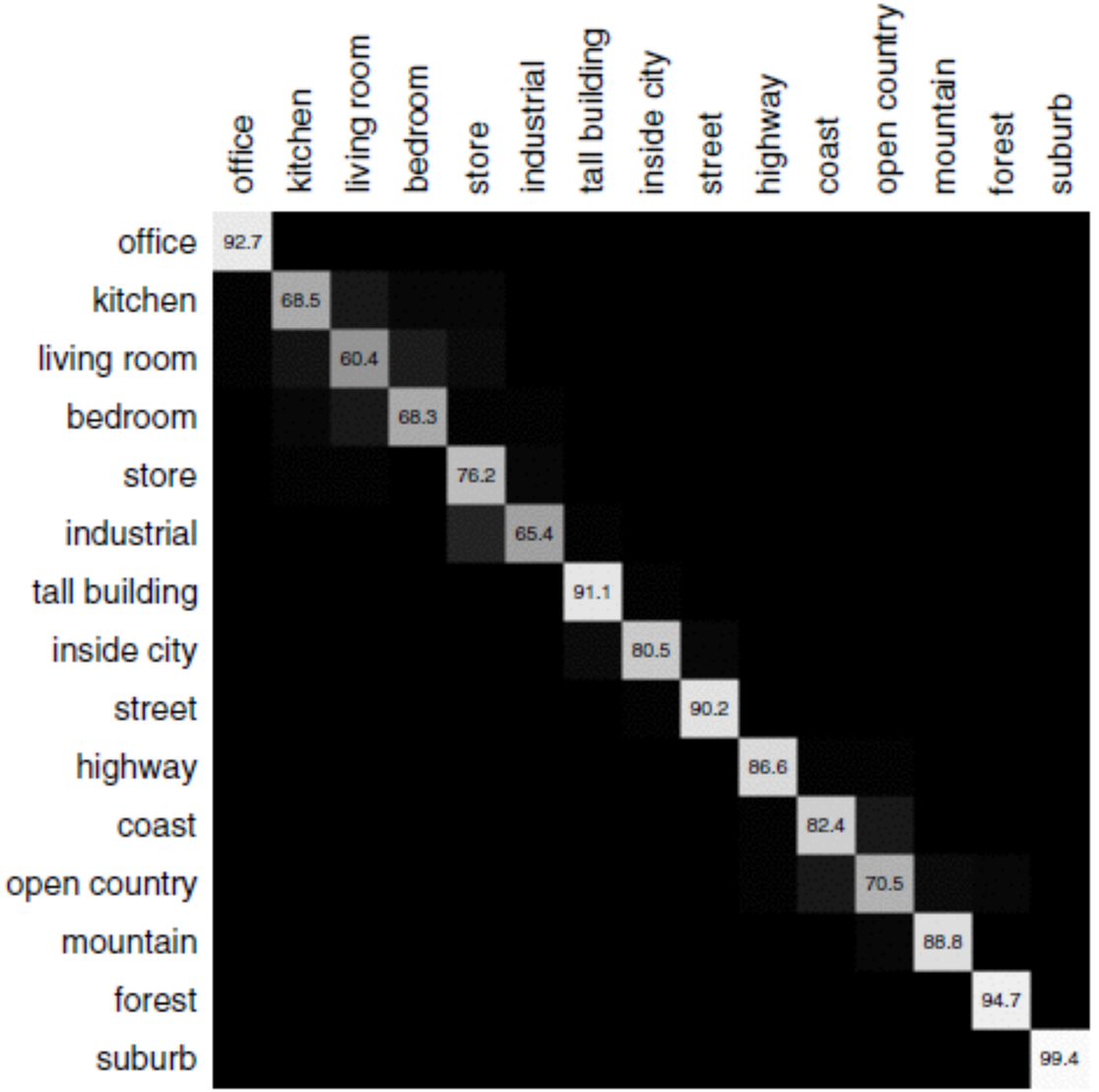


Davis and Goadrich. The Relationship Between Precision-Recall and ROC Curves. ICML (Intern. Conf. Machine Learning) 2006.

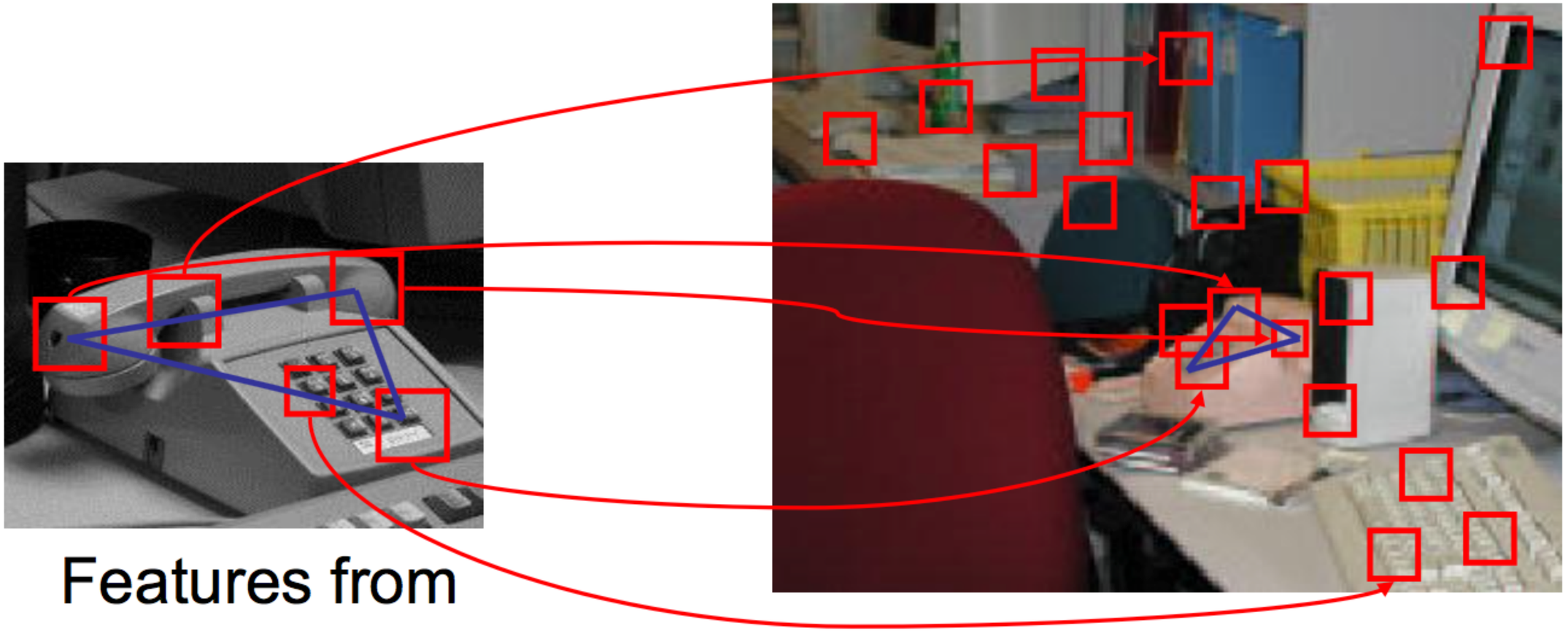
Confusion Matrix

- For forced choice classification tasks
- Accuracy = (correct samples)/(total samples)
- Different effect of class imbalance:
  - Per-class accuracy
  - Total overall accuracy

	Build-ing	Grass	Tree	Cow	Sky	Aero-plane	Face
Building	22			2		1	1
Grass		62	3				
Tree			28				
Cow				22			1
Sky					44		
Aeroplane	1					14	
Face							15



Feature Matching & Geometric Relations

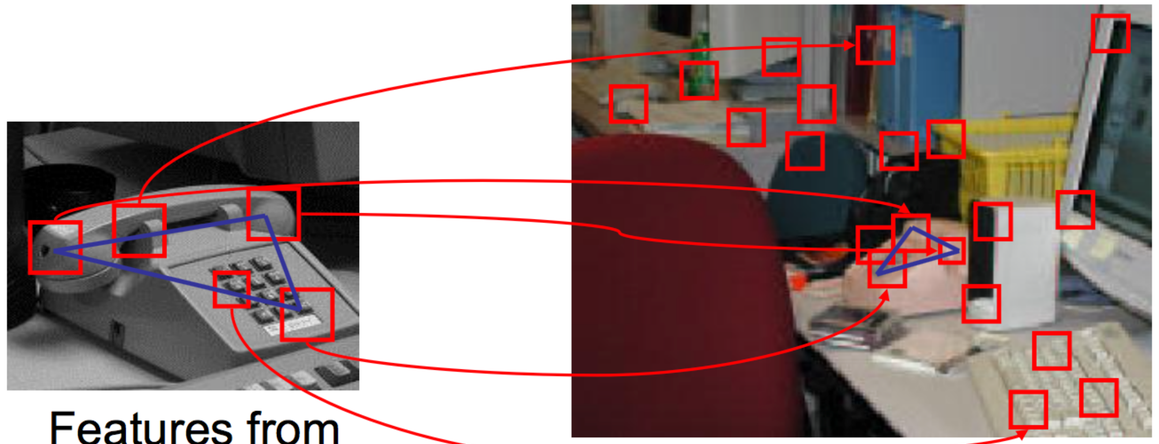


Features from training image(s)

Features from input image

Feature Matching & Geometric Relations

- Problem: Aspect is different between training and test images
- Solution: “Invariant” features
- Problem: Now Local feature similarity is not sufficient
- Solution: Use global geometric consistency
- Problem: Now we have a large number of features
- Solution: Define distance in feature space + efficient indexing

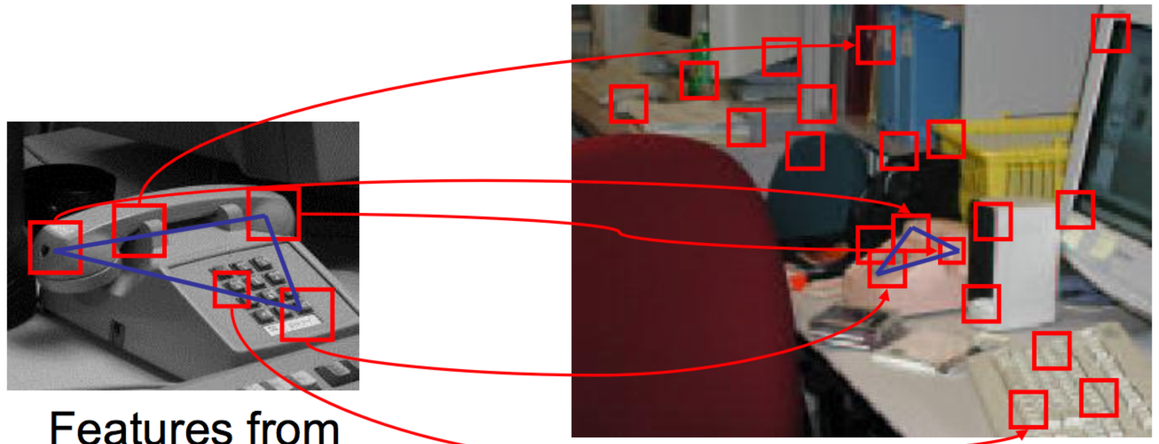


Features from training image(s)

Features from input image

Feature Matching & Geometric Relations

- Problem: Aspect is different between training and test images
- Solution: “Invariant” features
- Problem: Now Local feature similarity is not sufficient
- Solution: Use global geometric consistency
- Problem: Now we have a large number of features
- Solution: Define distance in feature space + efficient indexing



Features from training image(s)

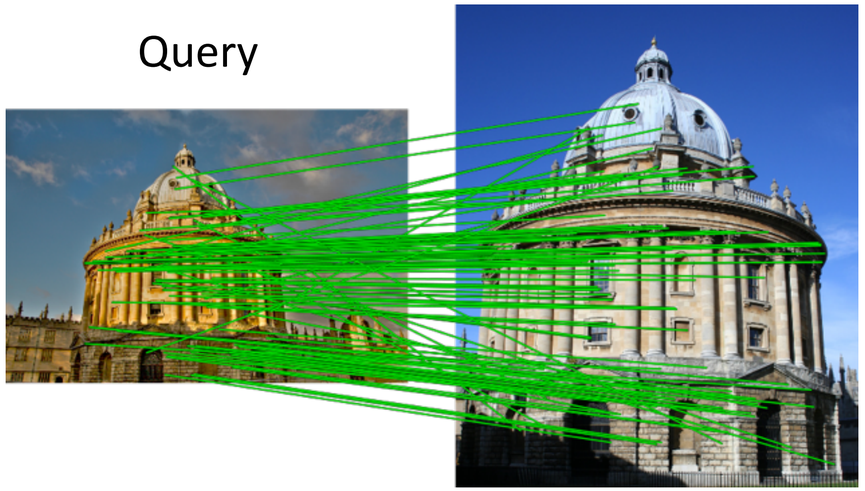
Features from input image

Are local features good enough for instance recognition?

Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement (geometric relations)?

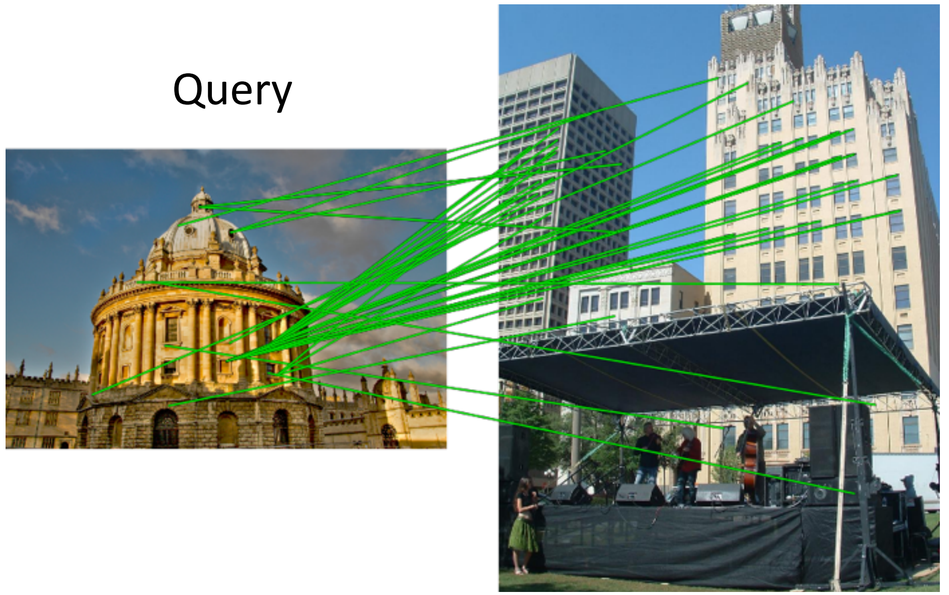
Spatial Verification

Query



DB image with high BoW similarity

Query



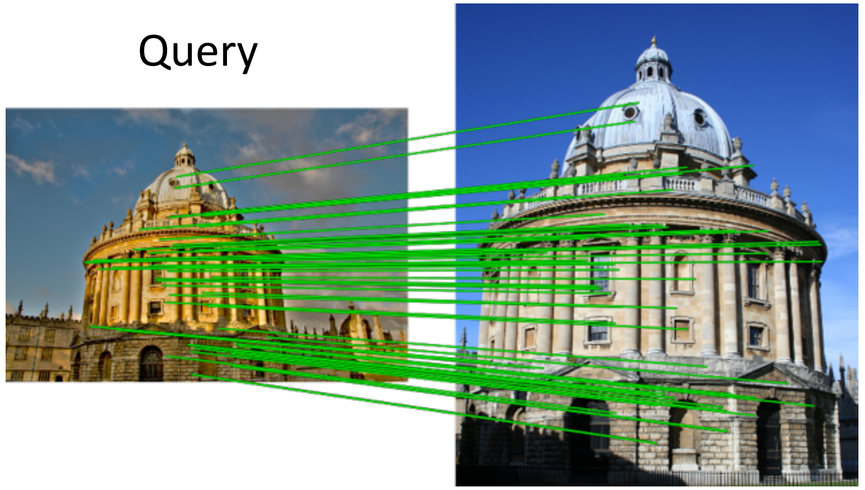
DB image with high BoW similarity

Both image pairs have many visual words in common.

Slide credit: Ondrej Chum

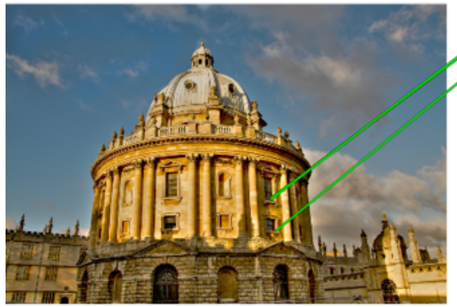
Spatial Verification

Query



DB image with high BoW similarity

Query



DB image with high BoW similarity

Only some of the matches are mutually consistent

Slide credit: Ondrej Chum

Spatial Verification: two basic strategies

- RANSAC
  - Use few examples to predict transformation…and then check
  - But will be discussed later

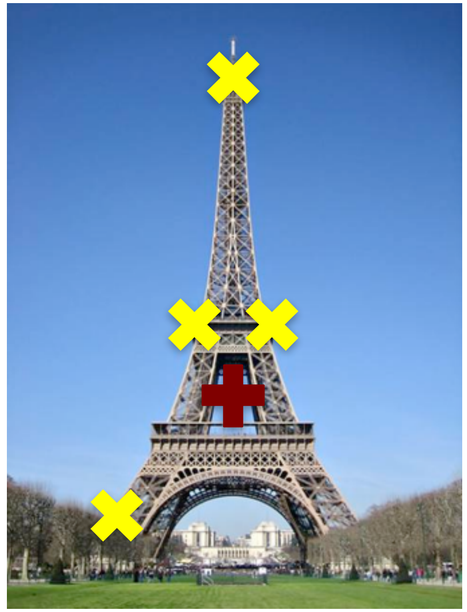
- Generalized Hough Transform

Spatial Verification: two basic strategies

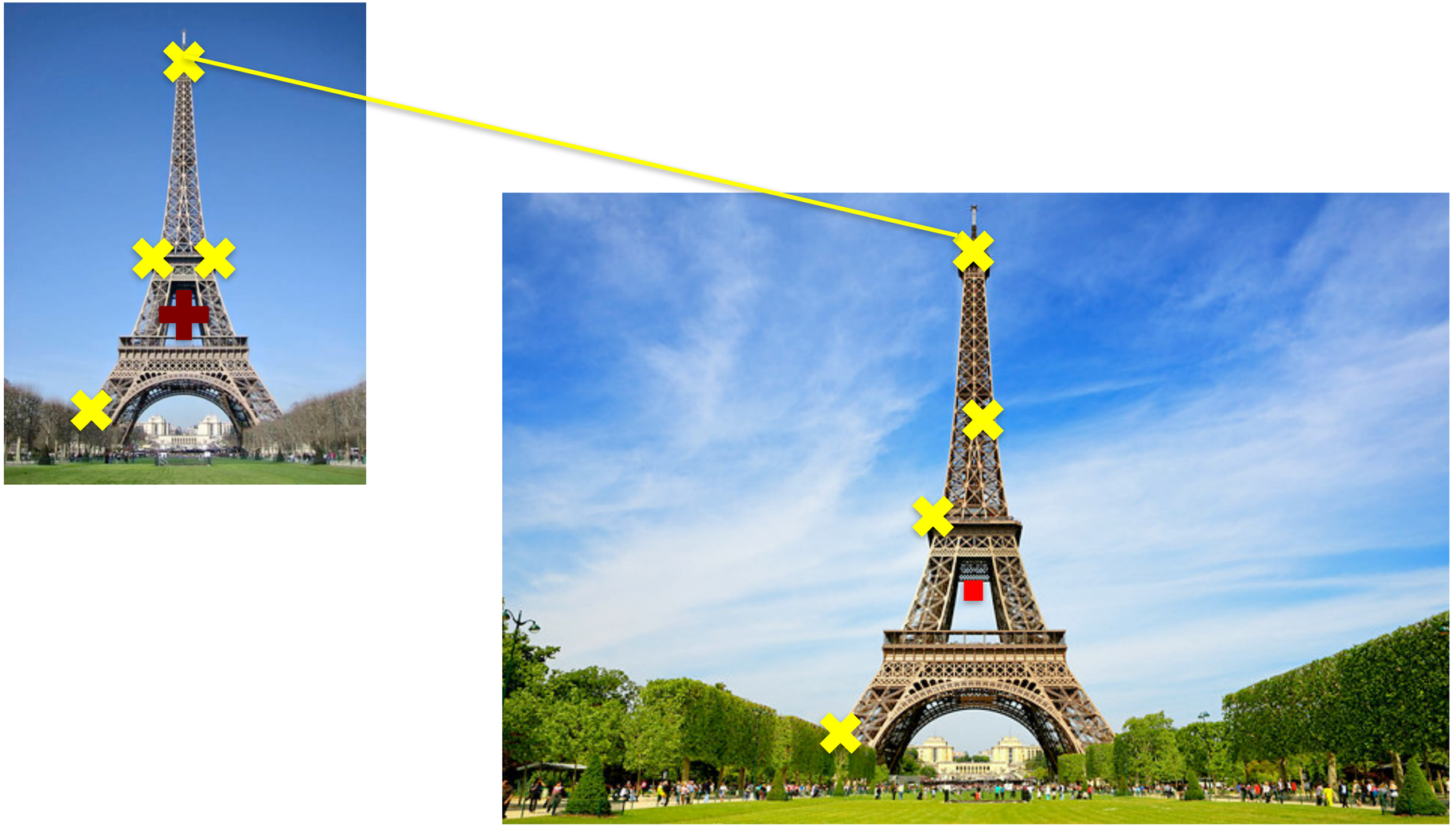
- RANSAC
  - Use few examples to predict transformation…and then check
  - But will be discussed later

- Generalized Hough Transform
  - Let each matched feature cast a vote on center location of the model object
  - Verify parameters with enough votes

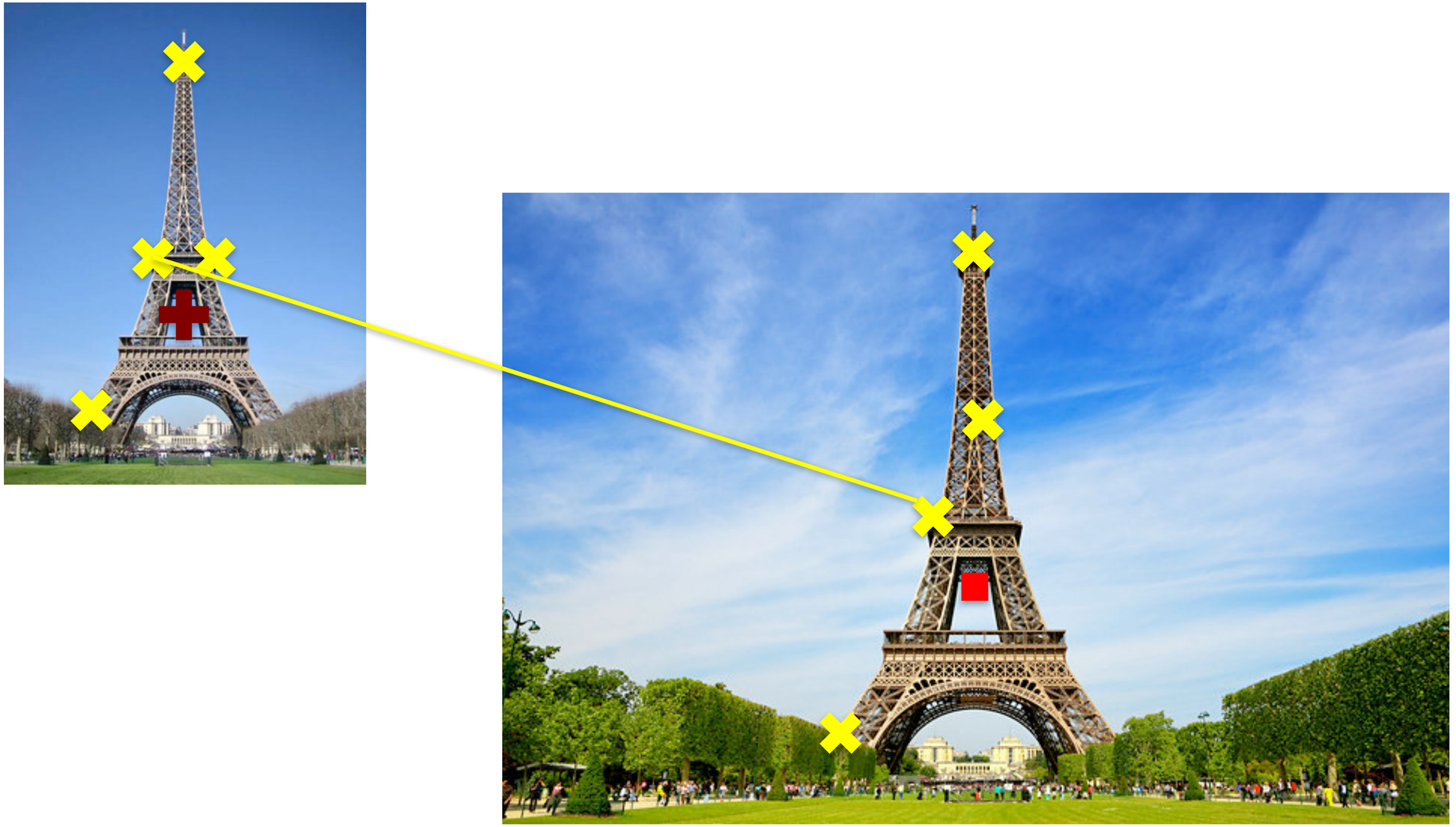
Generalized Hough Transform



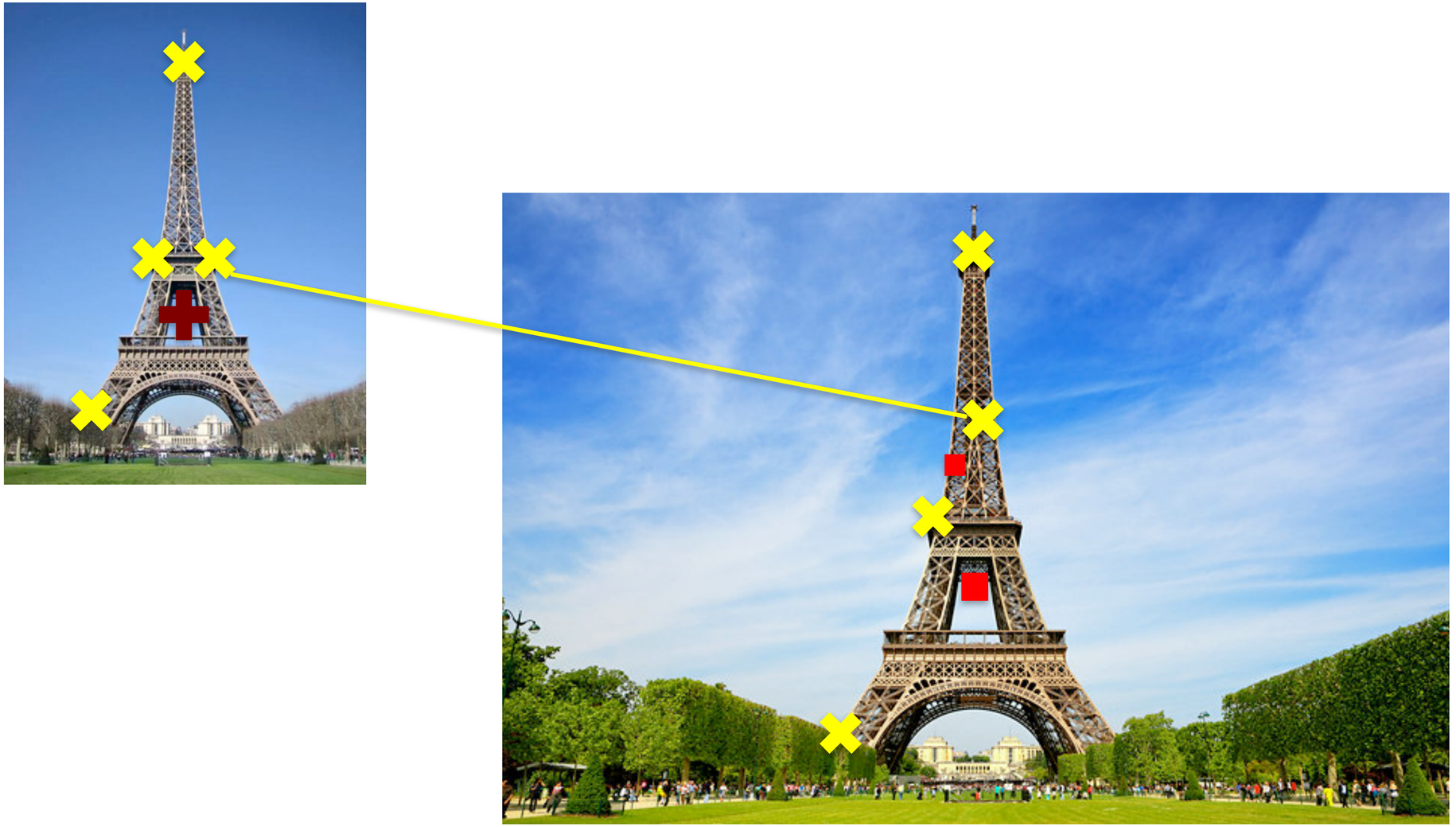
Generalized Hough Transform



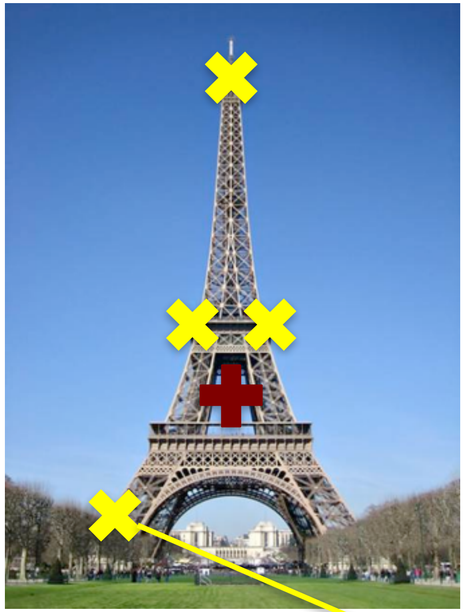
Generalized Hough Transform



Generalized Hough Transform

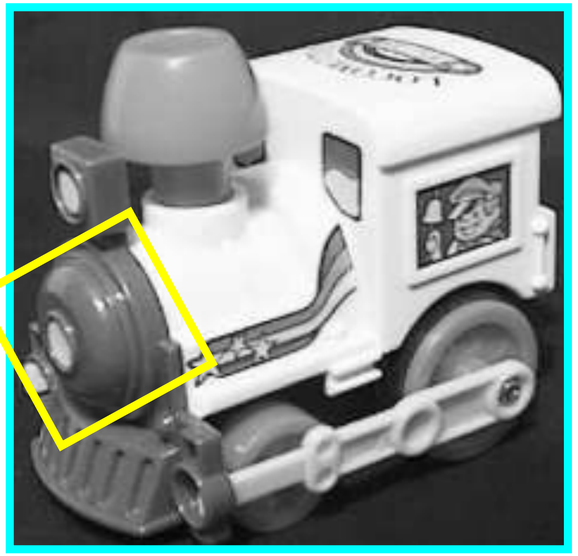


Generalized Hough Transform



Voting: Generalized Hough Transform

- If we use scale, rotation, and translation invariant local features, then each feature match gives an alignment hypothesis (for scale, translation, and orientation of model in image).



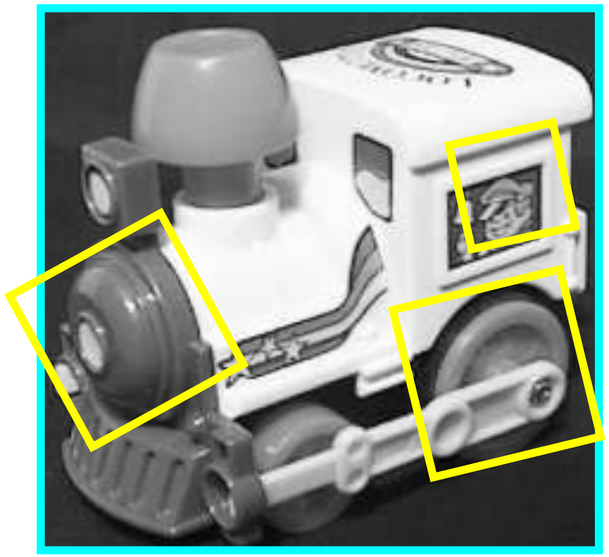
Model

Novel image

Adapted from Lana Lazebnik

Voting: Generalized Hough Transform

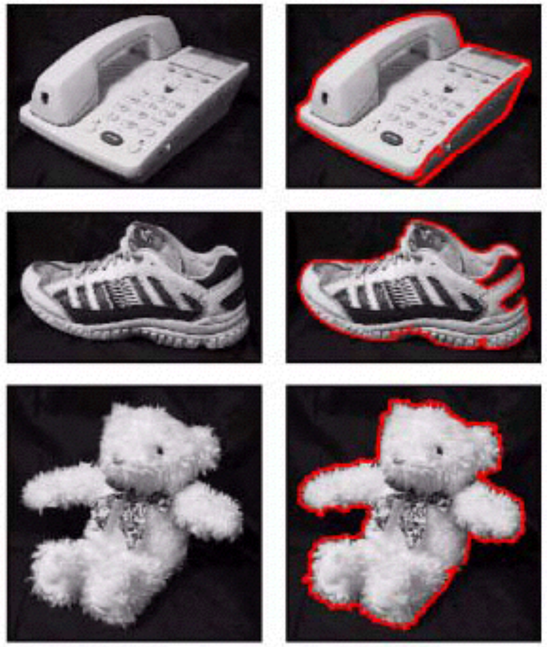
- A hypothesis generated by a single match may be unreliable,
- So let each match **vote** for a hypothesis in Hough space



Model

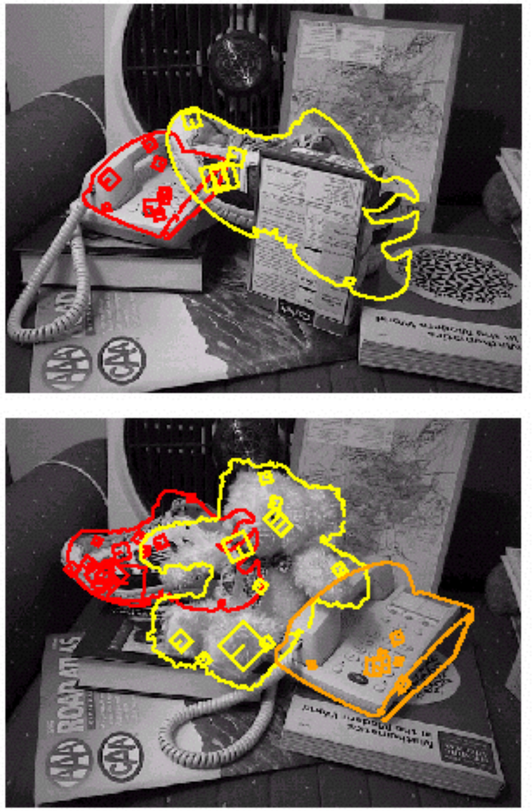
Novel image

Example result



Background subtract for model boundaries

Objects recognized,



Recognition in spite of occlusion

[Lowe]

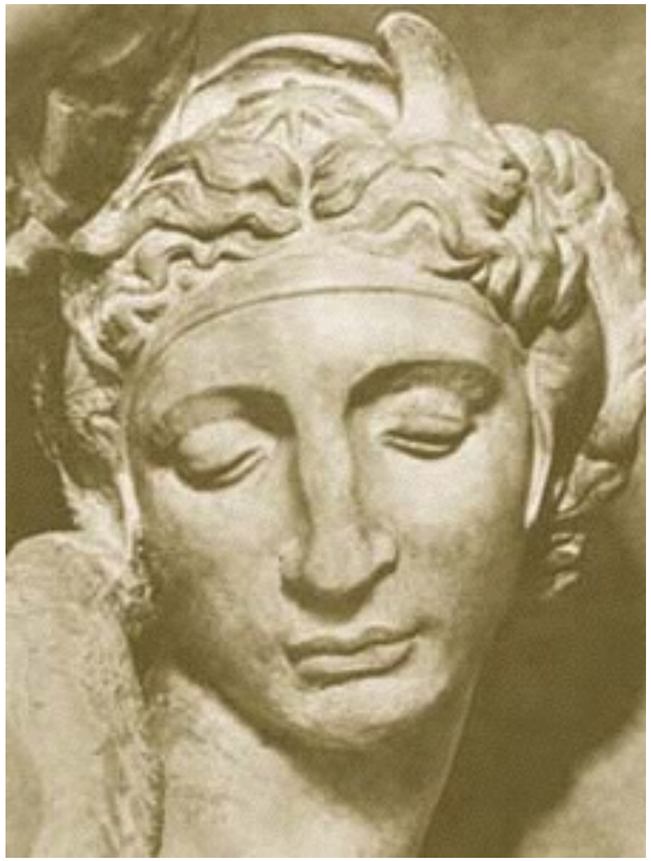
Recall: difficulties of voting

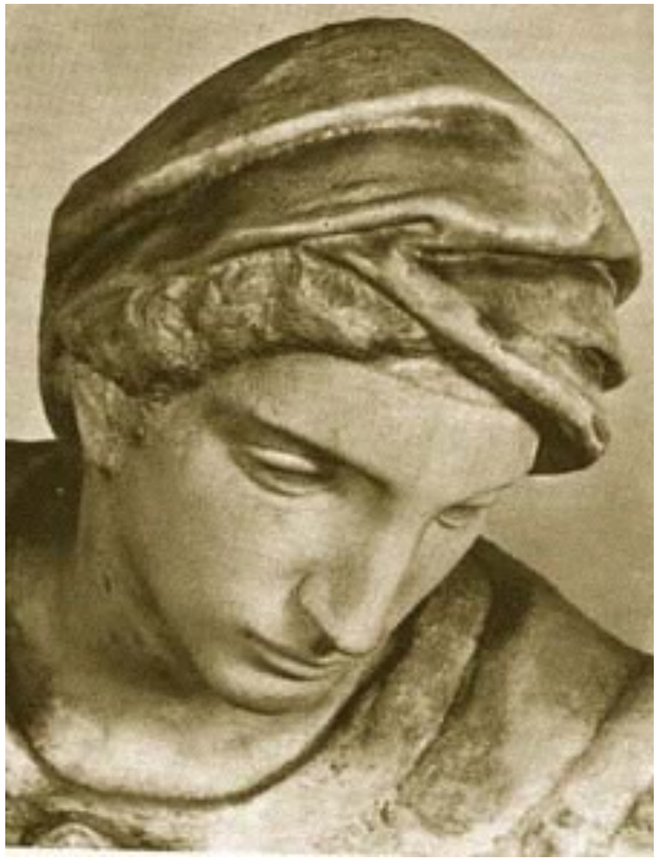
- Noise/clutter can lead to as many votes as true target
- Bin size for the accumulator array must be chosen carefully

- In practice, good idea to make broad bins and spread votes to nearby bins, since verification stage can prune bad vote peaks.

Why is recognition hard?

Challenges 1: view point variation

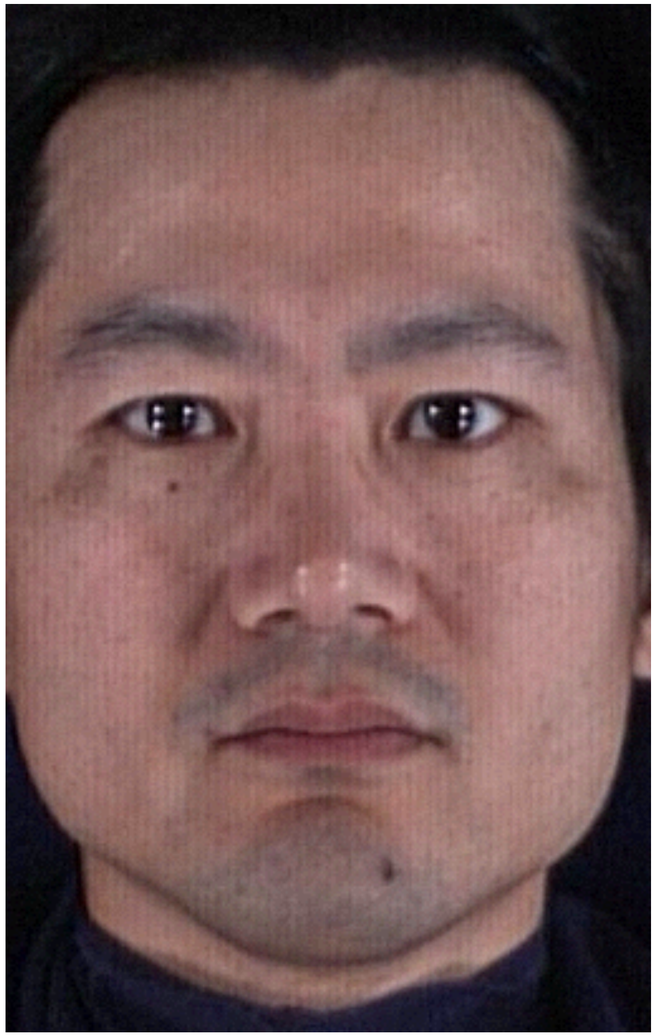




Michelangelo 1475-1564

slide by Fei Fei, Fergus & Torralba

Challenges 2: illumination



slide credit: S. Ullman

Challenges 3: occlusion

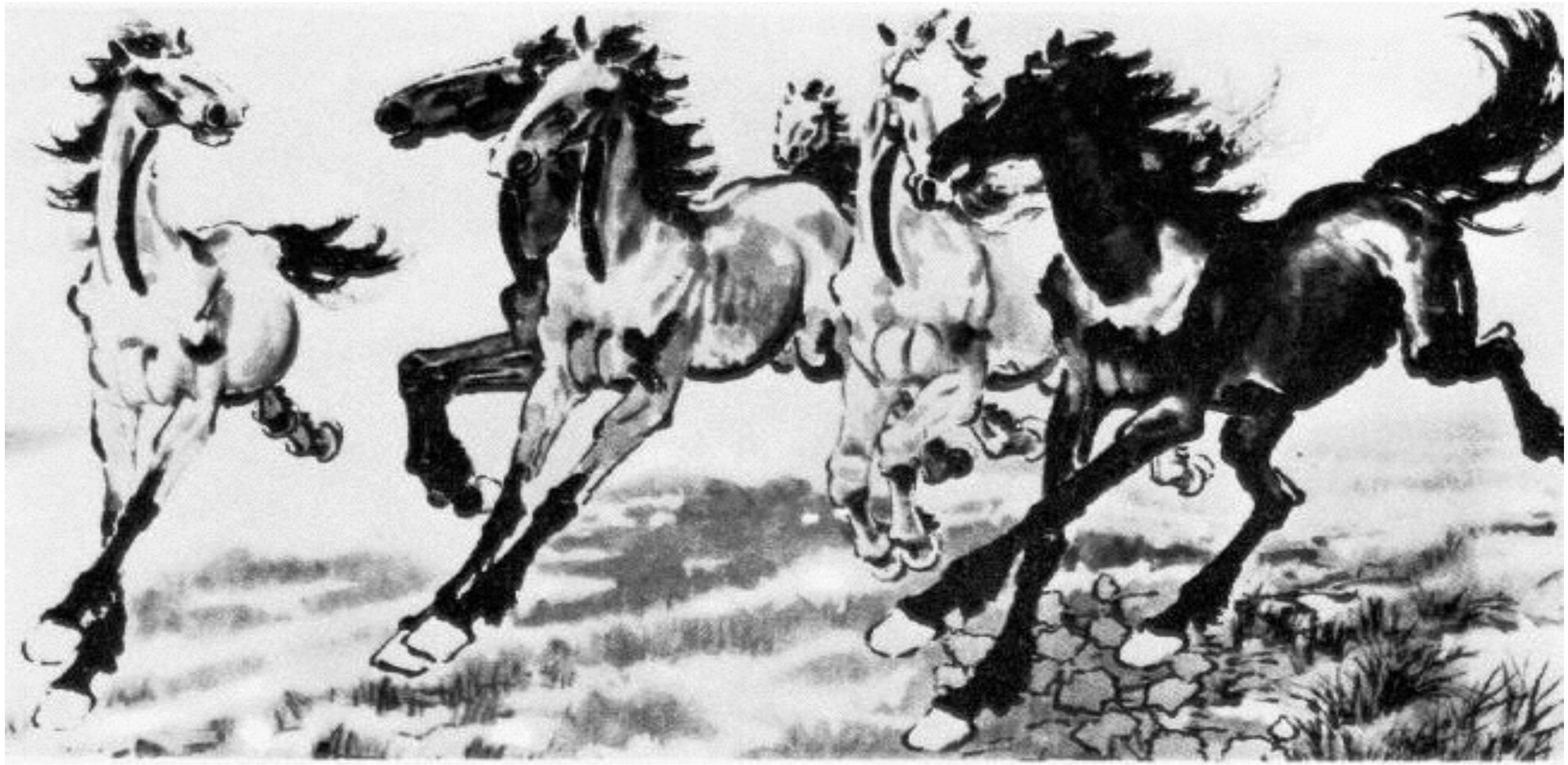
Magritte, 1957

slide by Fei Fei, Fergus & Torralba

Challenges 4: scale

slide by Fei Fei, Fergus & Torralba

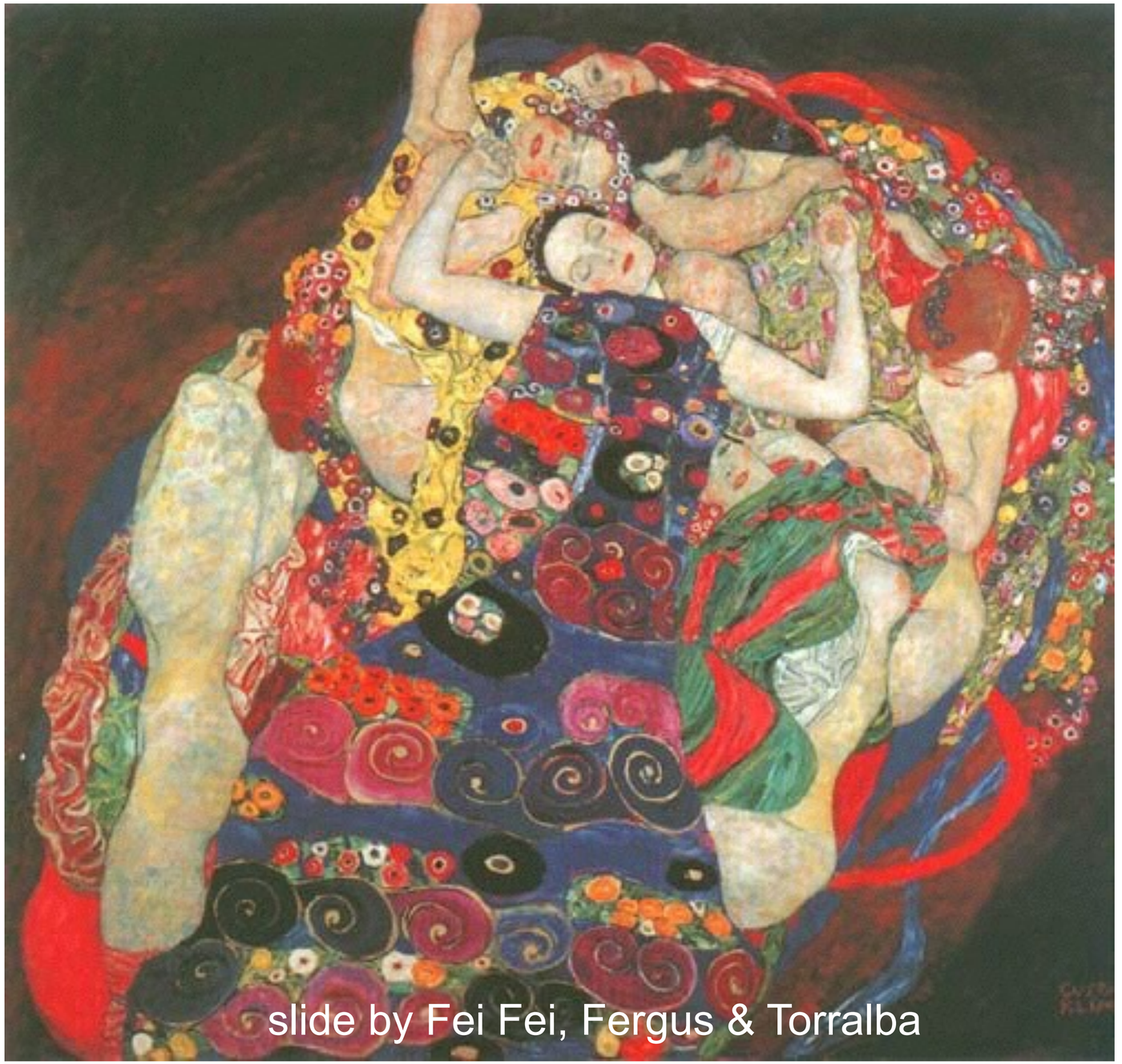
Challenges 5: deformation



slide by Fei Fei, Fergus & Torralba

Xu, Beihong 1943

Challenges 6: background clutter



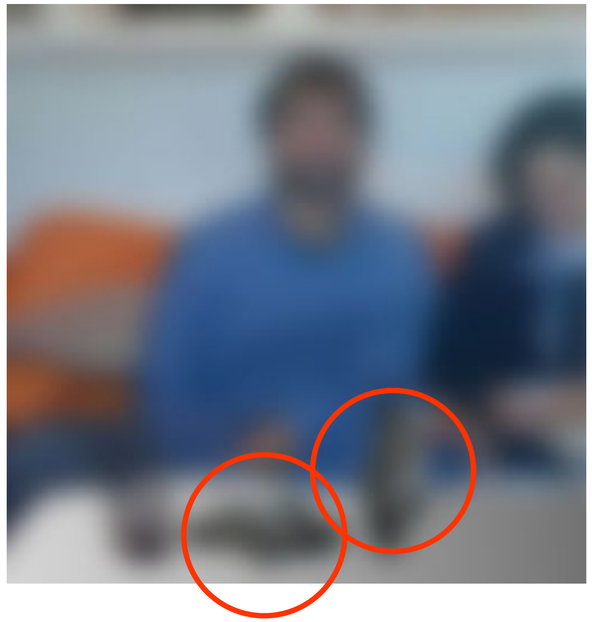
Klimt, 1913

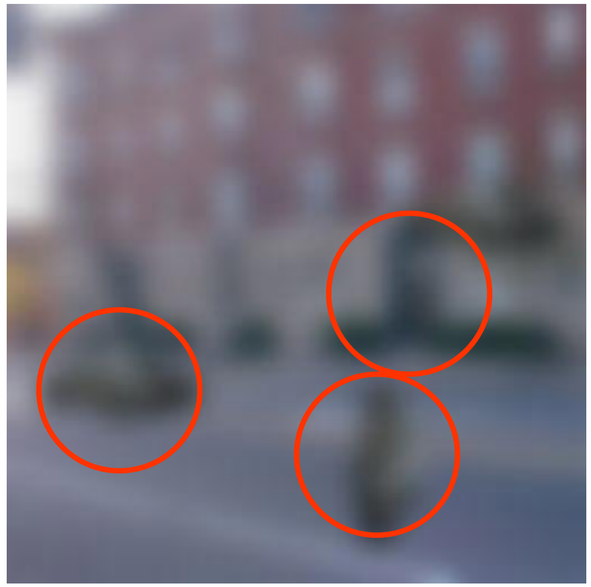
slide by Fei Fei, Fergus & Torralba

Challenges 7: object intra-class variation

slide by Fei-Fei, Fergus & Torralba

Challenges 8: local ambiguity





slide by Fei-Fei, Fergus & Torralba

Supervised Learning Paradigm

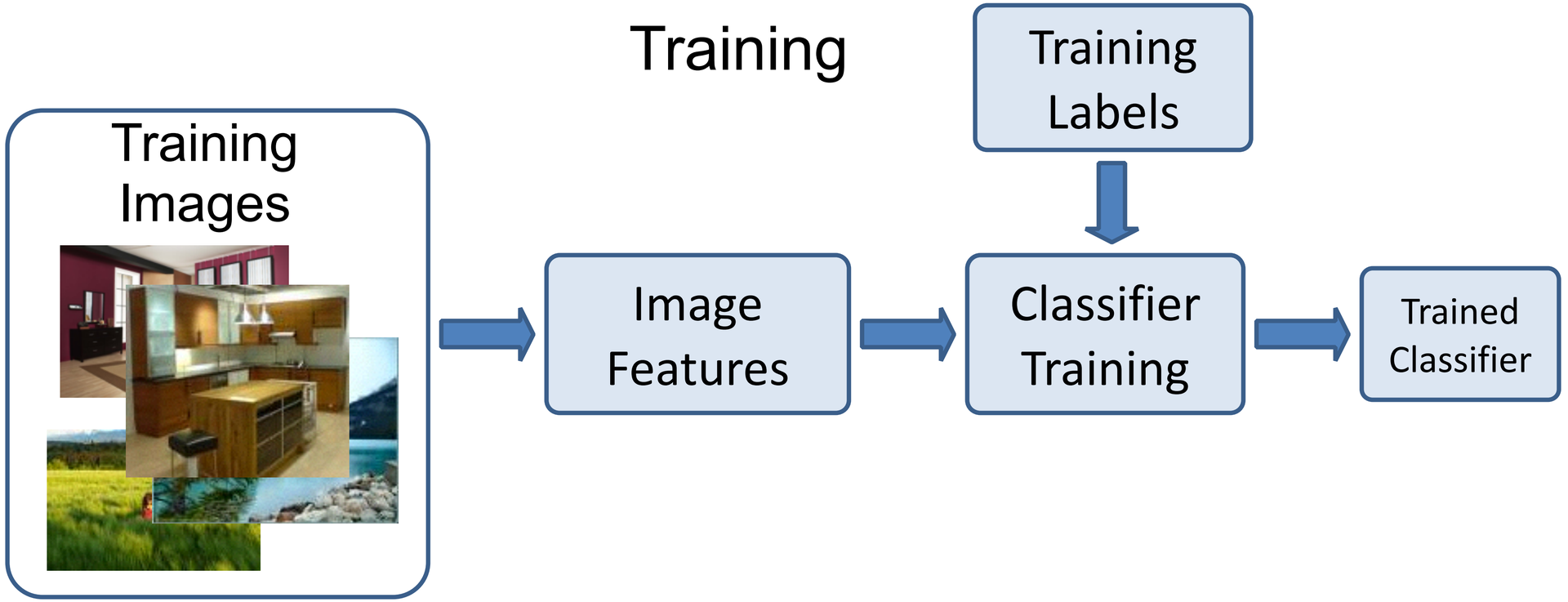
Positive Examples

Negative Examples

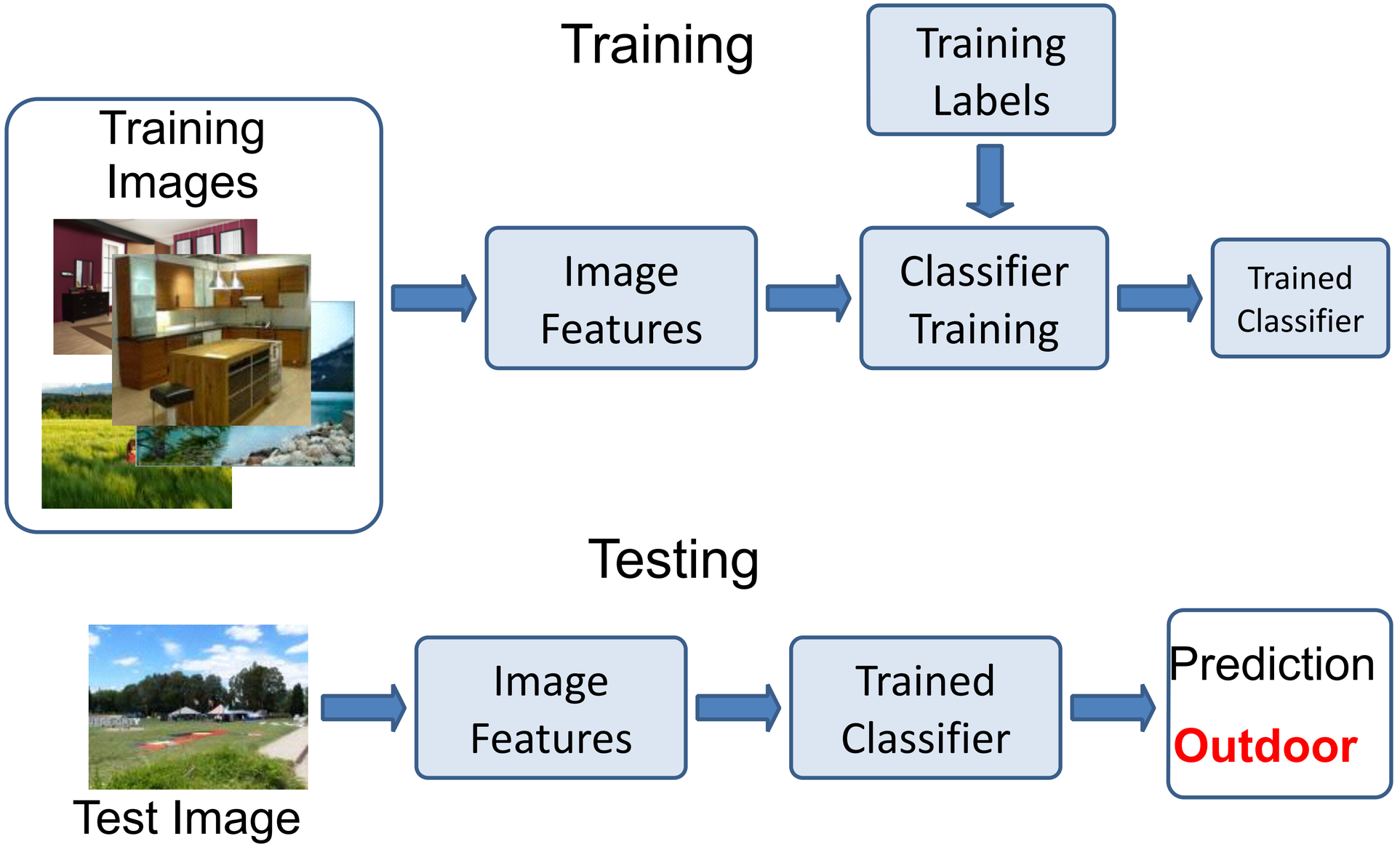
Training Time

Goal: Learn a function f(I), that given an input image I predicts whether it is a forest or not

Training phase

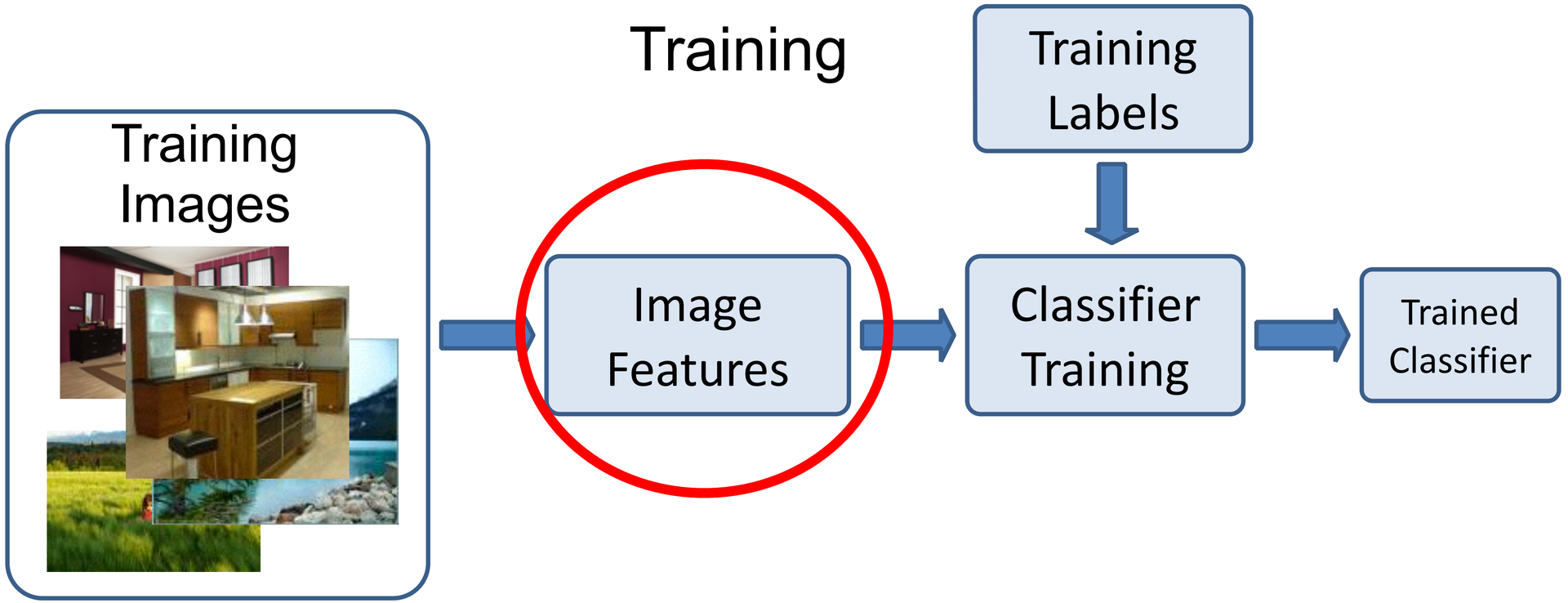


Testing phase



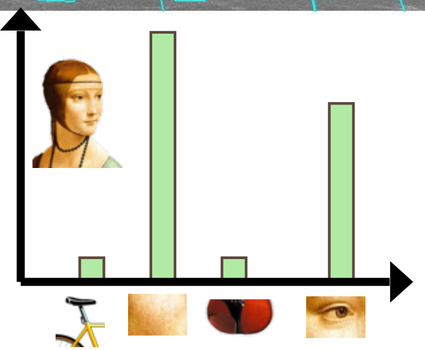
So, what matters in recognition algorithms?

Part I: Image features



Representation

- Pixels
- Edges
- SIFT
- Bag of Words
- ……



Google

medici_sketch.bmp × describe image here

Search

About 2 results (0.29 seconds)

Everything

Images

Maps

Videos

News

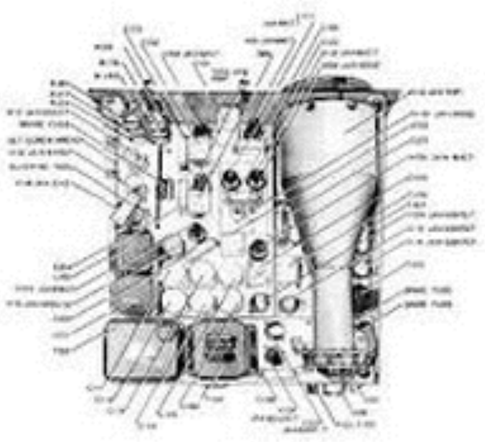
Shopping

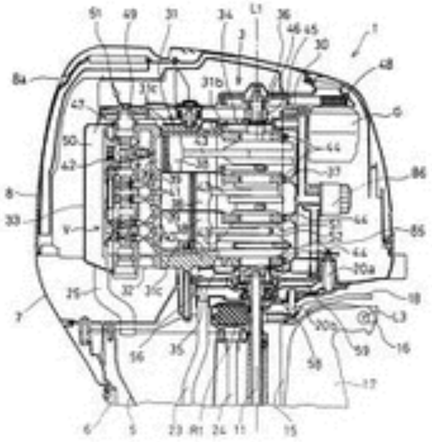
More

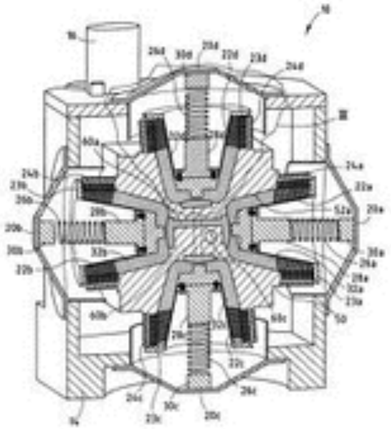
Image size:
443 × 482

No other sizes of this image found.

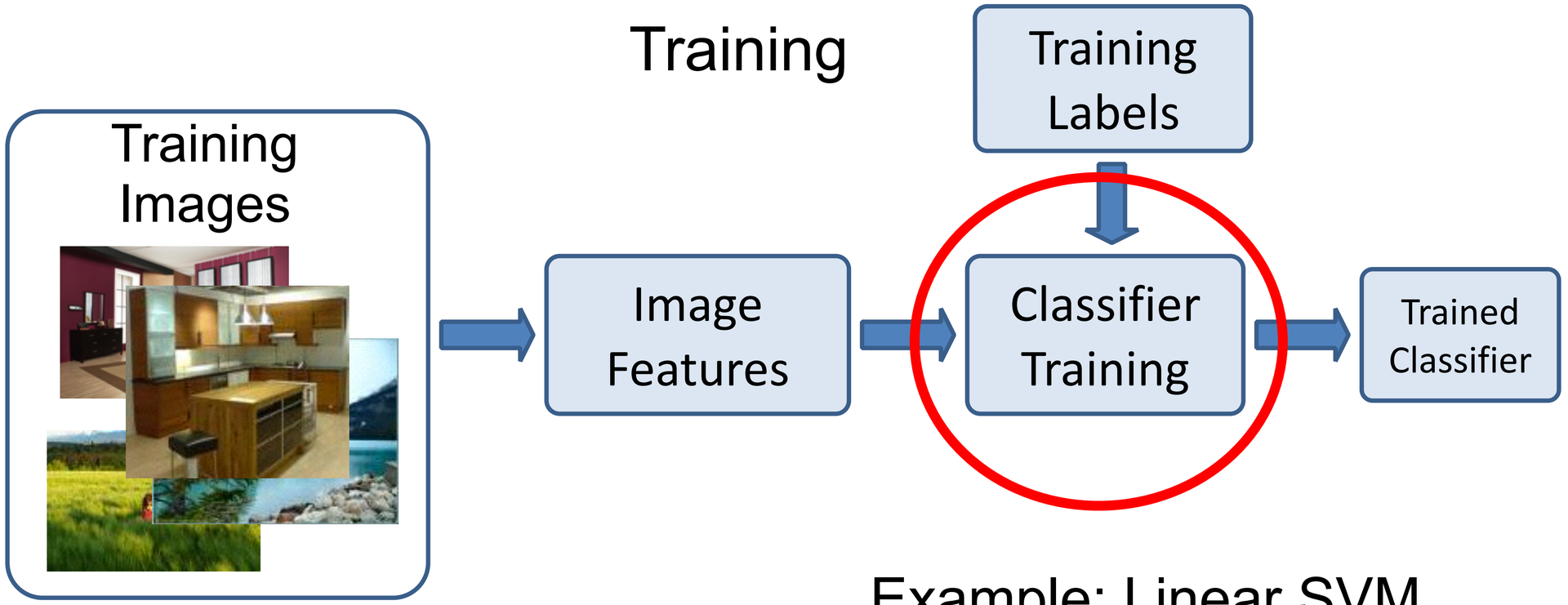
Visually similar





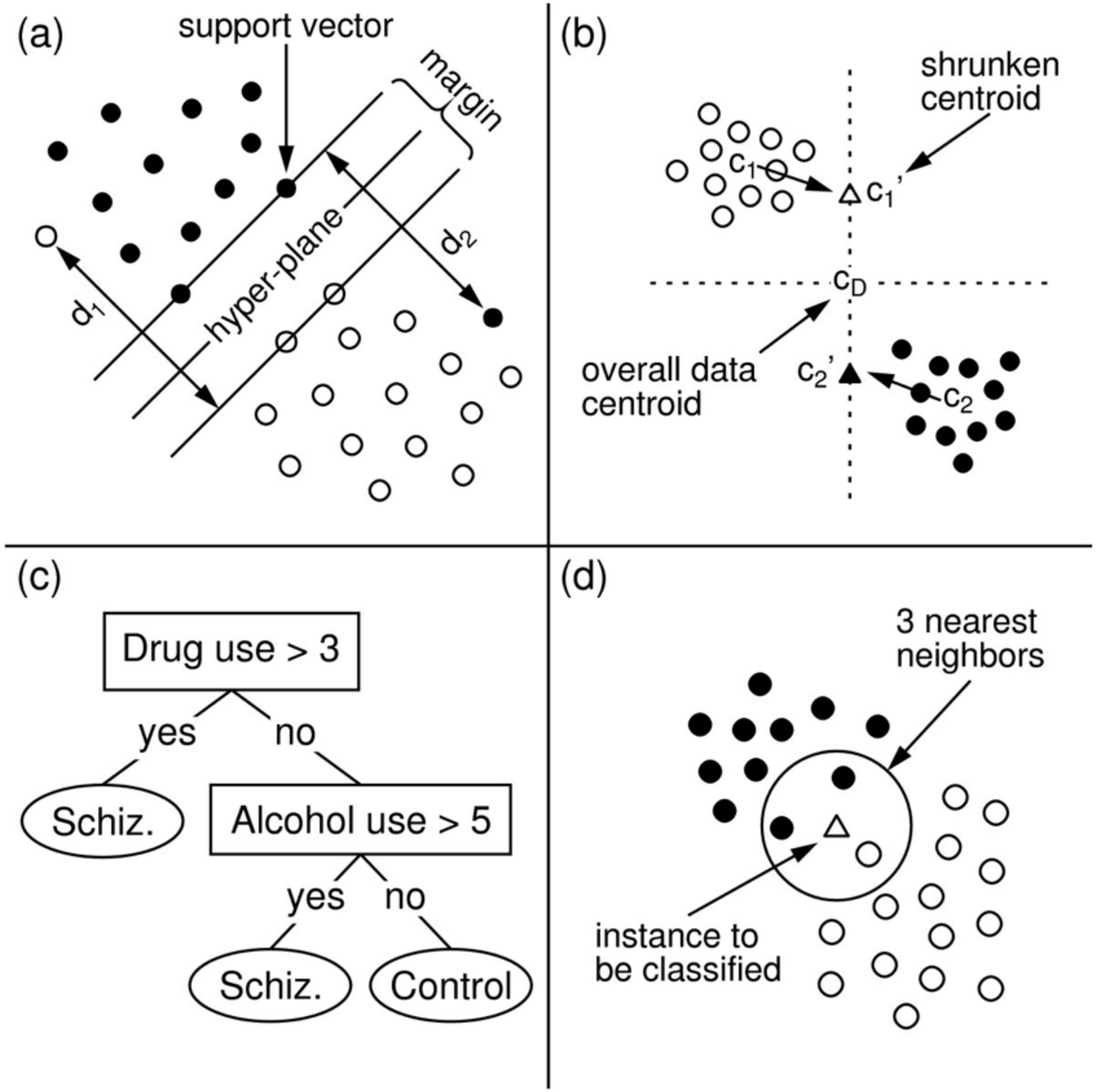


Part 2: Classifiers



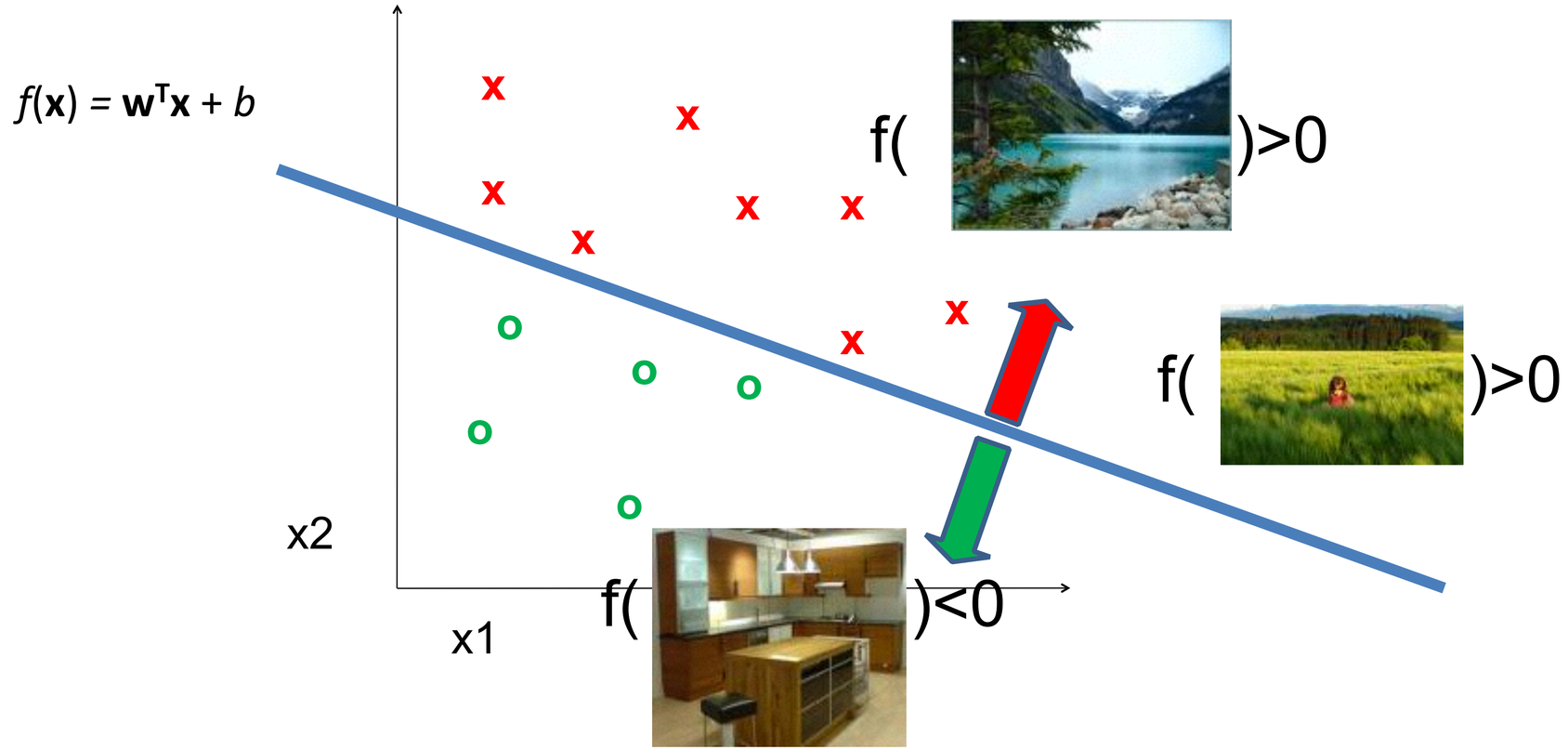
Example: Linear SVM

Learning Techniques

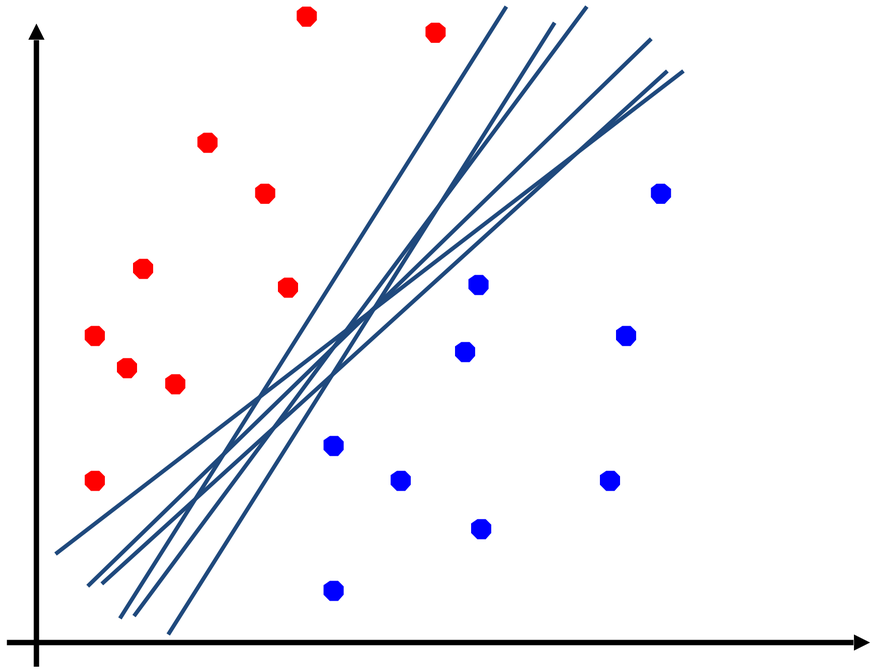


Choice of classifier or inference method

Linear classifier



Linear Separators

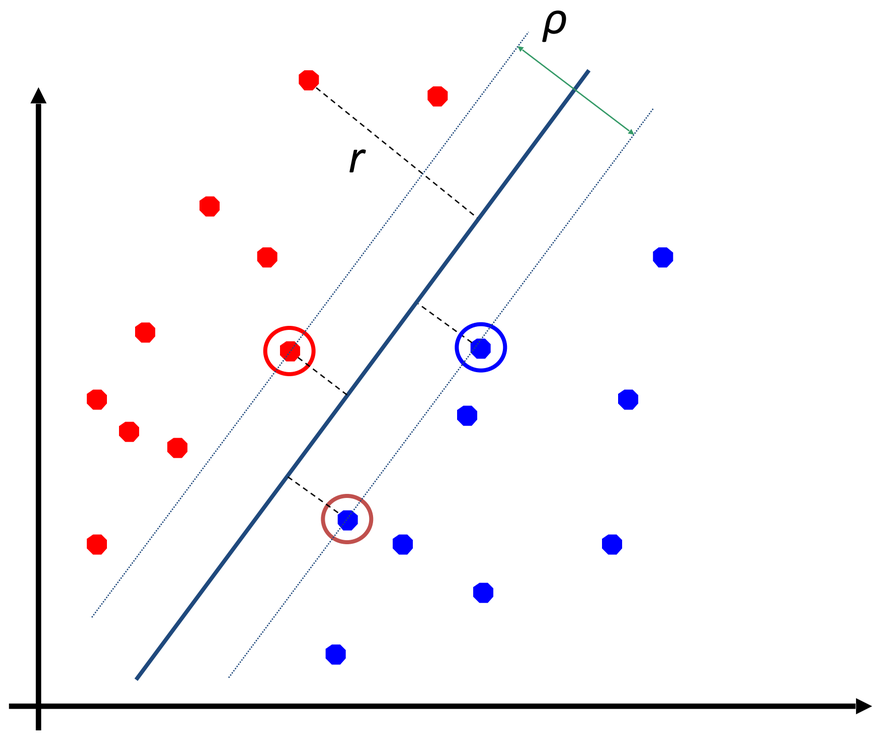


http://www.cs.utexas.edu/~mooney/cs391L/slides/**svm.ppt**

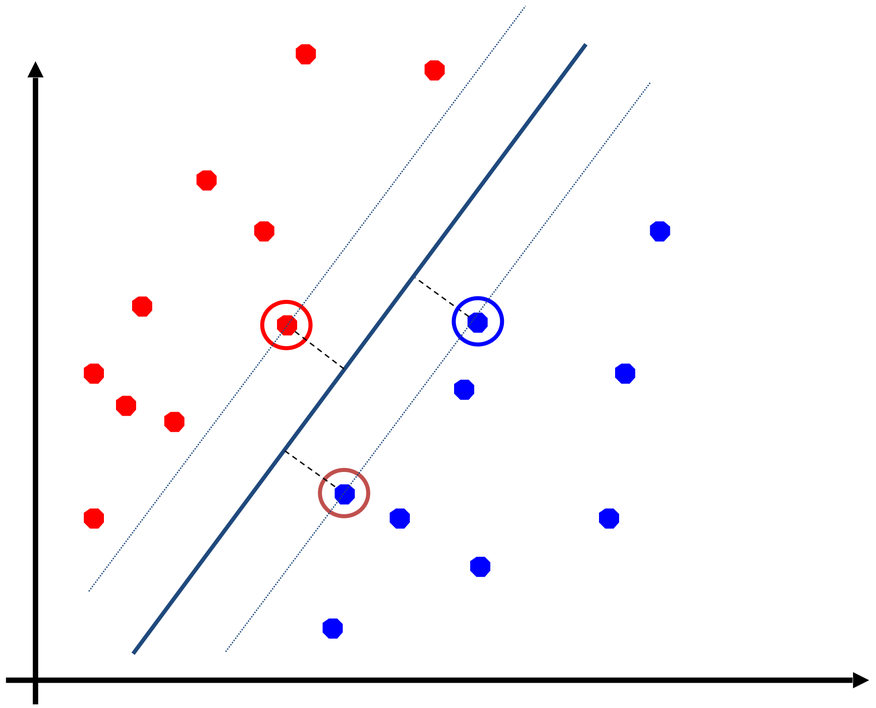
http://www1.cs.columbia.edu/~belhumeur/courses/biometrics/2010/svm.ppt

Classification Margin

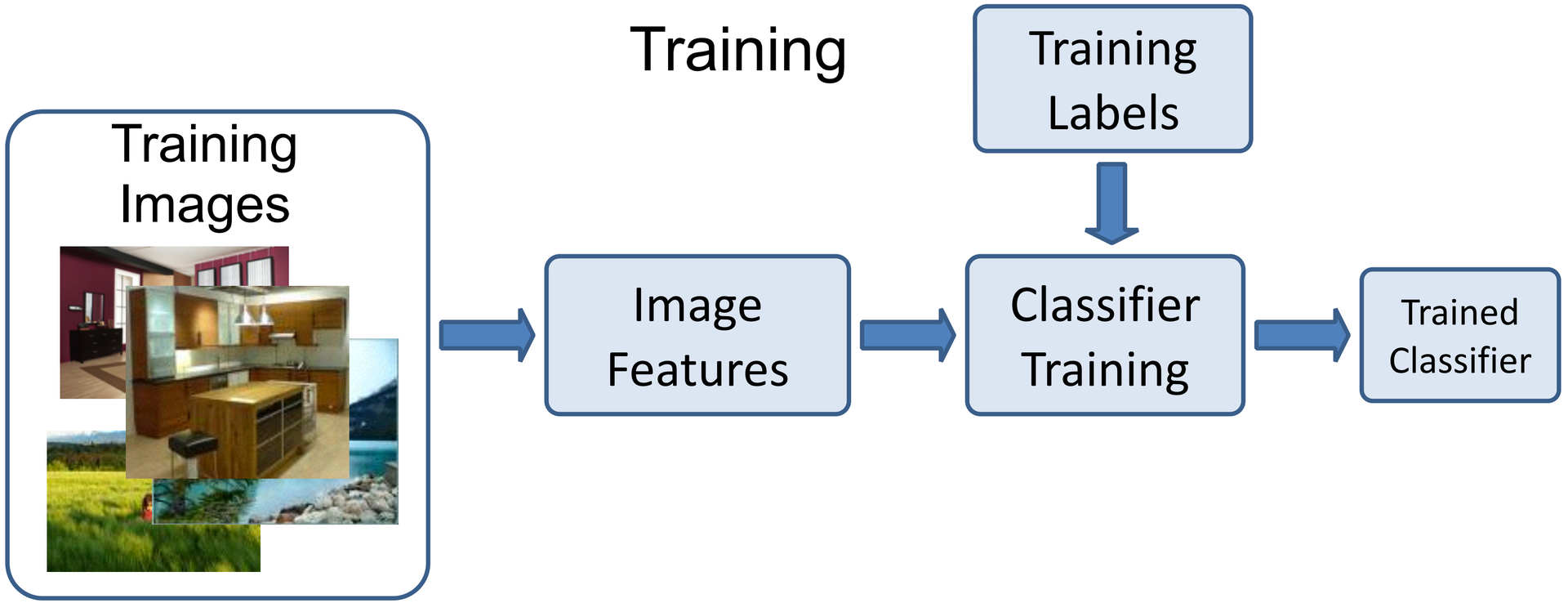
$$r = \frac{\mathbf{w}^T \mathbf{x}_i + b}{\|\mathbf{w}\|}$$



Maximum Margin Classification



Part 3: Data


Training
Training Images
Training Labels
Image Features
Classifier Training
Trained Classifier

Data

Testing Data

Training Data

How much data is enough? How many images do you need to learn the appearance of car?

Datasets in perspective

Number of images on my hard drive: 10^3

Number of images seen during my first 10 years: 10^8
(3 images/second * 60 * 60 * 16 * 365 * 10 = 630720000)

Number of images seen by all humanity: 10^{20}
106,456,367,669 humans[1] * 100 years * 3 images/second * 60 * 60 * 16 * 365 =
1 from http://www.prb.org/Articles/2002/HowManyPeopleHaveEverLivedonEarth.aspx

Number of all 32x32 images: 10^{7373}
$256^{32*32*3} \sim 10^{7373}$

Data

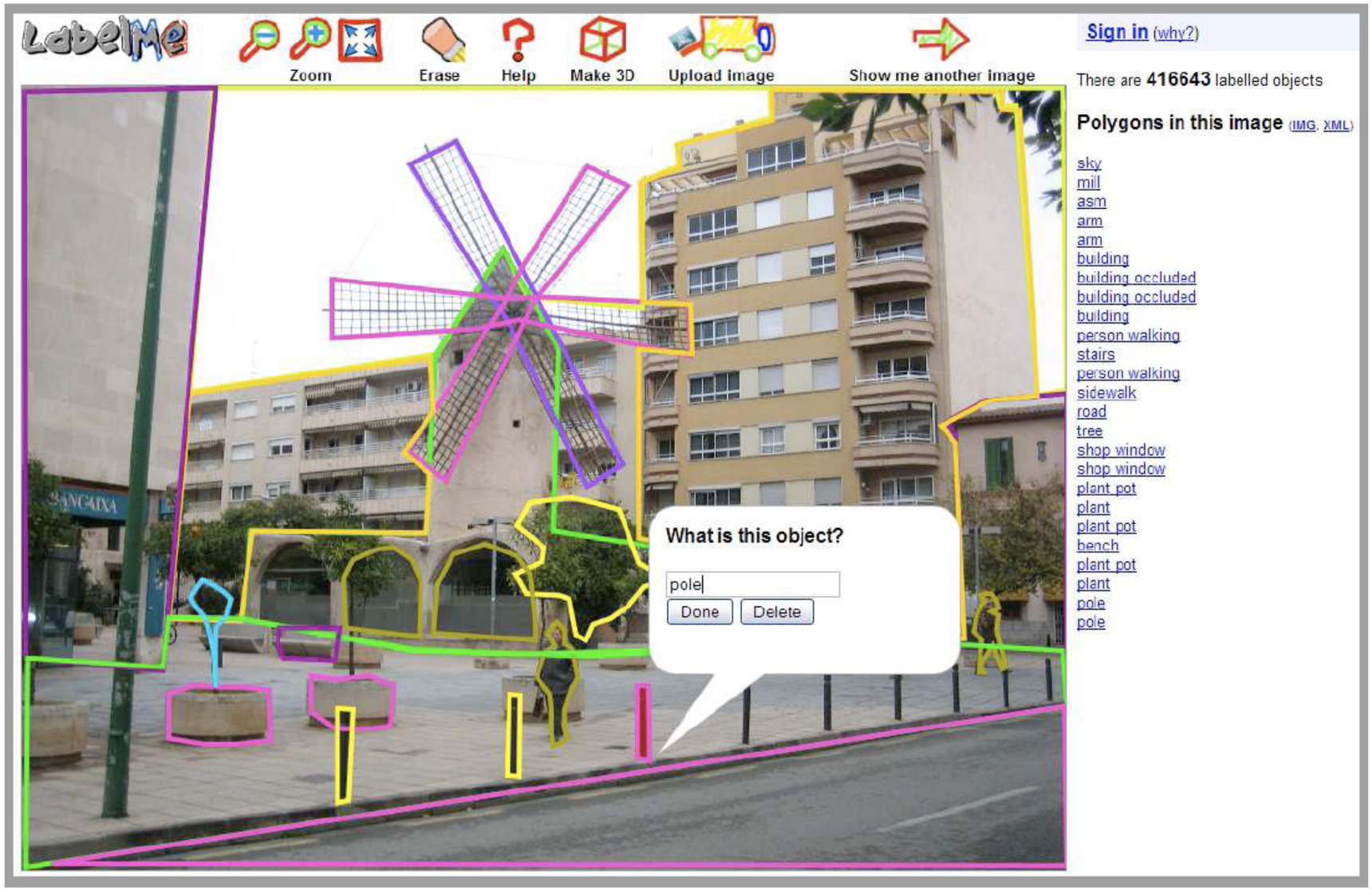
Testing Data

Training Data

How much data is enough? How many images do you need to learn the appearance of car?

- The more the better
- But data requires annotations…

Creating Data : LabelMe



Tool went online July 1st, 2005
530,000 object annotations collected

Labelme.csail.mit.edu

B.C. Russell, A. Torralba, K.P. Murphy, W.T. Freeman, IJCV 2008

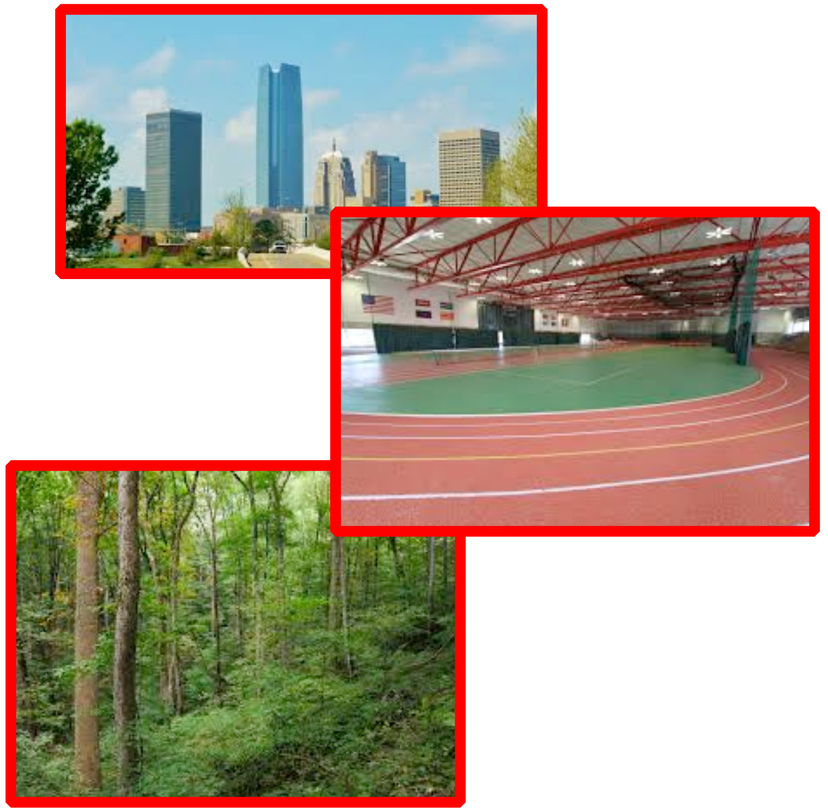
What Matters in Recognition?

Data

Learning

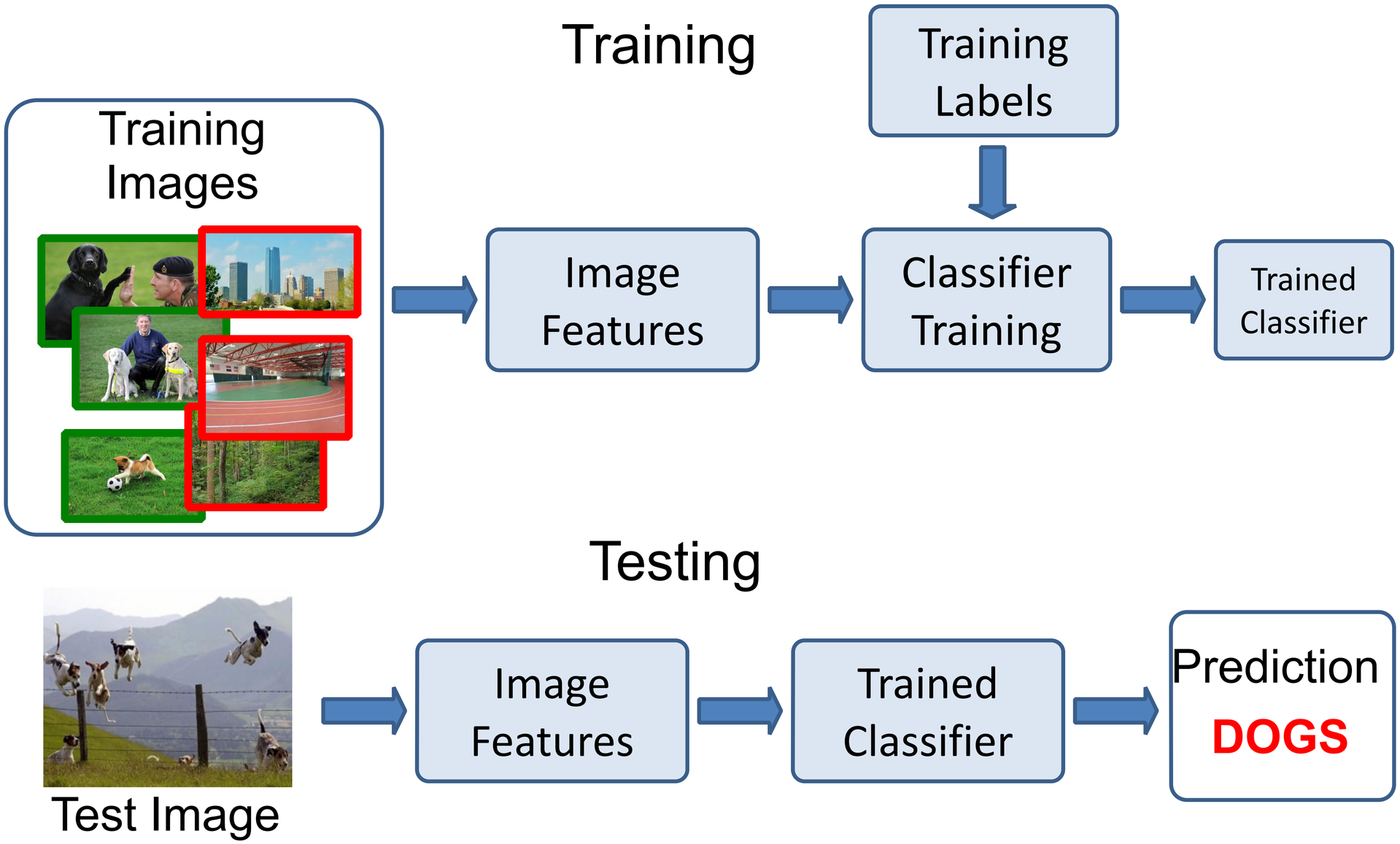
Representation

From Scenes to Objects

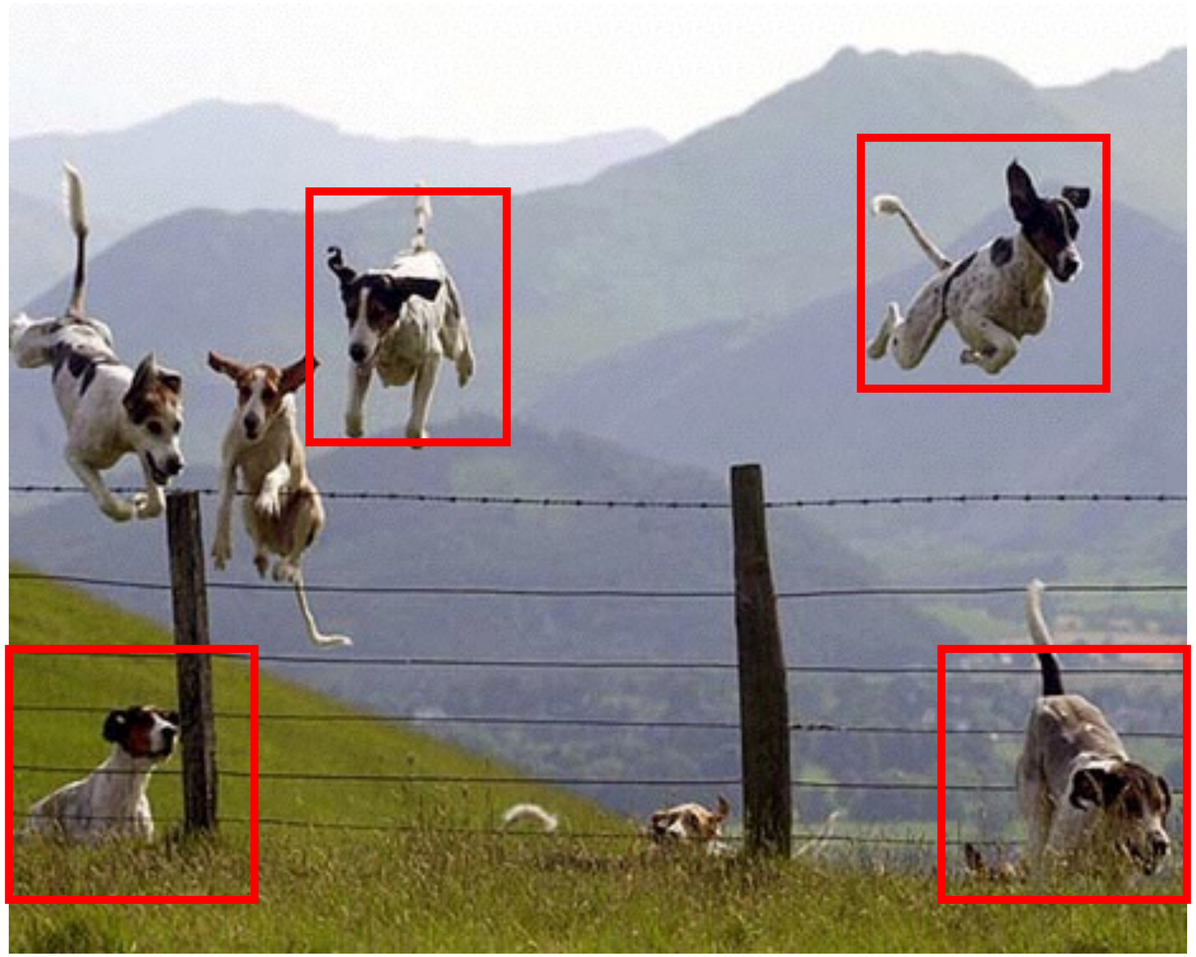


Classifier: Is there a dog in the image?

Same Pipeline!



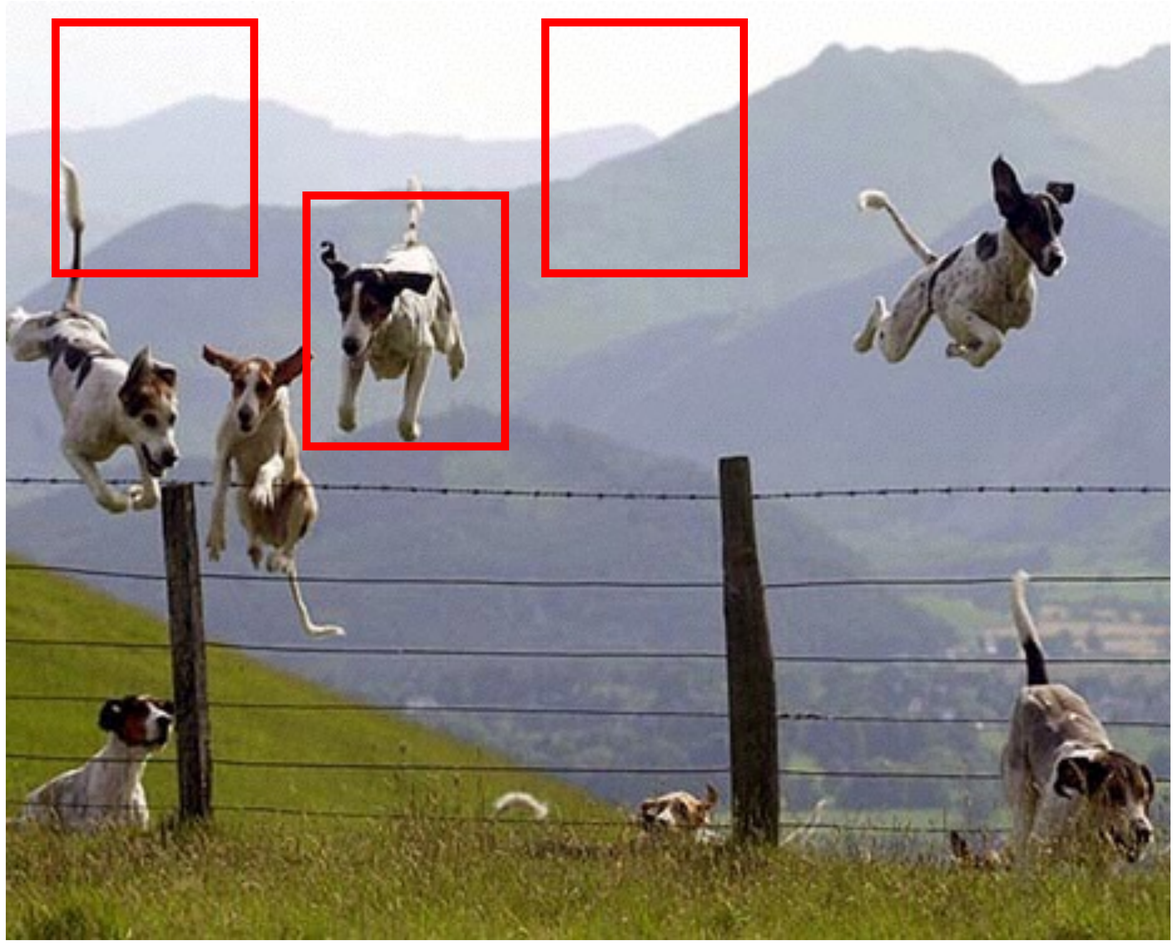
But

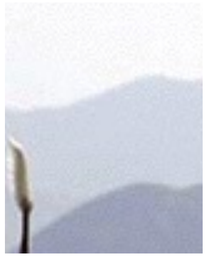


“Where is it?”

Idea

- How about we classify each patch in the image?



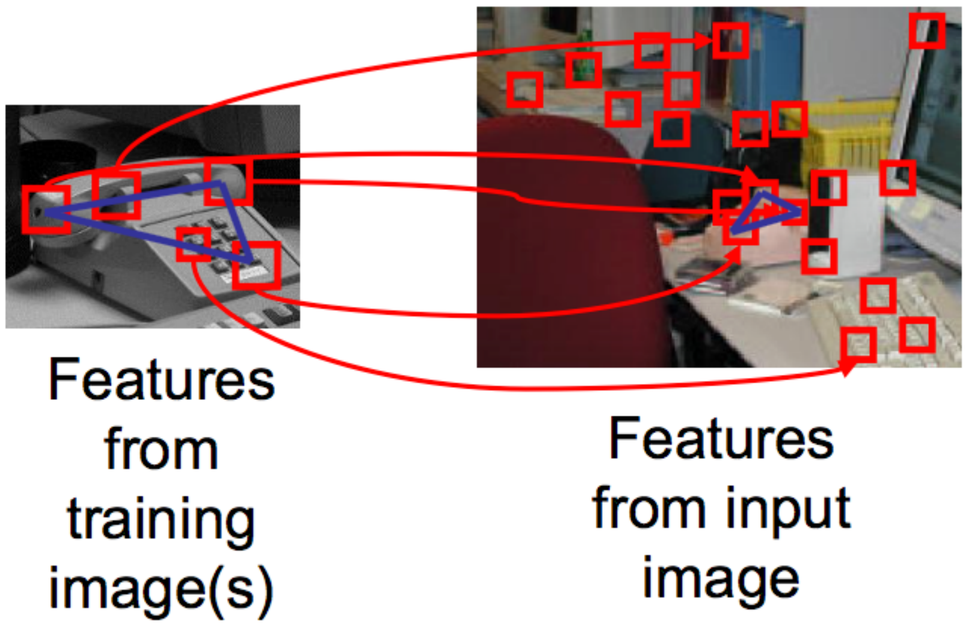


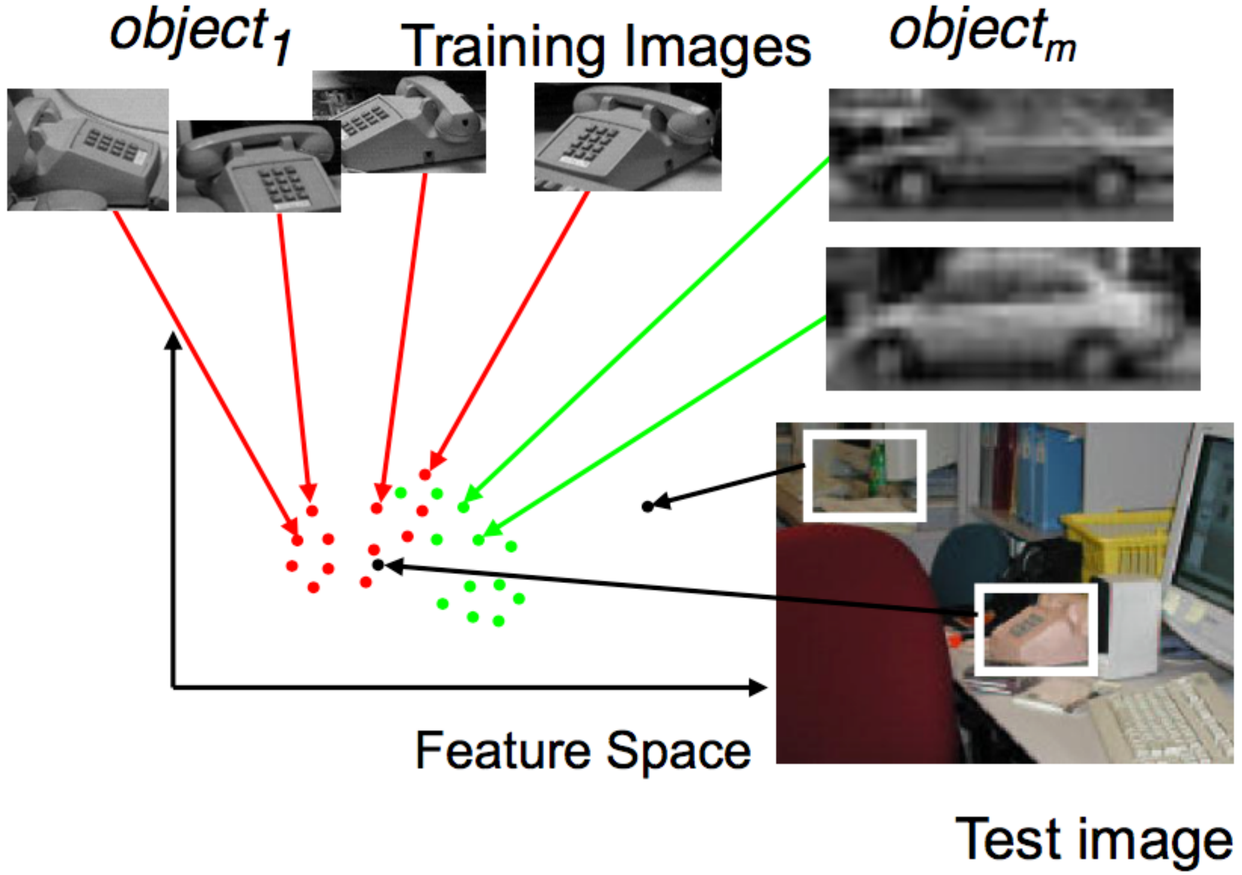
No Dog

Dog

No Dog

2 Common Approaches



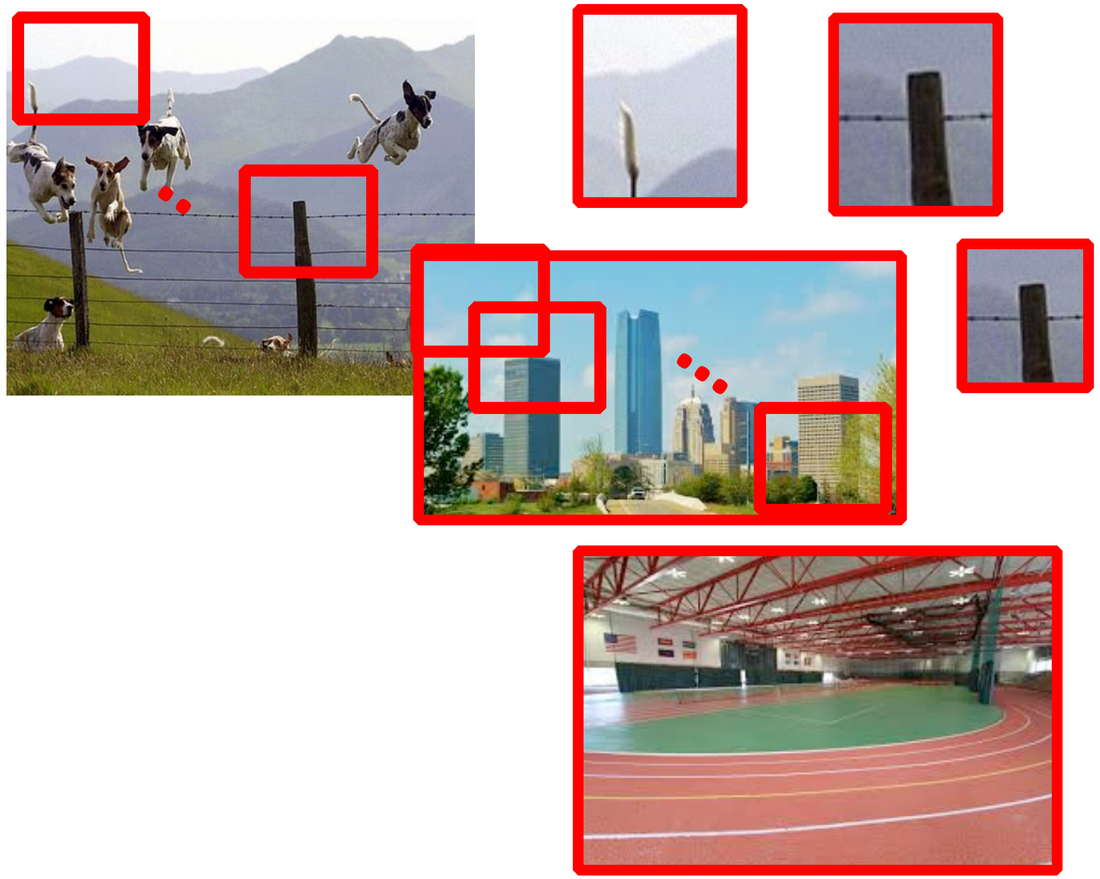


Approaches based on using feature matches and geometric relations

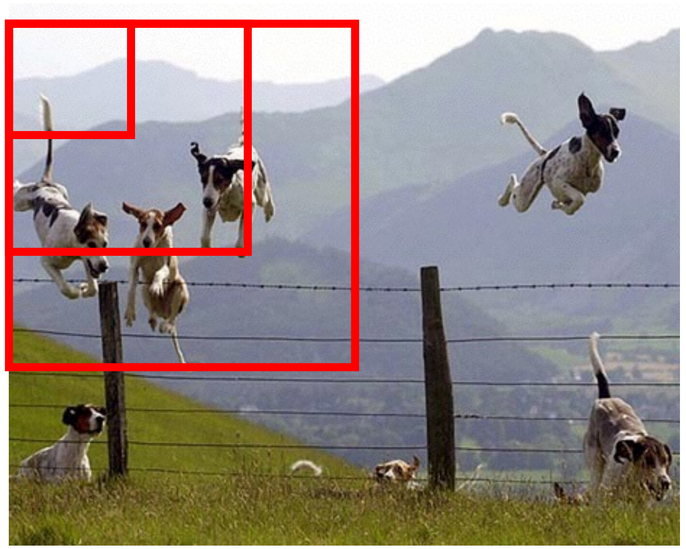
Approaches based on classifying/matching image patches (windows)

Training Time

- Build templates that quickly differentiate object patch from background patch

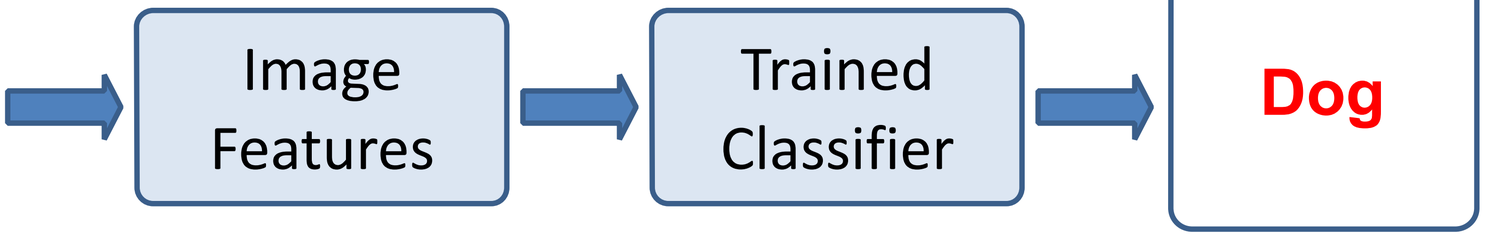


Sliding Window



Test patch at each location and scale

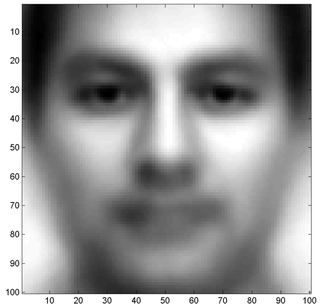
Test Patch



Given an image and a template-face how do I find the faces?

400 × 200
(RGB)

+



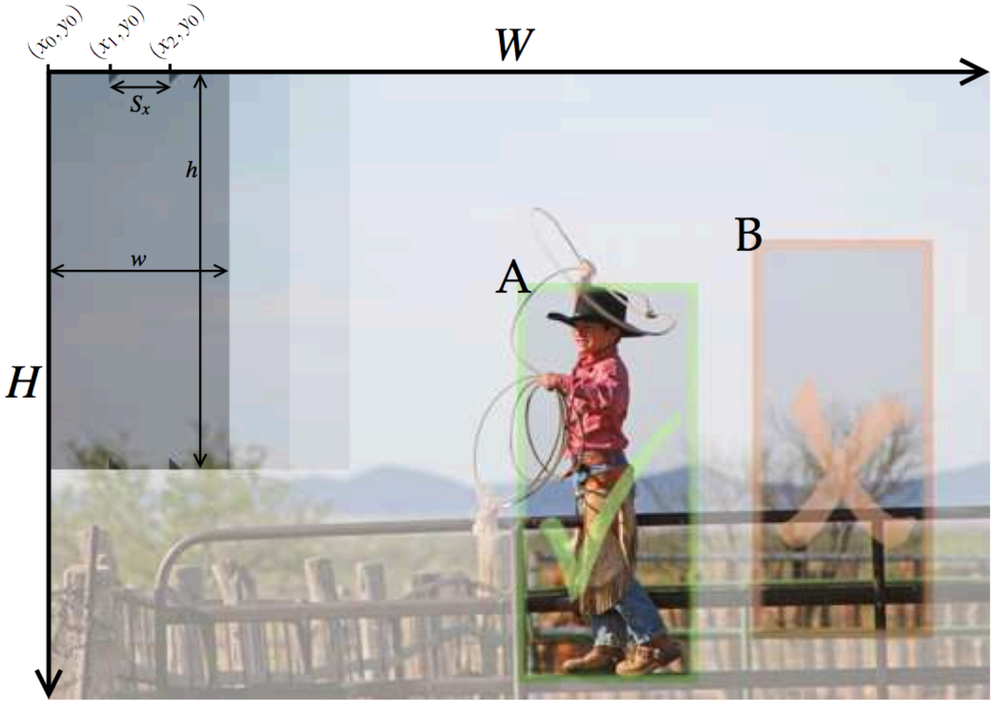
100 × 100

+

Finding Faces in an Image

- Picture is larger than the face template
  - E.g. face template is 100x100, picture is 600x800
- First convert to greyscale
  - R + G + B
  - Not very useful to work in color

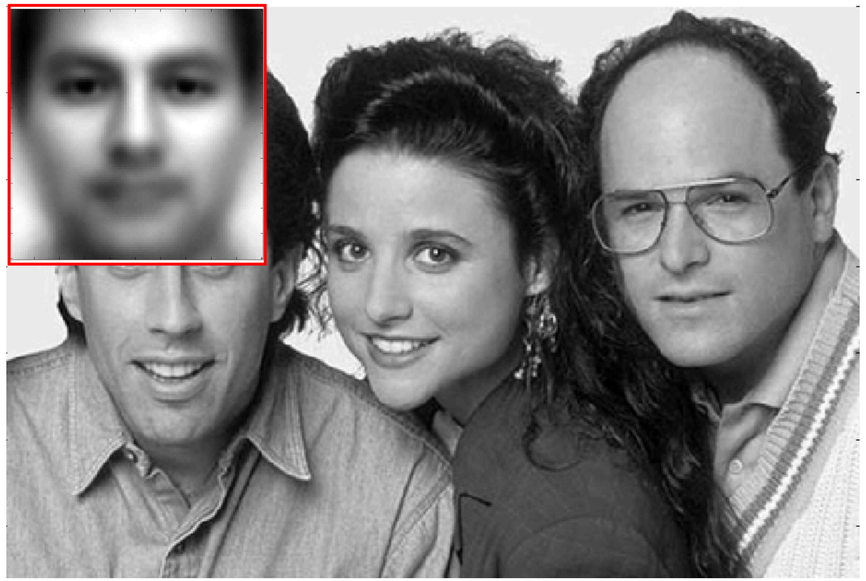
Sliding Windows



slidingWindows$(I, W, H, w, h, S_x, S_y)$
  for $y = 0, S_y, 2S_y, 3S_y, \ldots, H - h$
    for $x = 0, S_x, 2S_x, 3S_x, \ldots, W - w$
      Query Pedestrian at I(x,y)
    endfor
  endfor

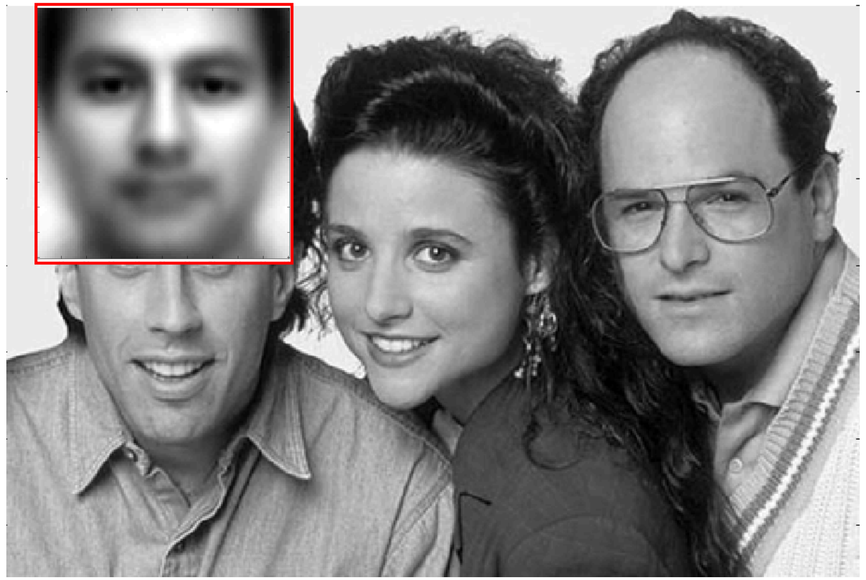
Figure 1.3: The Sliding Windows Methodology. In order to find an object instance, a detector must be run over multiple sub-regions of the image. Consequently the detector must use a minimal amount of processing for each individual window. To the right of the main image we show the basics of a sliding windows algorithm. The algorithm takes as parameters, the input image I, its associated width W and height H, the width w and height h of the detection window, and the windowing step sizes S_x and S_y. For exhaustive searching of the image a step size of 1 is used in both dimensions.

Finding Faces in an Image



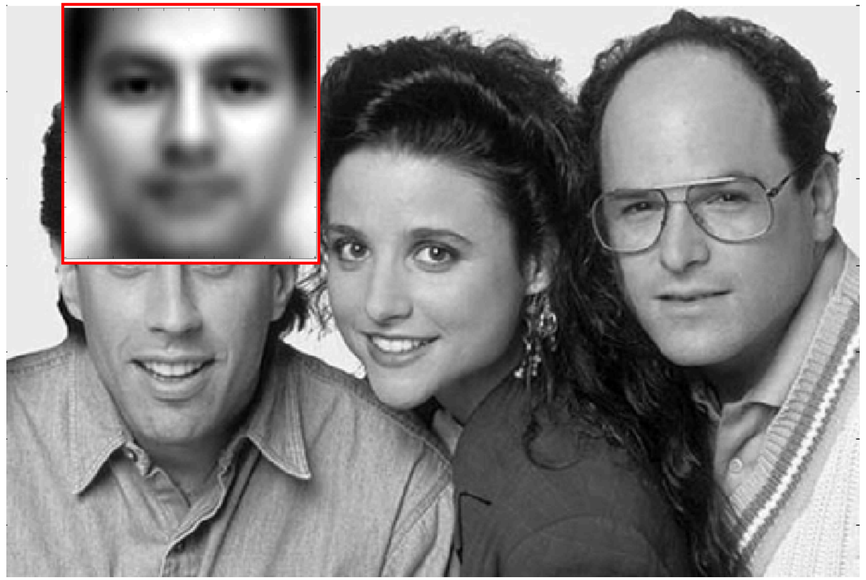
- Try to “match” the face template at each location in the larger picture

Finding Faces in an Image



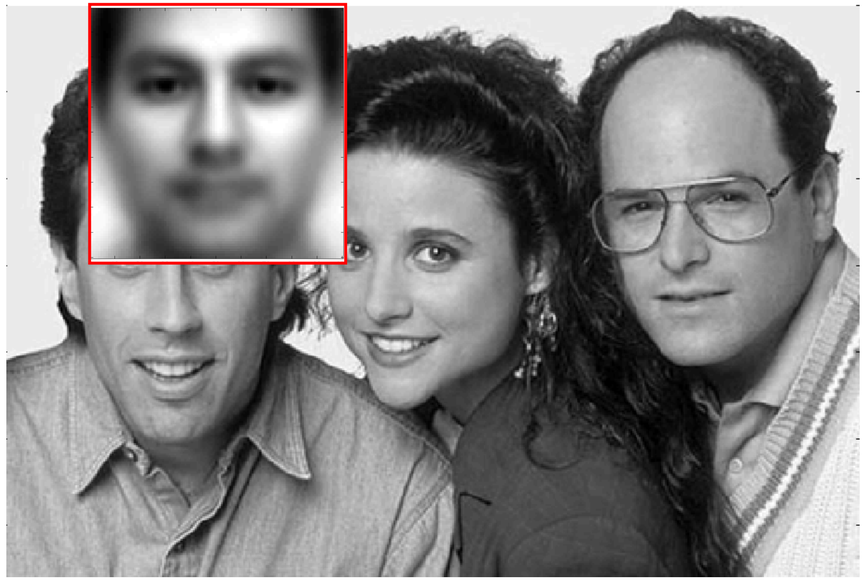
- Try to “match” the face template at each location in the larger picture

Finding Faces in an Image



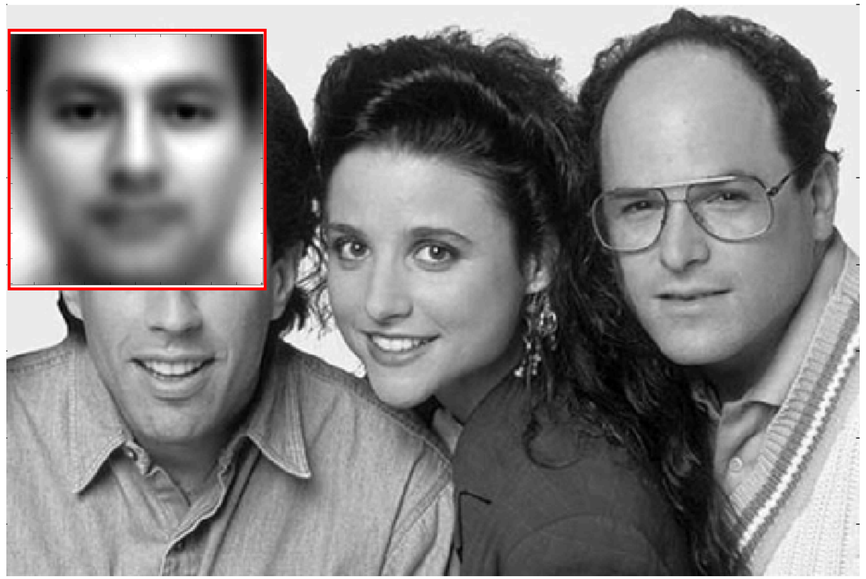
- Try to “match” the face template at each location in the larger picture

Finding Faces in an Image



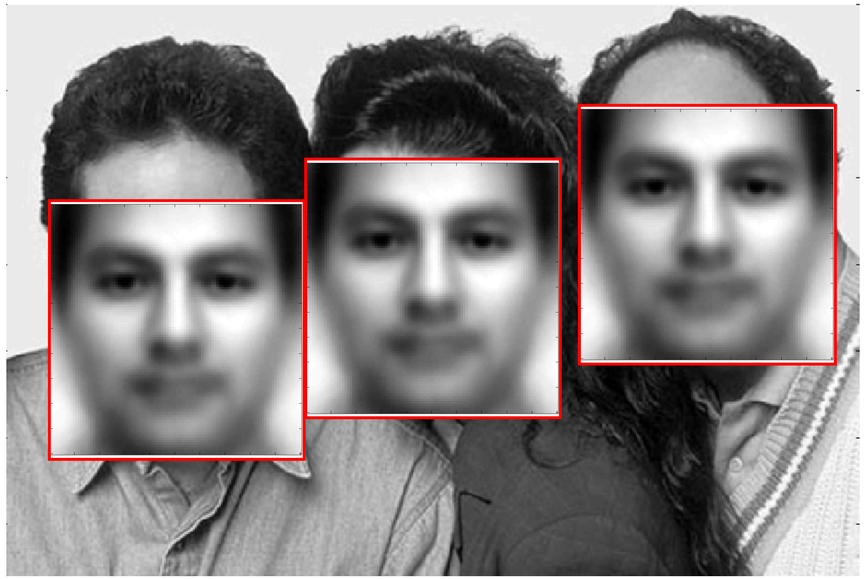
- Try to “match” the face template at each location in the larger picture

Finding Faces in an Image



- Try to “match” the face template at each location in the larger picture

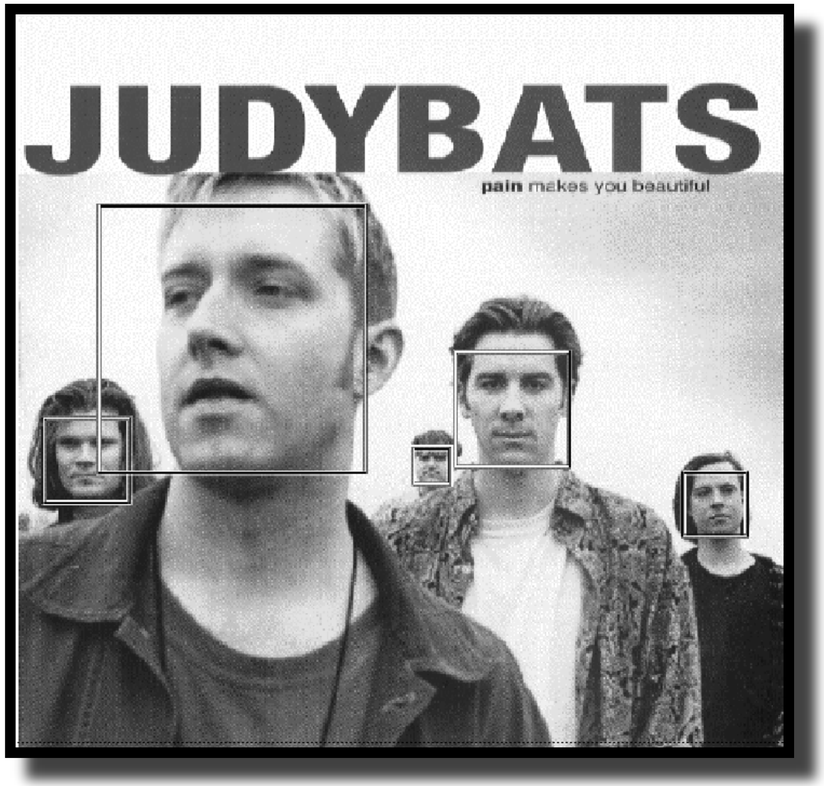
Finding Faces in an Image



- Try to “match” the face template at each location in the larger picture

Sliding windows solves only the issue of location – what about scale?

- Not all faces are the same size
- Some people have bigger faces
- The size of the face on the image changes with perspective
- Our “typical face” only represents one of these sizes



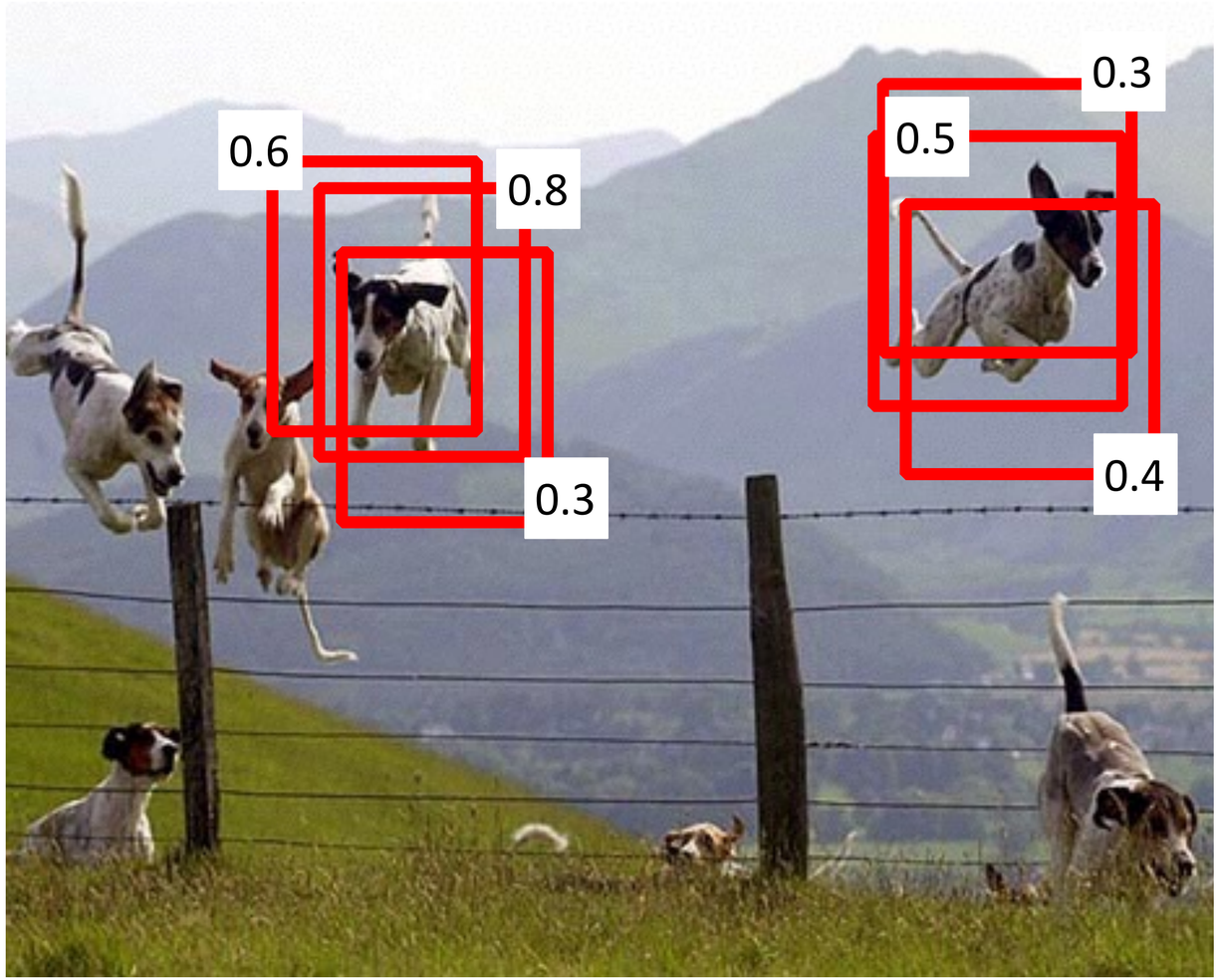
Scale-Space Pyramid

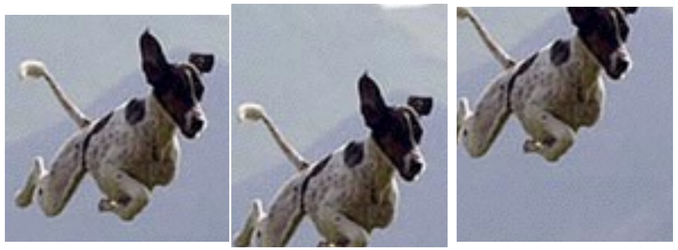
Figure 1.4: The Scale-Space Pyramid. The detector is run using the sliding windows approach over the input image at various scales. When the scale of the person matches the detector scale the classifier will (hopefully) fire yielding an accurate detection.

Speed concerns

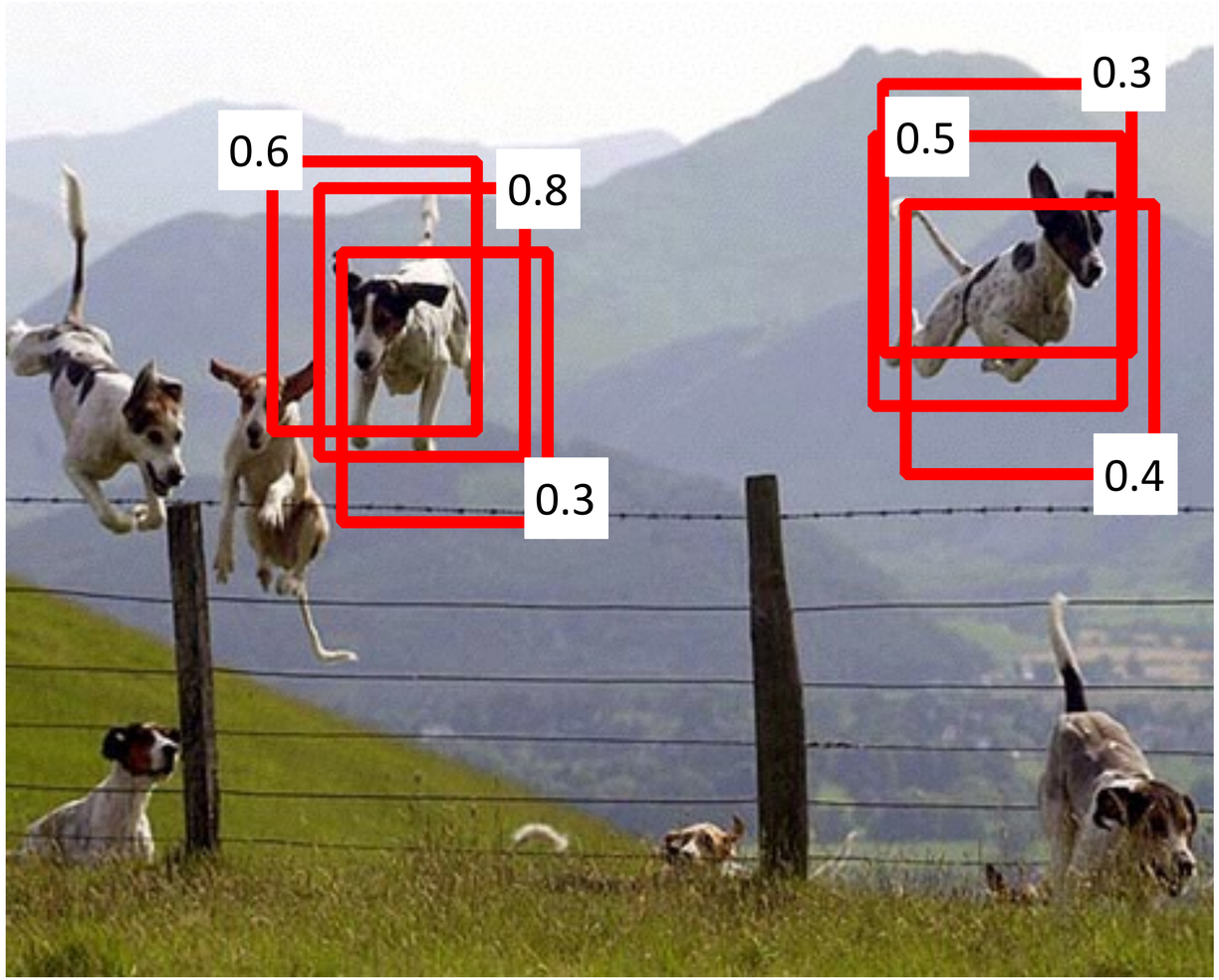
- Not all faces are the same size
- Some people have bigger faces
- The size of the face on the image changes with perspective
- Our face template only represents one of these sizes

Issues: Multiple Detections



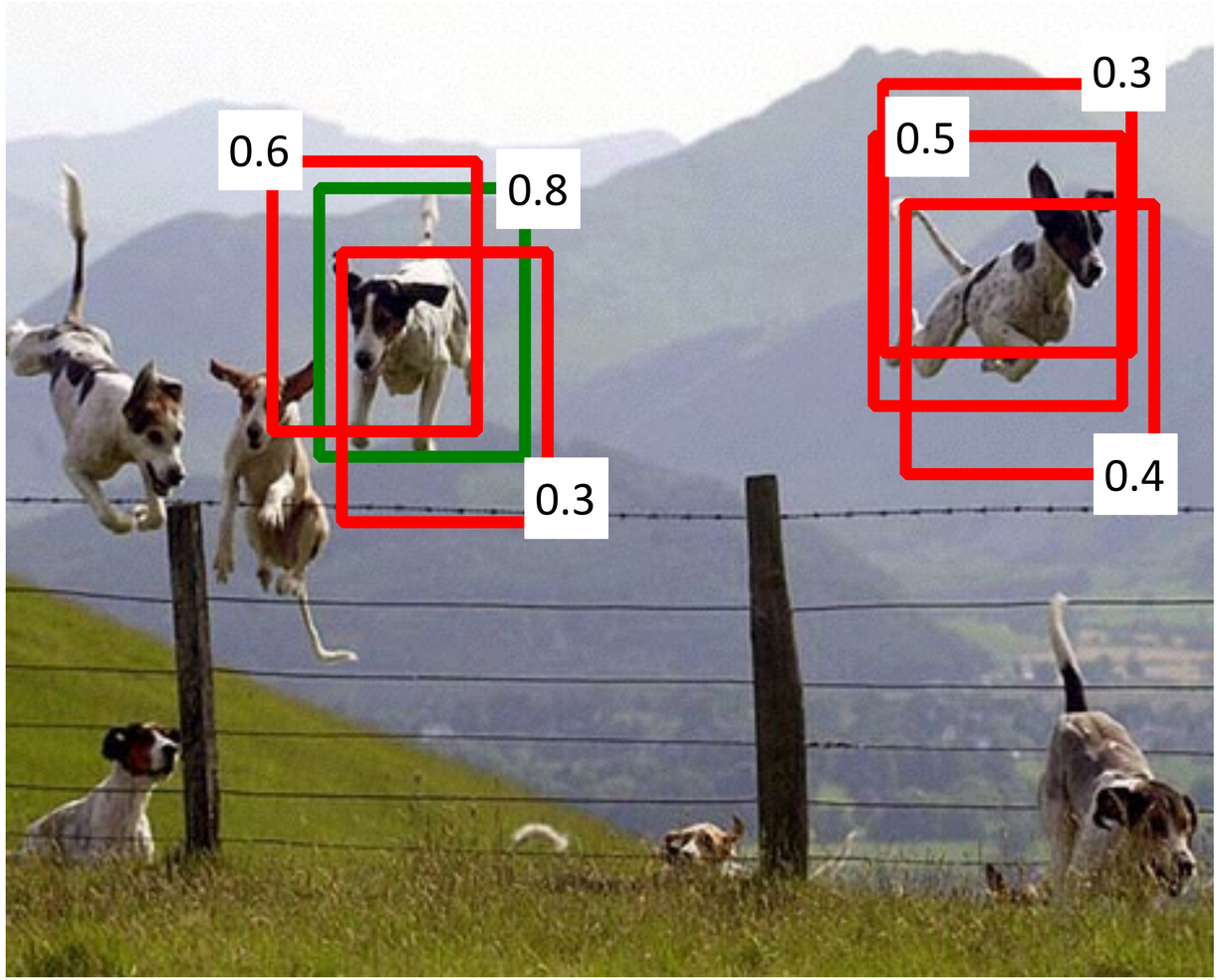


Greedy Non-max



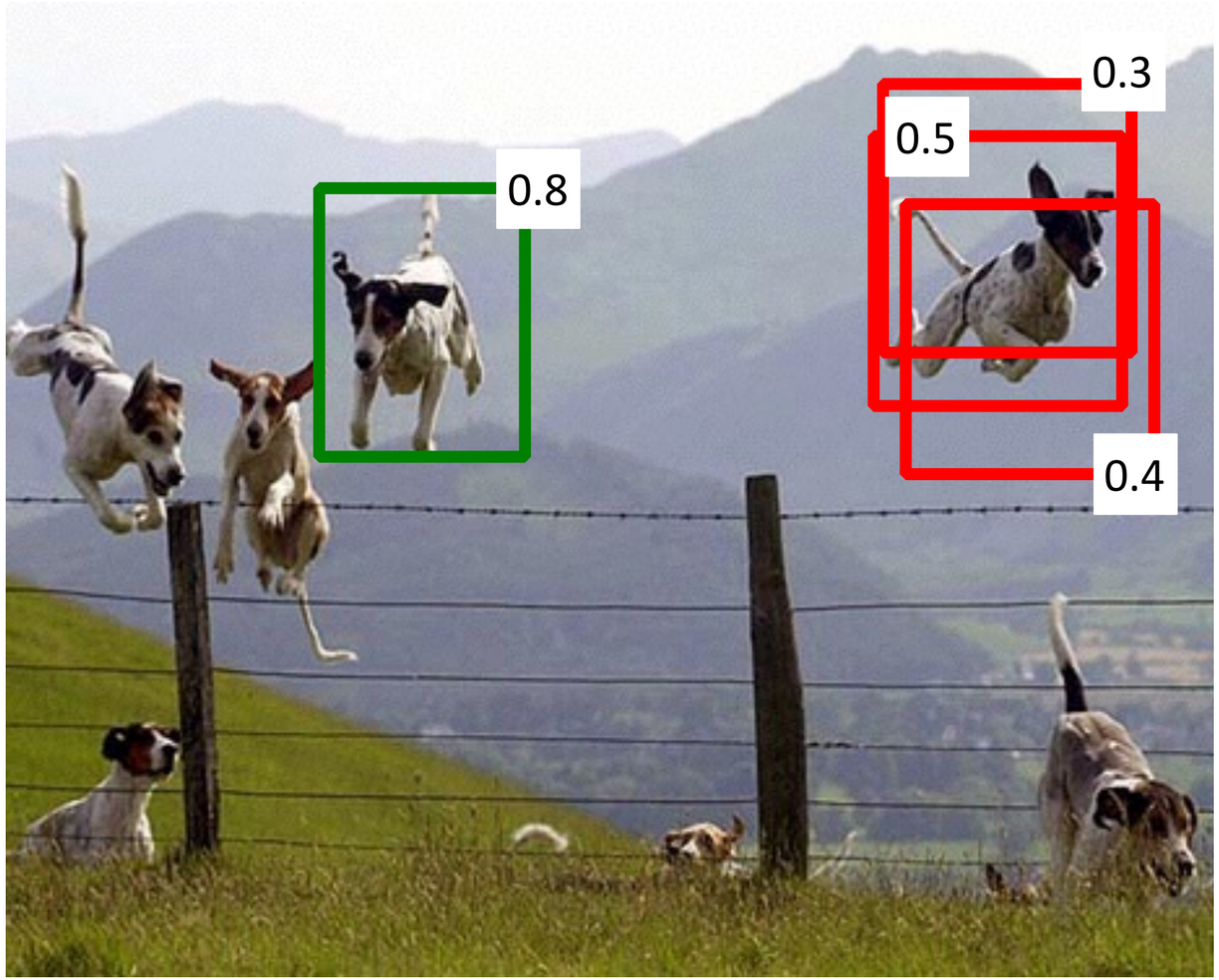
- Choose the window with maximum score.
- Reject the windows with high overlap with selected window.
- Repeat until all windows selected or rejected

Greedy Non-max



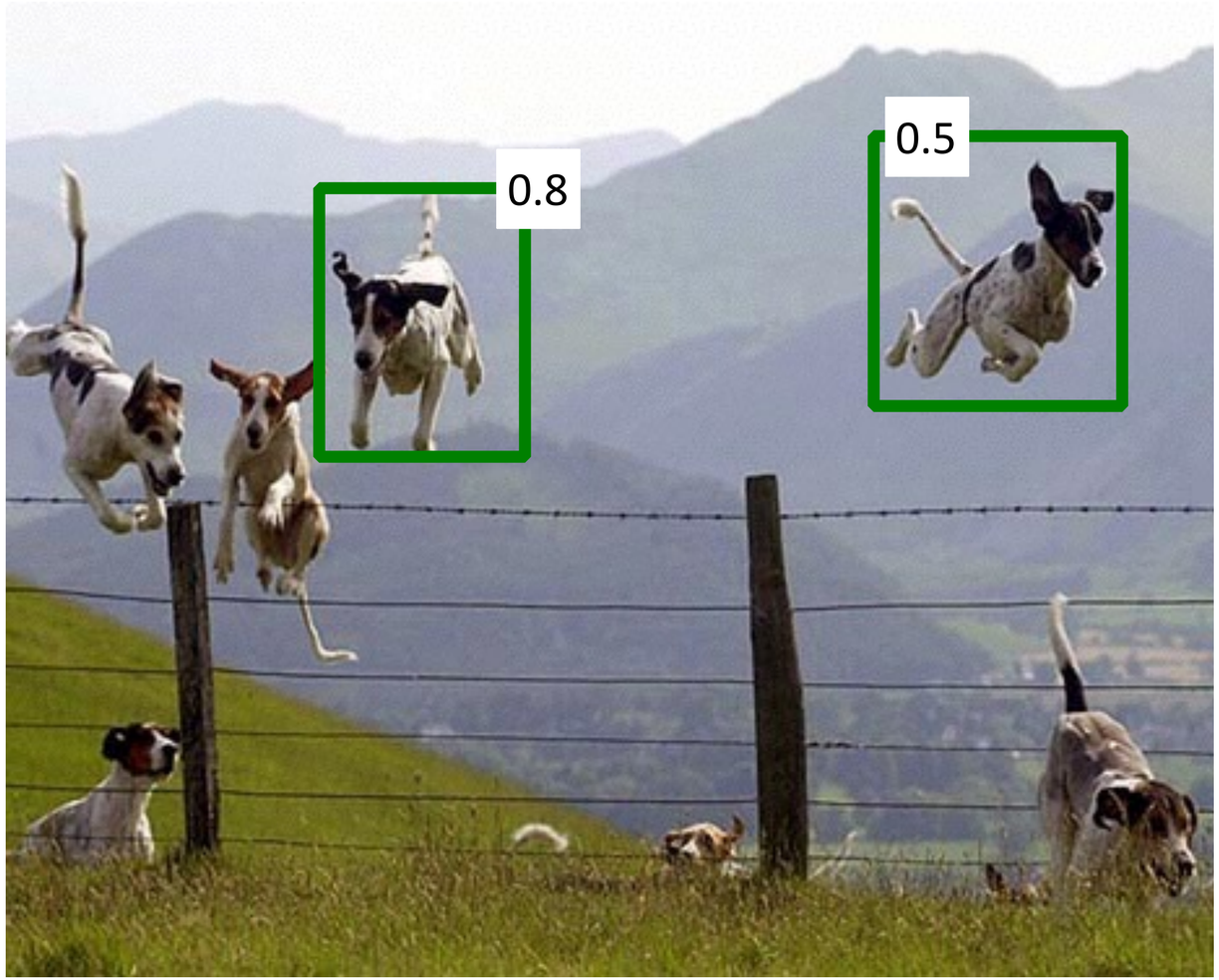
- Choose the window with maximum score.
- Reject the windows with high overlap with selected window.
- Repeat until all windows selected or rejected

Greedy Non-max



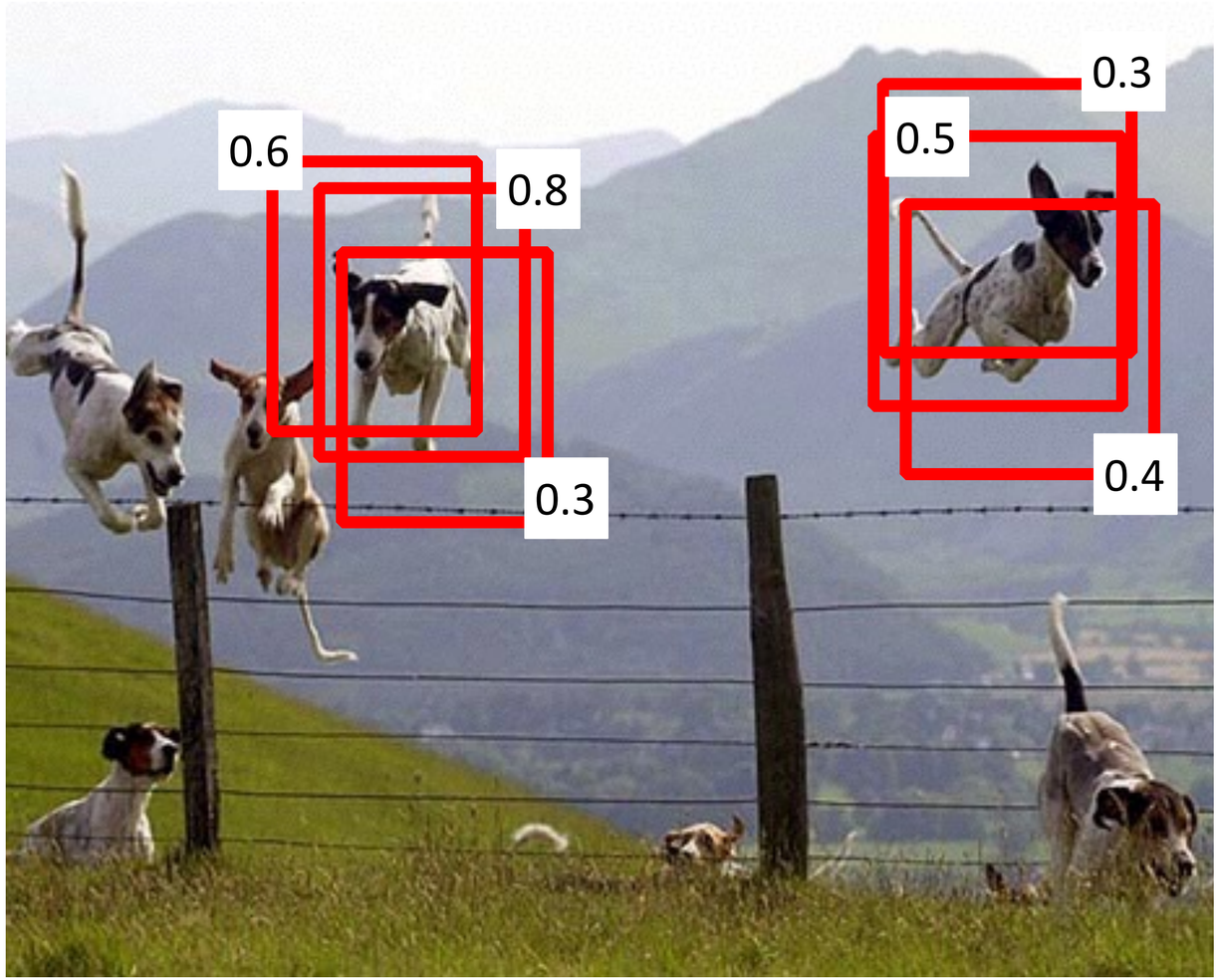
- Choose the window with maximum score.
- Reject the windows with high overlap with selected window.
- Repeat until all windows selected or rejected

Greedy Non-max



- Choose the window with maximum score.
- Reject the windows with high overlap with selected window.
- Repeat until all windows selected or rejected

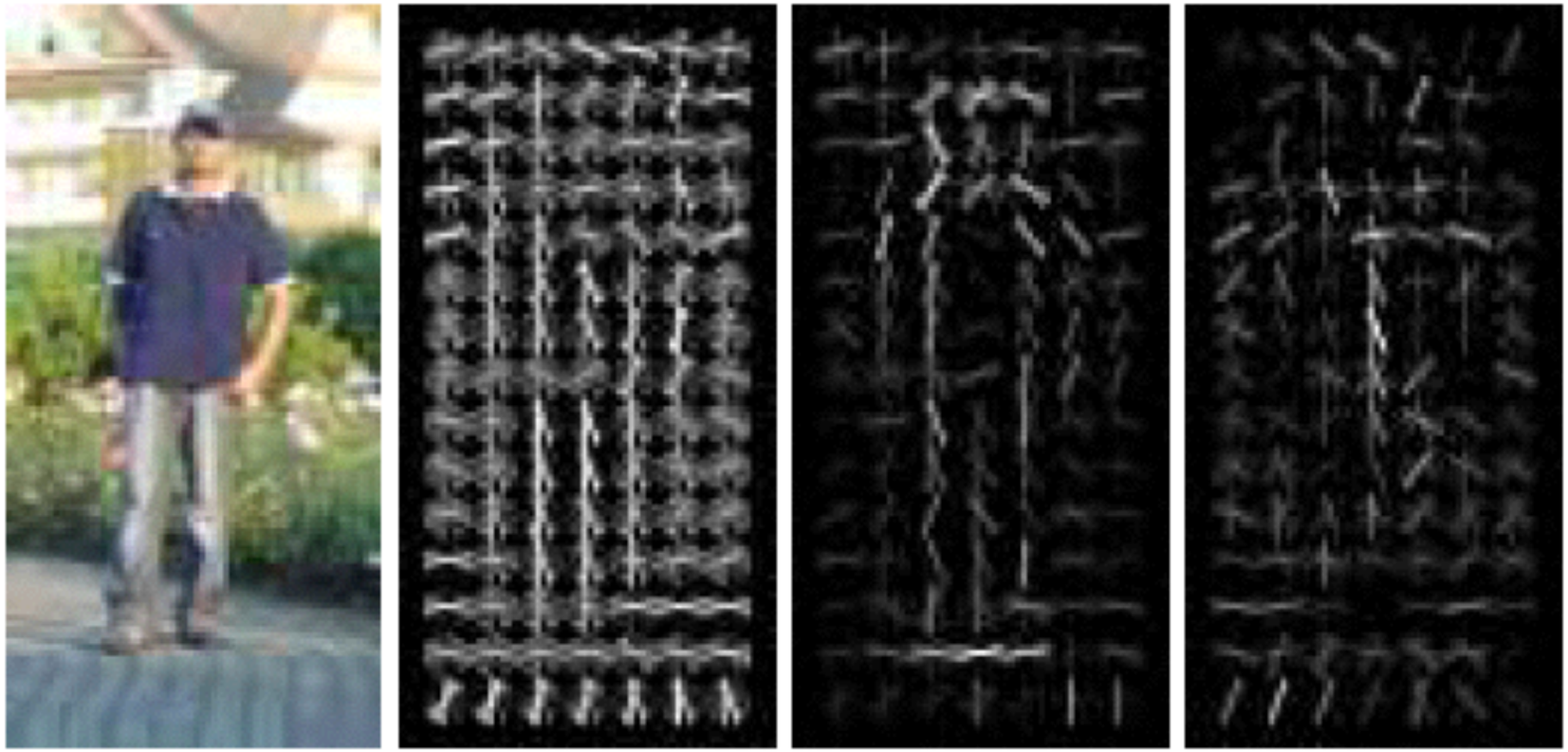
Clustering: Next HW



- Represent each window with X,Y coordinates of center and width, height.
- Perform Clustering and represent final score and detection window based on cluster members

How to Evaluate the Windows?

Case Study: HOG based Object Detection



HOG Pedestrian Detection

- Detect & localize upright people in static images

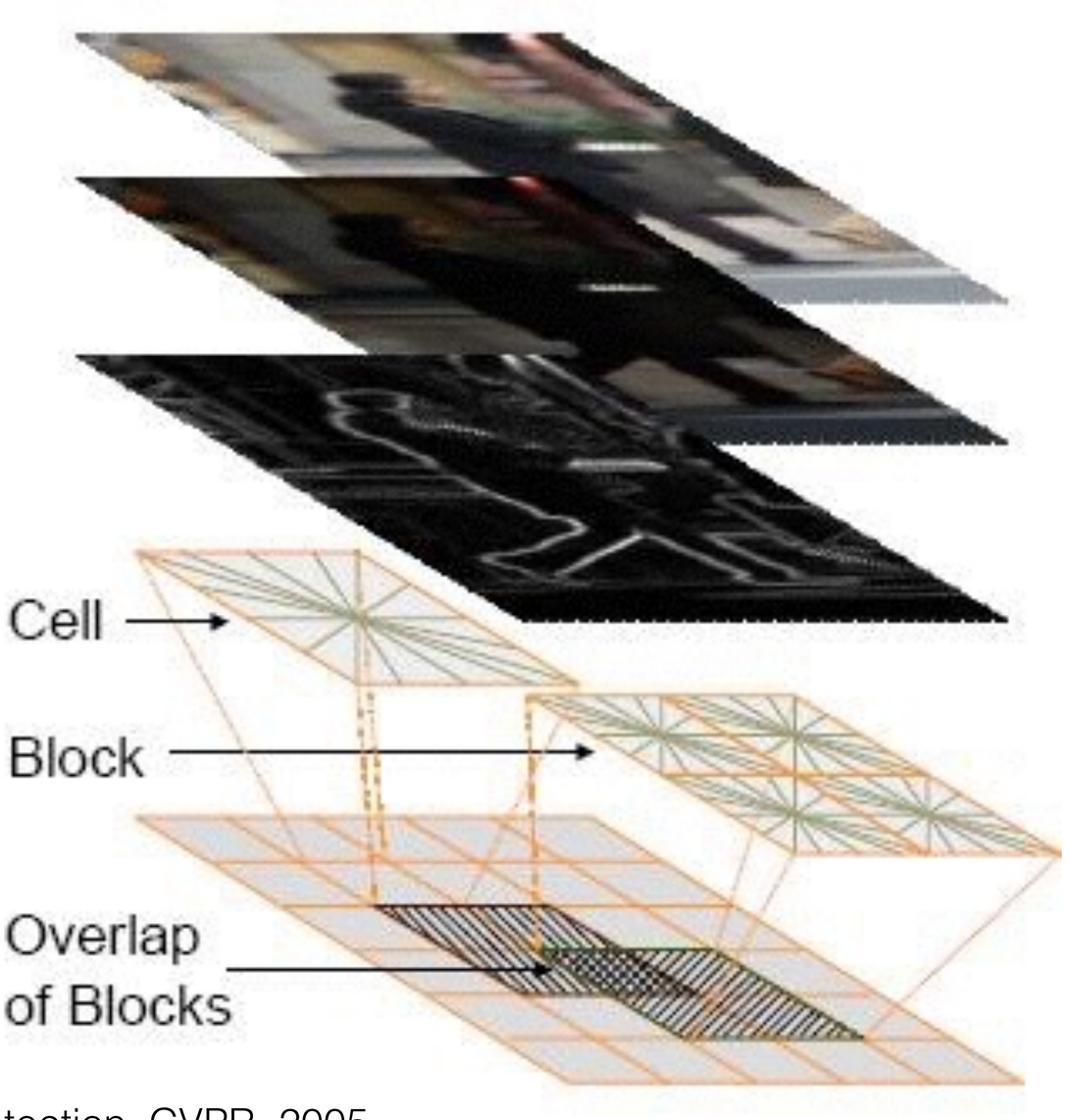
Data Set

Train	*Test*
614 positive images 1218 negative images	288 positive images 453 negative images
1208 positive windows	566 positive windows
Overall 1774 human annotations + reflections	

SVM Example for HOG Pedestrian Detection

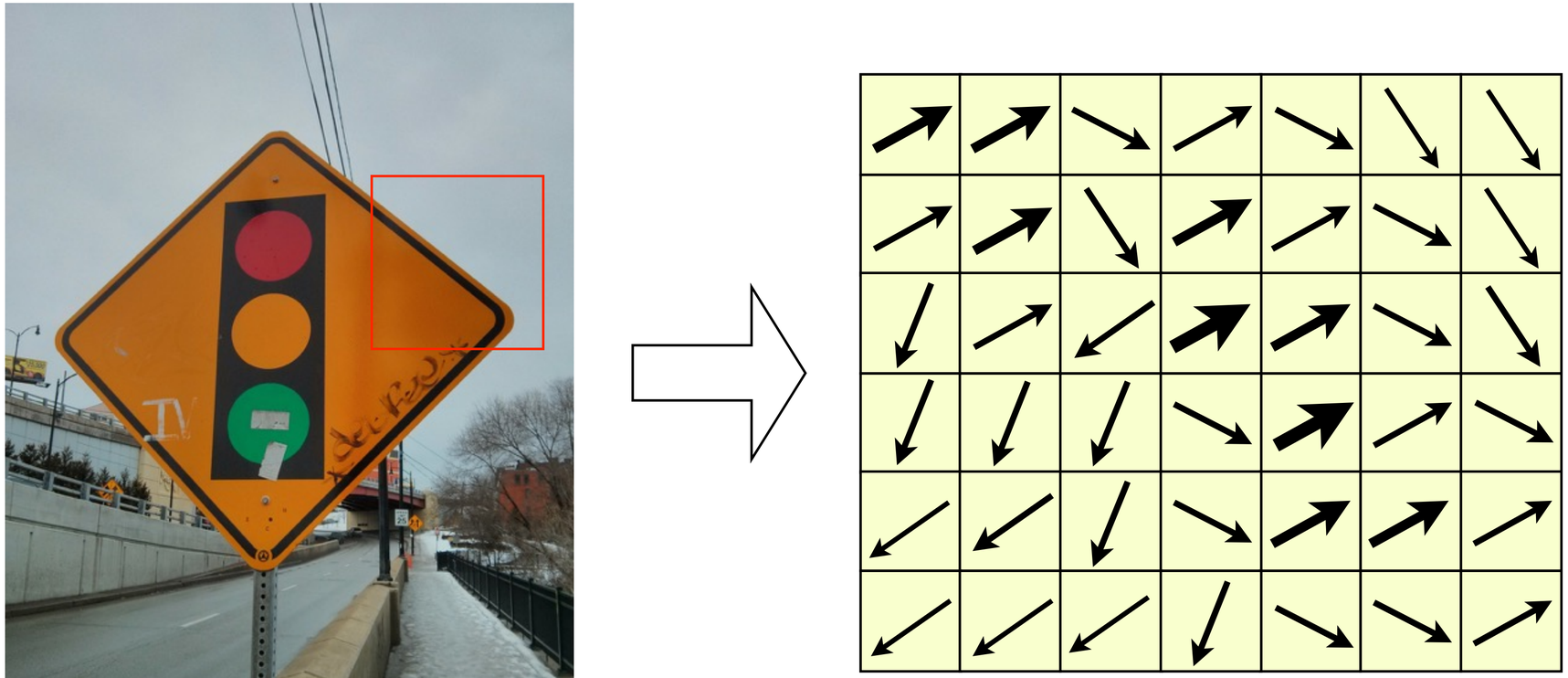
Summary:

- Compute Gradients
- HOG Features in Regular Grid
- Combine/Concatenate into a single vector
- Apply SVM Algorithm

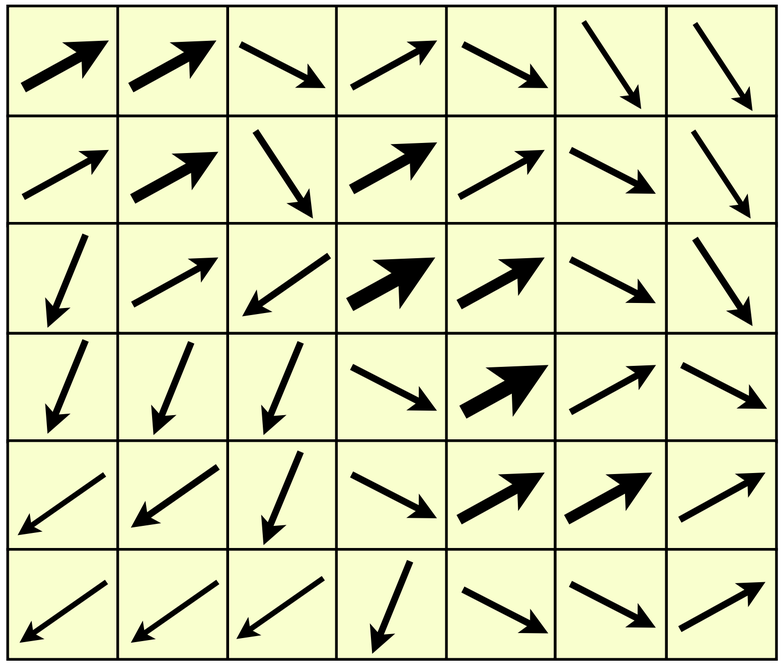


N. Dalal and B. Triggs . Histograms of Oriented Gradients for Human Detection. CVPR, 2005

Primer : Histograms of Oriented Gradients

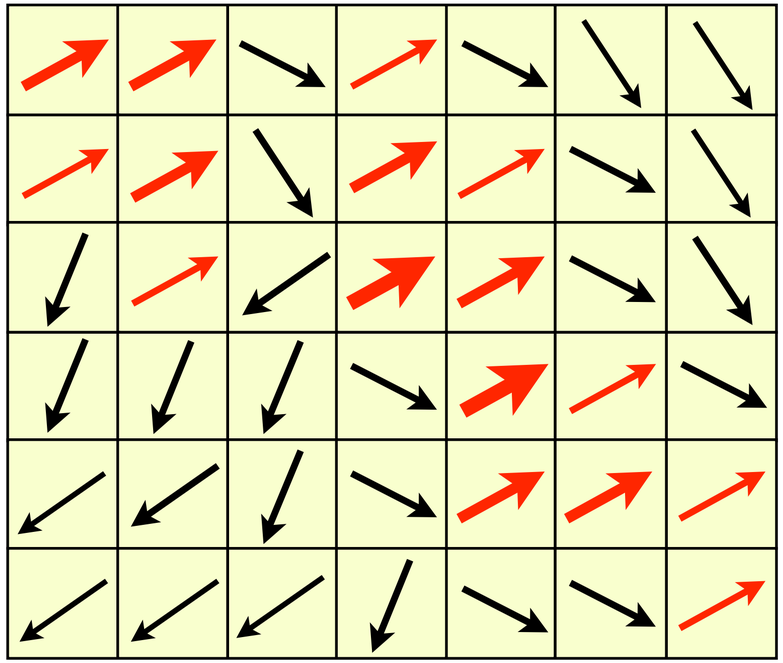


Primer : Histograms of Oriented Gradients

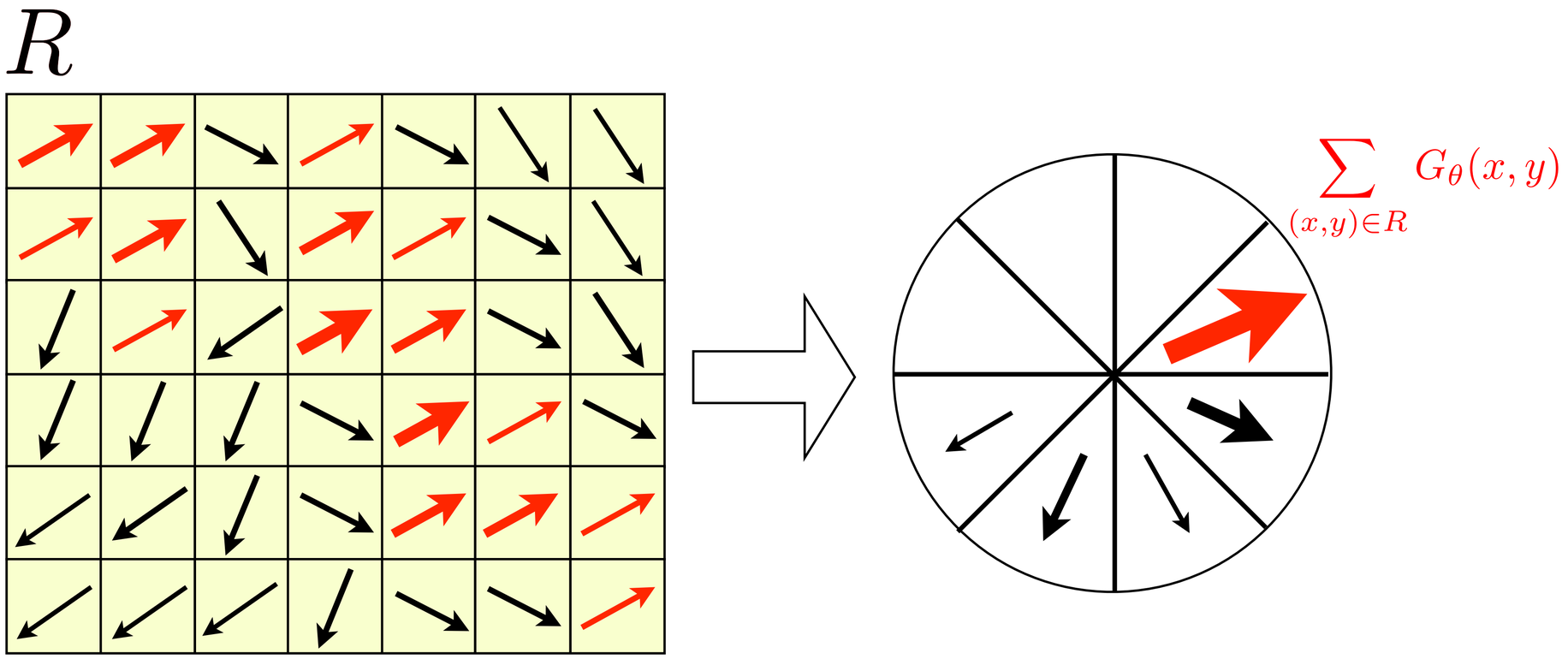


Primer : Histograms of Oriented Gradients

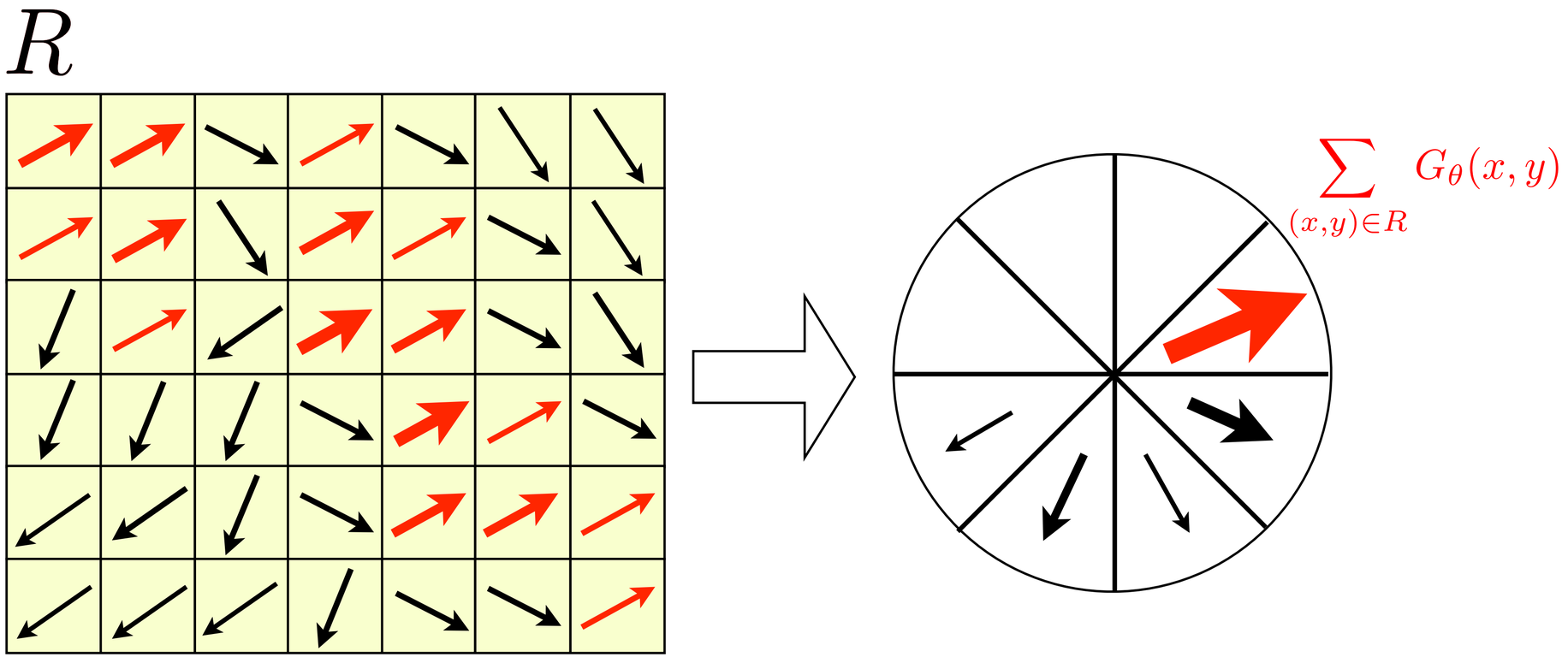
R



Primer : Histograms of Oriented Gradients



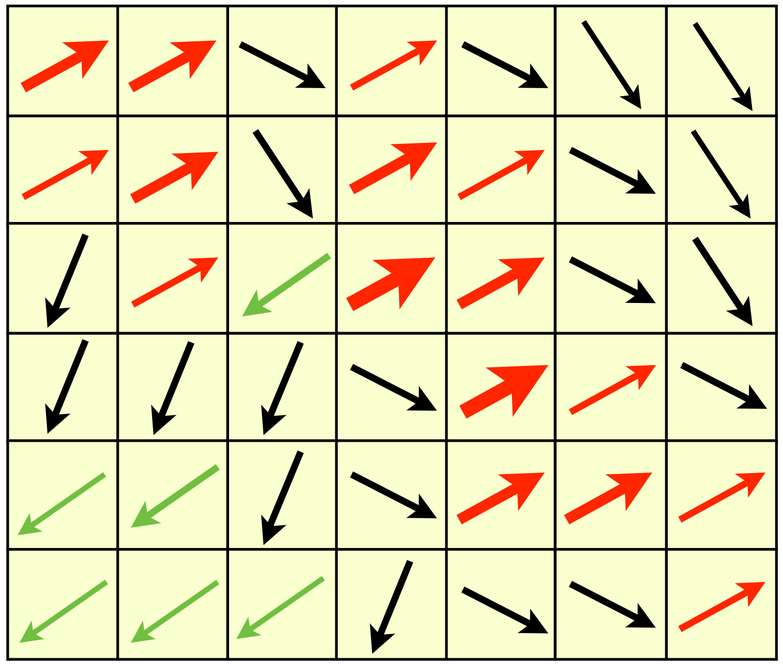
HOG : Histograms of Oriented Gradients

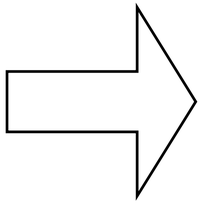


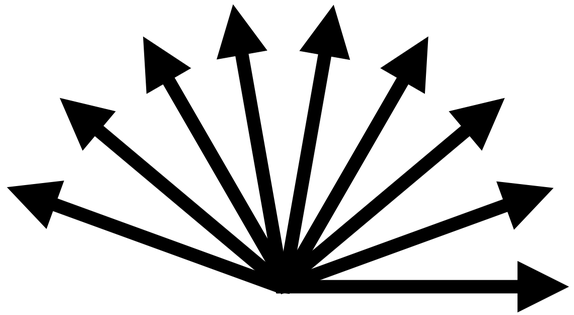
- In this case an 8-vector histogram of oriented gradients
- But there are many other configurations

HOG : Unsigned Gradients

R





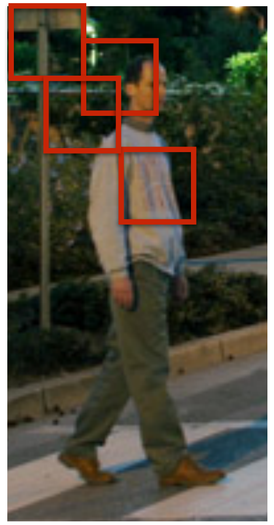


- In the original published version a 9-vector histogram of **unsigned** oriented gradients

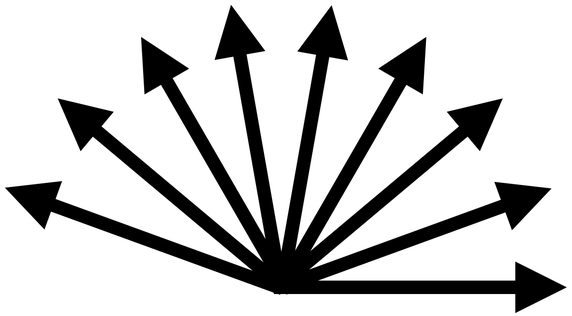
HOG - Pedestrian Detection

Scan Input Image

Resize
64x128



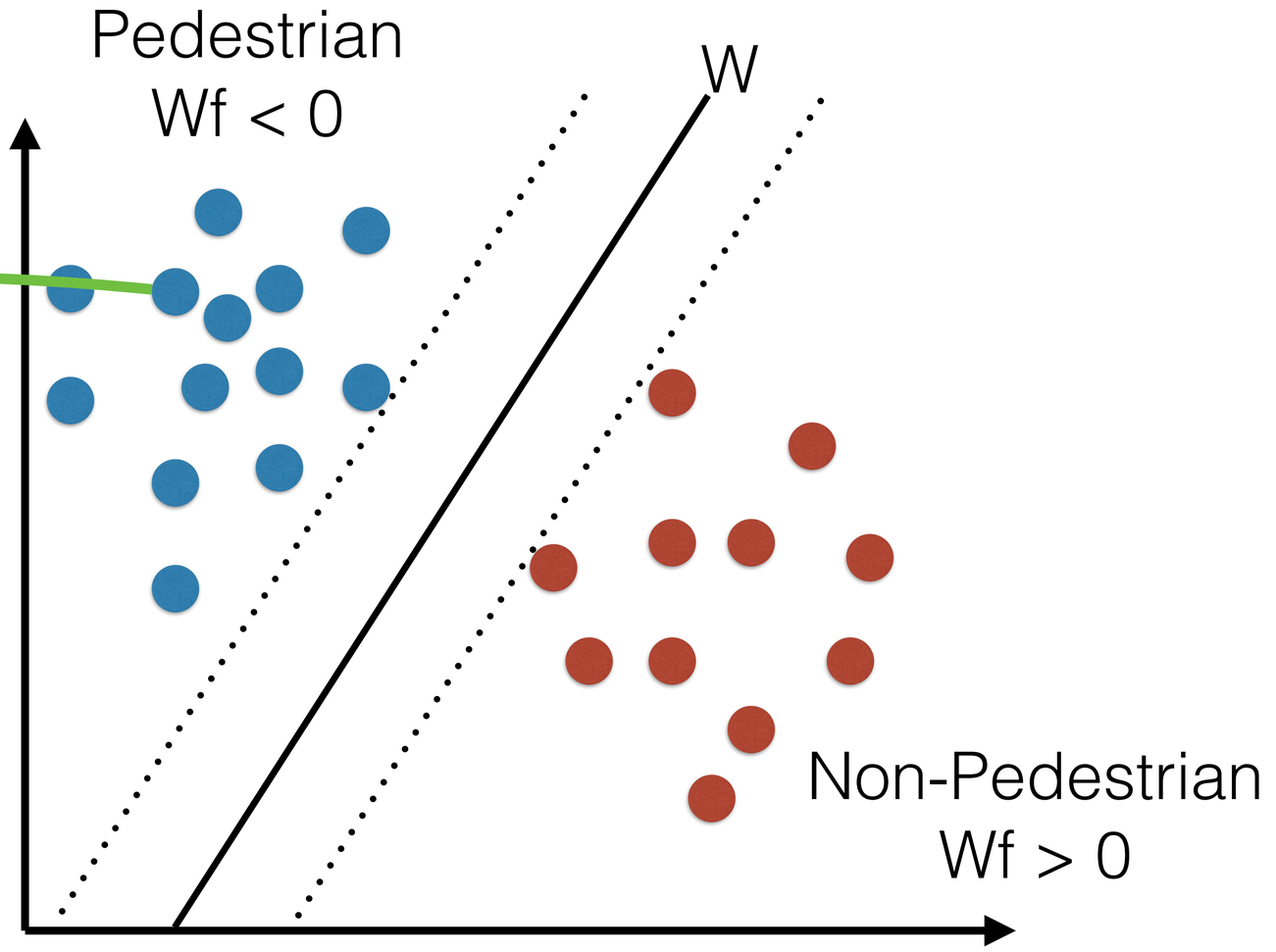
16x16
Overlapping
Blocks



Compute Histogram
over 9 directions

Final feature vector f = 7x15x9 = 945

HOG + Linear SVM



Training

Learning phase

Input: Annotations on training images

Create fixed-resolution normalised training image data set

Encode images into feature spaces

Learn binary classifier

Resample negative training images to create hard examples

Encode images into feature spaces

Learn binary classifier

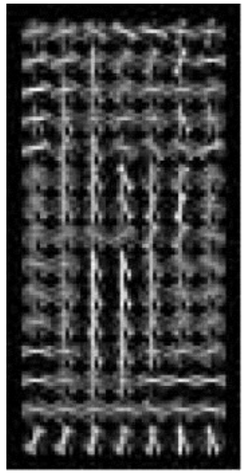
Object/Non-object decision

Retraining reduces false positives by an order of

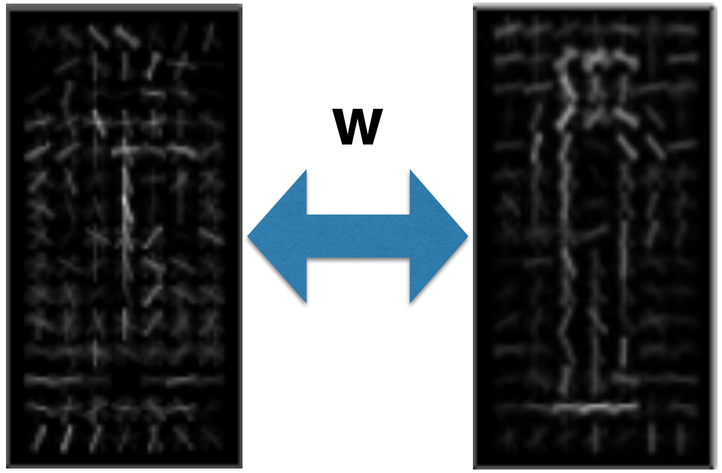
Histogram of Oriented Gradients (HOG)

- **f** = vector of gradient histograms
- Input detection window is classified as pedestrian if:
  w.f > 0

Example Detection Window



HOG Descriptor **f**



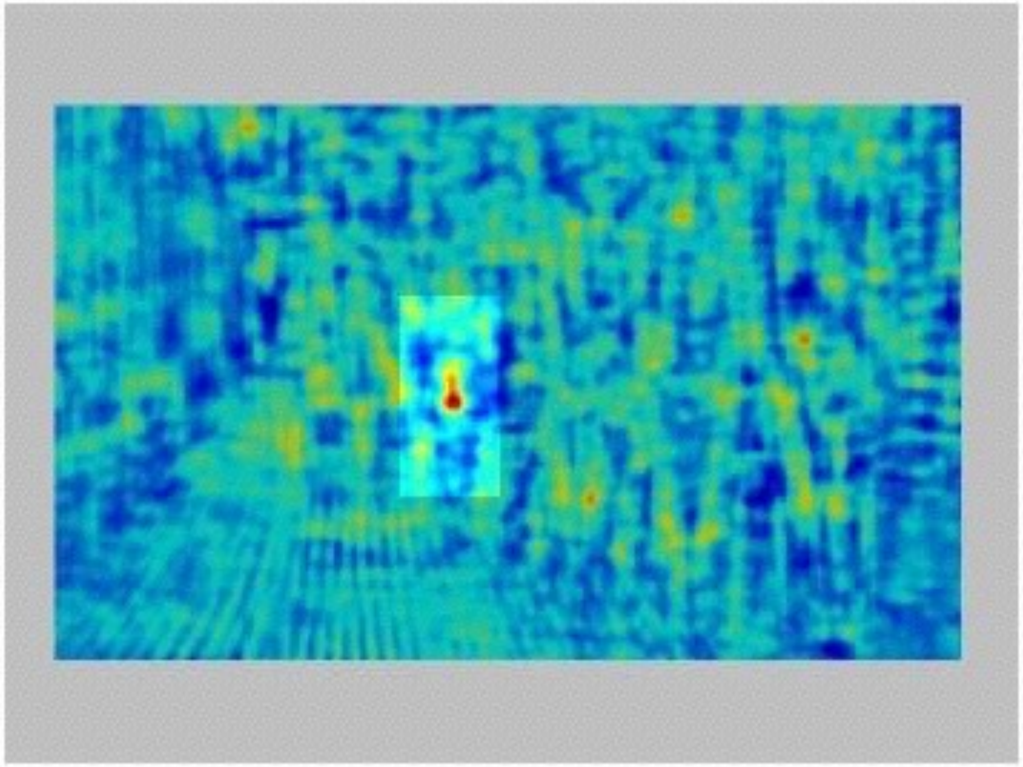
Pedestrian Class

Non-Pedestrian Class

N. Dalal and B. Triggs . Histograms of Oriented Gradients for Human Detection. CVPR, 2005

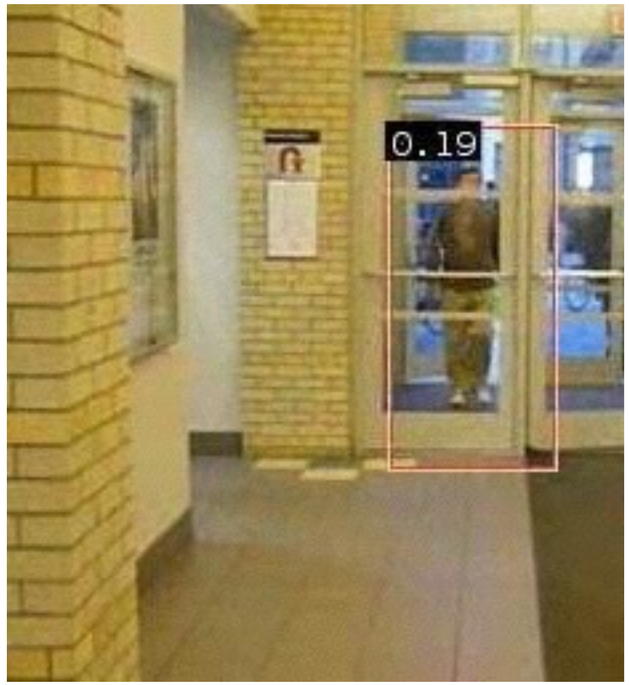
Localizing Detections

Input Image



Detection Confidence
Find local-maxima

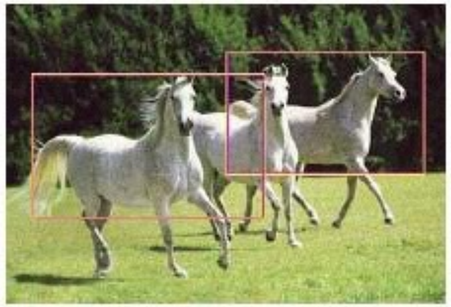
Examples

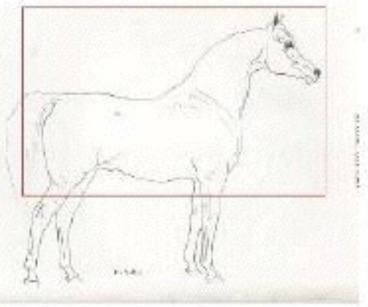


Examples

Other Objects

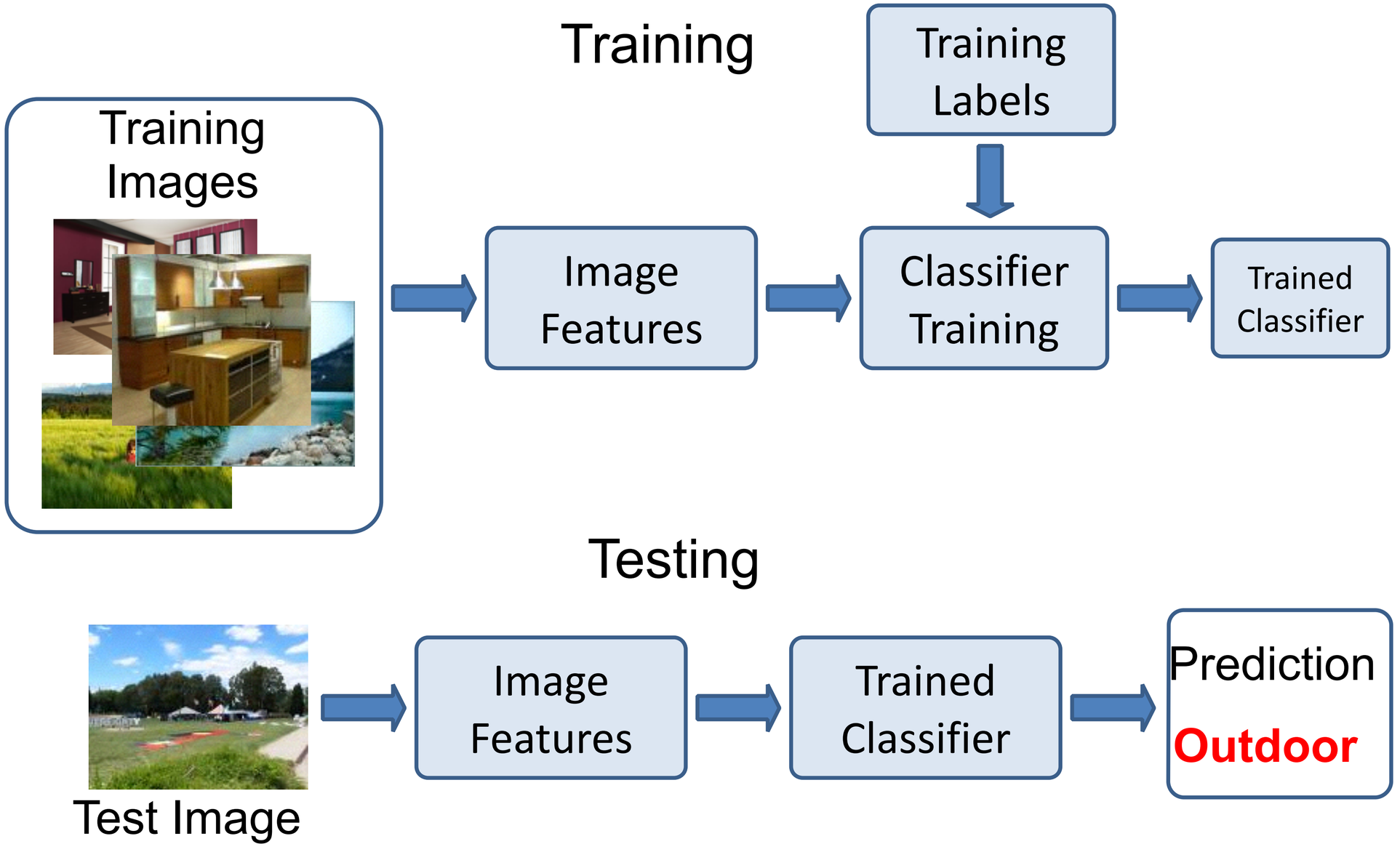


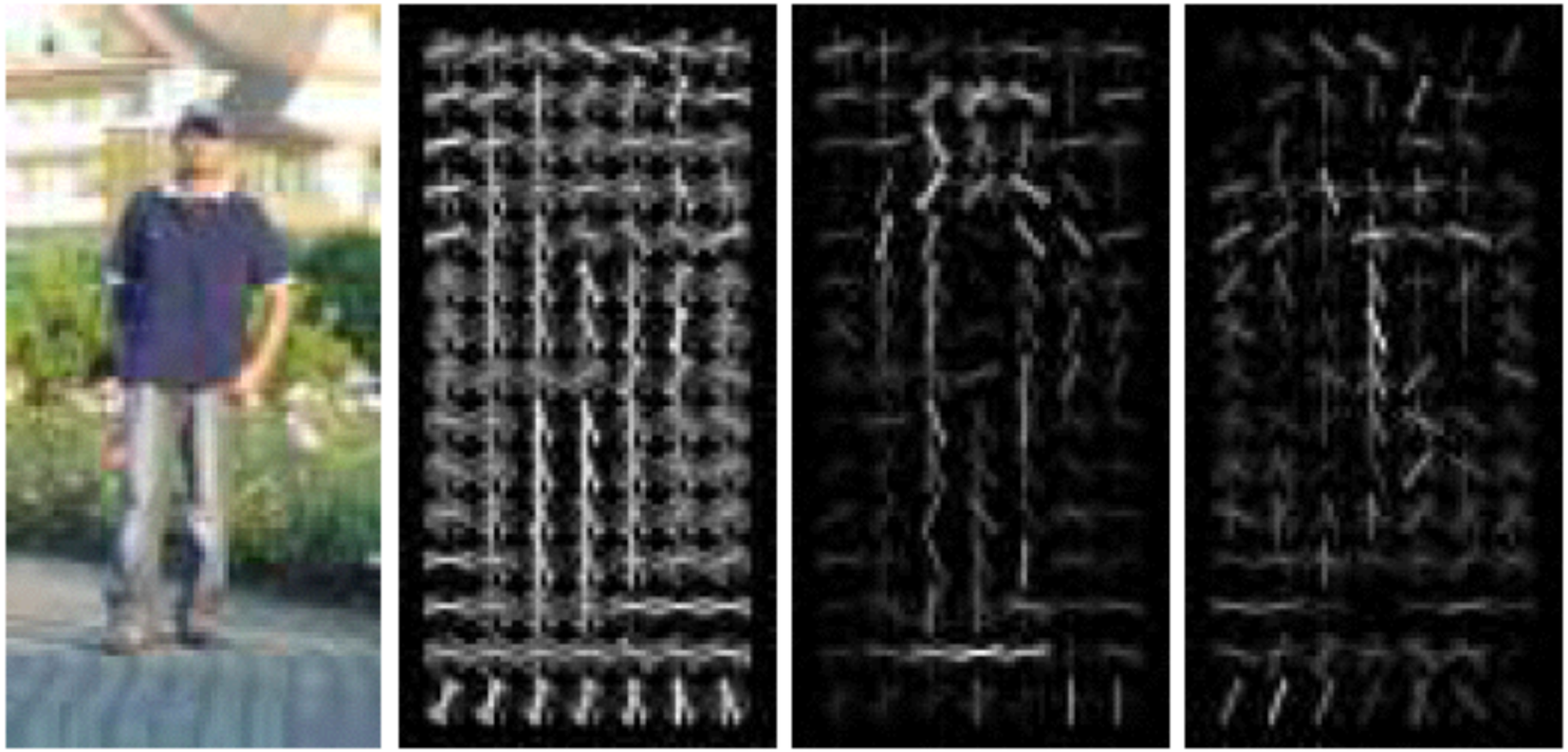



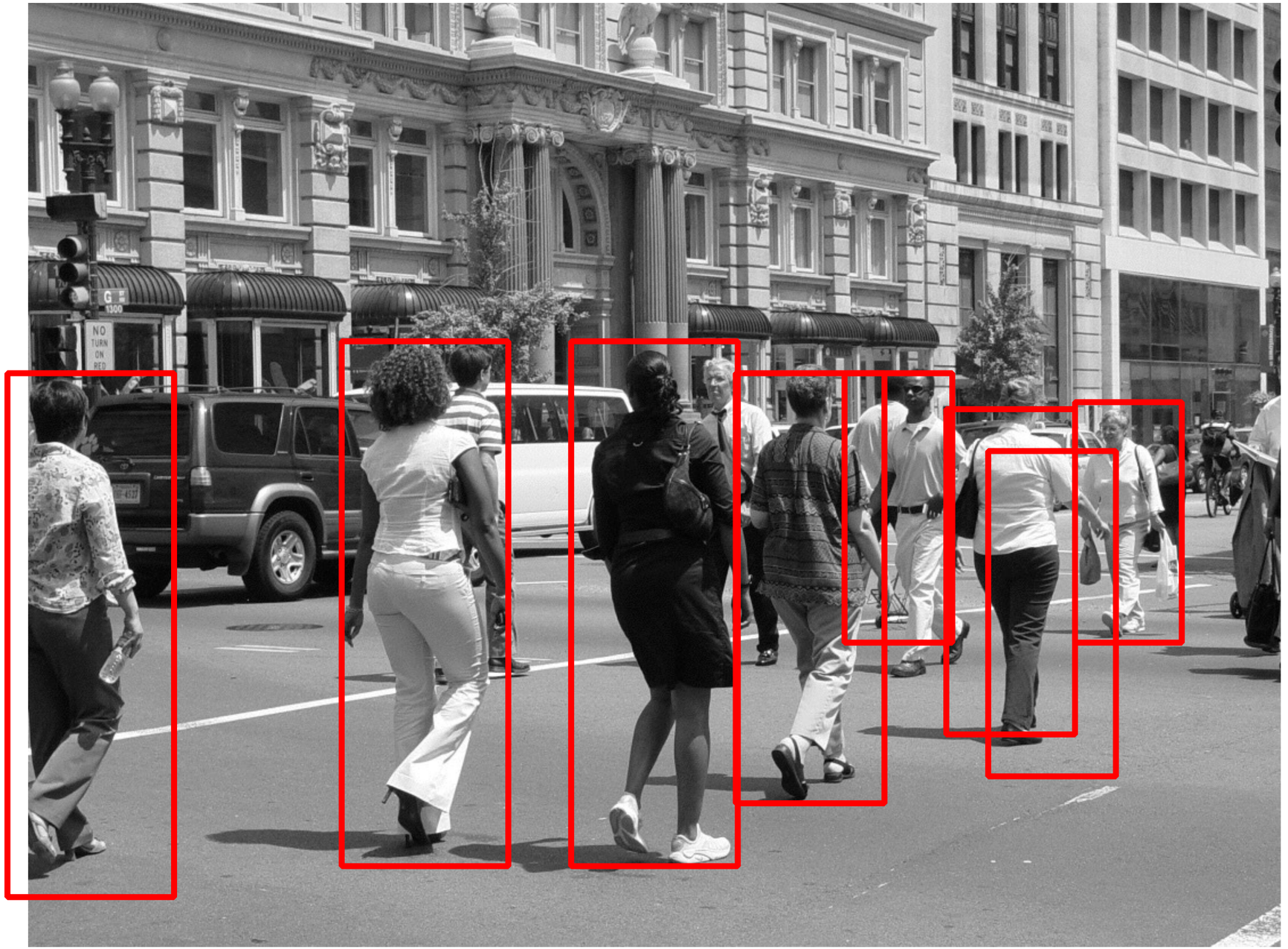
Recognition - Object Detection Continued

Testing phase



Case Study: HOG based Object Detection




NO
TURN
ON
RED

Strengths and Weaknesses of Statistical Template Approach

Strengths

- Works very well for non-deformable objects: faces, cars, upright pedestrians
- Fast detection

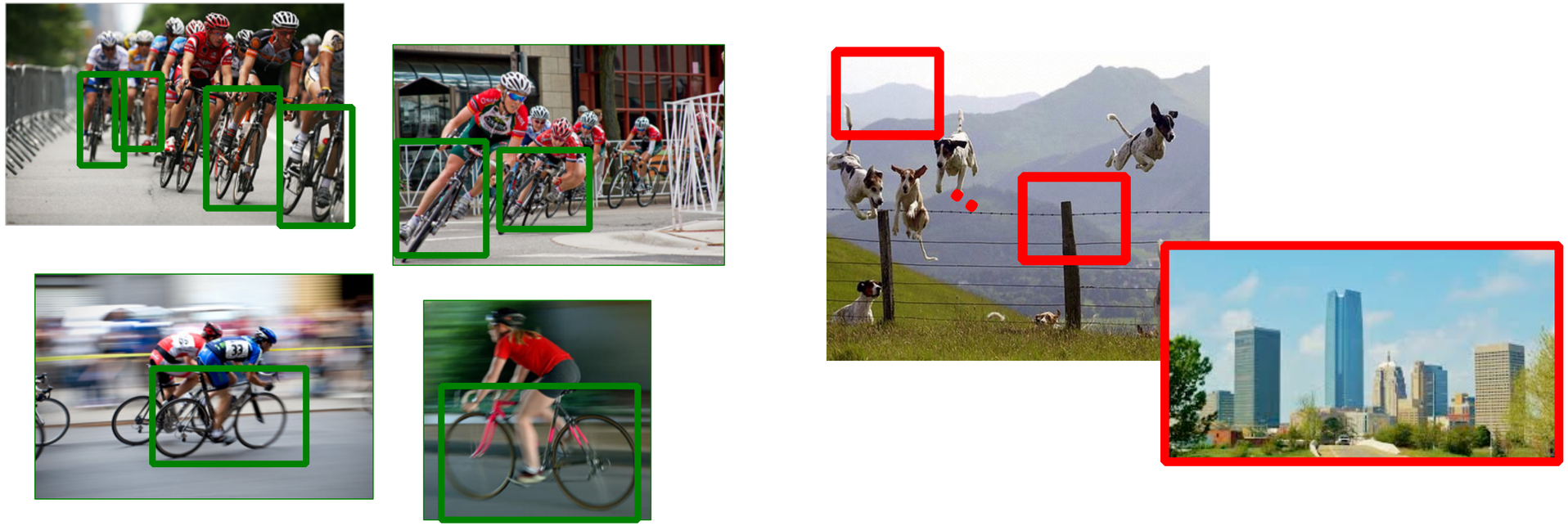
Weaknesses

- Not so well for highly deformable objects
- Not robust to occlusion
- Requires lots of training data

Tricks of the trade

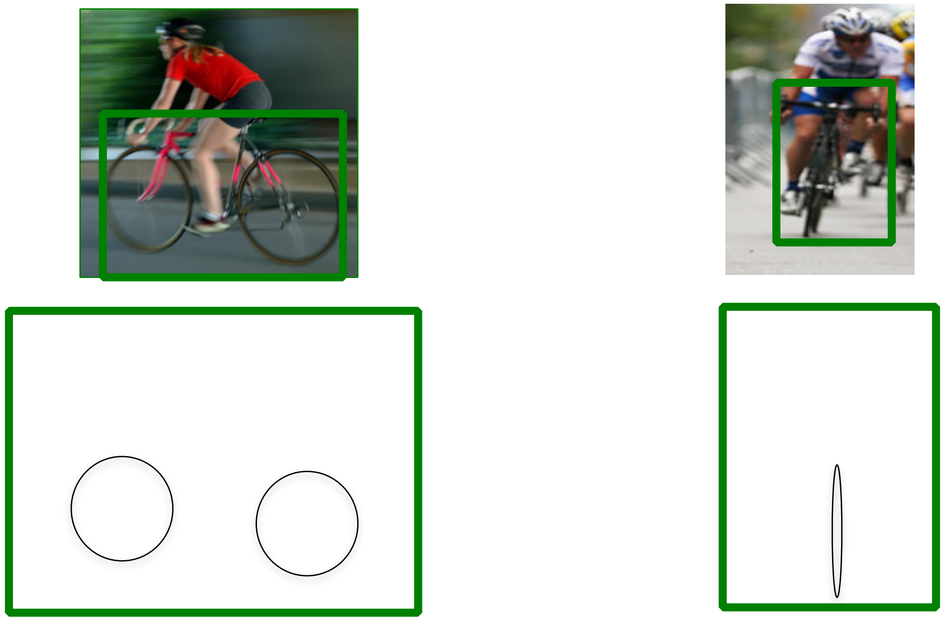
- Details in feature computation really matter
  - E.g., normalization in Dalal-Triggs improves detection rate by 27% at fixed false positive rate
- “Jittering” to create synthetic positive examples
  - Create slightly rotated, translated, scaled, mirrored versions as extra positive examples
- Bootstrapping to get hard negative examples
  1. Randomly sample negative examples
  2. Train detector
  3. Sample negative examples that score > -1
  4. Repeat until all high-scoring negative examples fit in memory

How about a bicycle detector?



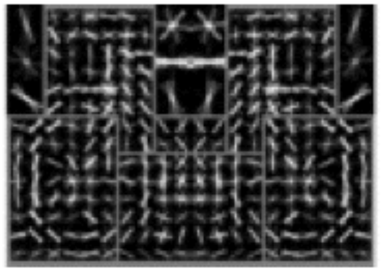
But what should a detector learn?

Side vs. Frontal Bikes

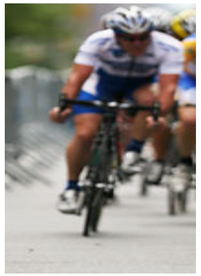


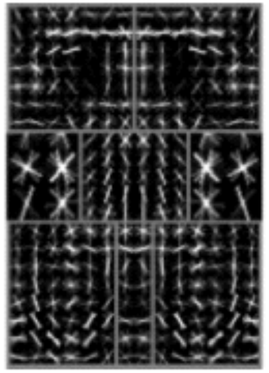
How about we divide bike into two classes? Side-Bike and Frontal Bike

Visual Subcategories

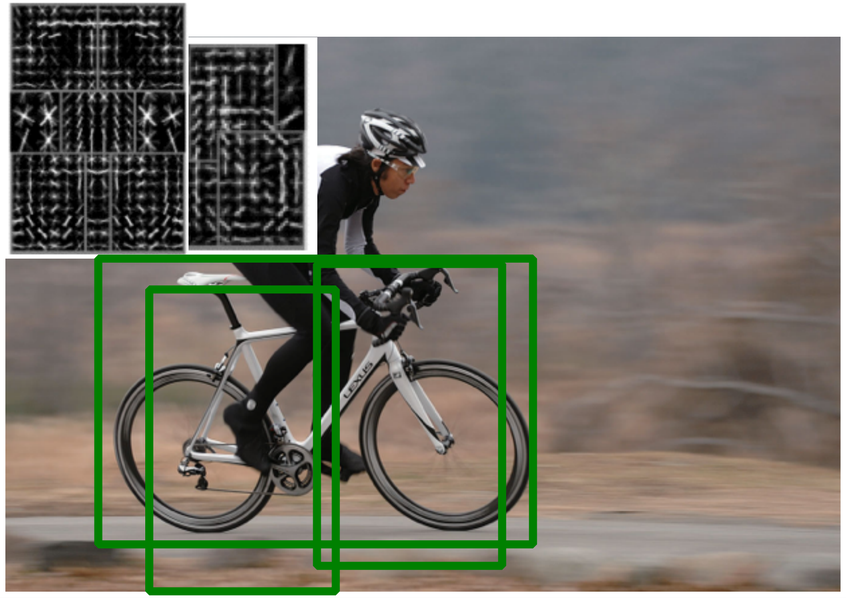


Side-Bike Detector



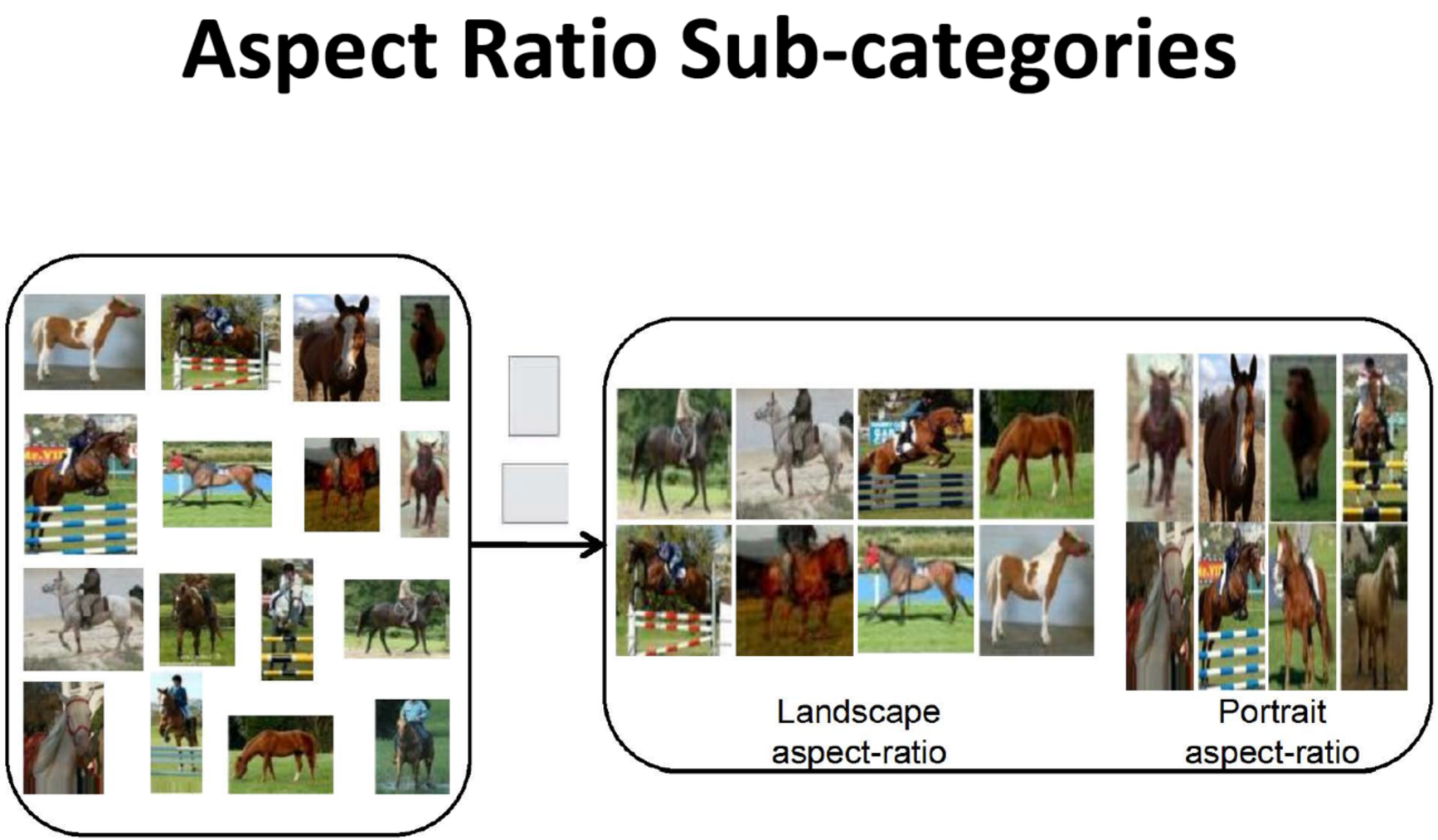


Front-Bike Detector

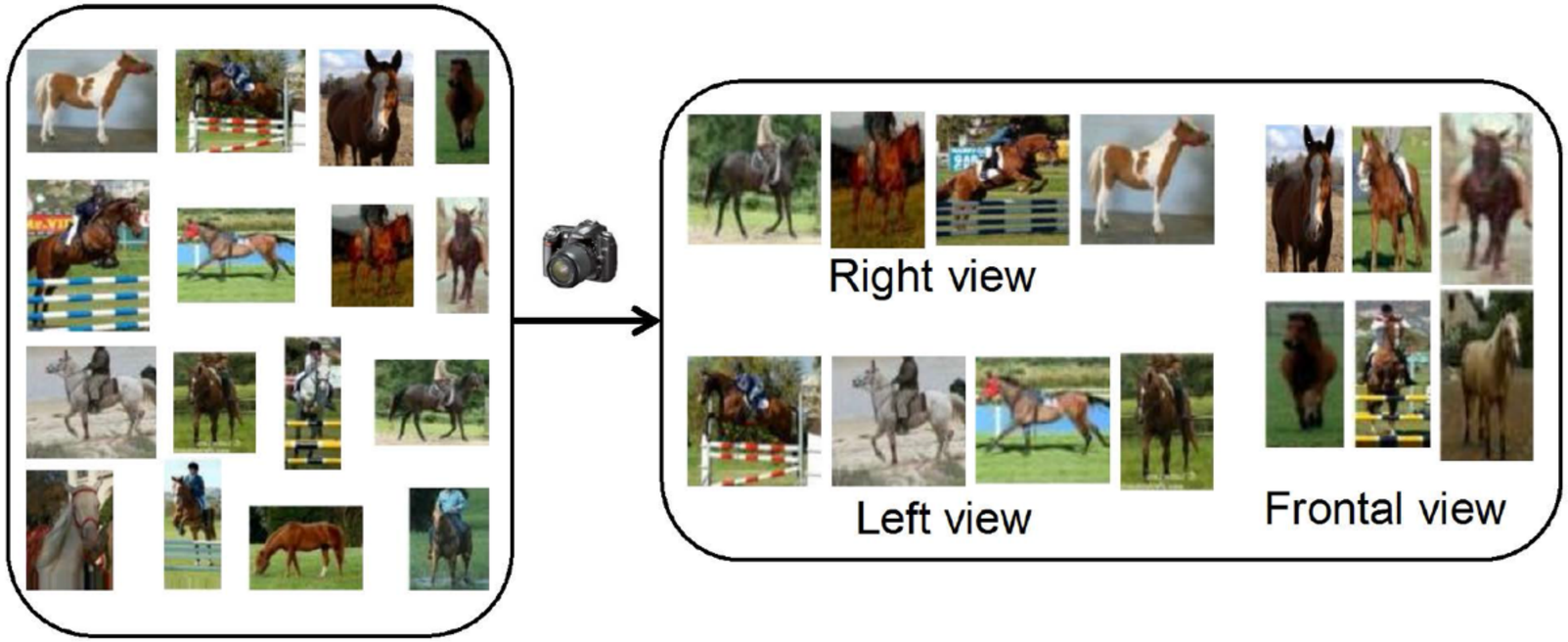


Competing Non-Max Suppression

What are the right subcategories?

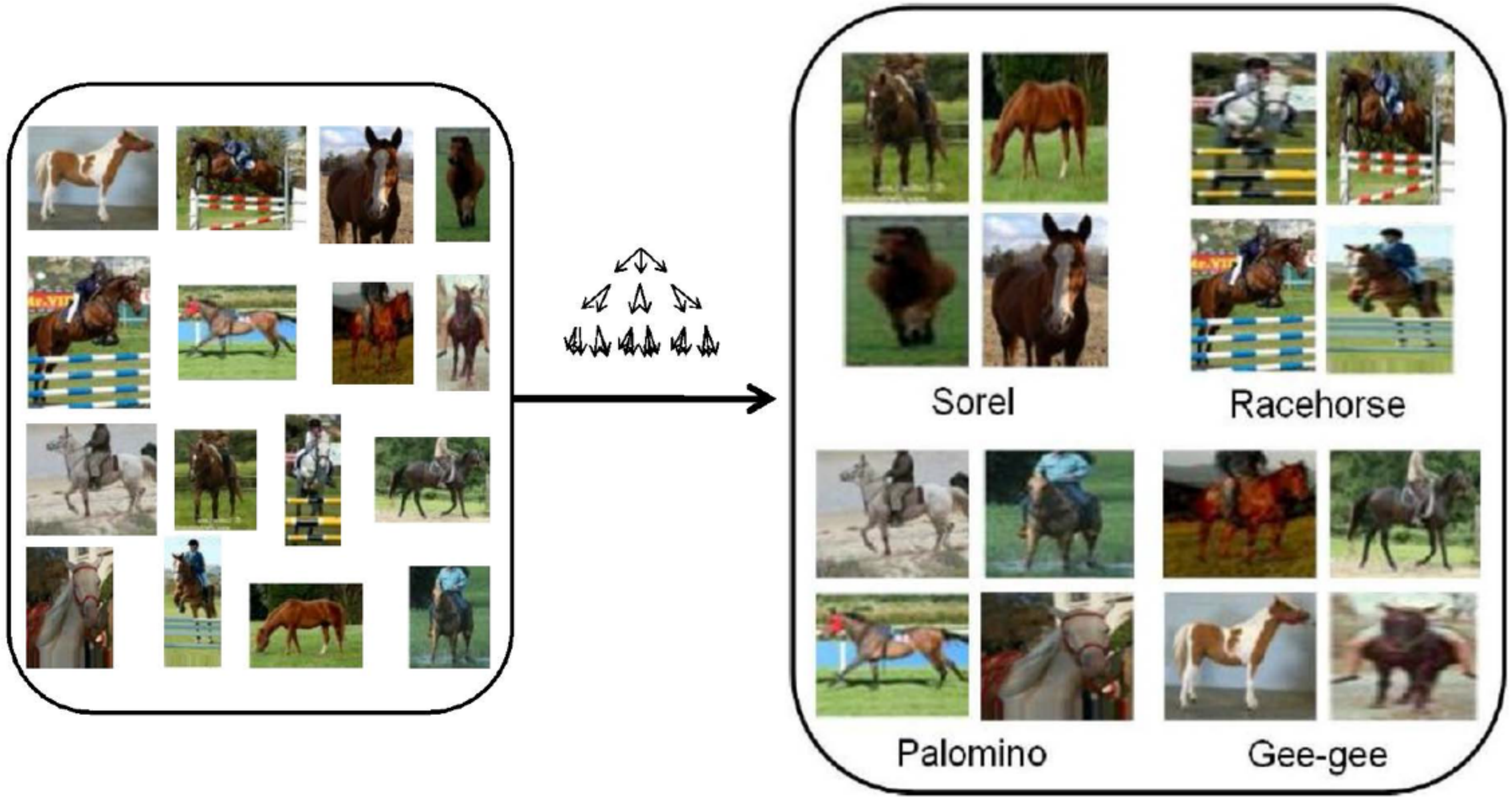


What are the right subcategories?



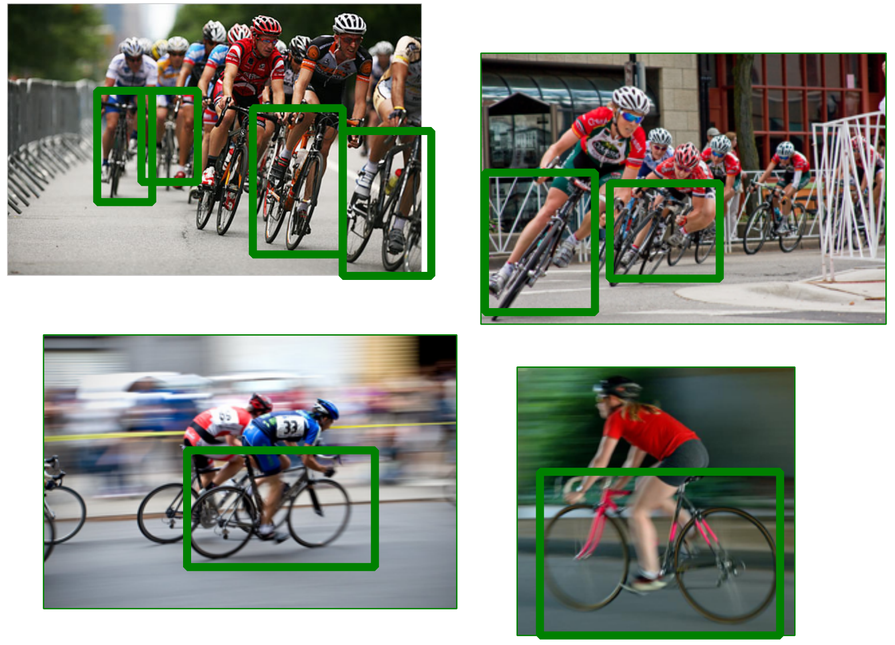
Chum & Zisserman 2007, Harzallah and Schmid 2008

What are the right subcategories?



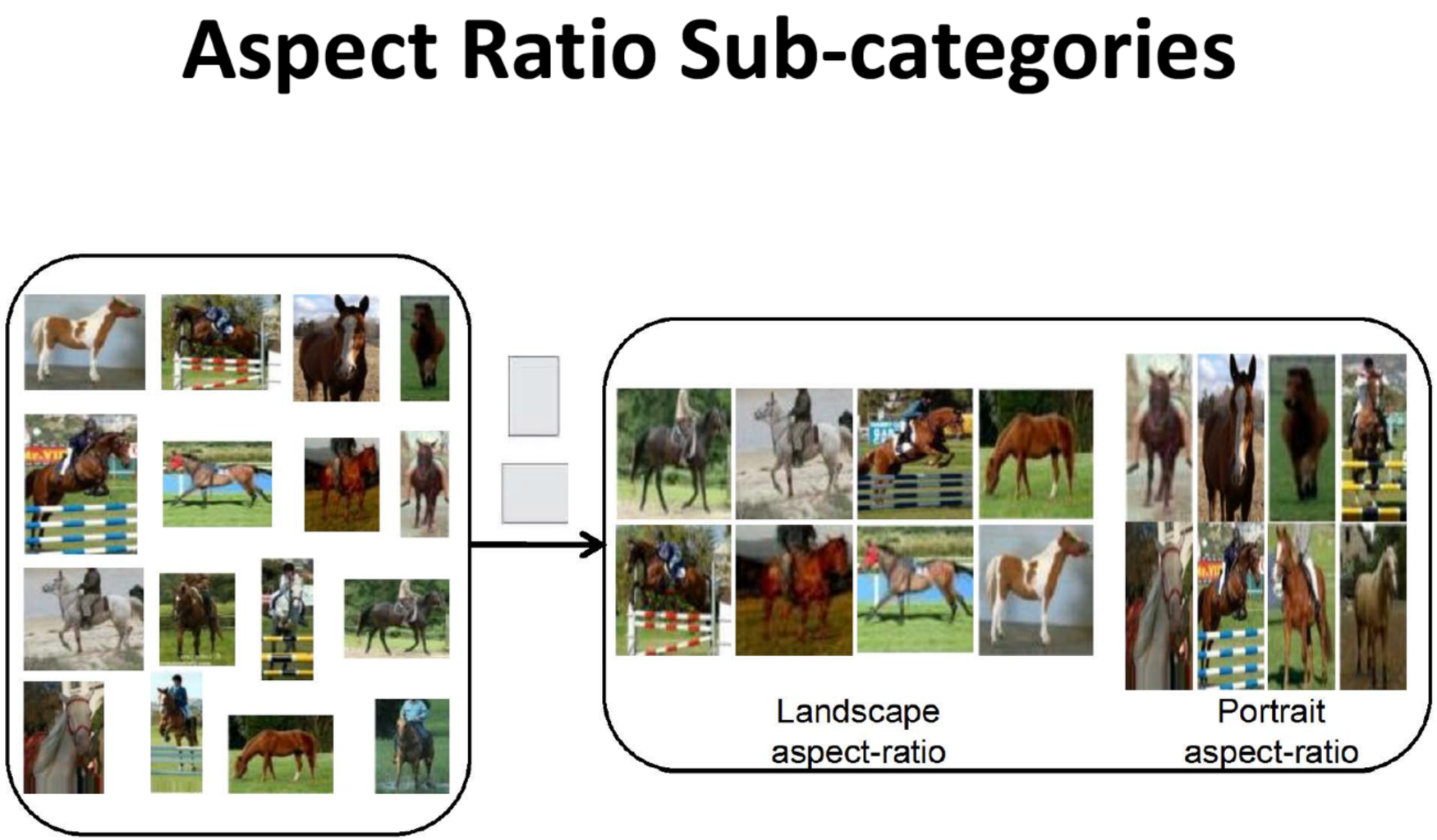
"ImageNet", Deng et al., 2009

But…

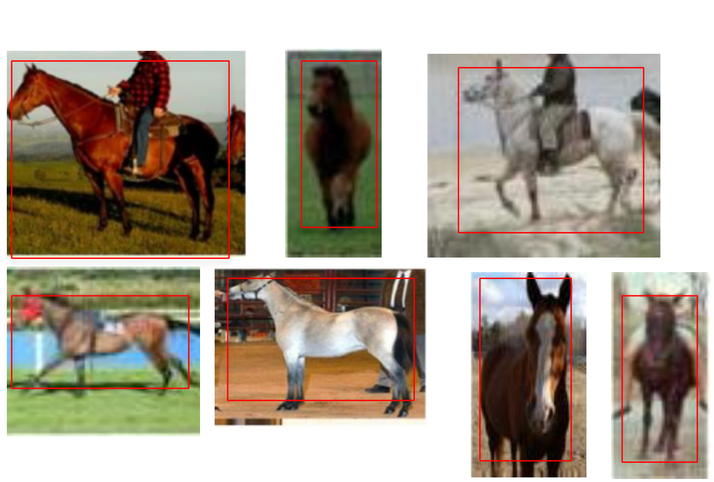


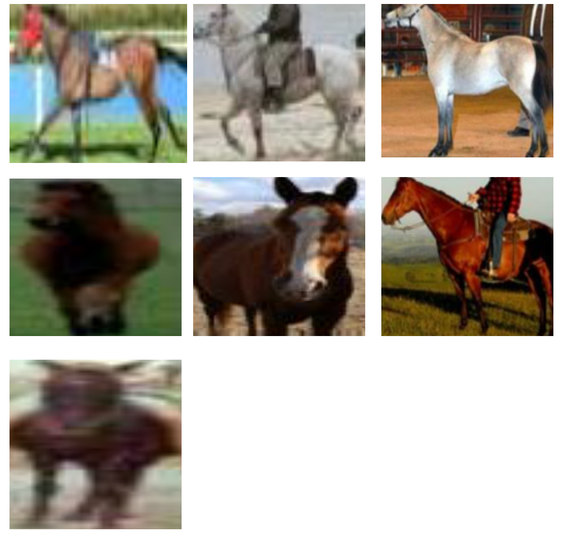
No frontal or side labels provided..

Possibility 1: Use Bounding Boxes

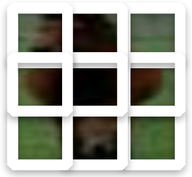


Possibility 2: Use appearances itself!



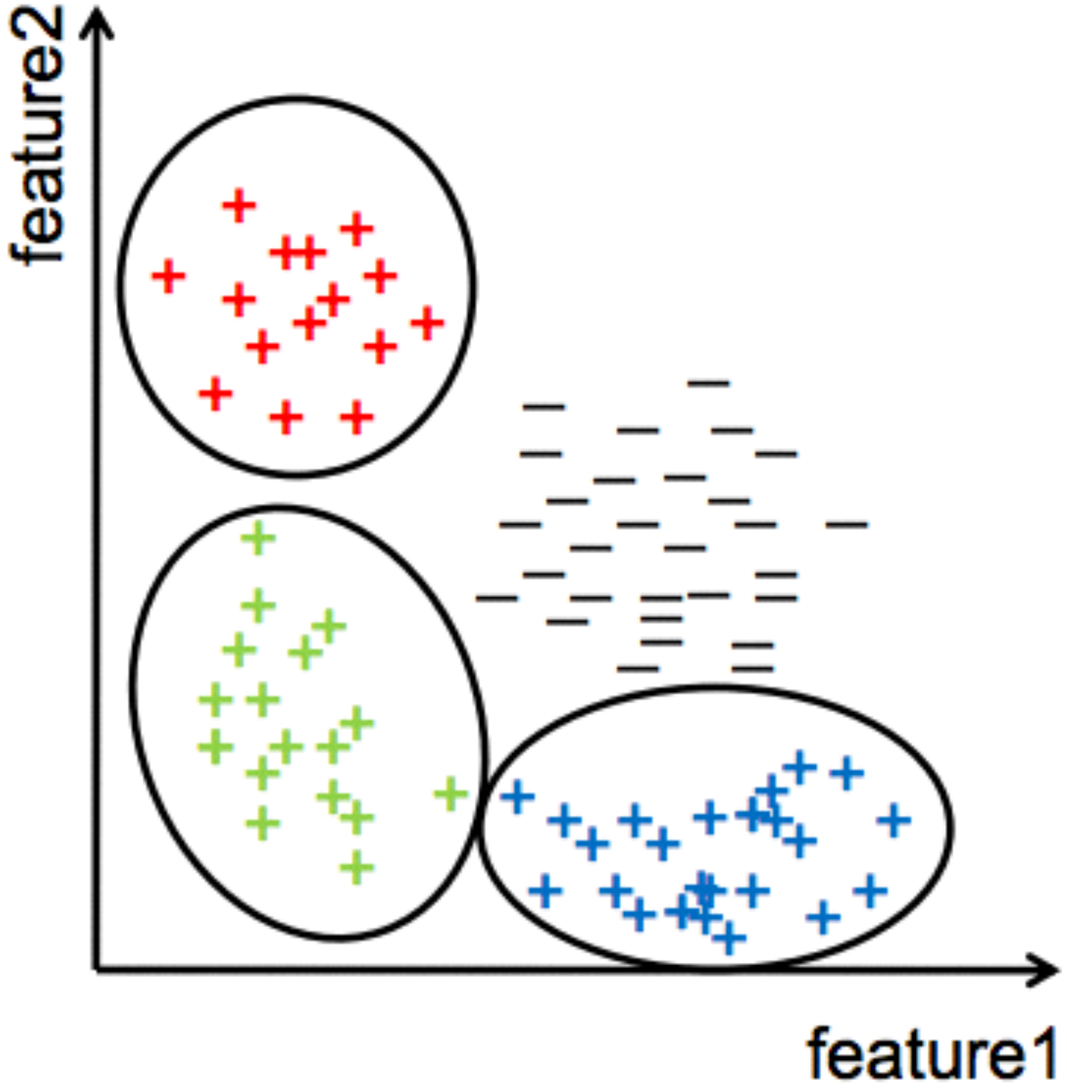


Resize Image to Canonical Size

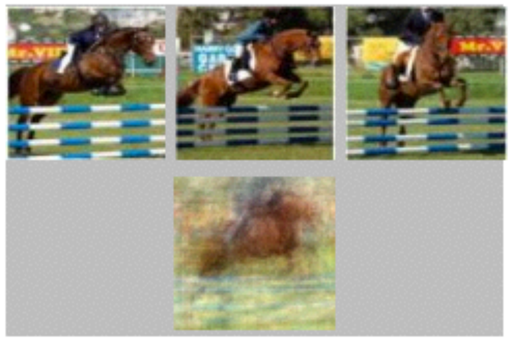


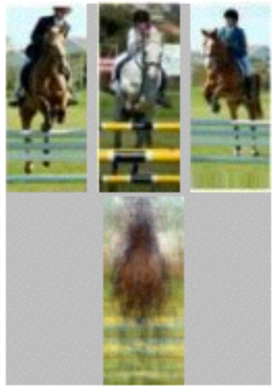
NxNx31 HOG Feature (Color Histogram, Dense SIFT, Filter Responses)

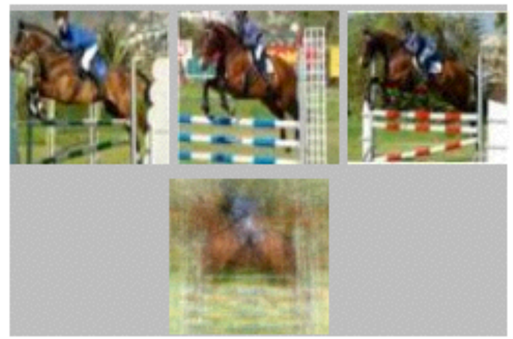
Subcategories

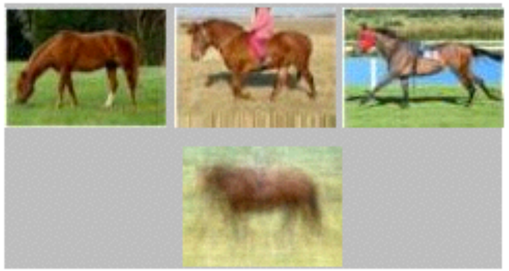


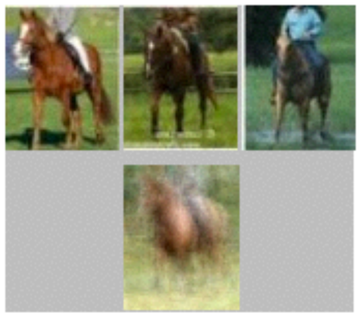
Visual Subcategories : Horse

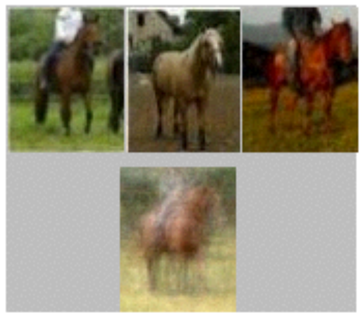


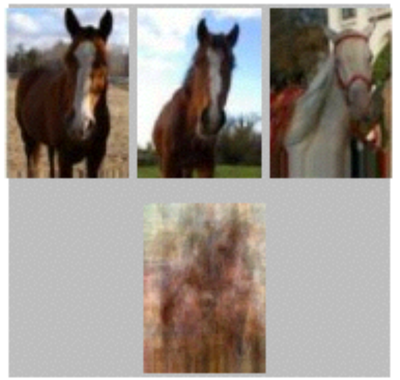






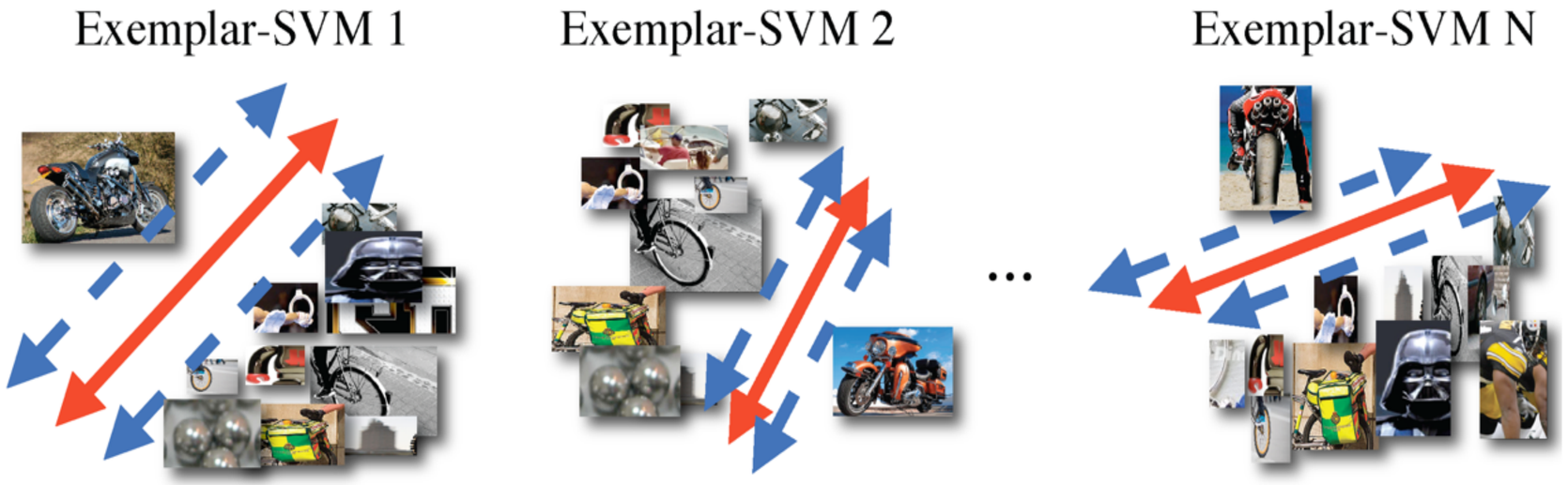




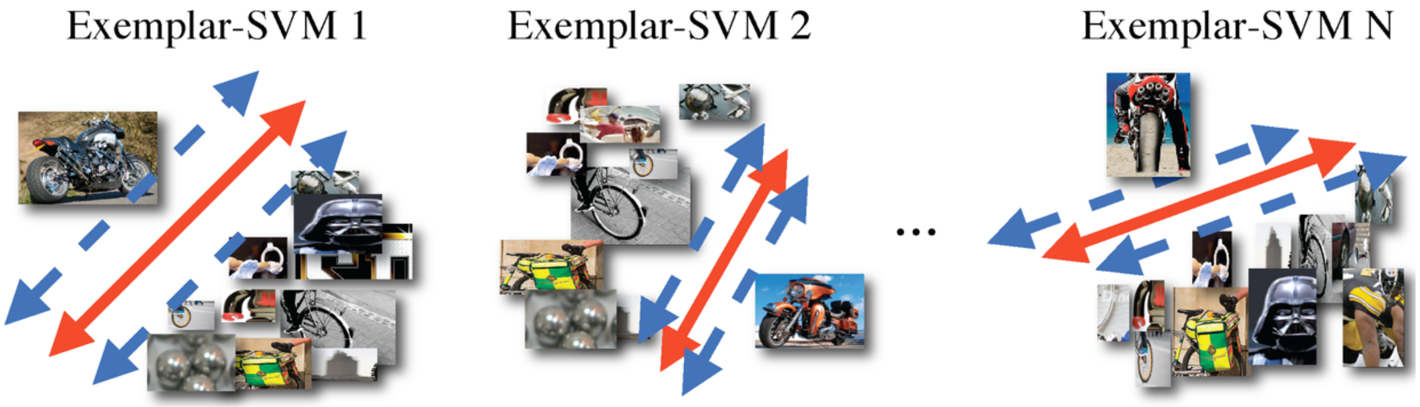


How about an extreme?

- Every instance (example) is a subcategory in itself.
- 100 training examples => 100 subcategories.
- For each example train an SVM: exemplar-SVM.



exemplar-SVMs



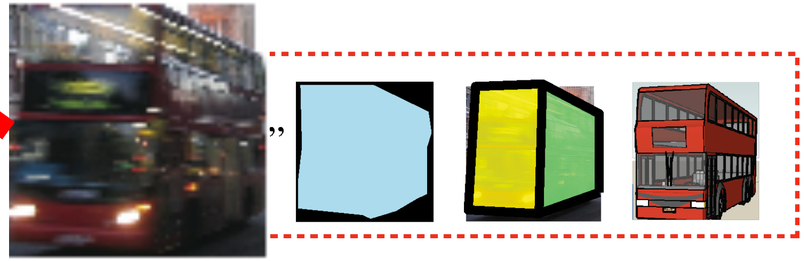
- Learn a separate linear SVM for each instance (exemplar) in the dataset (PASCAL VOC)

Each Exemplar-SVM is trained with a **single** positive instance and **millions** of negatives

Each Exemplar-SVM is more defined by "*what it is not*"

specificity of exemplar-SVMs

why associations



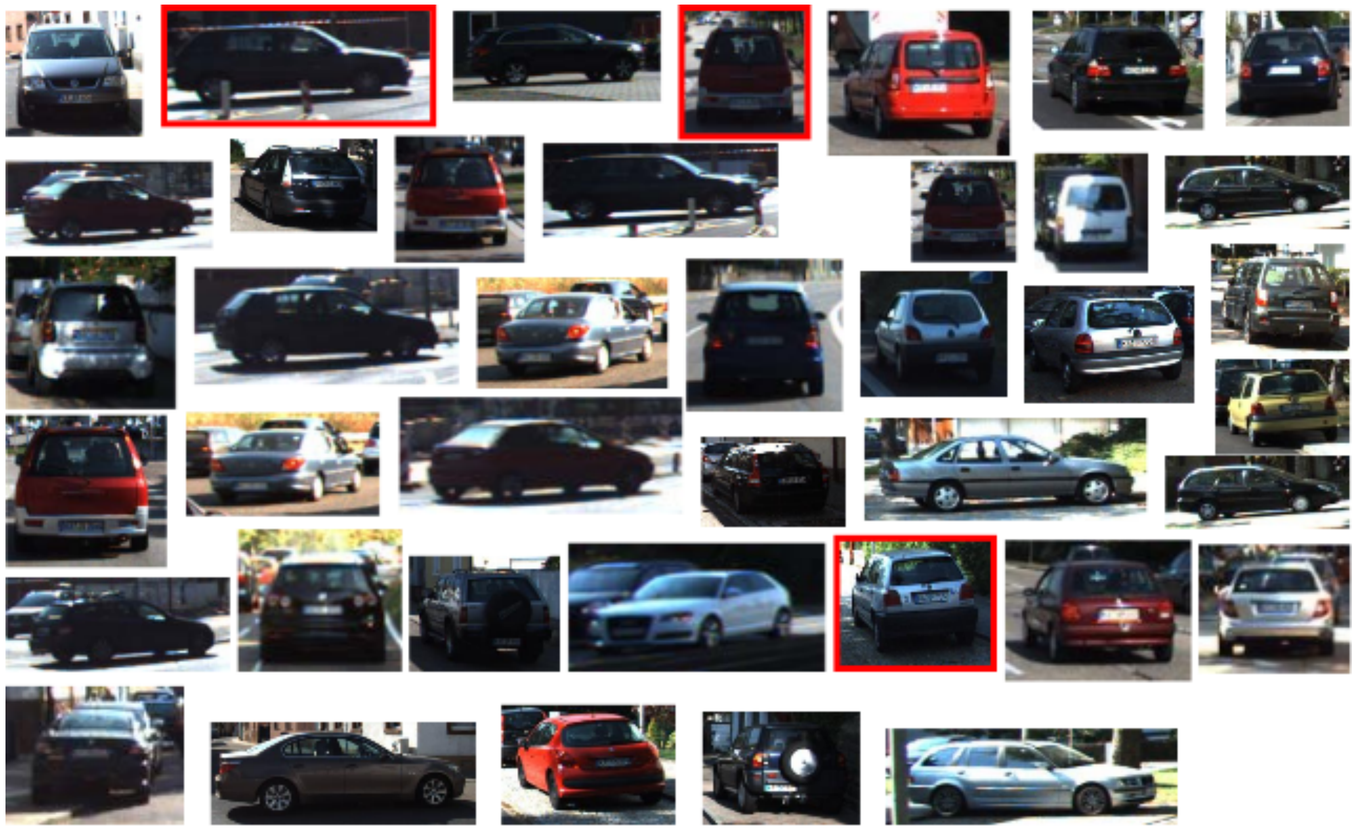
Double-Decker BUS

Does this scale: Training time?

Test time?

Next Assignment

- Identify most informative training examples

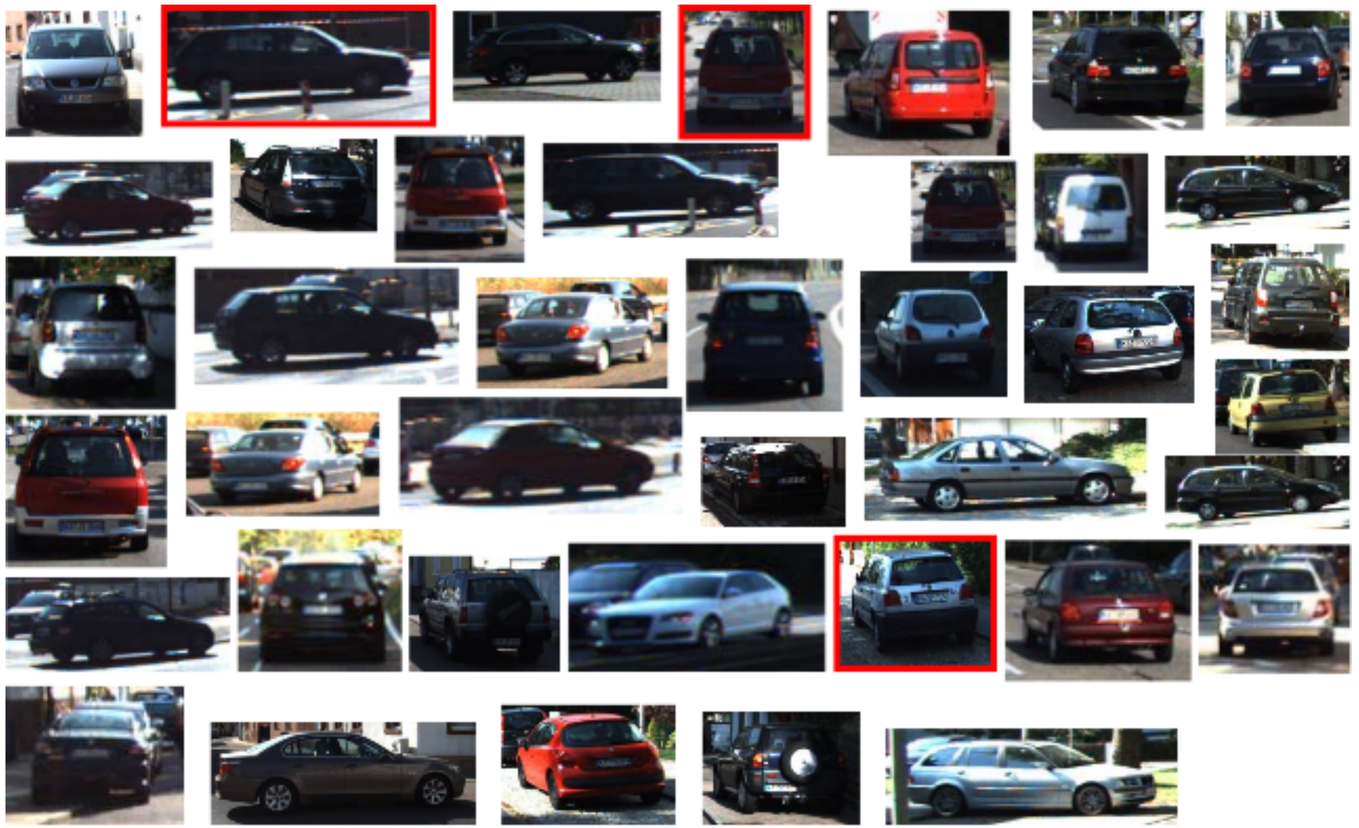


(a) Input Exemplars

(b) Selected Exemplars

Next Assignment

- Identify most informative training examples
  - Train detectors only on these examples

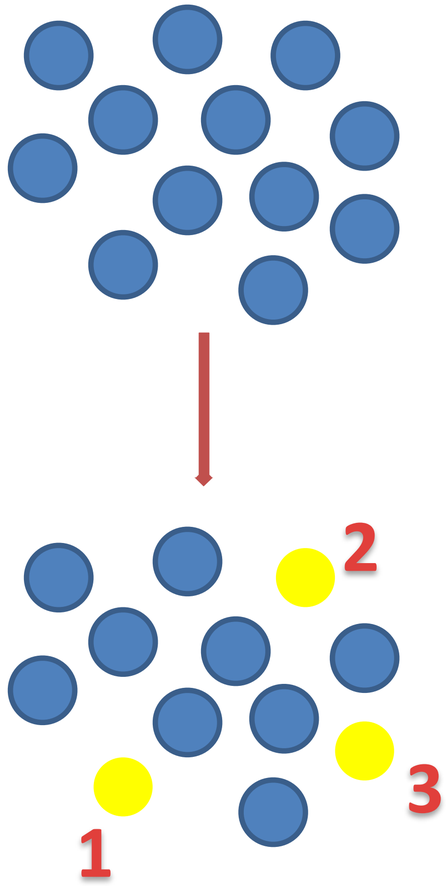


(a) Input Exemplars

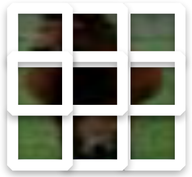
(b) Selected Exemplars

Problem statement

- Input
  - Set of labeled images (100)
- Output
  - Subset of images
  - Subset should be much smaller

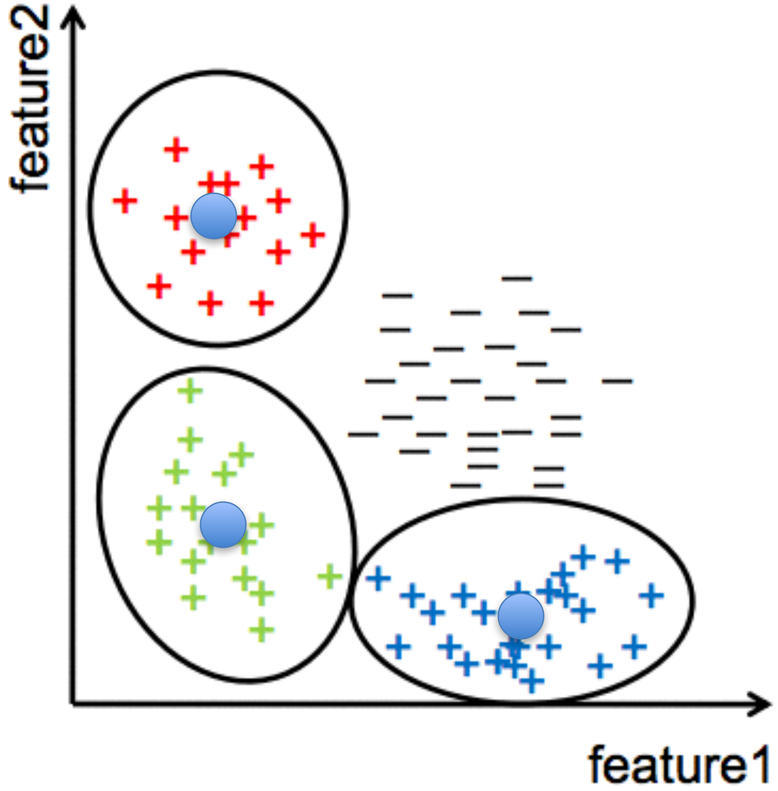


Clustering to the rescue!



NxNx31 HOG Feature (Color Histogram, Dense SIFT, Filter Responses)

Image Closest to the N cluster centers can be used to train N exemplars



But how do you detect Occluded objects?

Parts-and-shape models

- Model:
  - Object as a set of parts
  - Relative locations between parts
  - Appearance of part

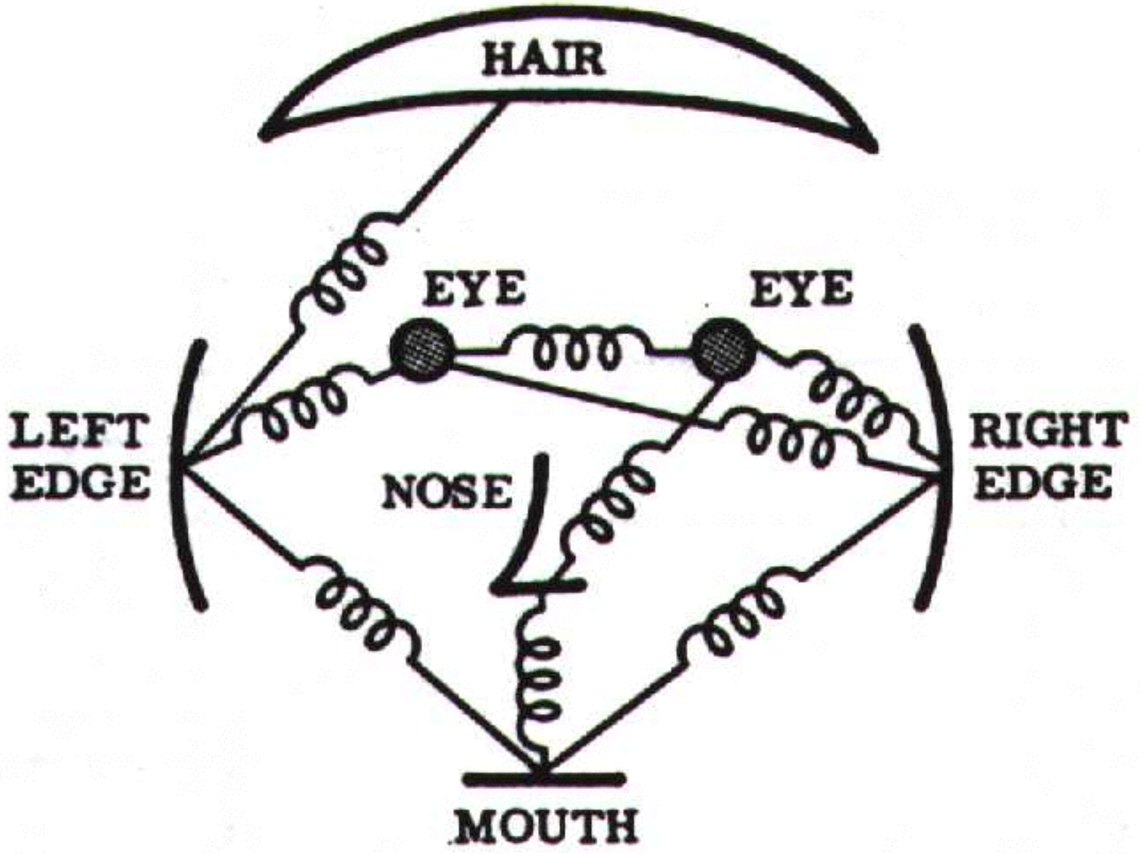
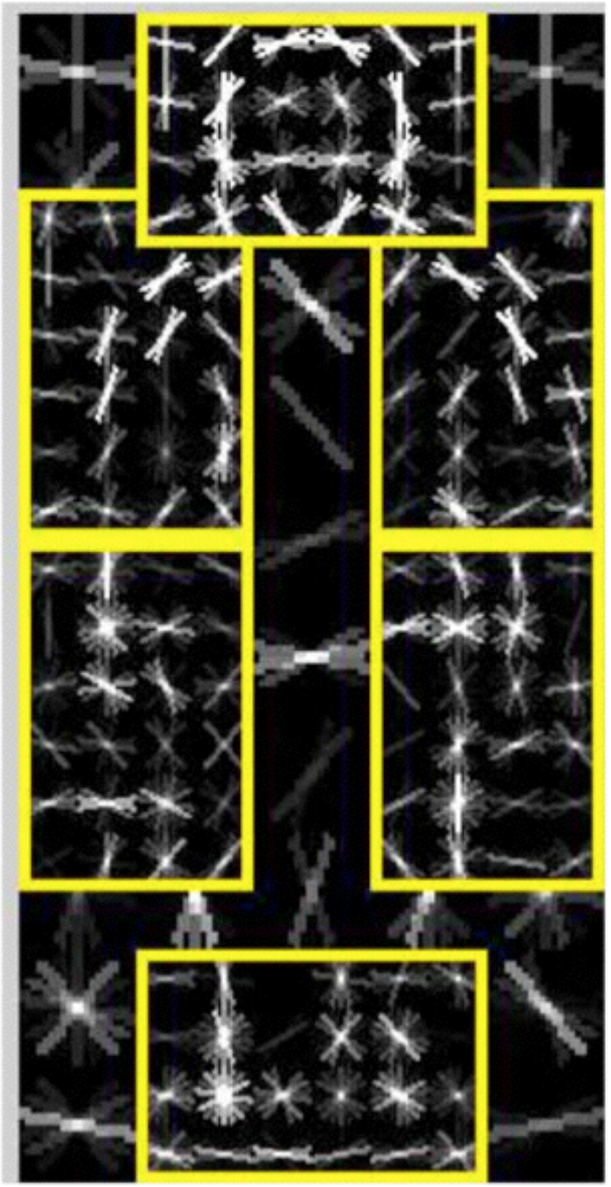


Figure from [Fischler & Elschlager 73]

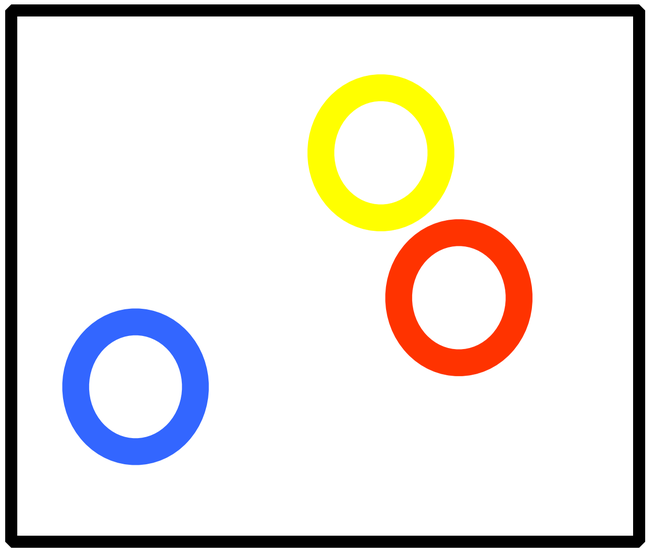
Deformable Model

- Break object into smaller **parts** that can **move around**
- But **enforce constraints** to prevent large deformations



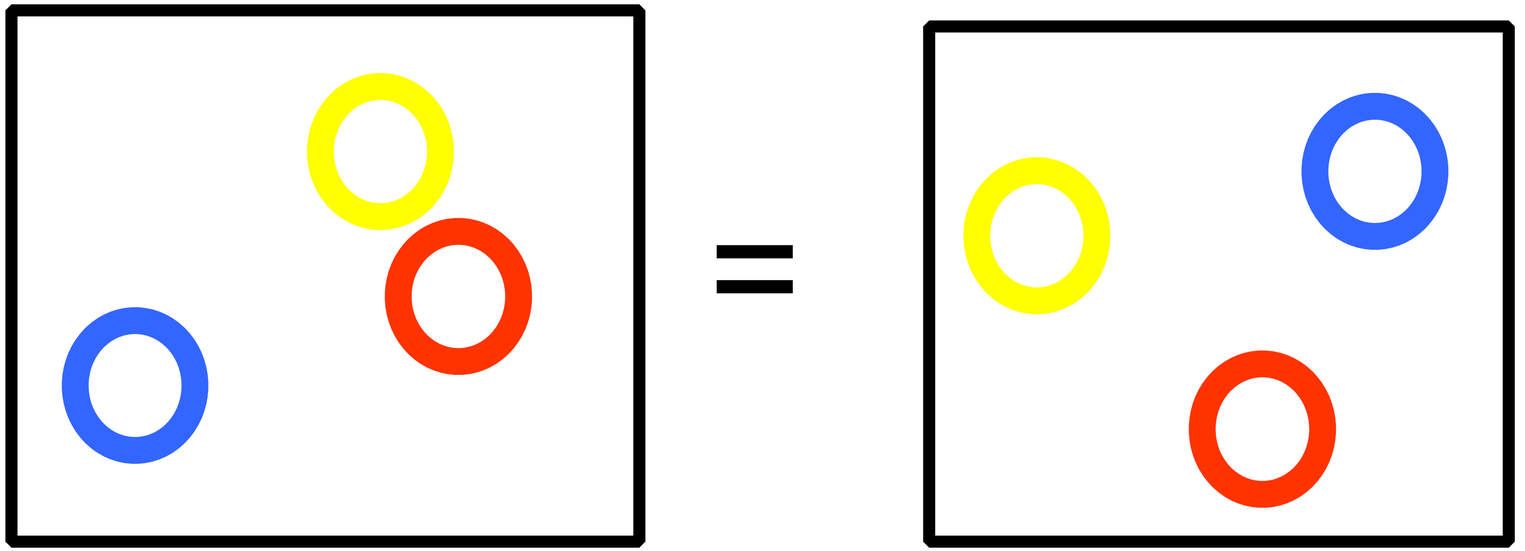
How to model spatial relations?

- One extreme: fixed template



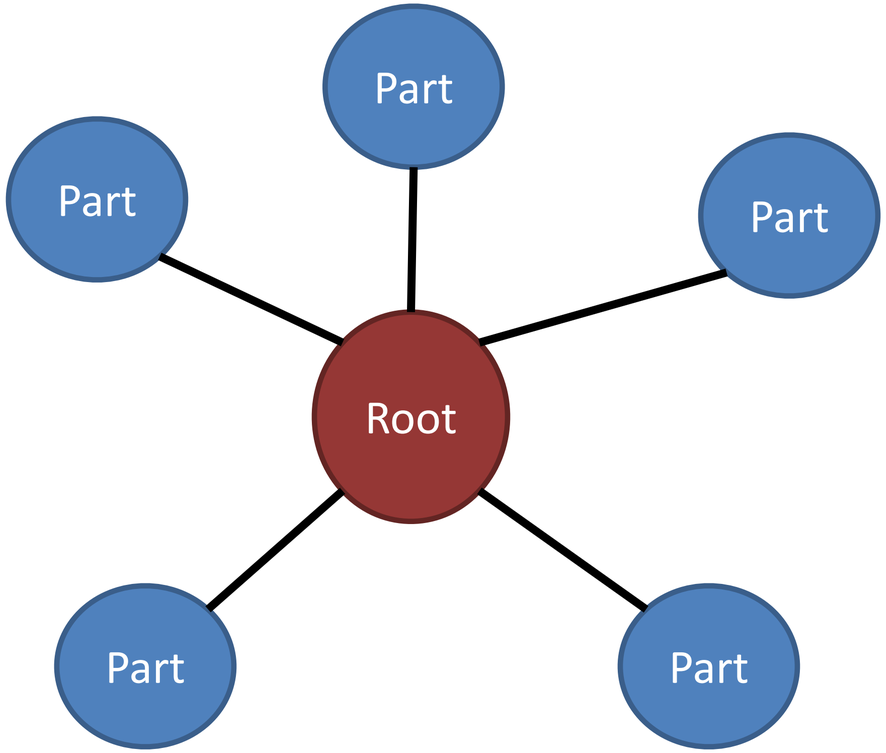
How to model spatial relations?

- Another extreme: bag of words



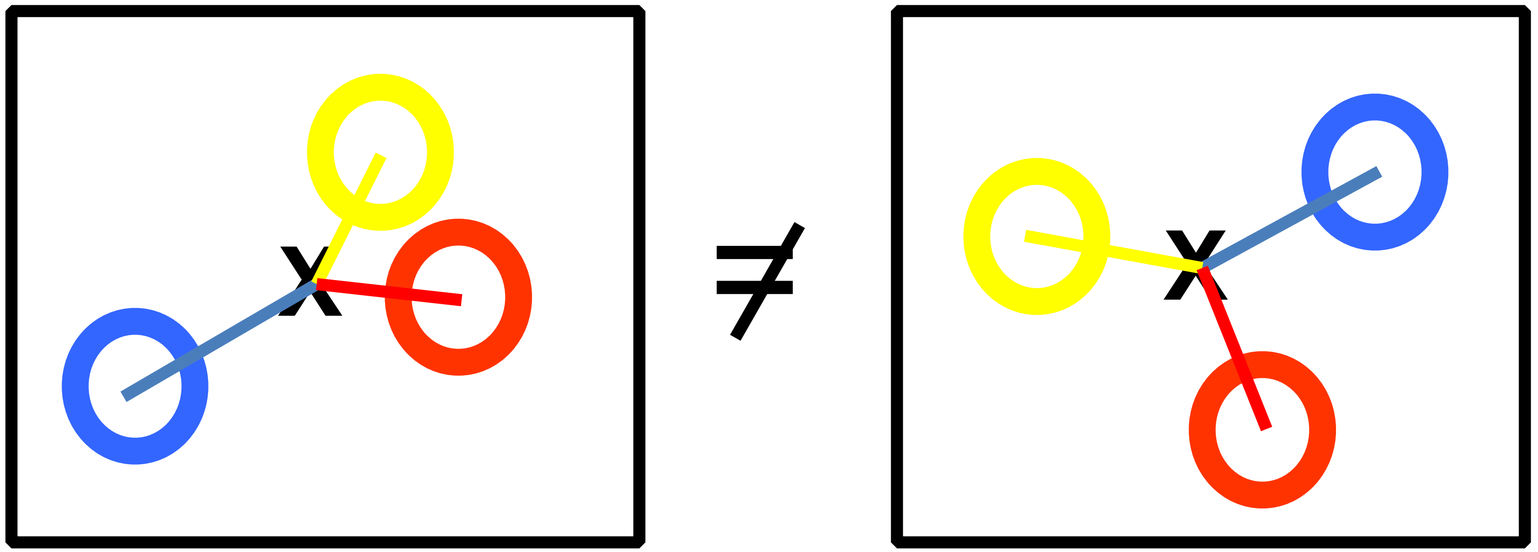
How to model spatial relations?

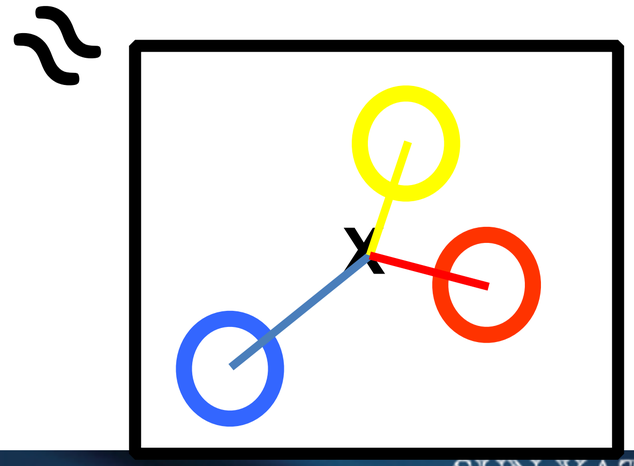
- Star-shaped model

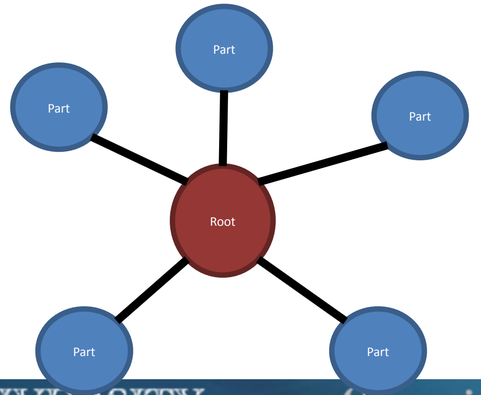


How to model spatial relations?

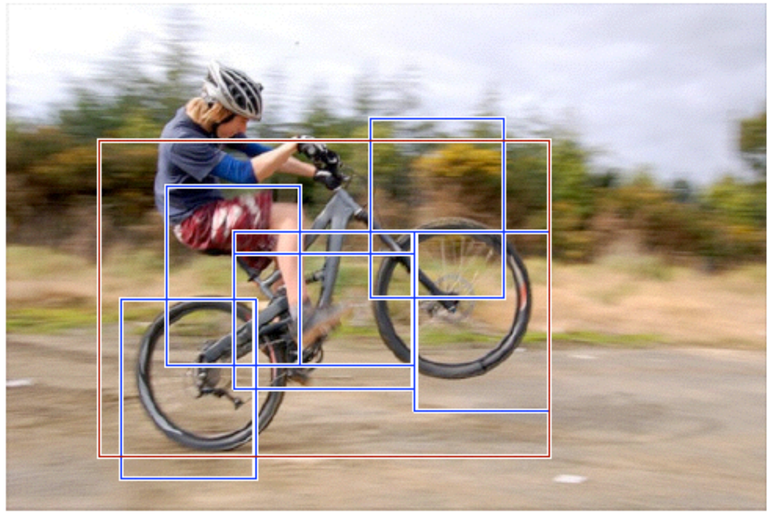
- Star-shaped model

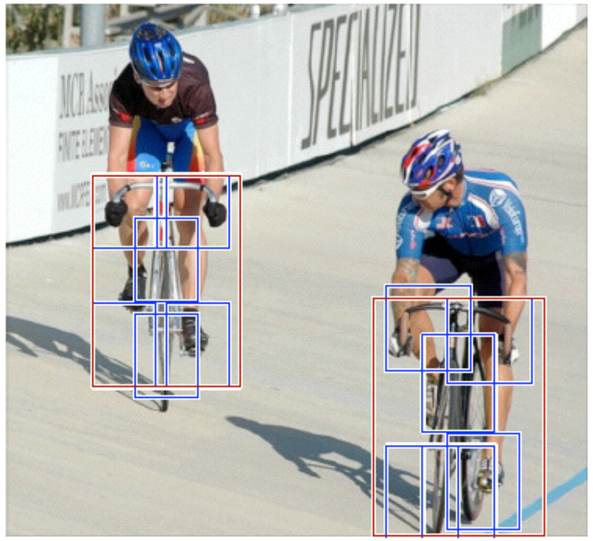


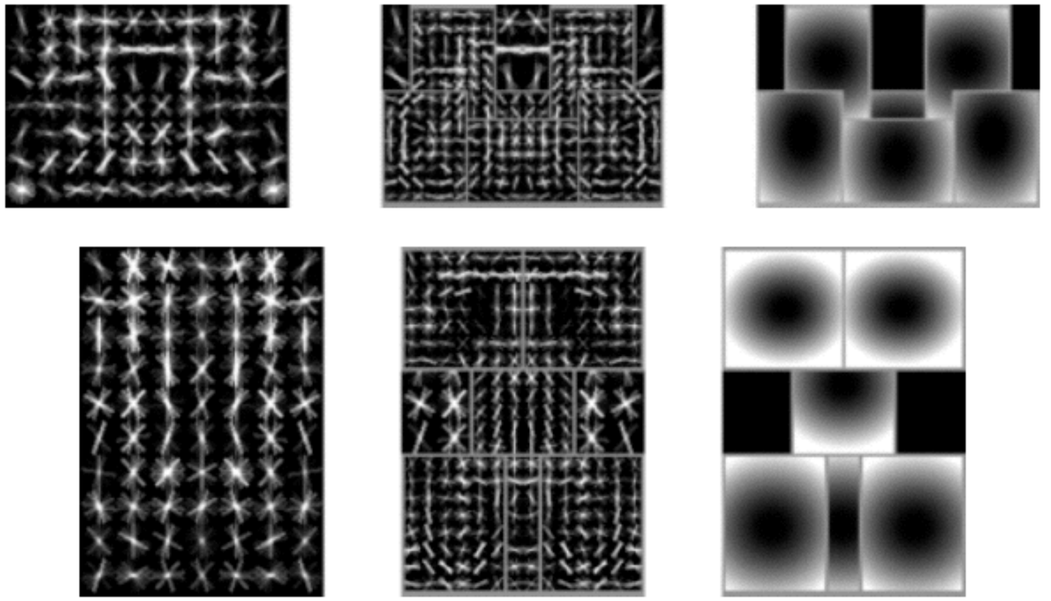




Representing a Model



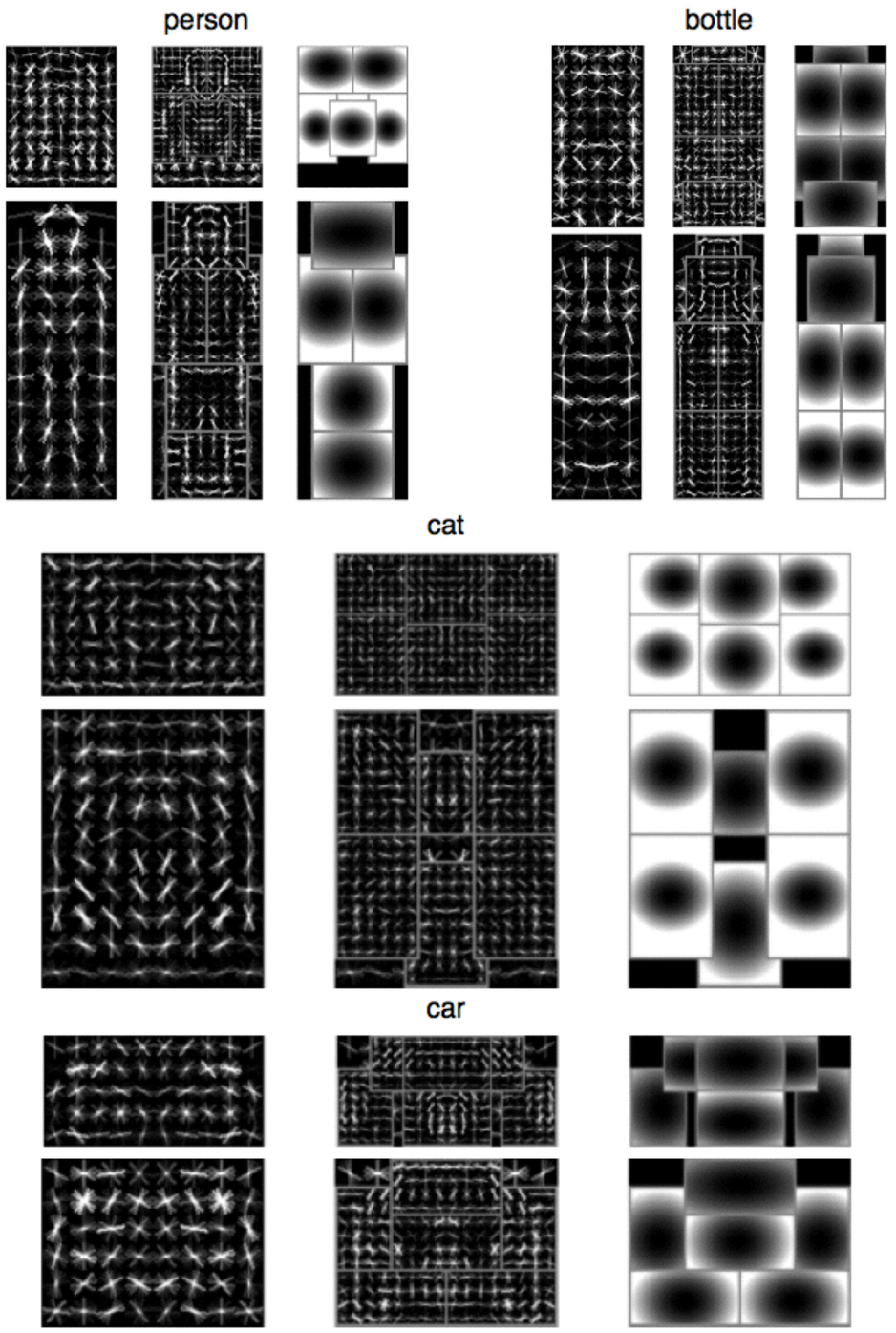




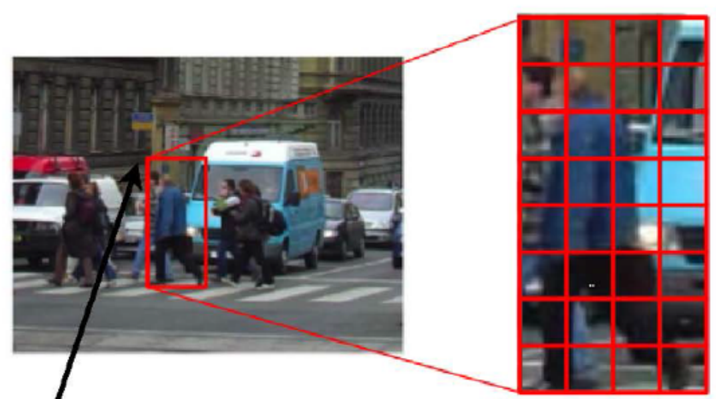
root filters
coarse resolution

part filters
finer resolution

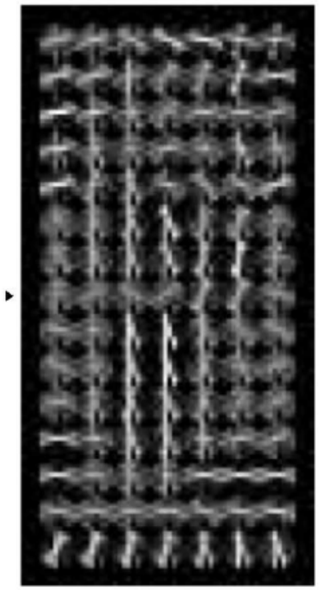
deformation
models


person
bottle
cat
car

Step back



$p = (x, y)$



$\cdot$

$\varphi(p) \cdot w$

Scoring a Detection

$$\max_{p} \left(w^T \phi(p) \right)$$

Weight learnt by SVM

HOG feature at location p

$$p = (x, y)$$

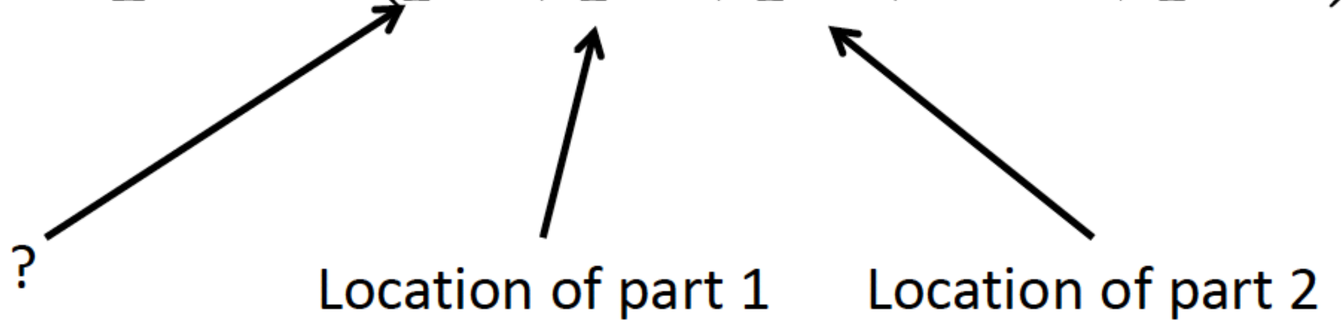
Slide from Varun Ramakrishna.

Part Configuration

Configuration: $\mathbf{p} = (p_0, p_1, p_2, \ldots, p_N)$

Part Configuration

Configuration: $\mathbf{p} = (p_0, p_1, p_2, \ldots, p_N)$



Scoring a Configuration

$$\mathtt{score}(\mathbf{p}) = \sum_{i=0}^{N} w_i^T \phi(p_i) + \sum_{ij} w_{ij}^T \psi(p_i, p_j)$$

Scoring a Configuration

$$\mathtt{score}(\mathbf{p}) = \sum_{i=0}^{N} w_i^T \phi(p_i) + \sum_{ij} w_{ij}^T \psi(p_i, p_j)$$

Weight learnt by SVM

Scoring a Configuration

$$\mathtt{score}(\mathbf{p}) = \sum_{i=0}^{N} w_i^T \phi(p_i) + \sum_{ij} w_{ij}^T \psi(p_i, p_j)$$

Weight learnt by SVM

HOG at location $\mathbf{p_i}$

Scoring a Configuration

$$\mathtt{score}(\mathbf{p}) = \sum_{i=0}^{N} w_i^T \phi(p_i) + \sum_{ij} w_{ij}^T \psi(p_i, p_j)$$

Weight learnt by SVM

HOG at location $\mathbf{p}_i$

Deformation parameter between part i & j

Scoring a Configuration

$$\mathtt{score}(\mathbf{p}) = \sum_{i=0}^{N} w_i^T \phi(p_i) + \sum_{ij} w_{ij}^T \psi(p_i, p_j)$$

Weight learnt by SVM

HOG at location $\mathbf{p}_i$

Deformation parameter between part i & j

Deformation function (usually quadratic)

Scoring a Configuration

$$\mathbf{score}(\mathbf{p}) = \sum_{i=0}^{N} w_i^T \phi(p_i) + \sum_{ij} w_{ij}^T \psi(p_i, p_j)$$

Weight learnt by SVM

HOG at location $\mathbf{p}_i$

Deformation parameter between part i & j

Deformation function (usually quadratic)

$$\psi(p_1, p_2) = (dx, dy, dx^2, dy^2)$$

$$dx = (p_1^x - p_2^x)$$

Best Configuration

$$\max_p \left(\texttt{score}(p)\right) = \max_p \left(\sum_{i=0}^{N} w_i^T \phi(p_i) + \sum_{ij} w_{ij}^T \psi(p_i, p_j) \right)$$

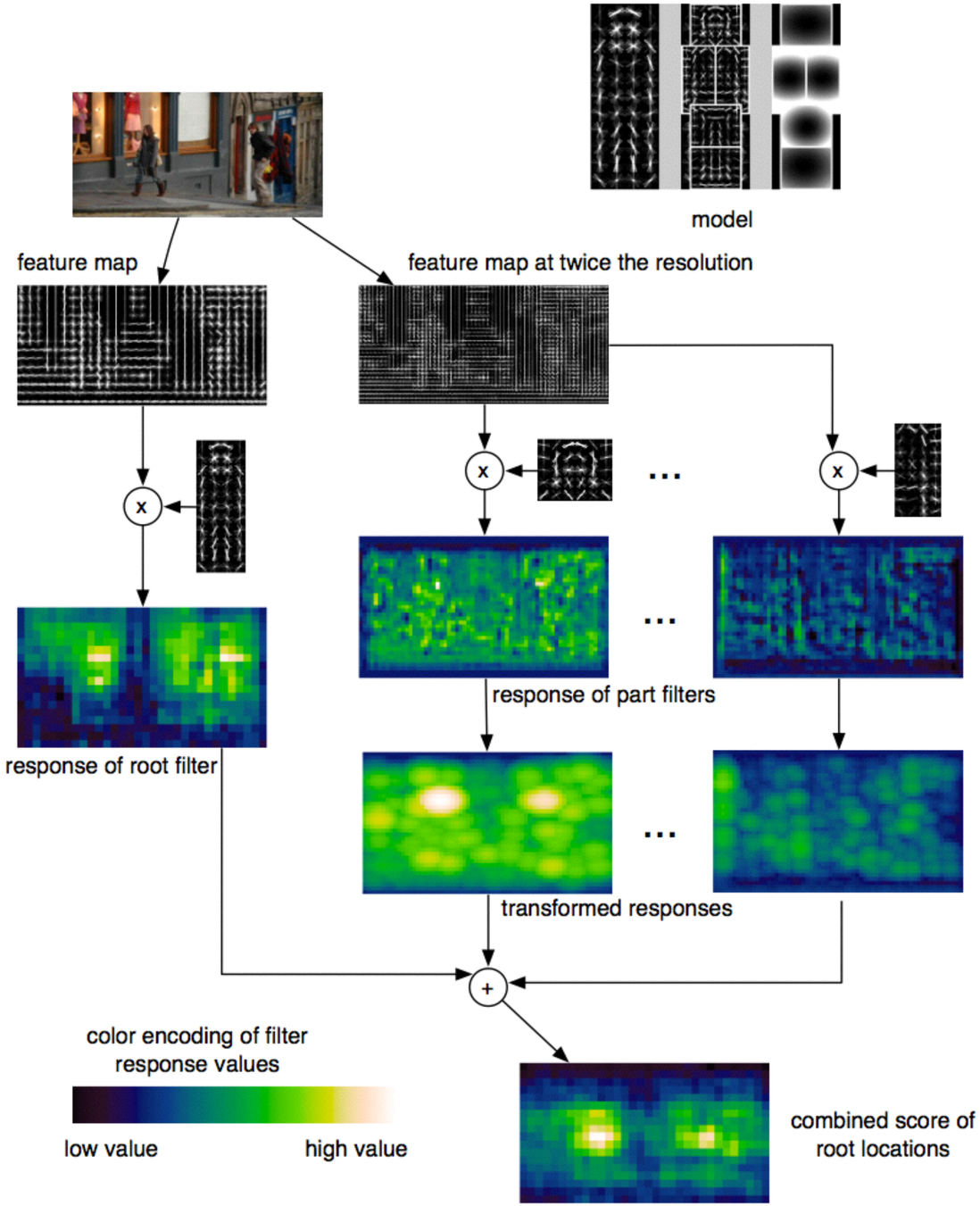
Best Configuration

$$\max_p \left(\texttt{score}(p)\right) = \max_p \left(\sum_{i=0}^{N} w_i^T \phi(p_i) + \sum_{ij} w_{ij}^T \psi(p_i, p_j) \right)$$

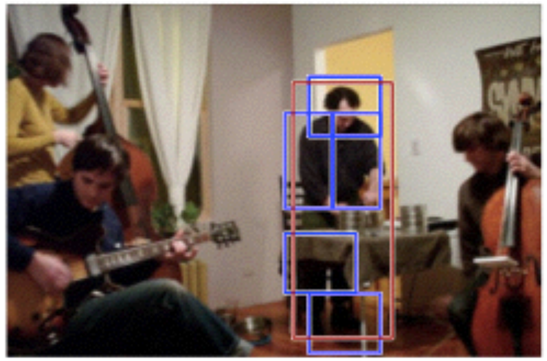
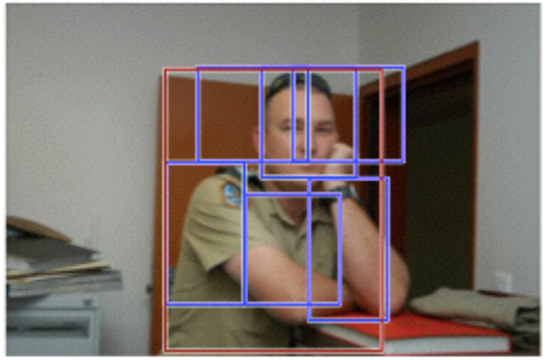
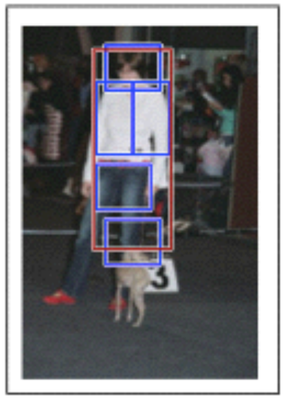
How many configurations?

For a 100x100 image, there are **(10^4)** possible locations for each part

No of configurations = **(10^4)^N**

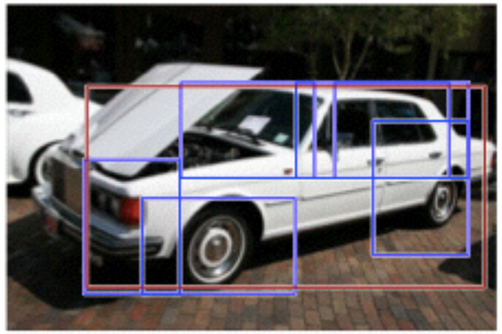
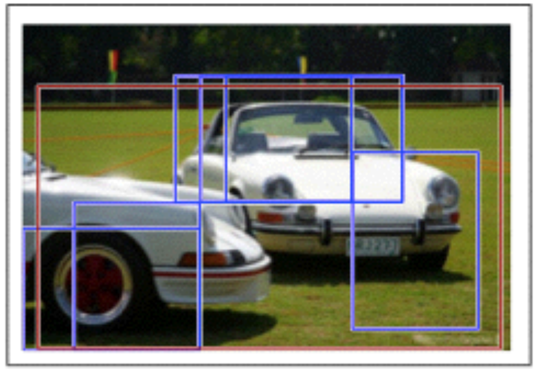
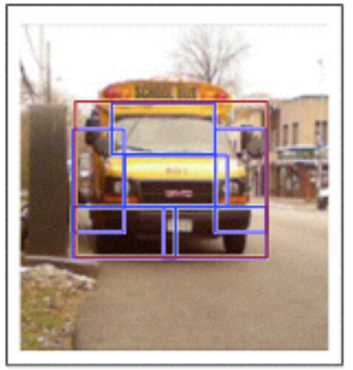

model
feature map
feature map at twice the resolution
response of part filters
response of root filter
transformed responses
color encoding of filter
response values
low value
high value
combined score of
root locations

person

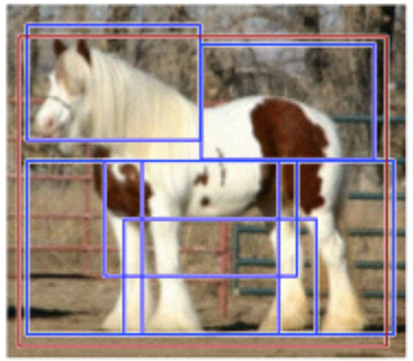
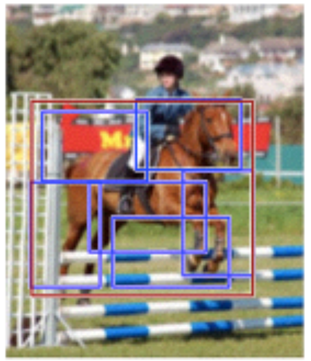
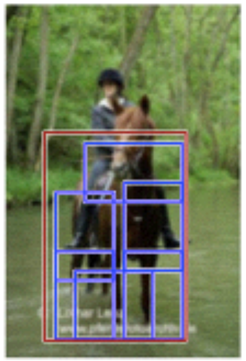
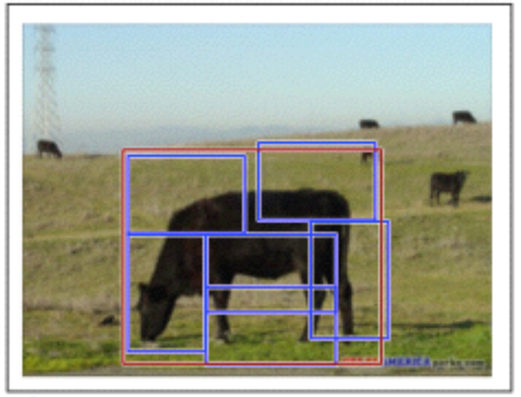
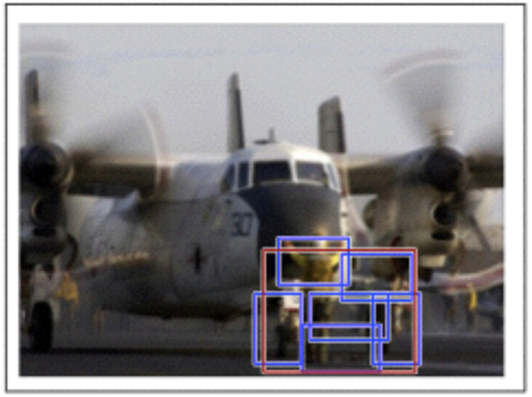






car



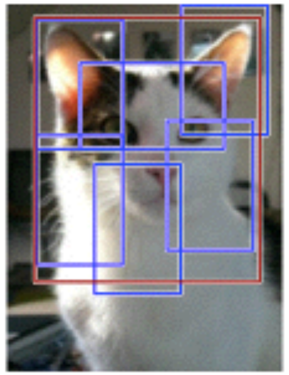
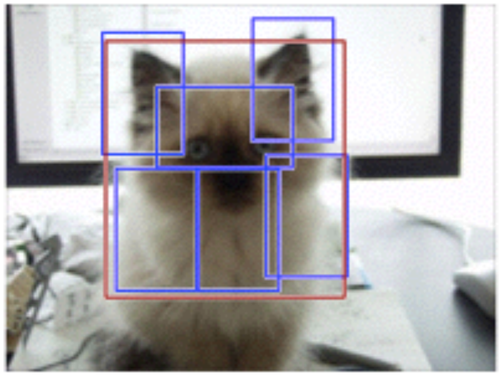
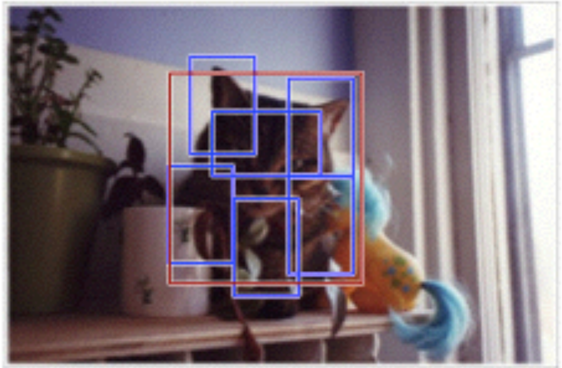
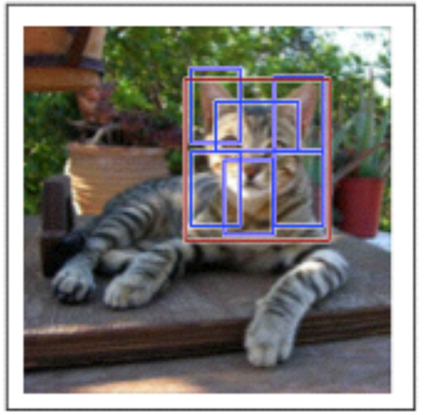
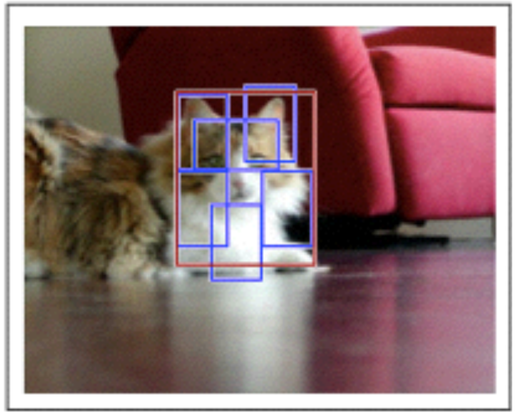
horse

bottle




cat

DPM

Detection results — PASCAL datasets

The models included with the source code were trained on the train+val dataset from each year and evaluated on the corresponding test dataset.
This is exactly the protocol of the "comp3" competition. Below are the average precision scores we obtain in each category.

	aero	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	*mean*
without context	45.6	49.0	11.0	11.6	27.2	50.5	43.1	23.6	17.2	23.2	10.7	20.5	42.5	44.5	41.3	8.7	29.0	18.7	40.0	34.5	*29.6*
with context	48.2	52.2	14.8	13.8	28.7	53.2	44.9	26.0	18.4	24.4	13.7	23.1	45.8	50.5	43.7	9.8	31.1	21.5	44.4	35.7	*32.2*
with context & extra octave	49.2	53.8	13.1	15.3	35.5	53.4	49.7	27.0	17.2	28.8	14.7	17.8	46.4	51.2	47.7	10.8	34.2	20.7	43.8	38.3	*33.4*
person detection grammar															49.9						

Table 1. PASCAL VOC 2010 comp3

	aero	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	*mean*
without context	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5	*33.7*
with context	36.6	62.2	12.1	17.6	28.7	54.6	60.4	25.5	21.1	25.6	26.6	14.6	60.9	50.7	44.7	14.3	21.5	38.2	49.3	43.6	*35.4*
person detection grammar															48.7						

Table 2. PASCAL VOC 2007 comp3