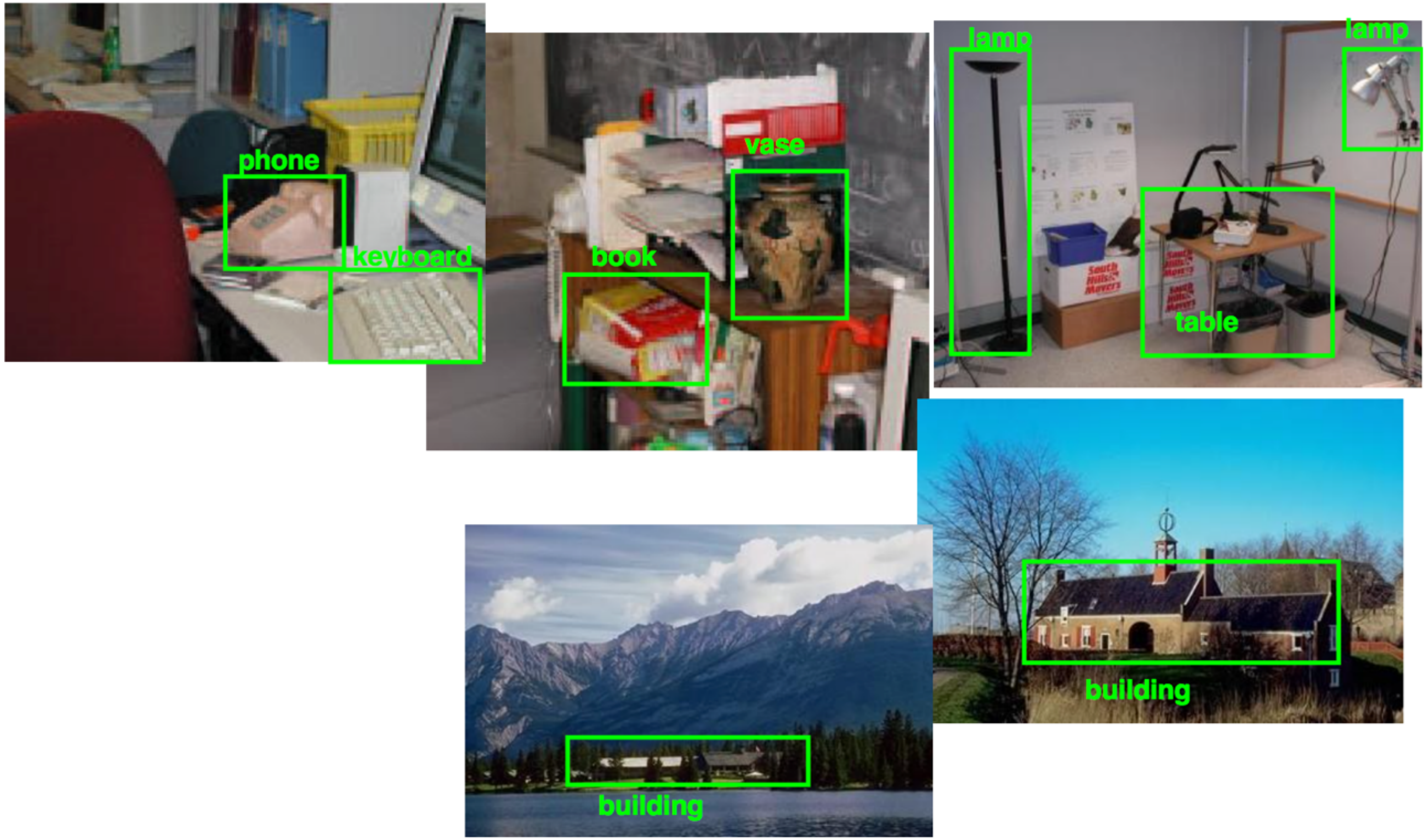
Recognition

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.

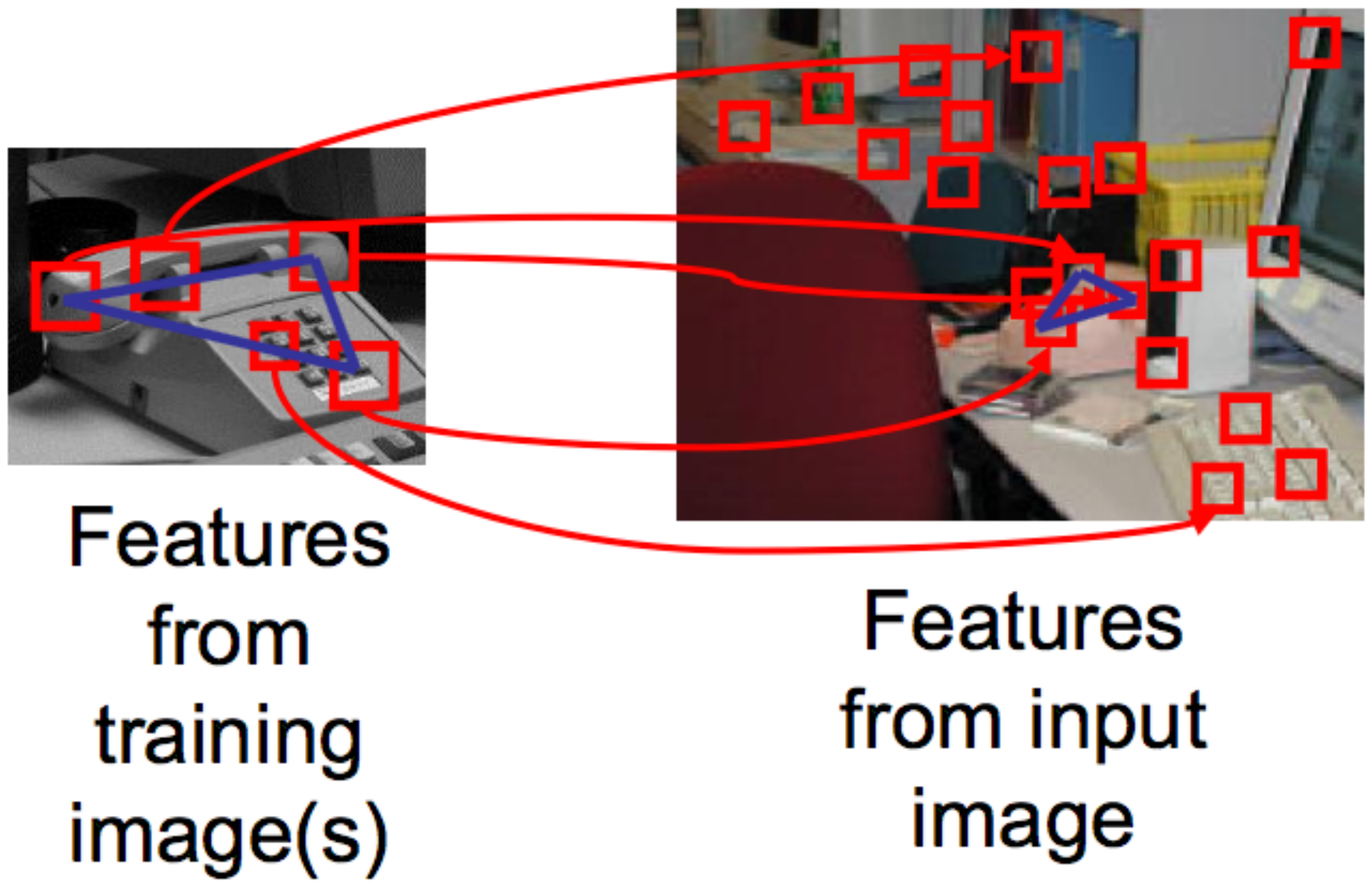
Query Image

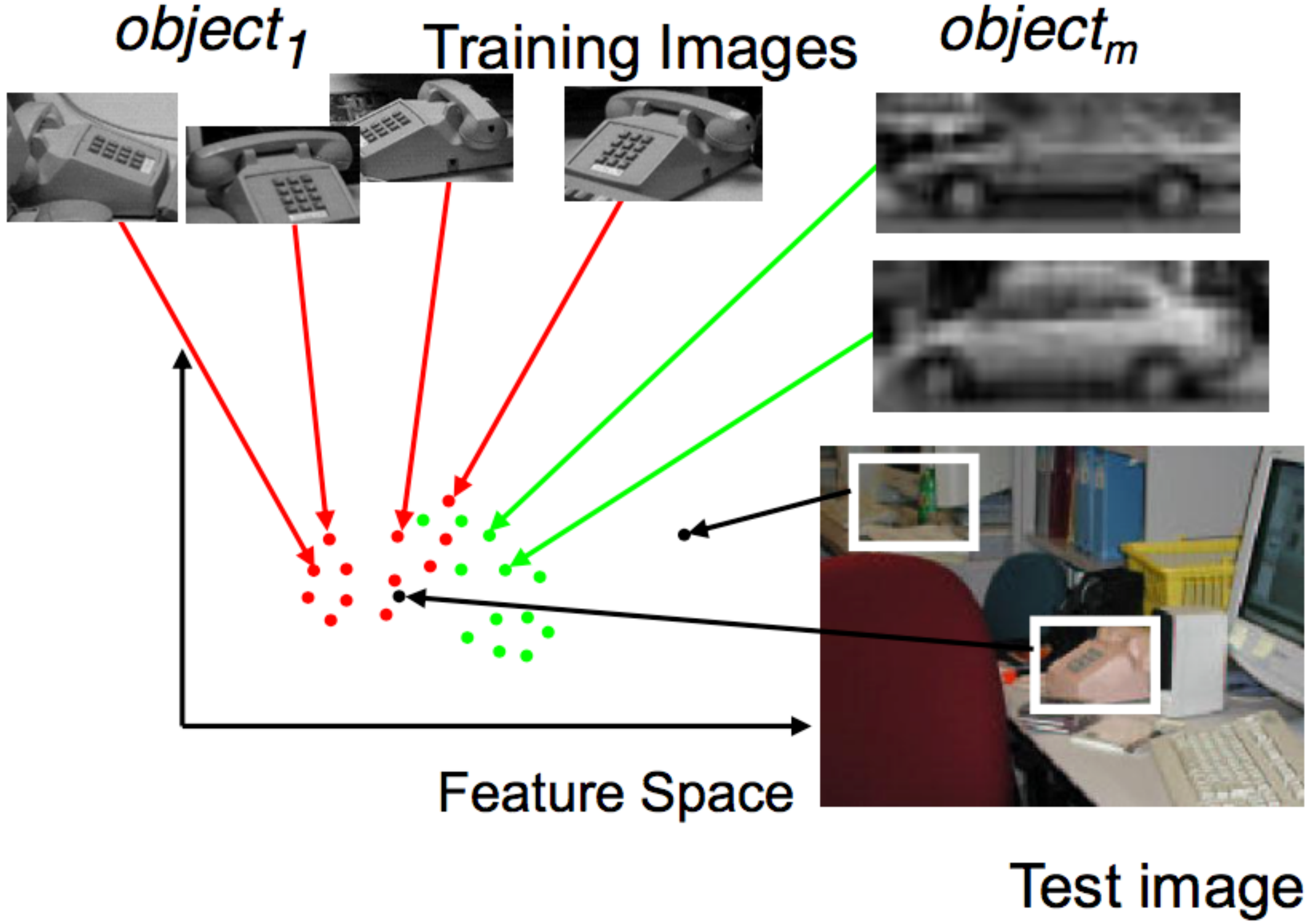
Output

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

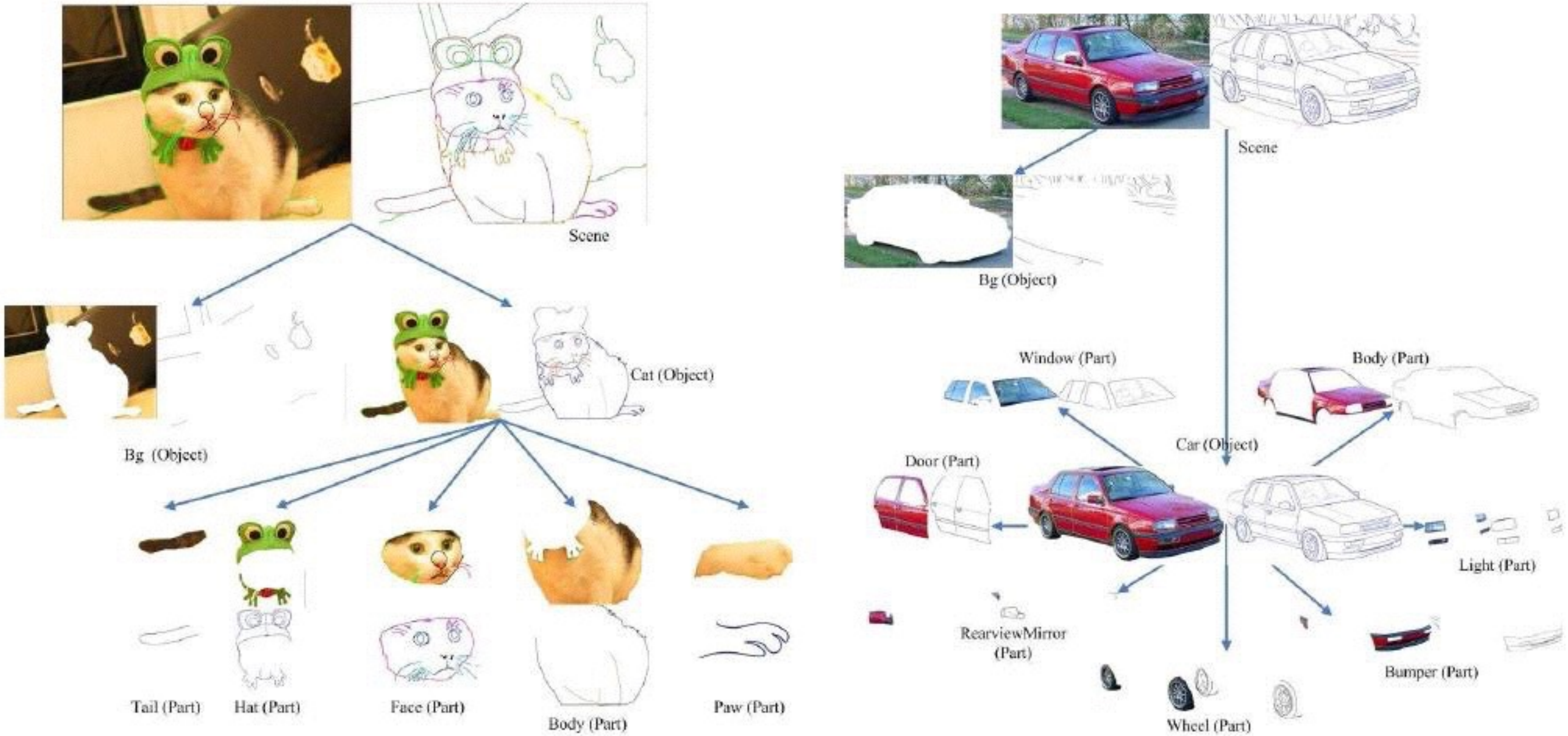


Figure 5: Two examples of the parse trees (cat and car) in the Lotus Hill Research Institute image corpus. From [87].

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

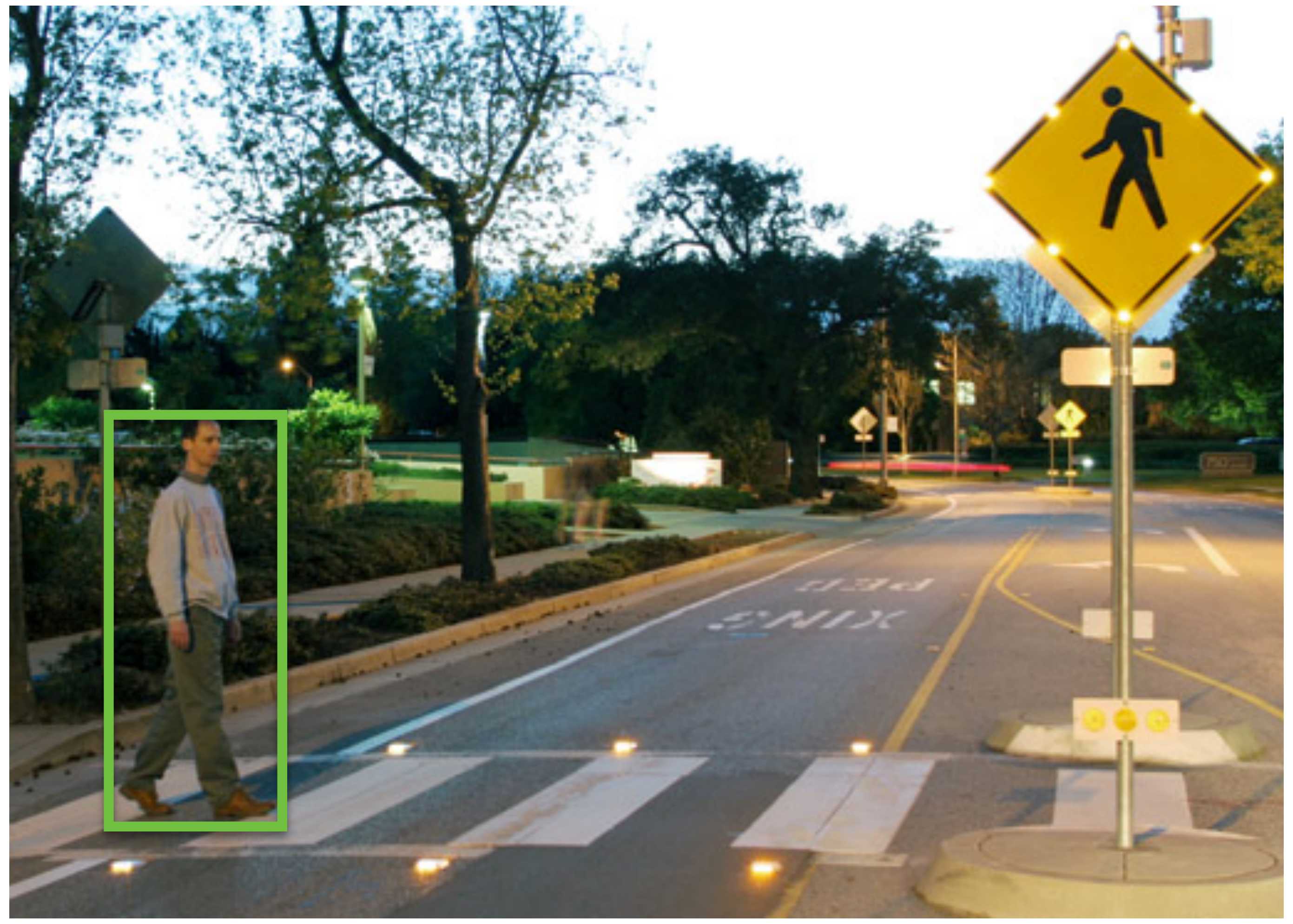
Intersection over Union

- Correct Detection if…

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

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

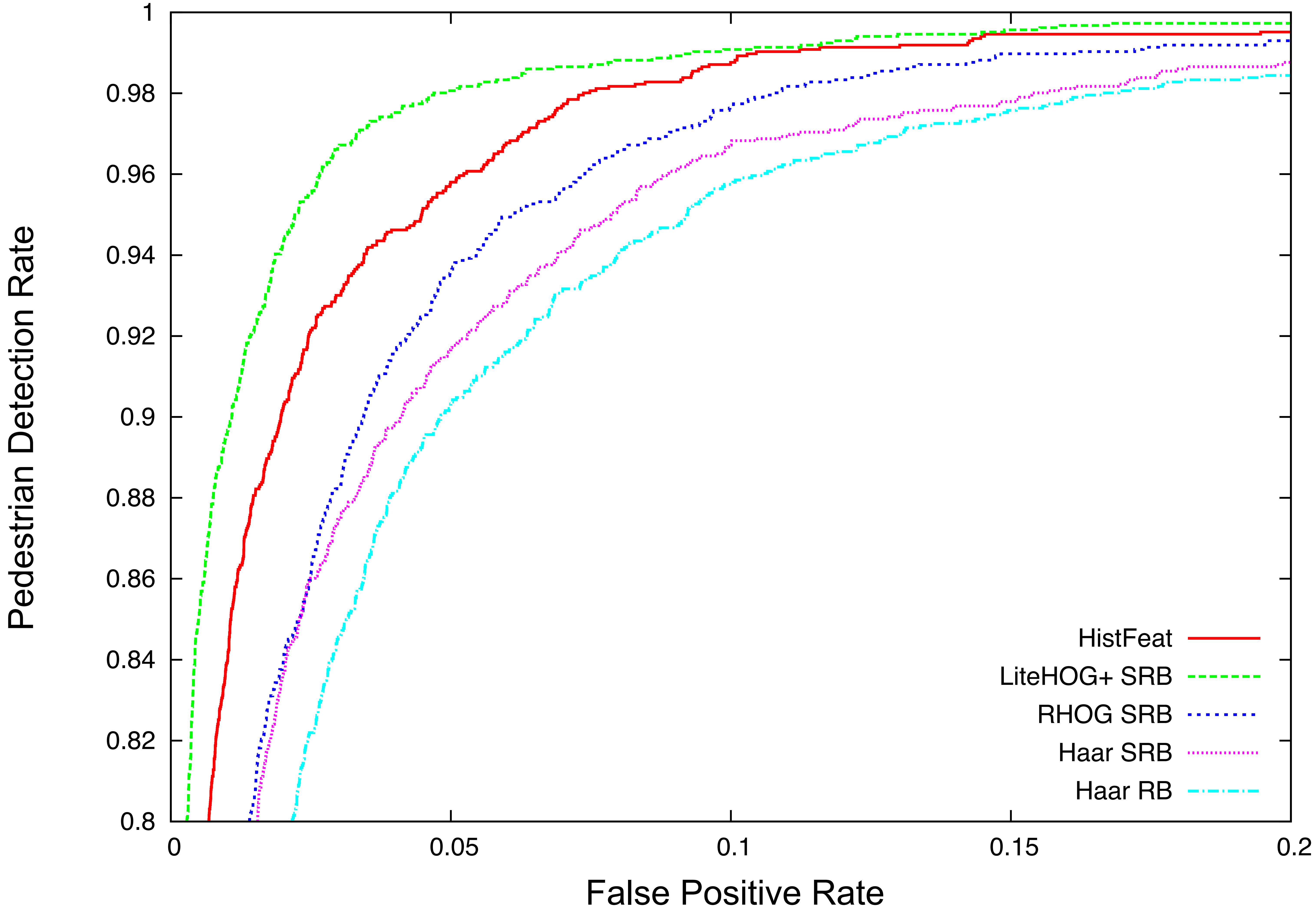
ROC Curve

Detection Rate

- TP/(TP-FN)

Prefer a Single Measure?

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



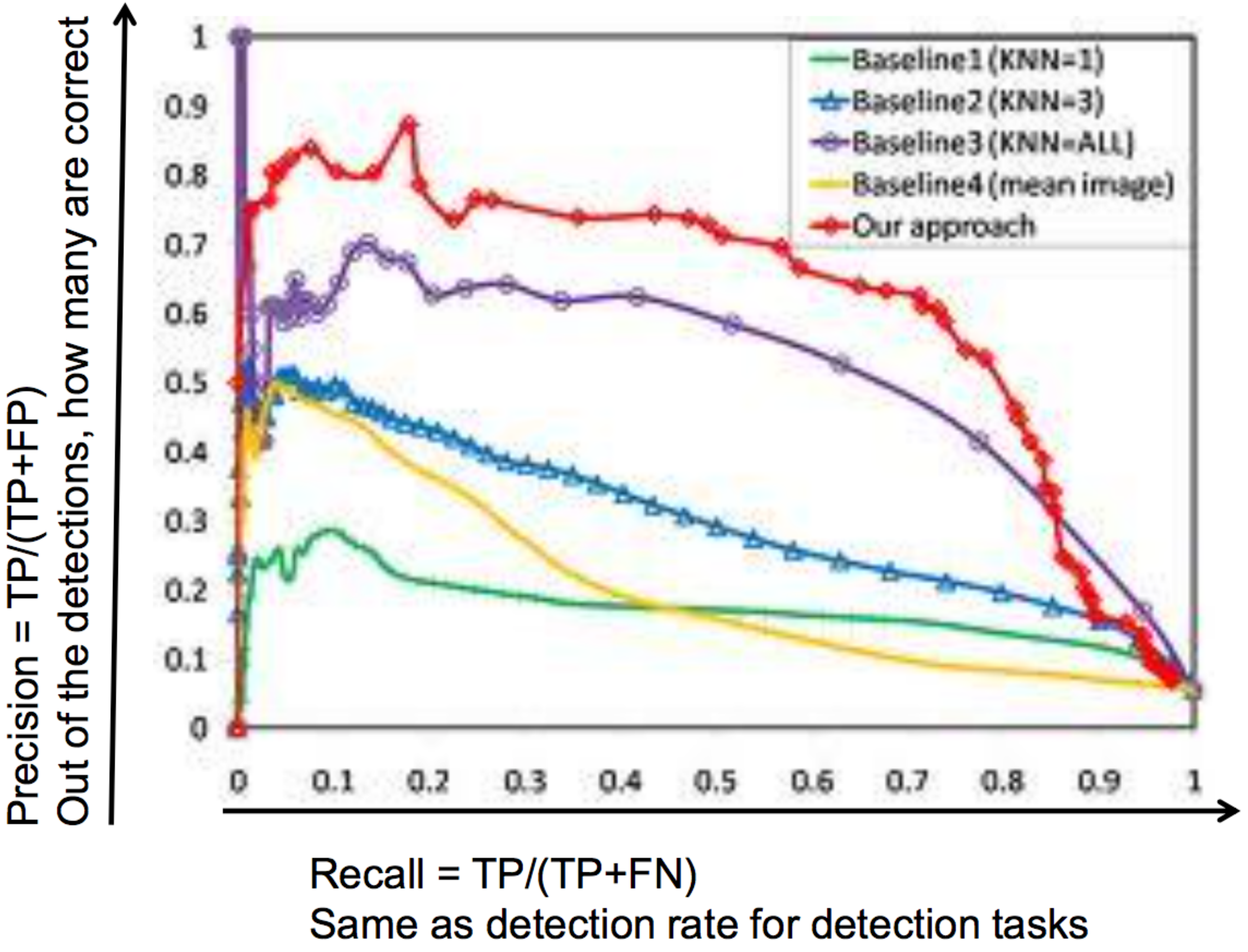
Precision-Recall Curve

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

Prefer a Single Measure?

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

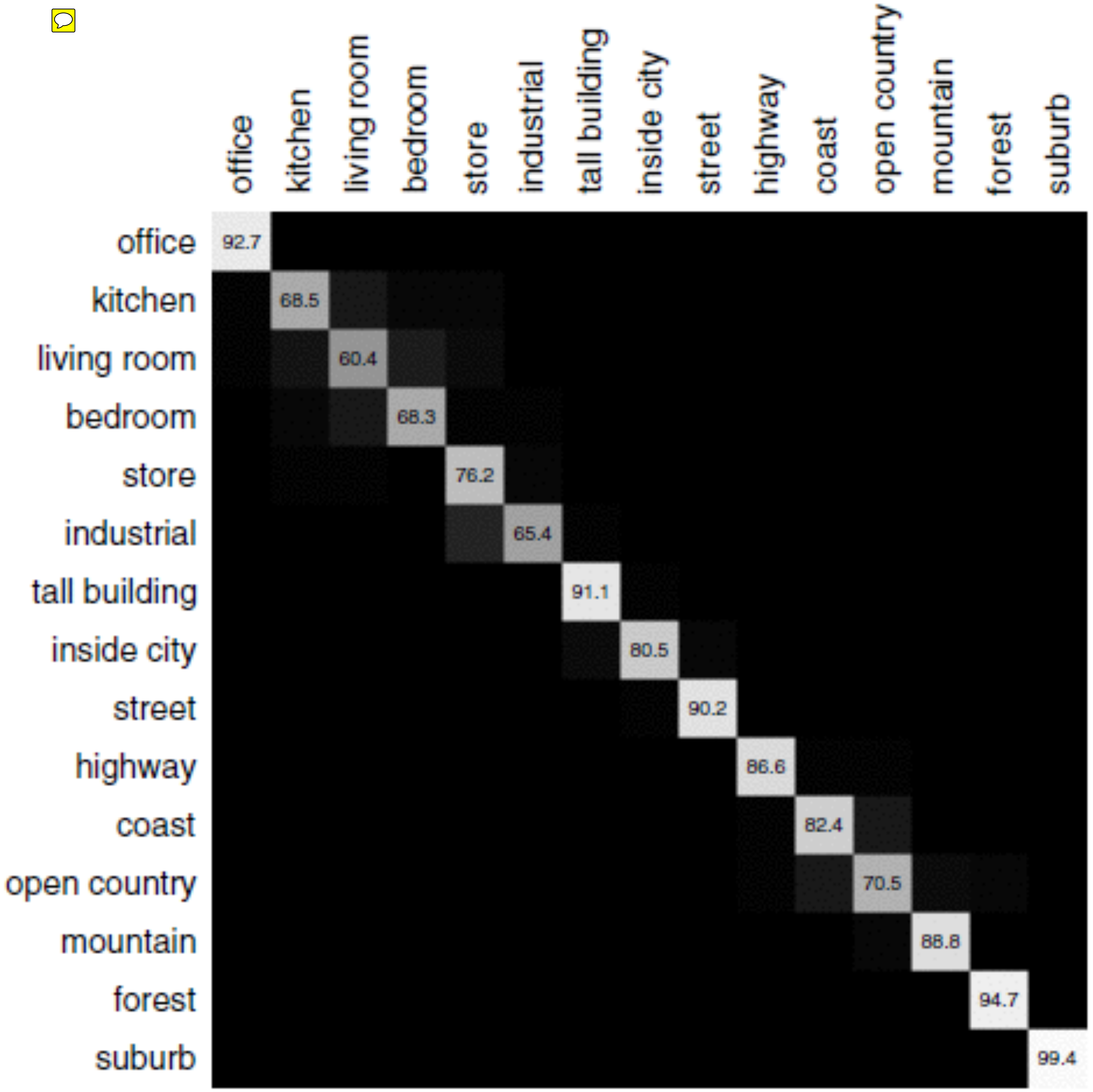
Davis and Goadrich. The Relationship Between Precision-Recall and ROC Curves. ICML (Intern. Conf. Machine Learning) 2006.



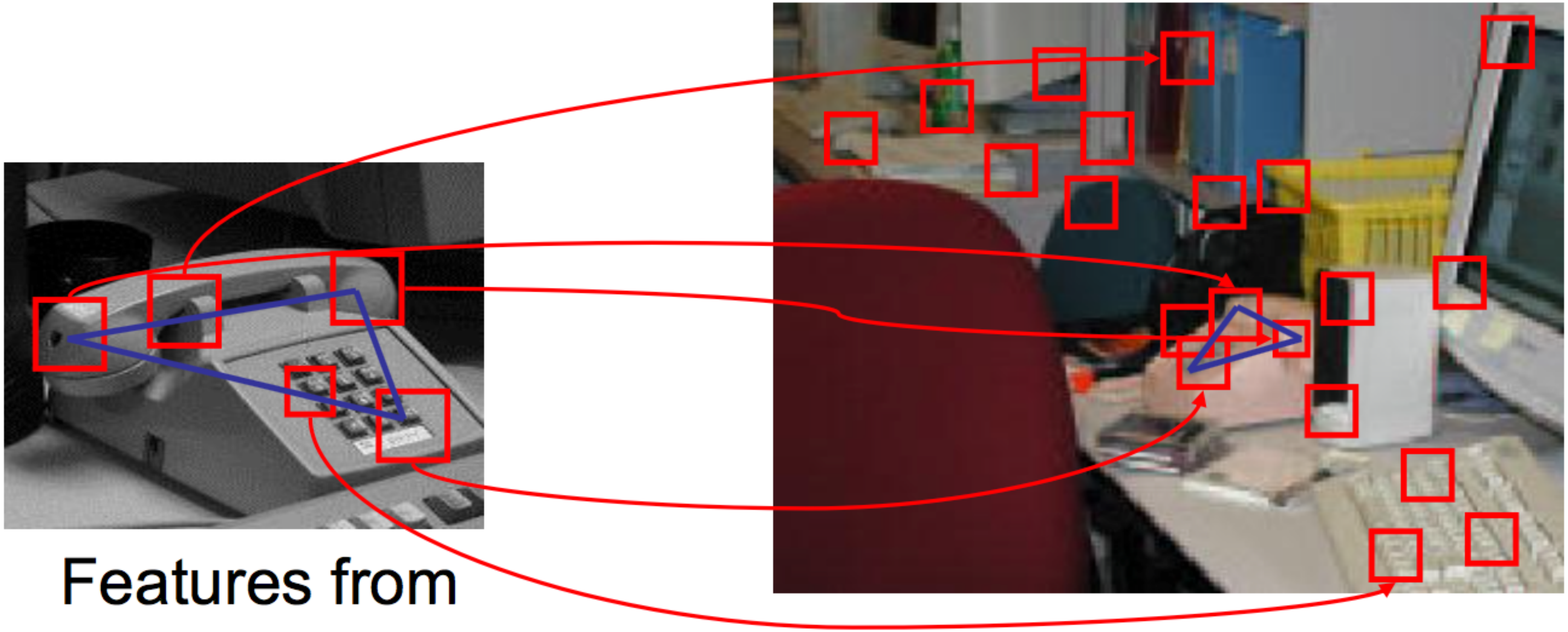
Confusion Matrix

- 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

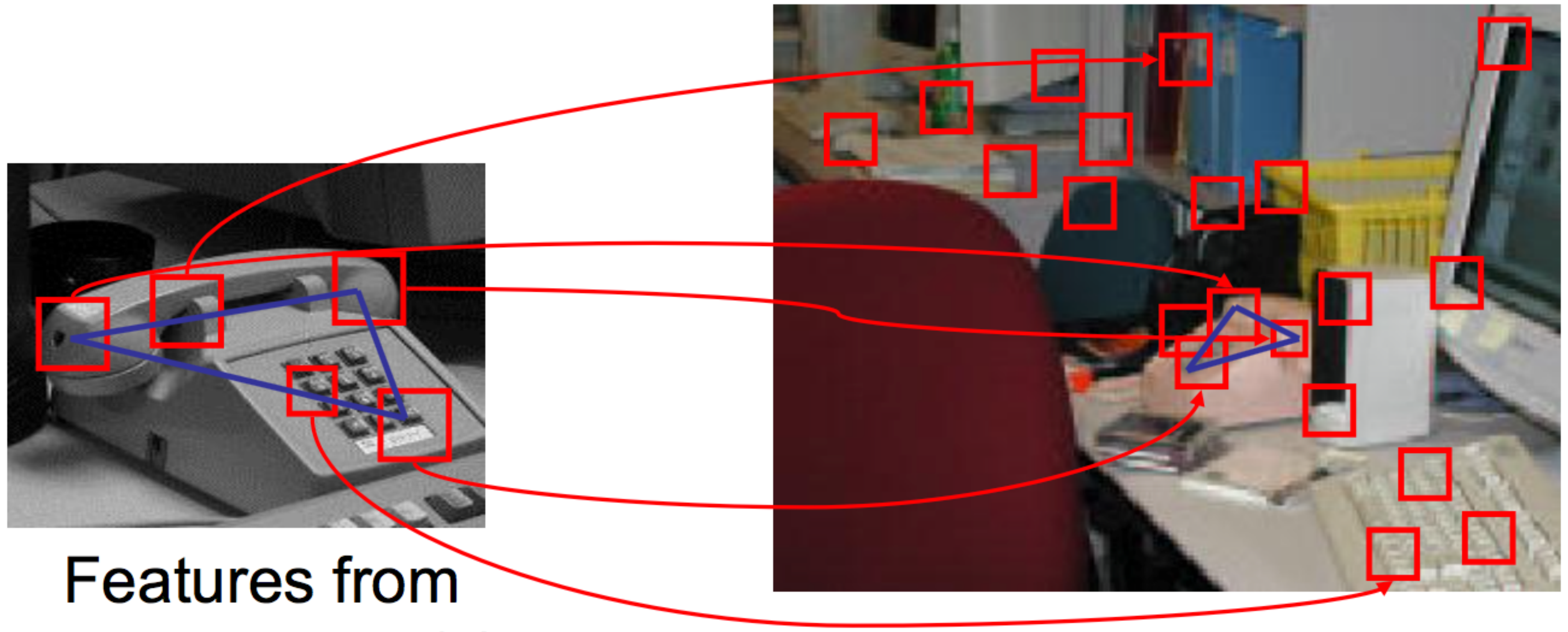


Features from training image(s)

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

Example Using SIFT

k=2

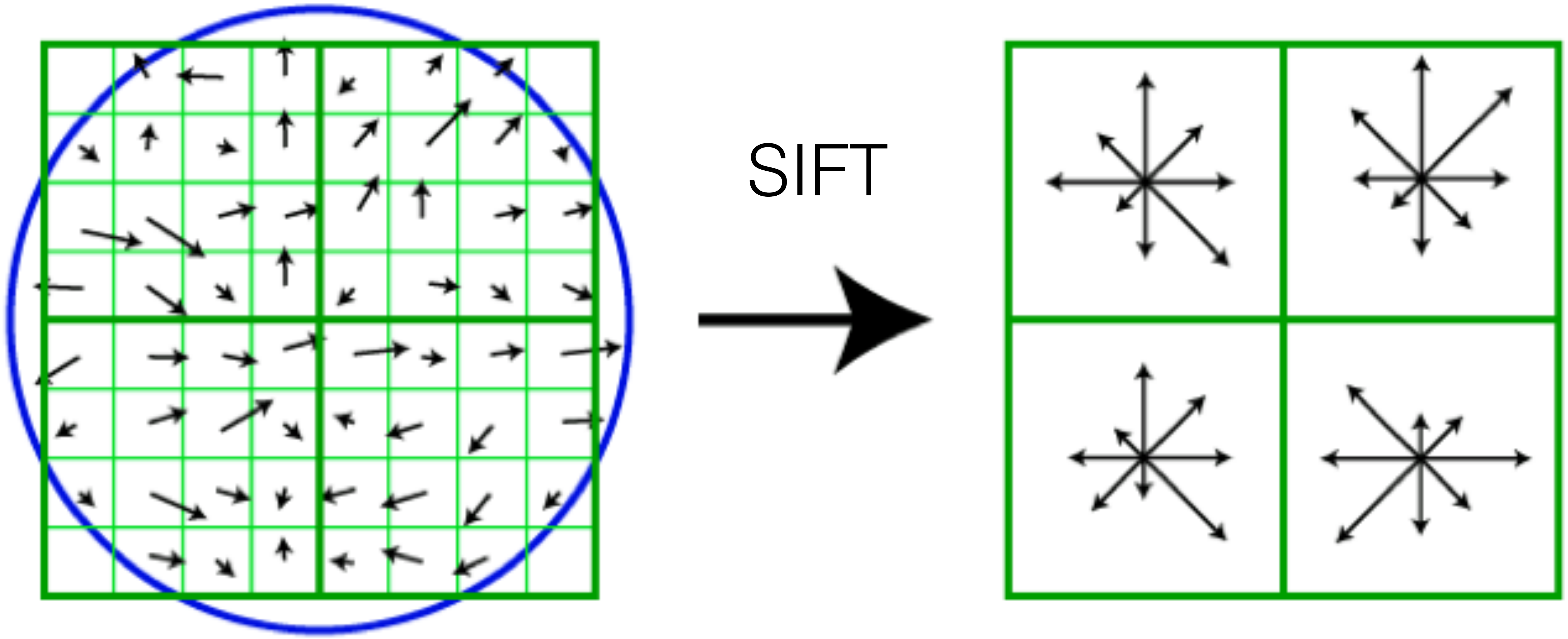


Image gradients

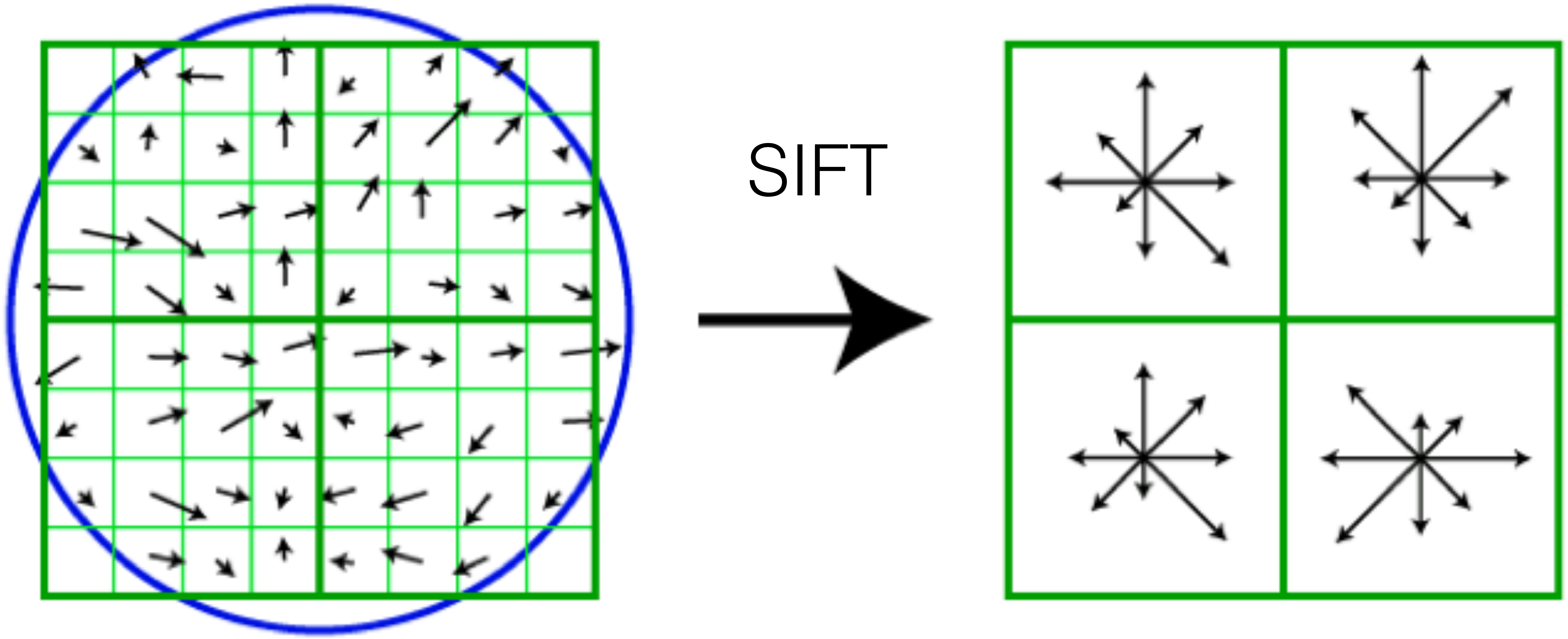
Keypoint descriptor

N=32

SIFT - Recap

- Image gradients are sampled over 4*k*x4*k* array of locations around interest point
- Create array of orientation histograms over 8 orientations in each of k^2 4x4 blocks
- 8 orientations x *k*x*k* histogram array $N = 8\ k^2$ dimensions
- In practice $k = 4$ therefore $N = 128$, i.e. a 128-vector

k=2



N=32

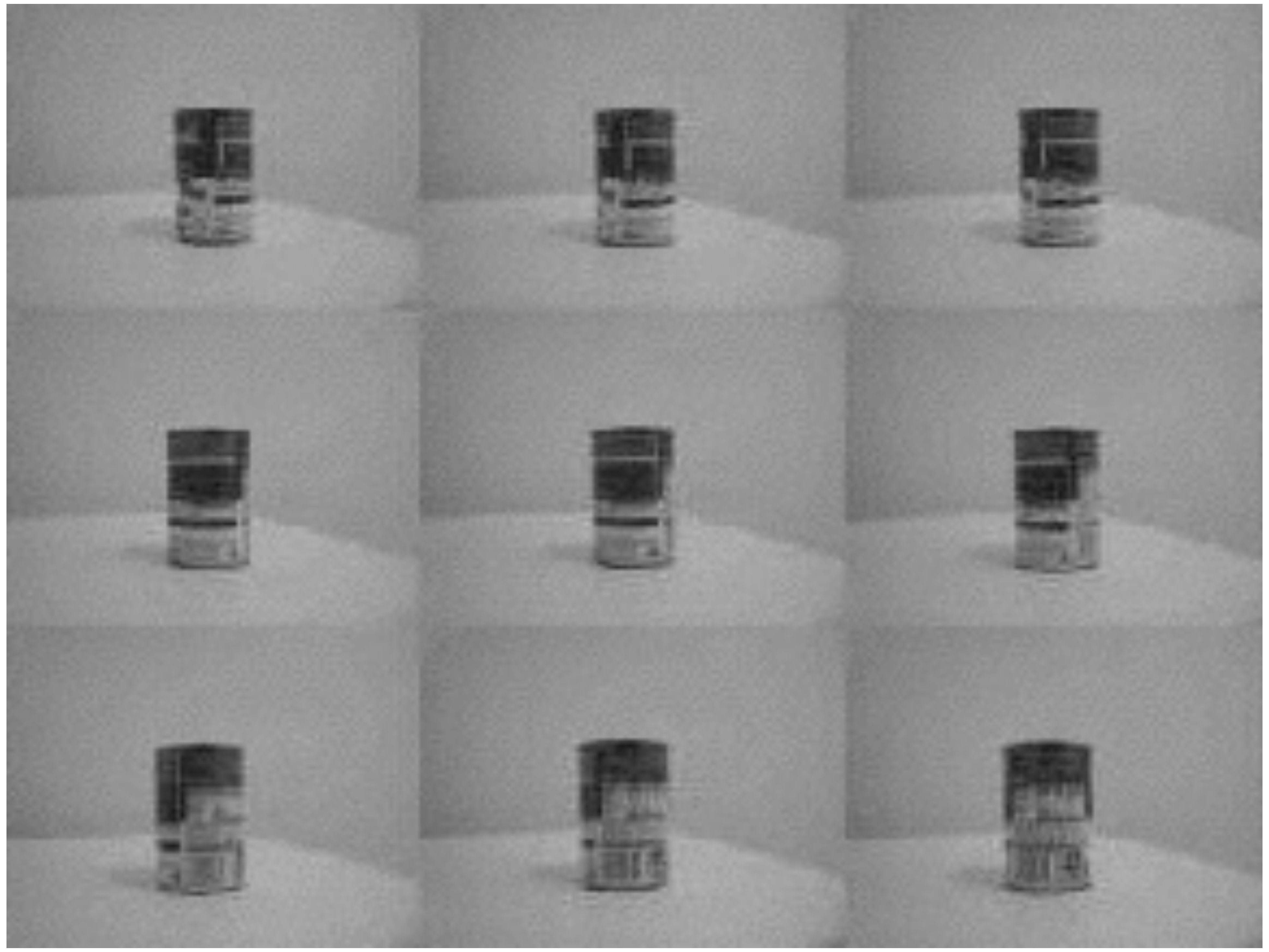
Image gradients   Keypoint descriptor

Example Using SIFT

- For each SIFT feature, find the closest feature in the reference set
- Keep match if distance is greater than $(1+\delta)$ times distance to next neighbor
- Apply RANSAC to set of potential correspondences to verify geometric consistency

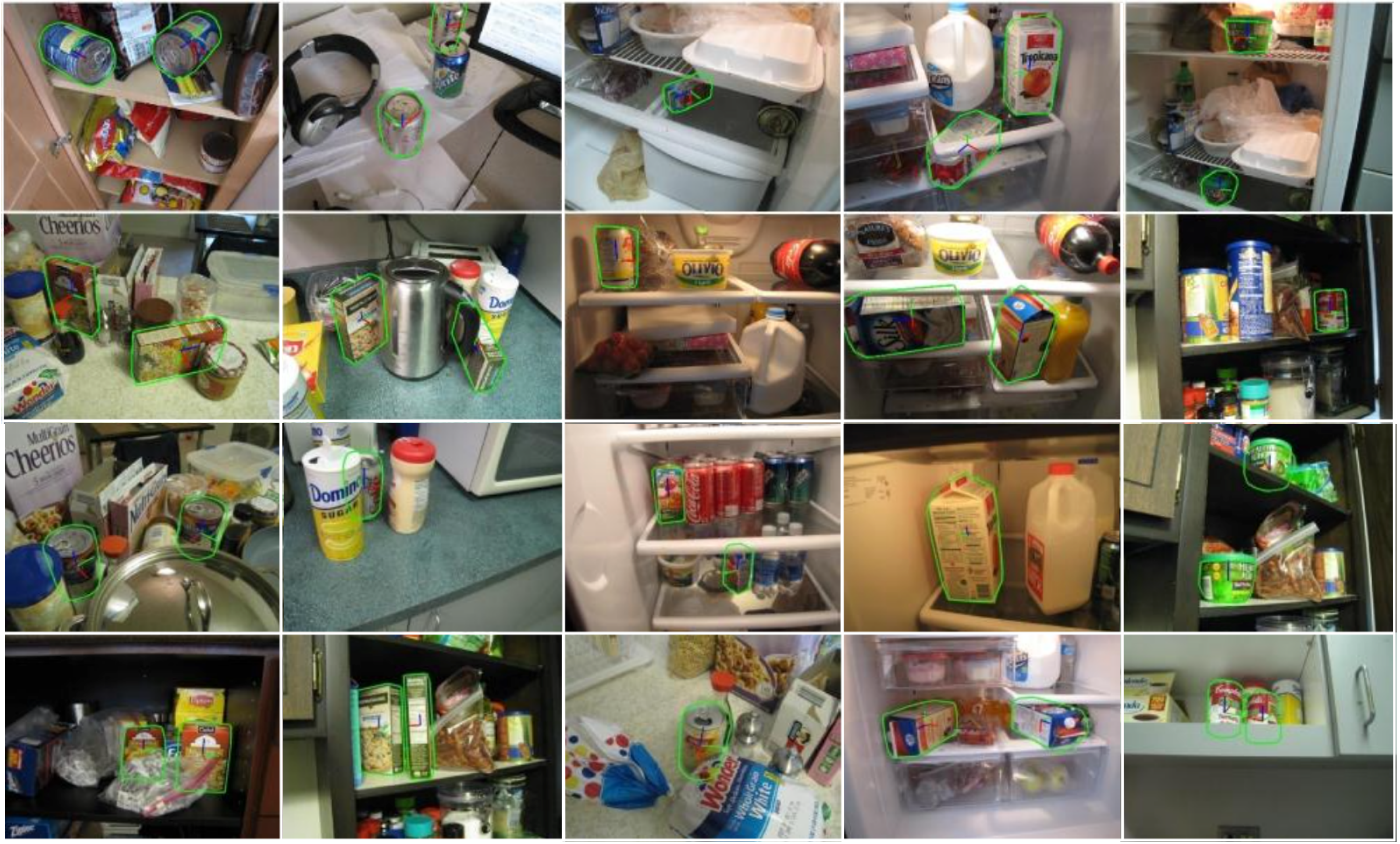
Example: For each SIFT Feature seek a match in the feature set

Example



Model: Limited set of views of the objects

Run-time input: Object in arbitrary pose and illumination conditions in uncontrolled environments


Cheerios
OLIVIO
Tropicana
MultiGrain Cheerios
Domino
SUGAR
Wonder
Whole Grain White

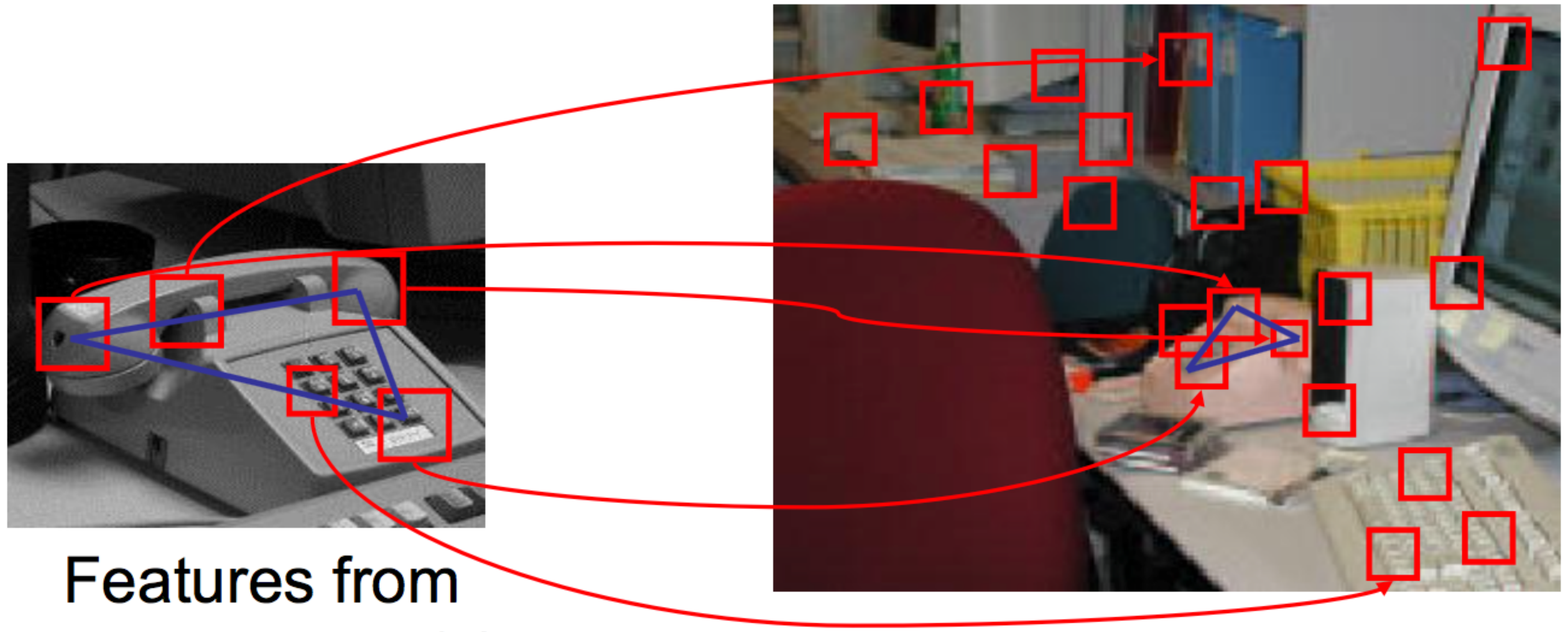
Feature Matching & Geometric Relations

Pros

- Relatively simple implementation
- Well-defined, efficient operations: Indexing, geometric verification (RANSAC), etc.

Cons

- Information reduced to a relatively small set of discrete features
- Generalization issues for broad categories How do I deal with recognizing the class of all the chairs?



Features from training image(s)

Features from input image

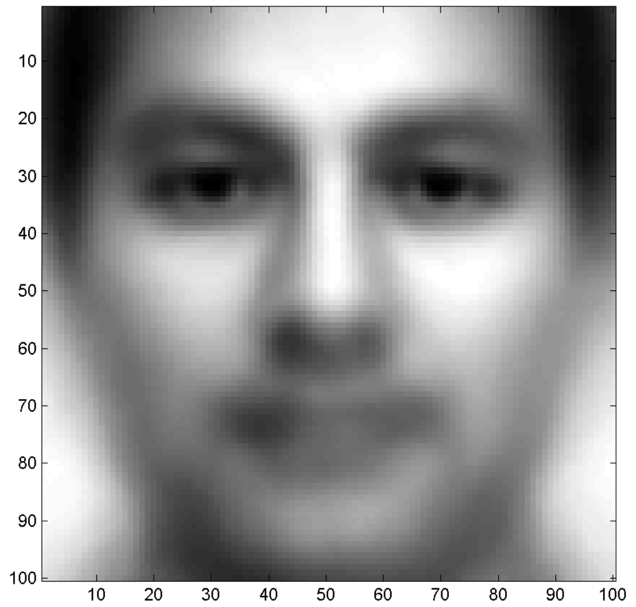
To Be Continued…

My personal favorite - Window Based Techniques :)

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

400×200
(RGB)

+



100×100

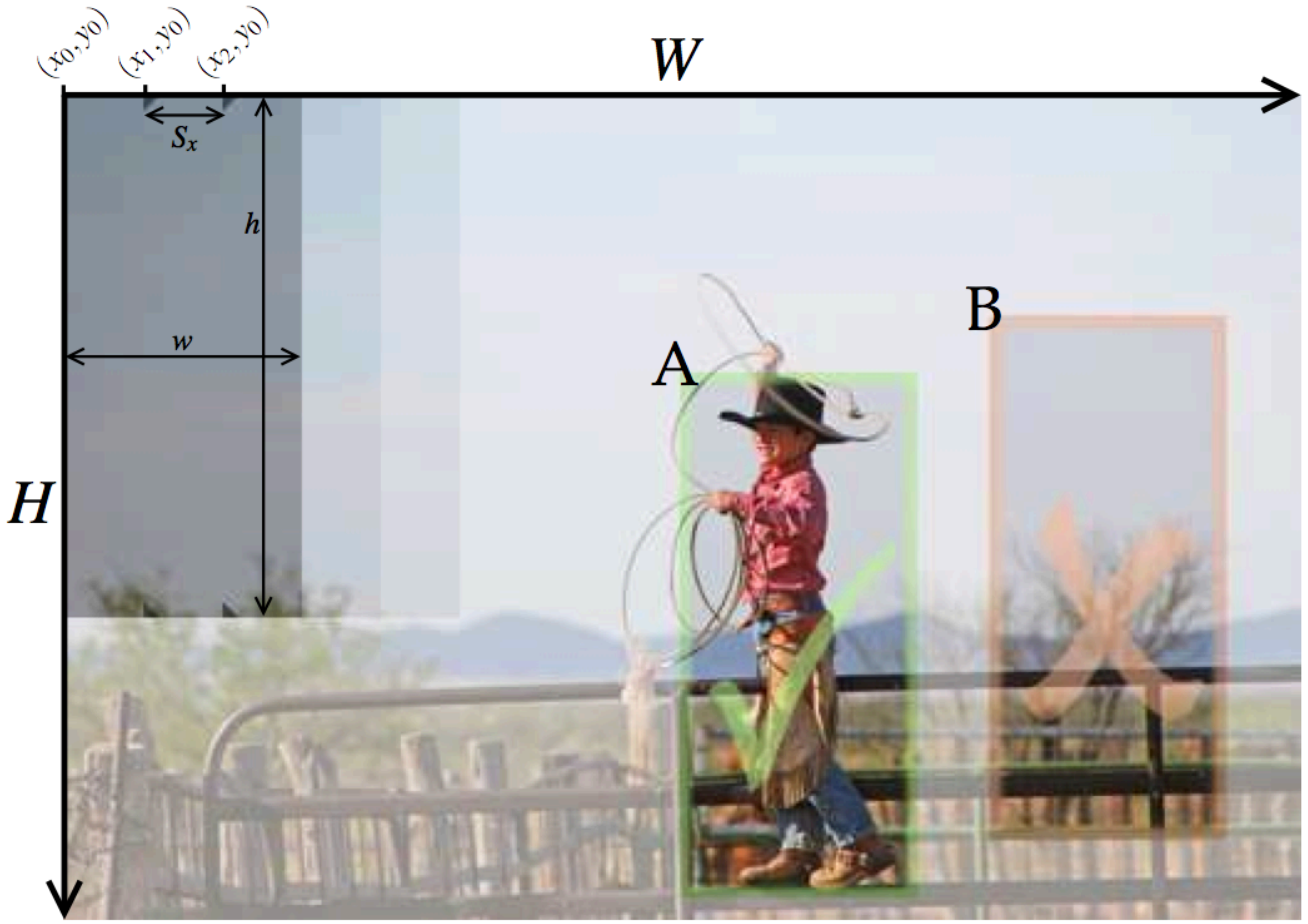
+

?

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

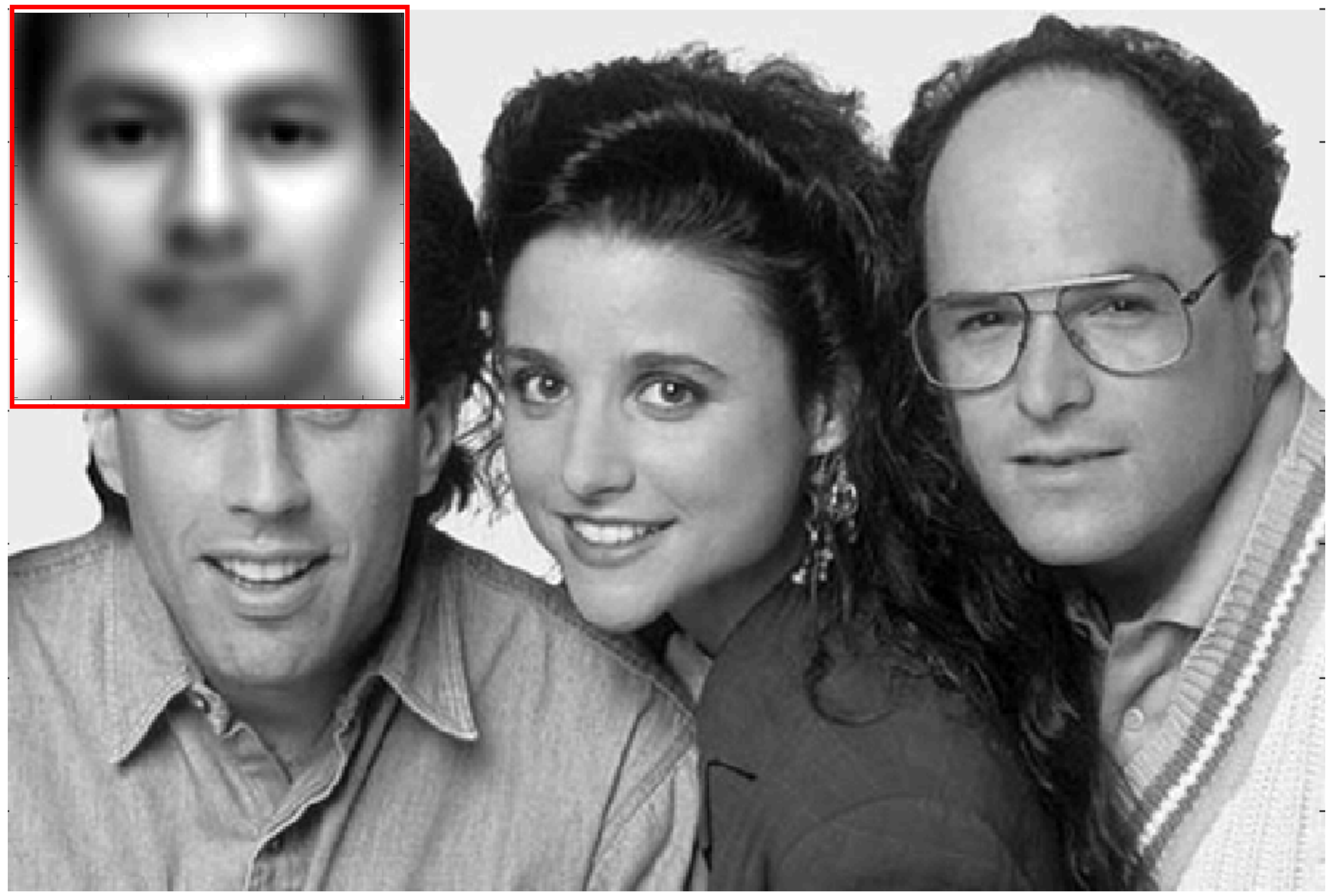
Sliding Windows



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

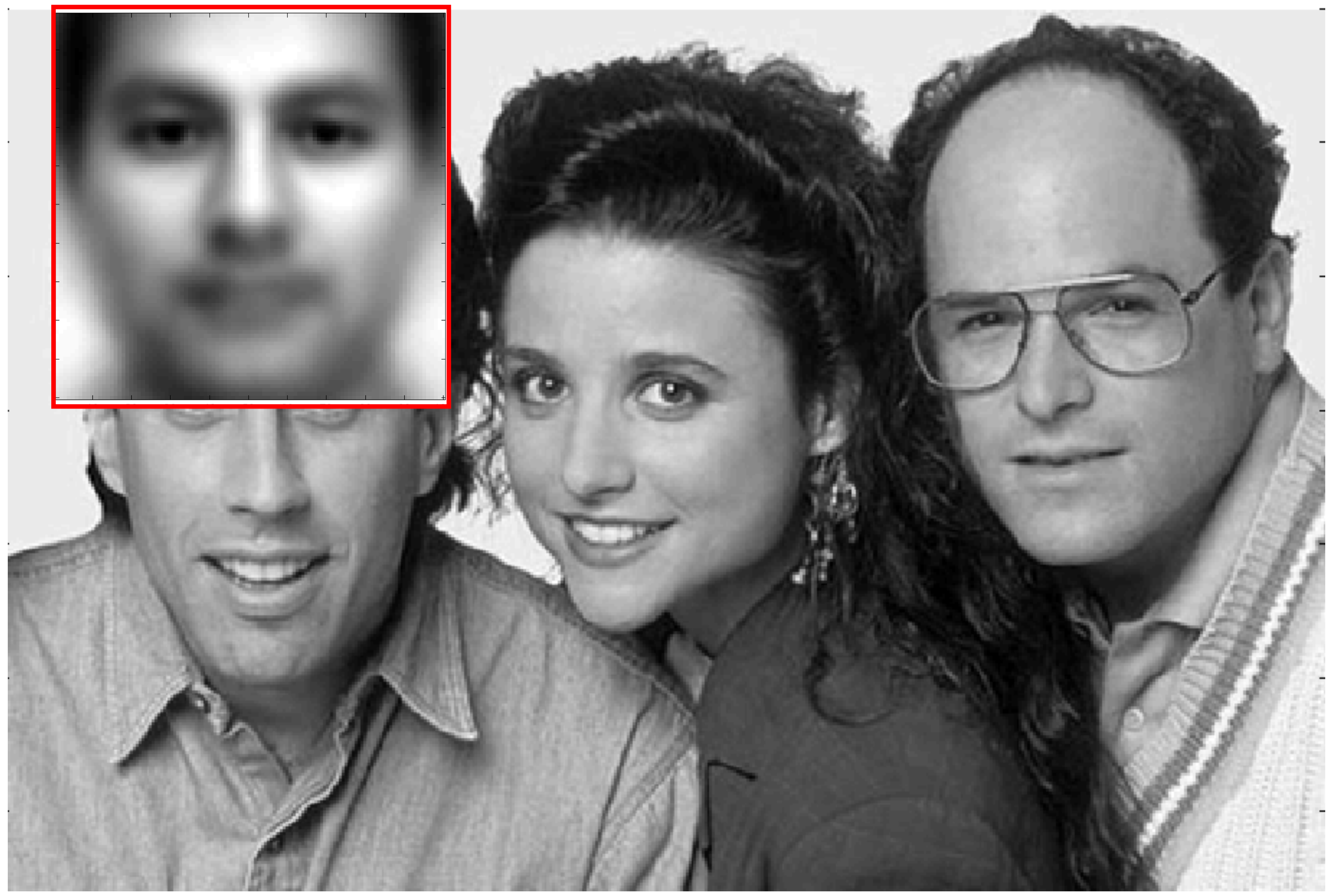
Figure 1.3: The Sliding Windows Methodology. In order to find an object instance, a detector must be run over multiple sub-regions of the image. Consequently the detector must use a minimal amount of processing for each individual window. To the right of the main image we show the basics of a sliding 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



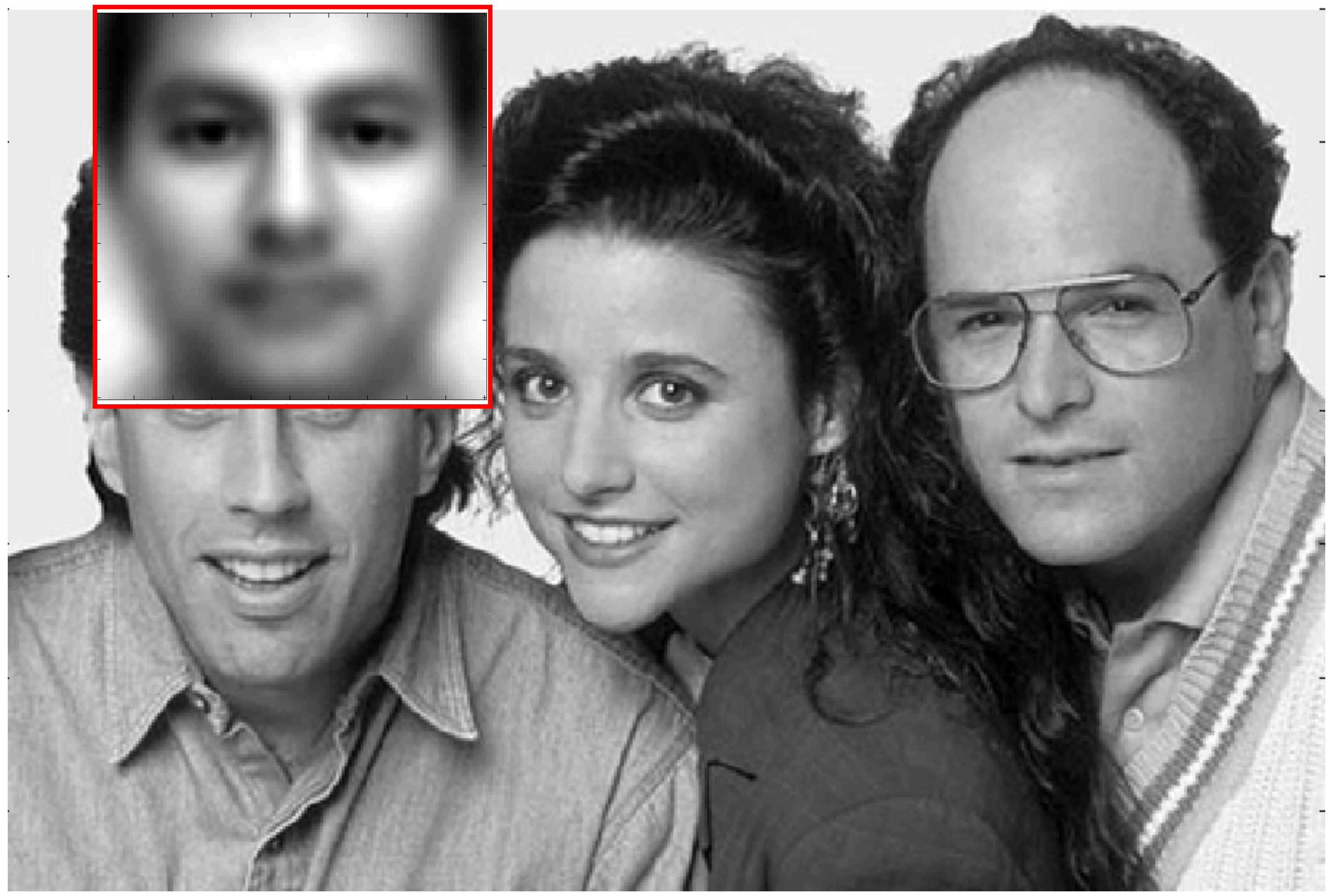
- Try to "match" the face template at each location in the larger picture

Finding Faces in an Image



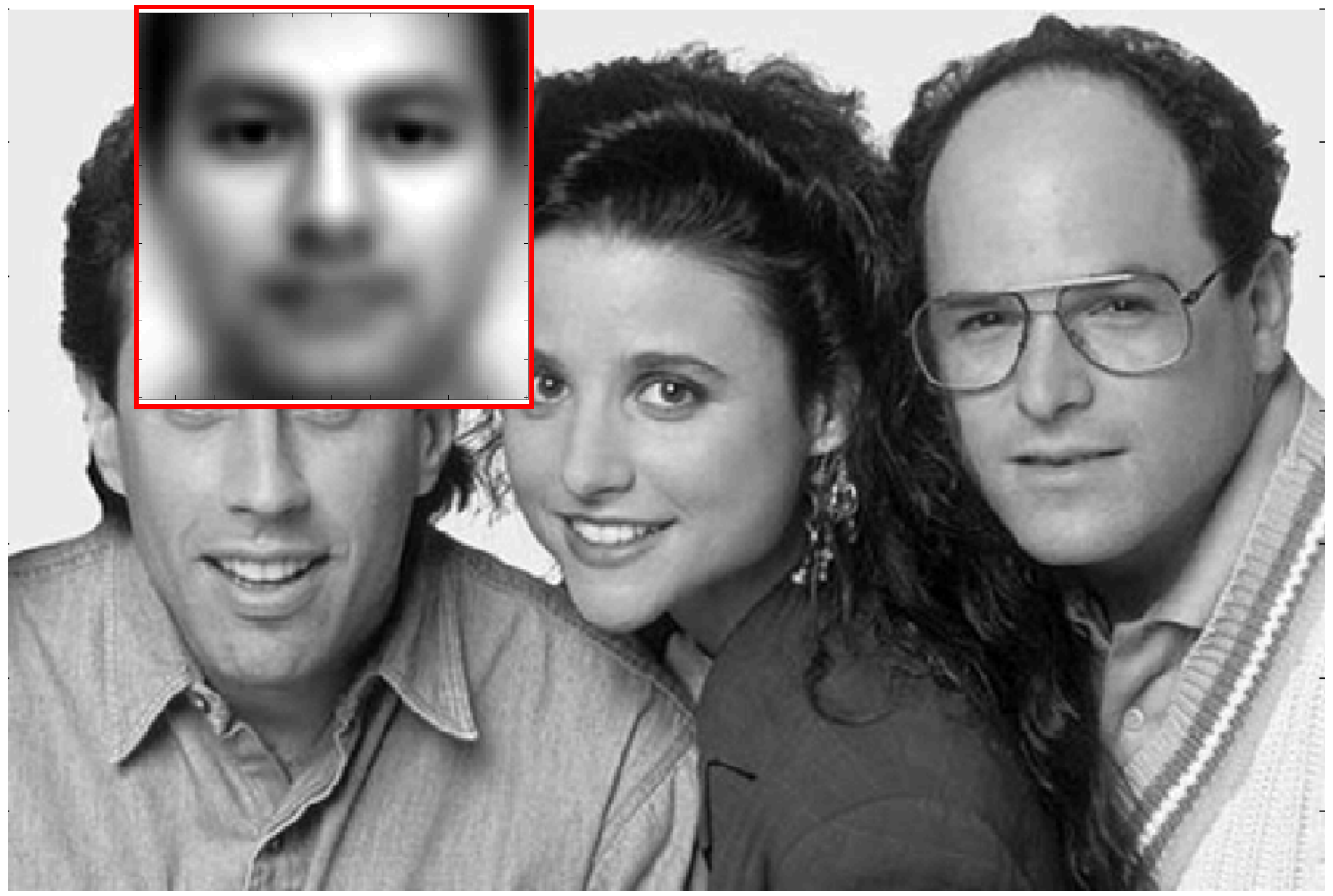
- Try to “match” the face template at each location in the larger picture

Finding Faces in an Image



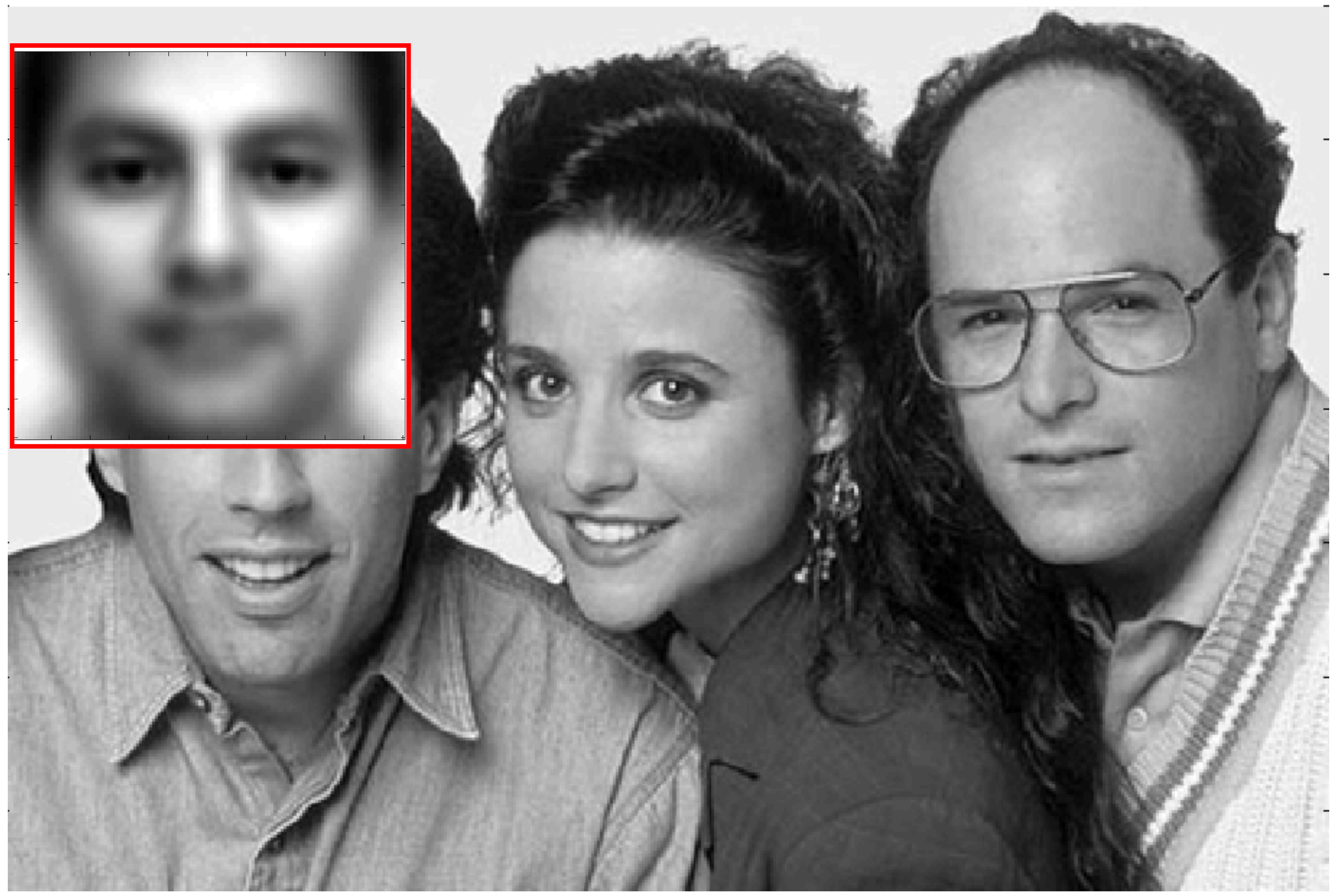
- Try to “match” the face template at each location in the larger picture

Finding Faces in an Image



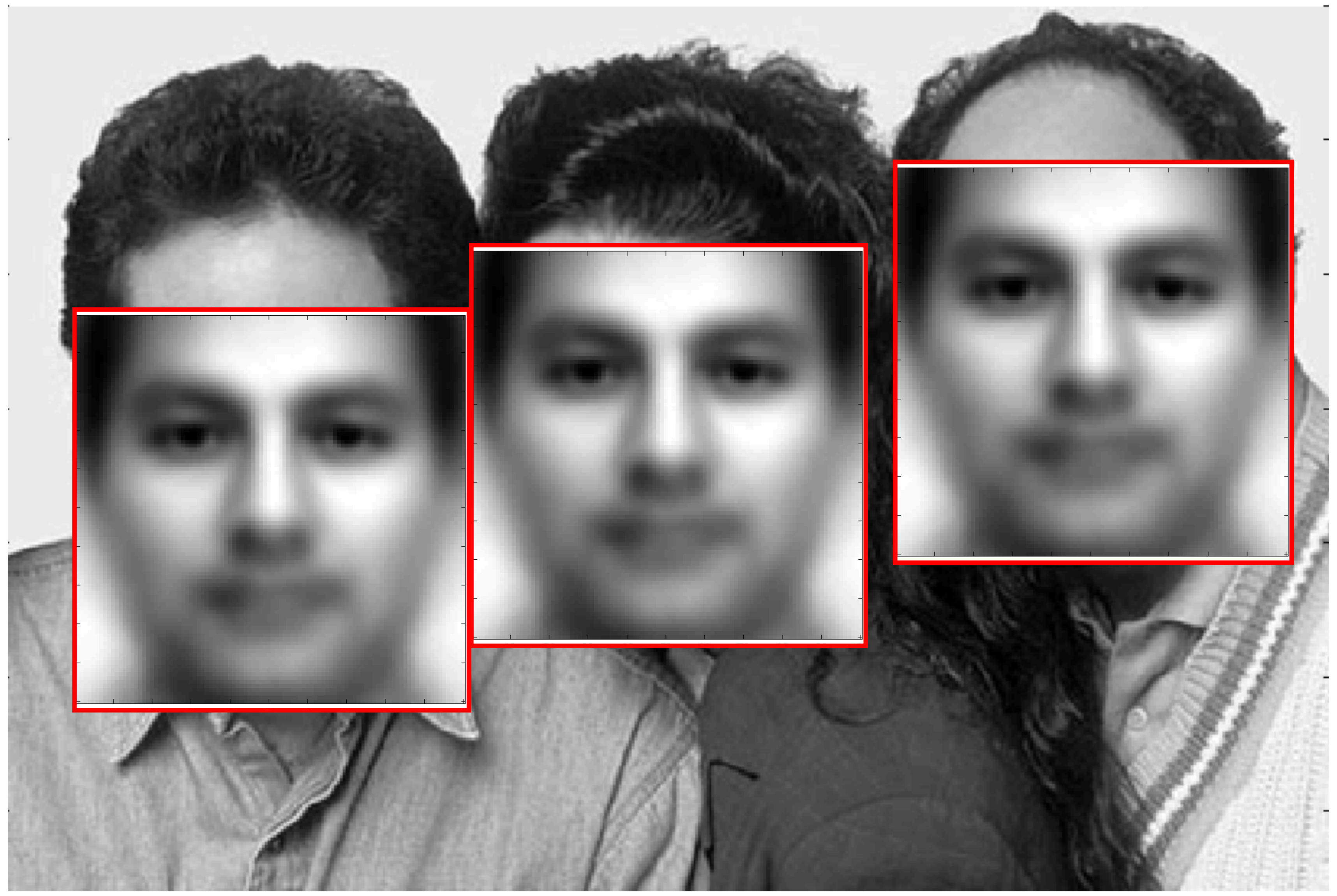
- Try to “match” the face template at each location in the larger picture

Finding Faces in an Image



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

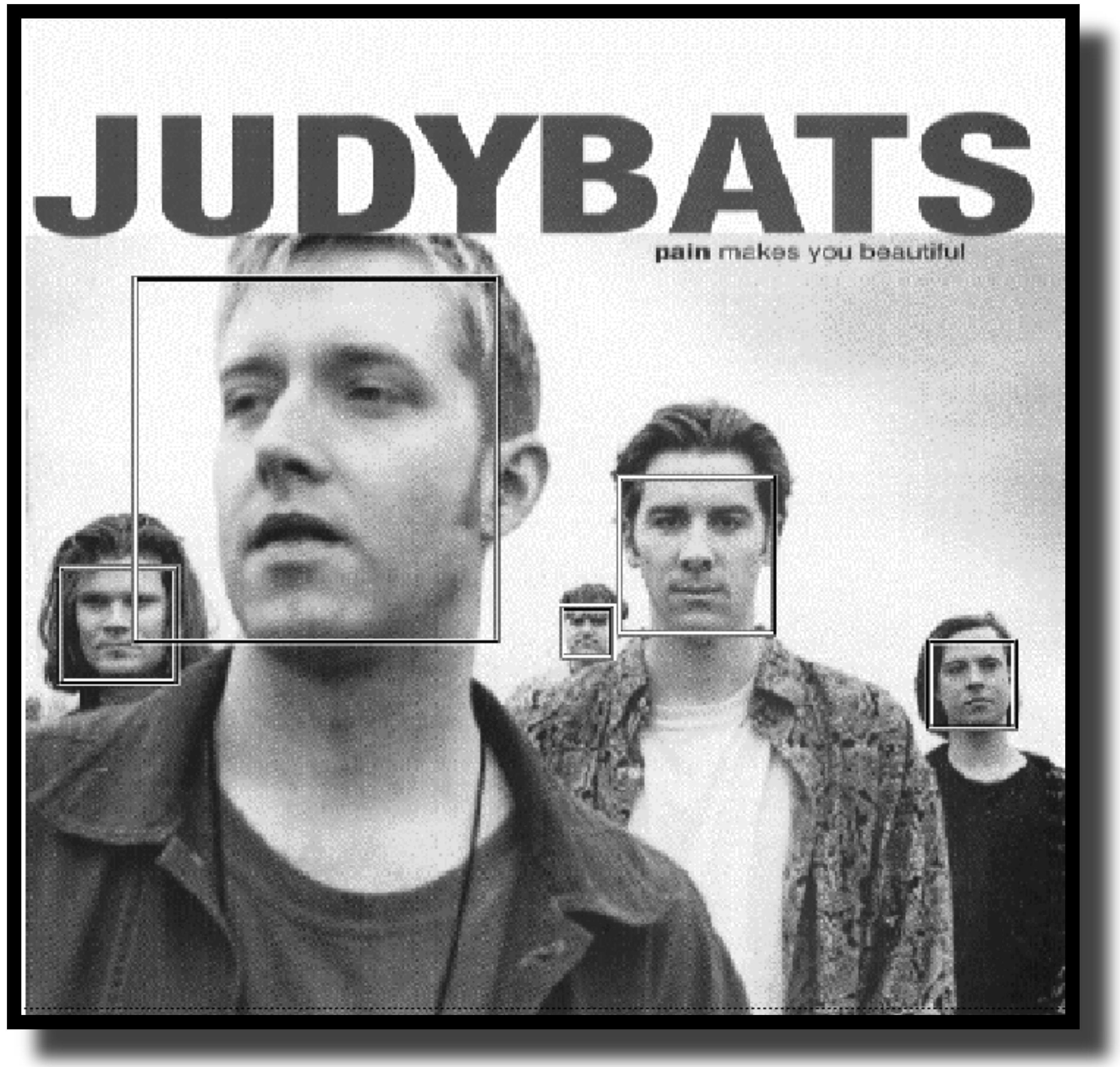
Finding Faces in an Image



- 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



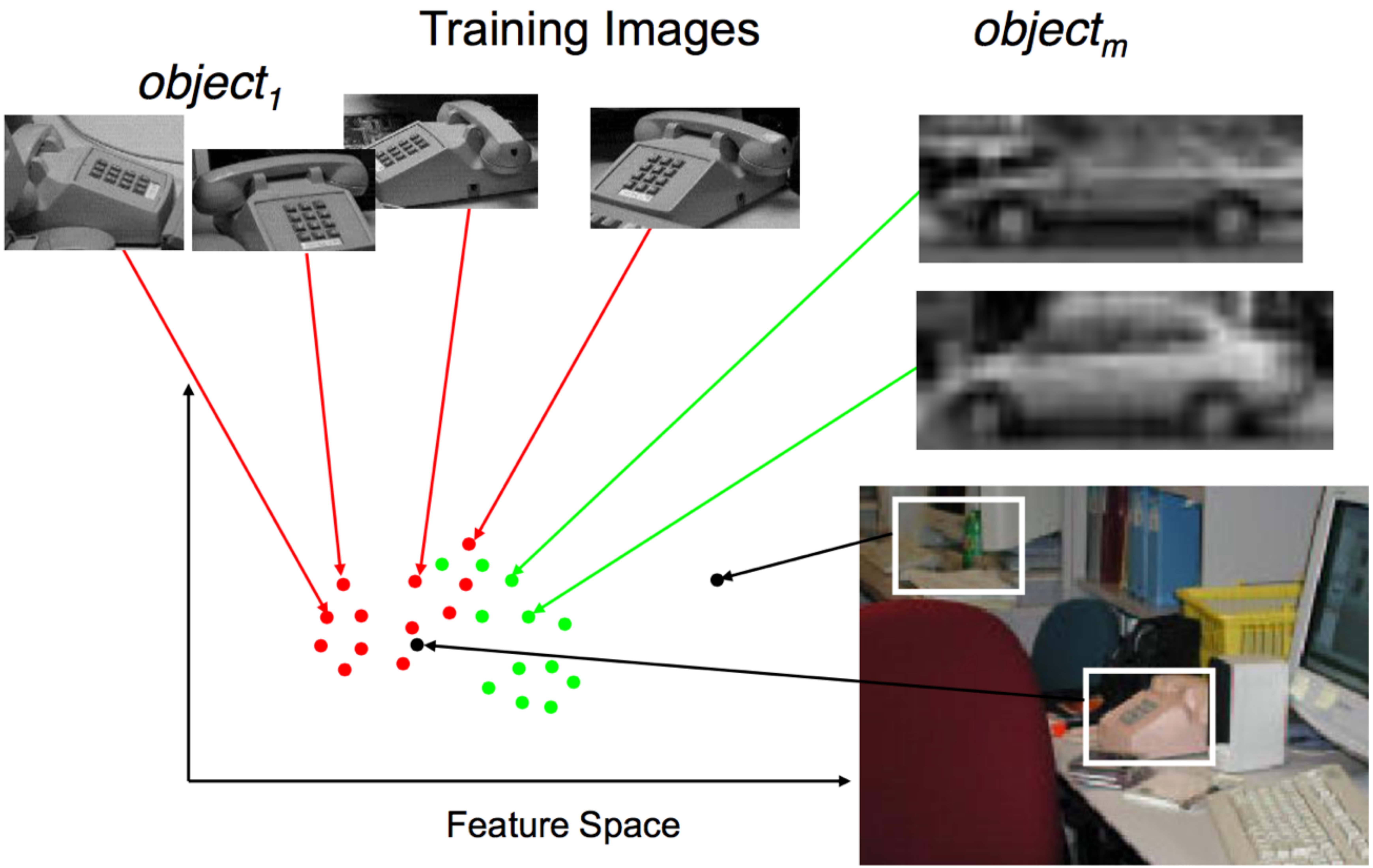
Scale-Space Pyramid

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

How to Evaluate the Windows?



What representation should we use?

Template Classification

- Support Vector Machines - SVMs
- Combination of simple classifiers (boosting)
- Neural networks, deep learning