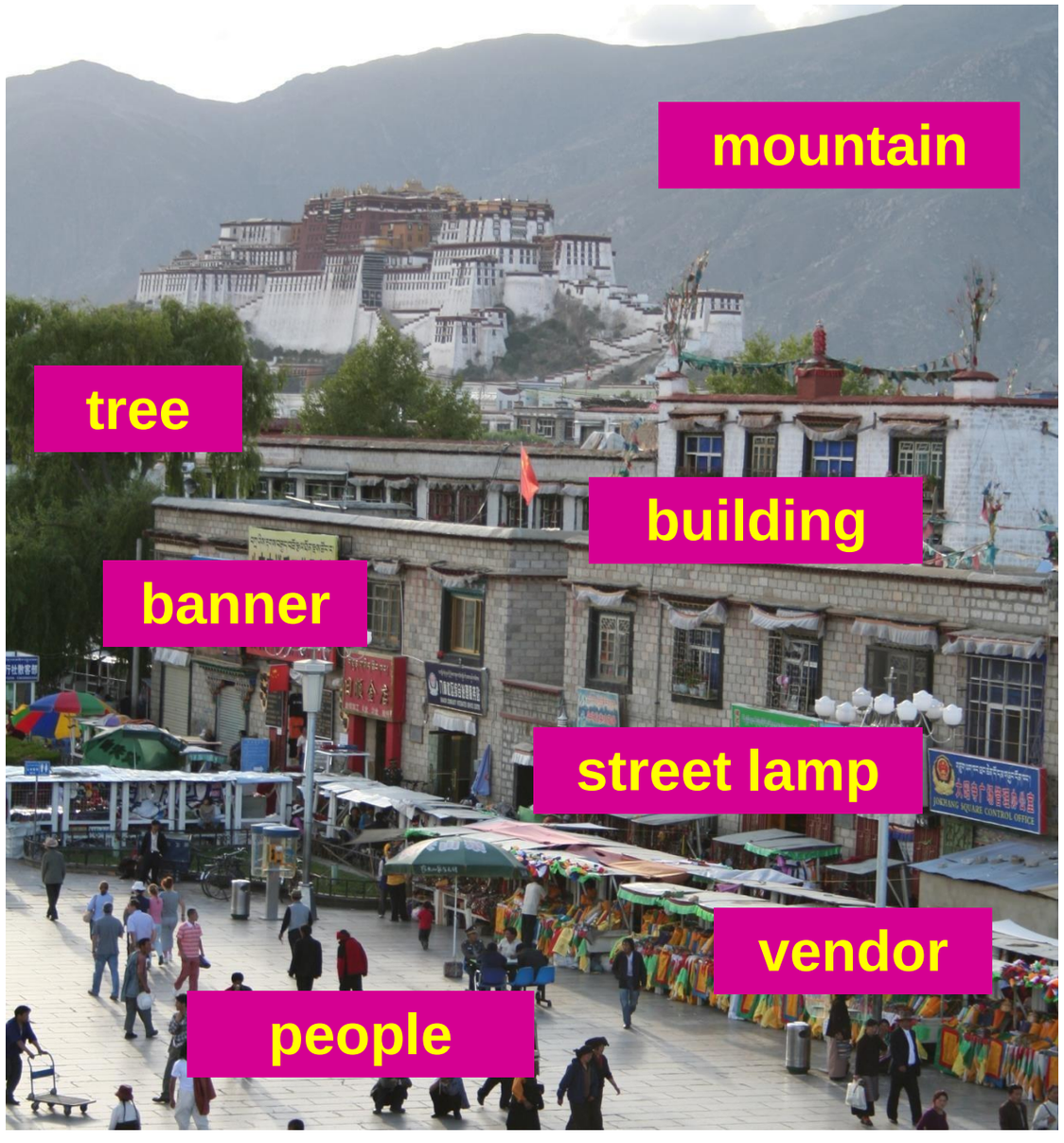
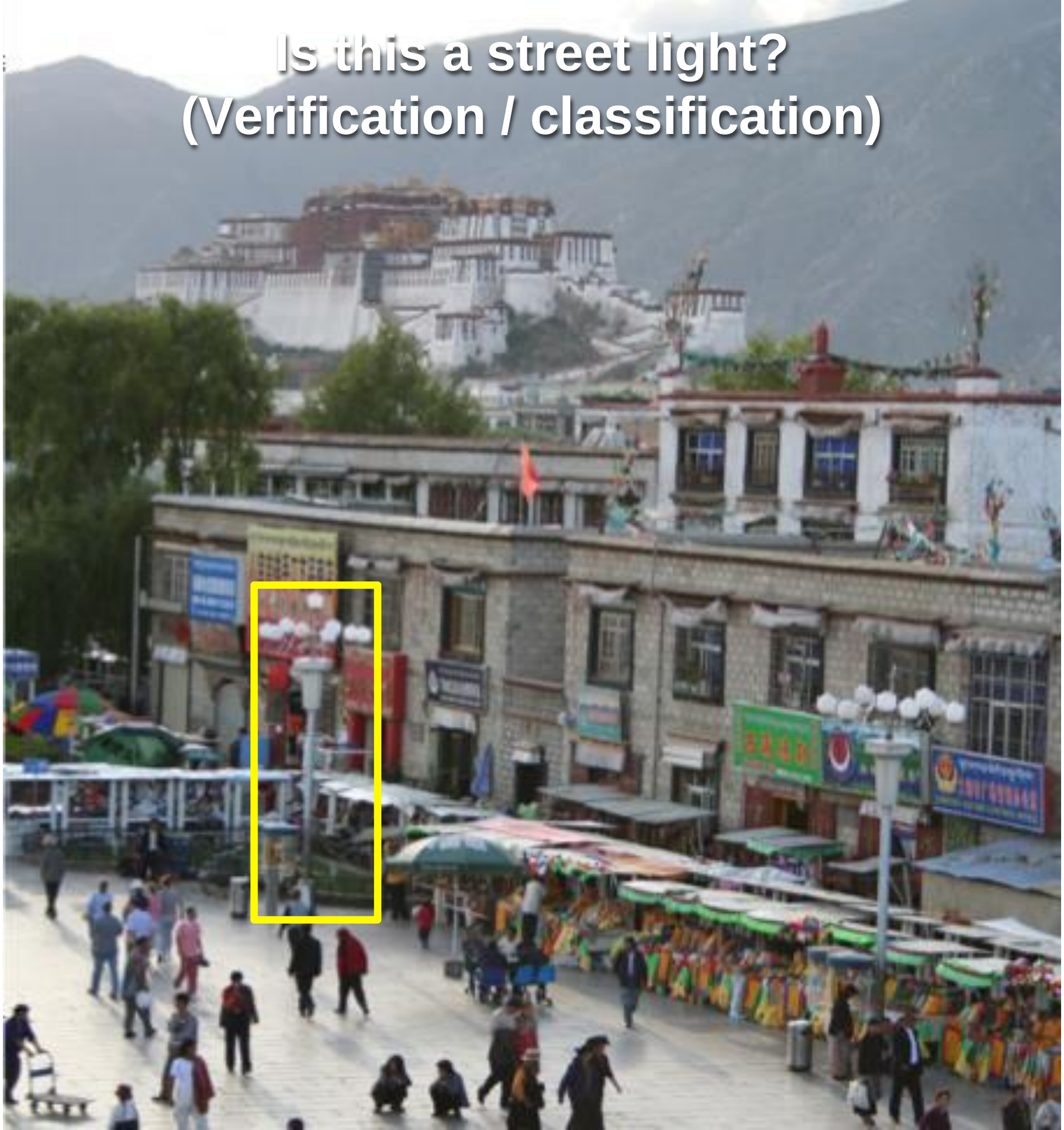
Introduction to object recognition



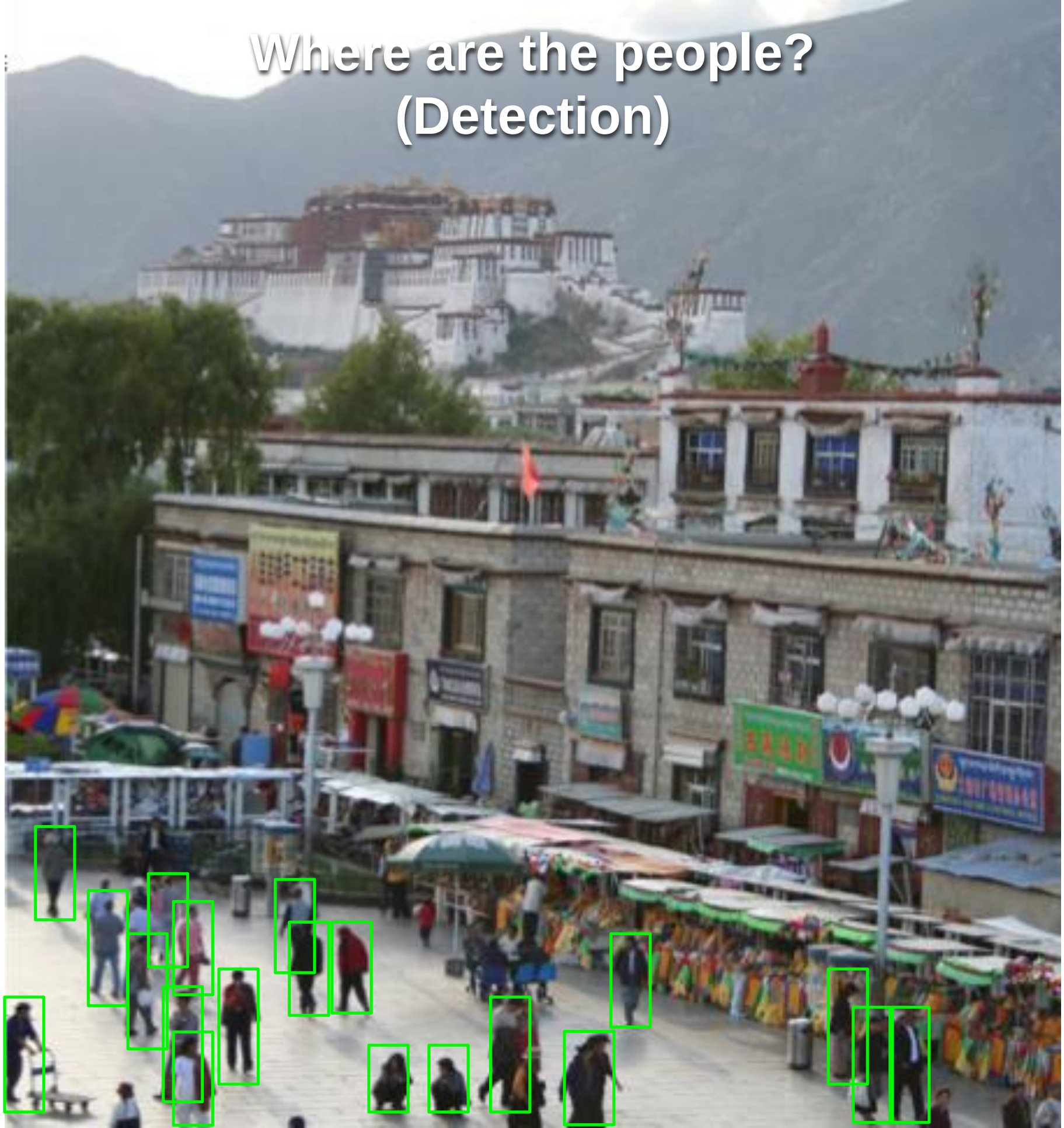
http://www.cs.cmu.edu/~16385/

What do we mean by ‘object recognition’?

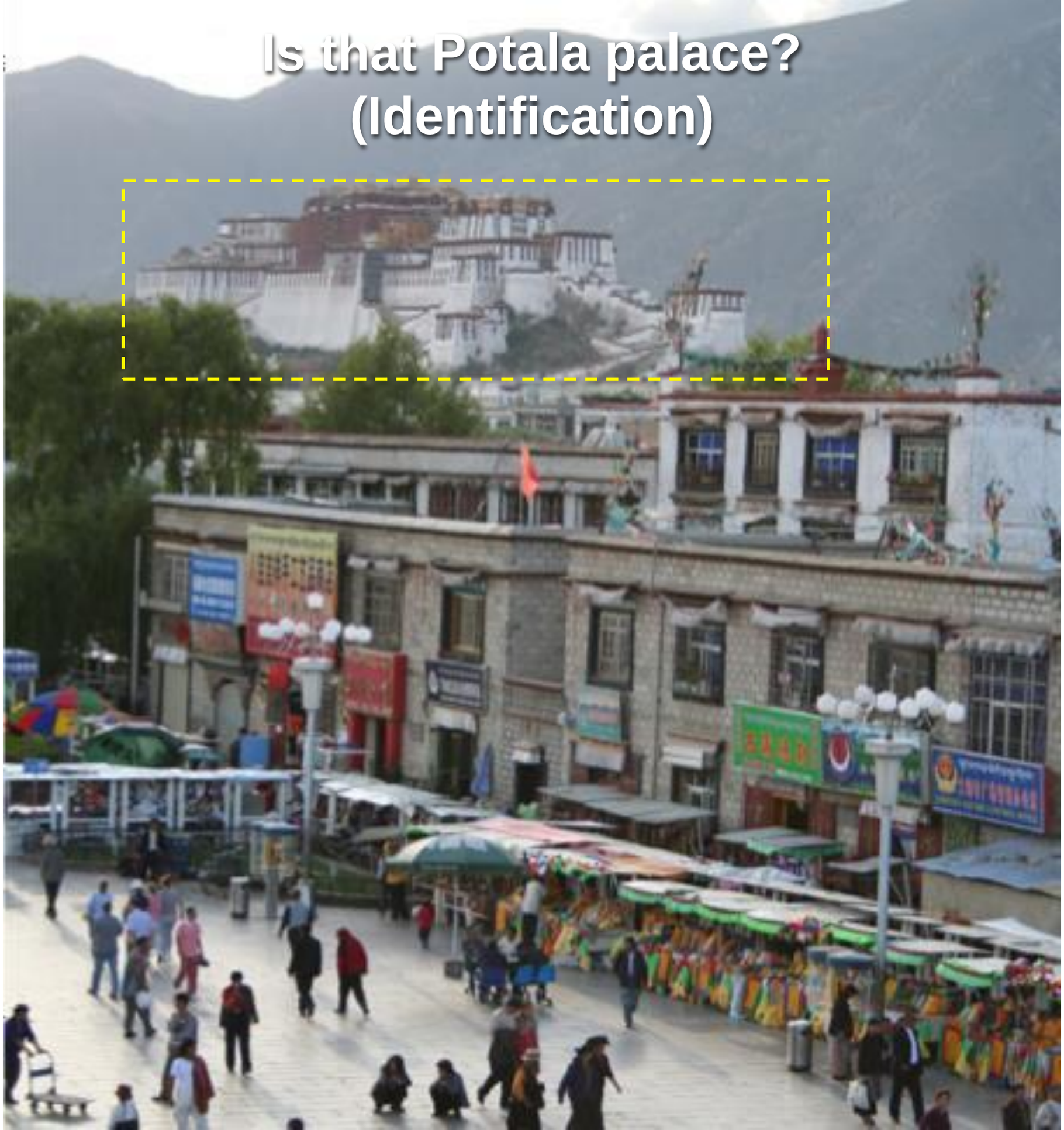
Is this a street light?
(Verification / classification)



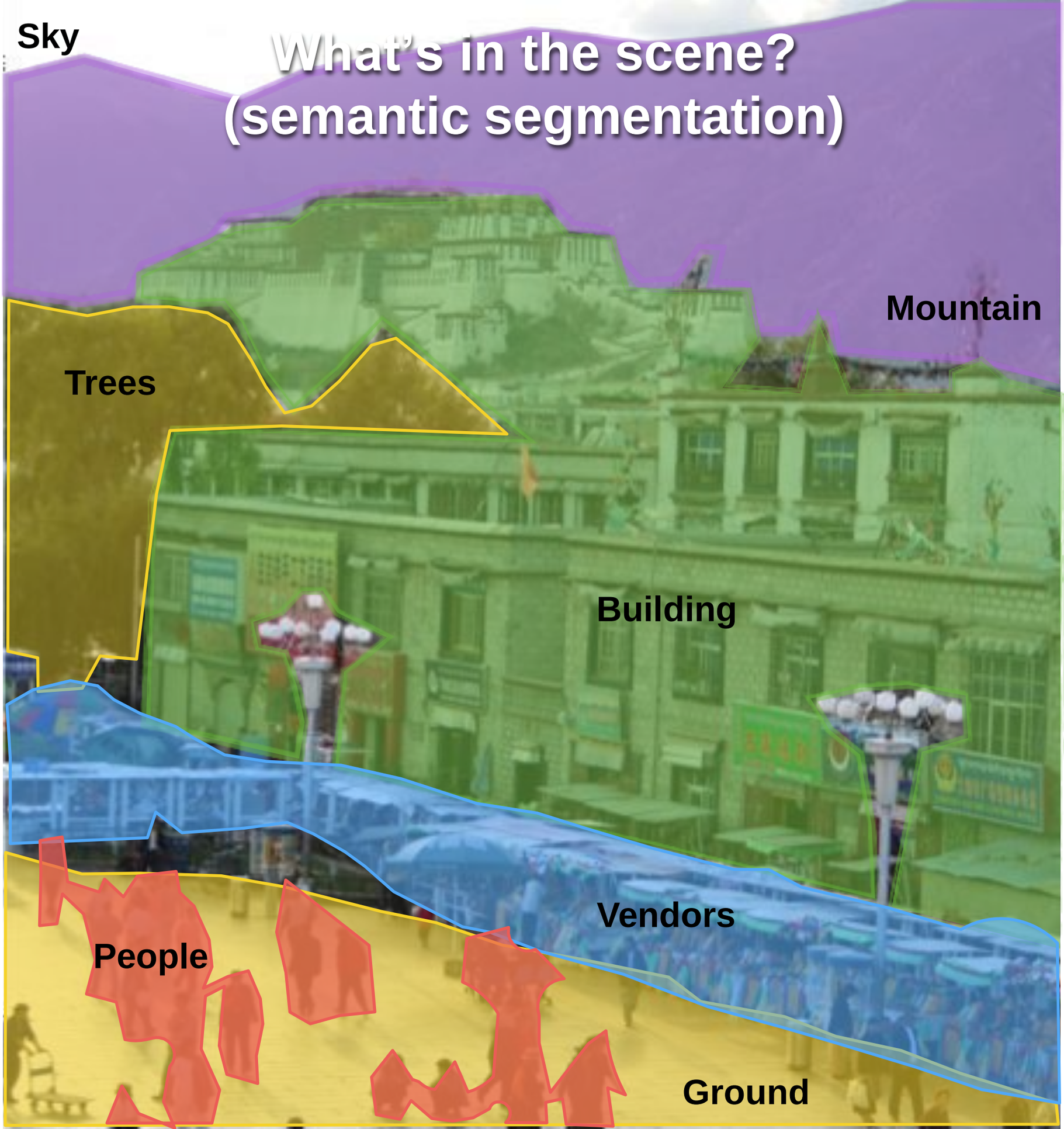
Where are the people? (Detection)



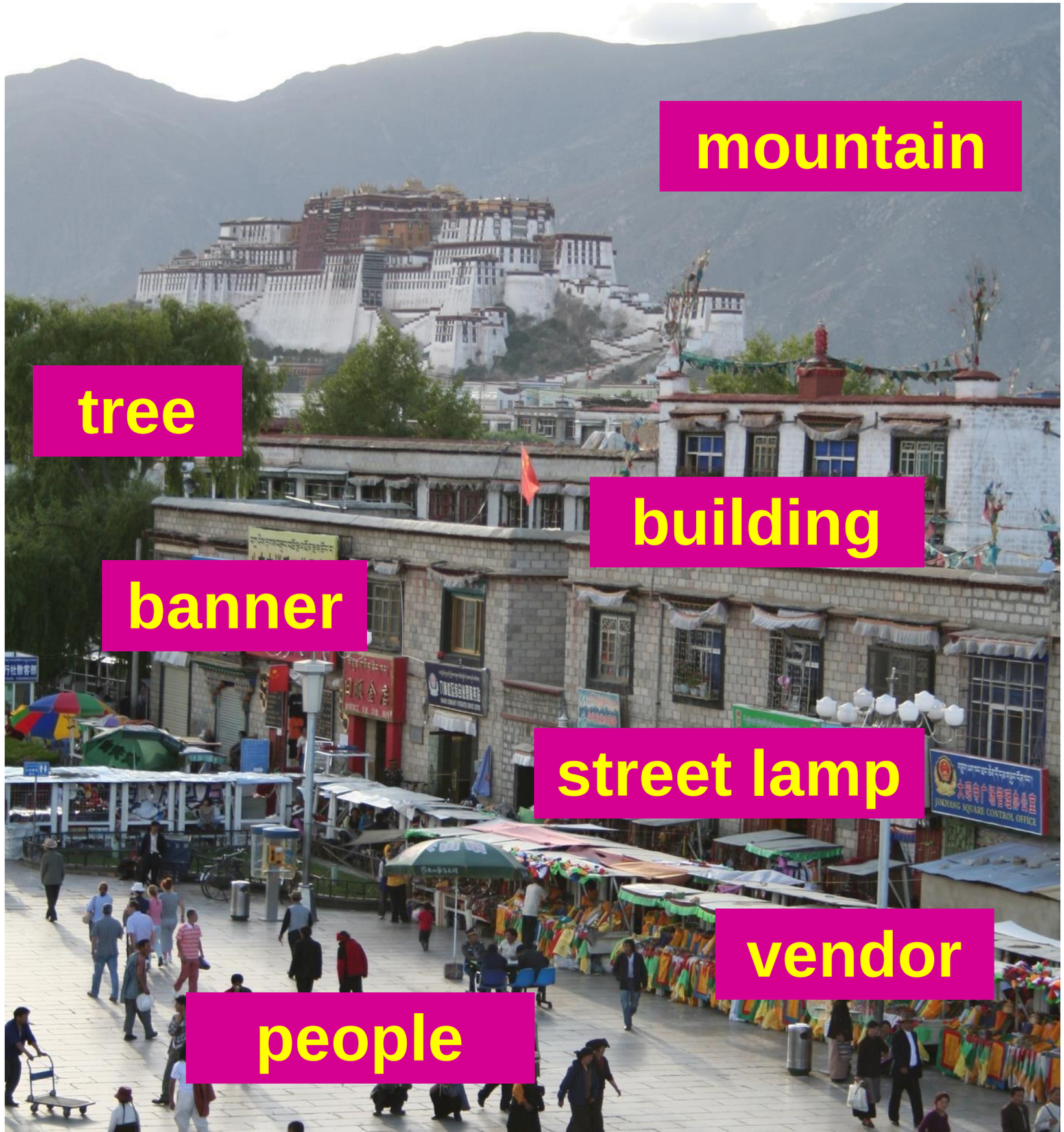
Is that Potala palace? (Identification)



What's in the scene? (semantic segmentation)



Object categorization



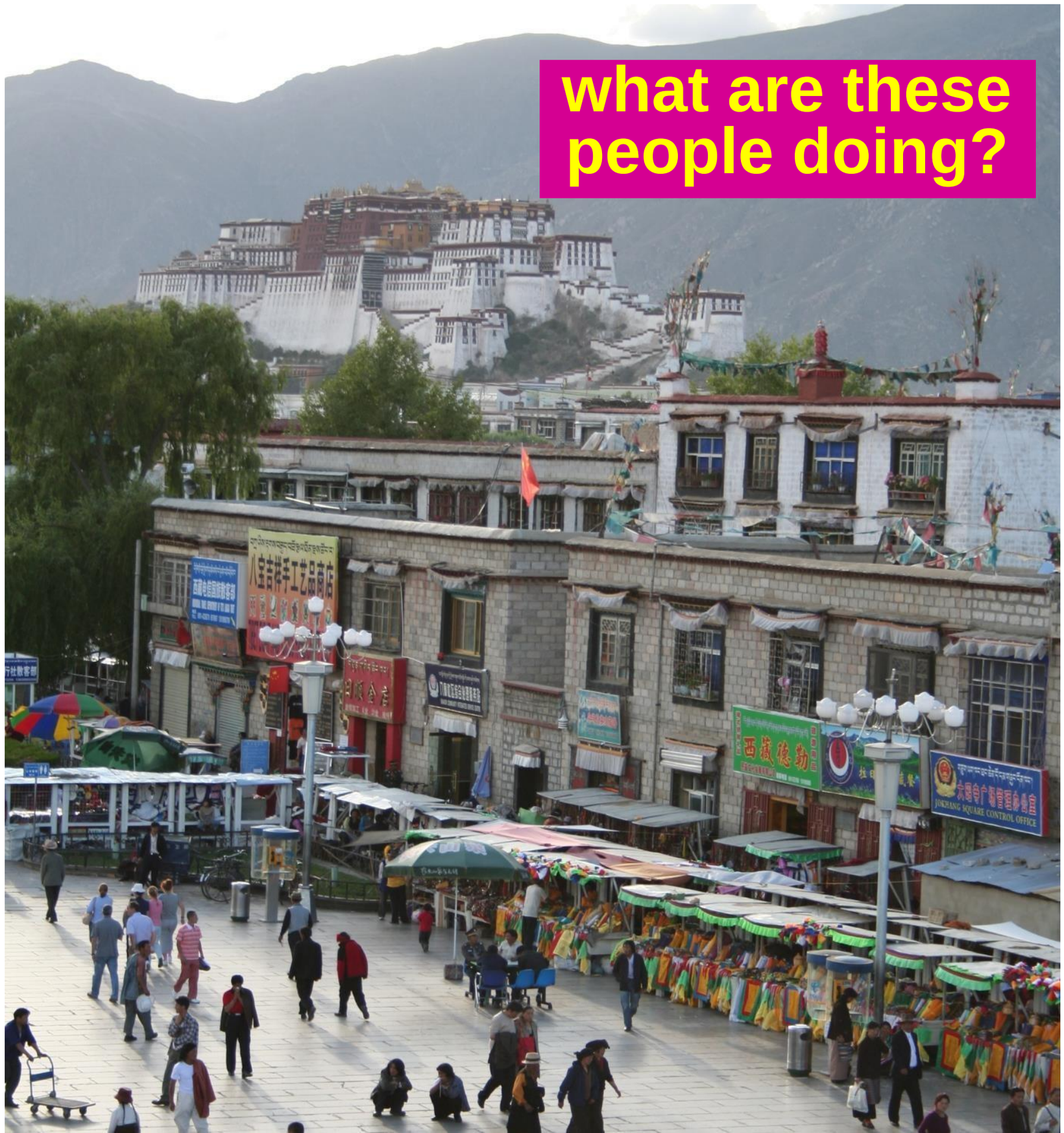
What type of scene is it?
(Scene categorization)

Outdoor

Marketplace

City

Activity / Event Recognition

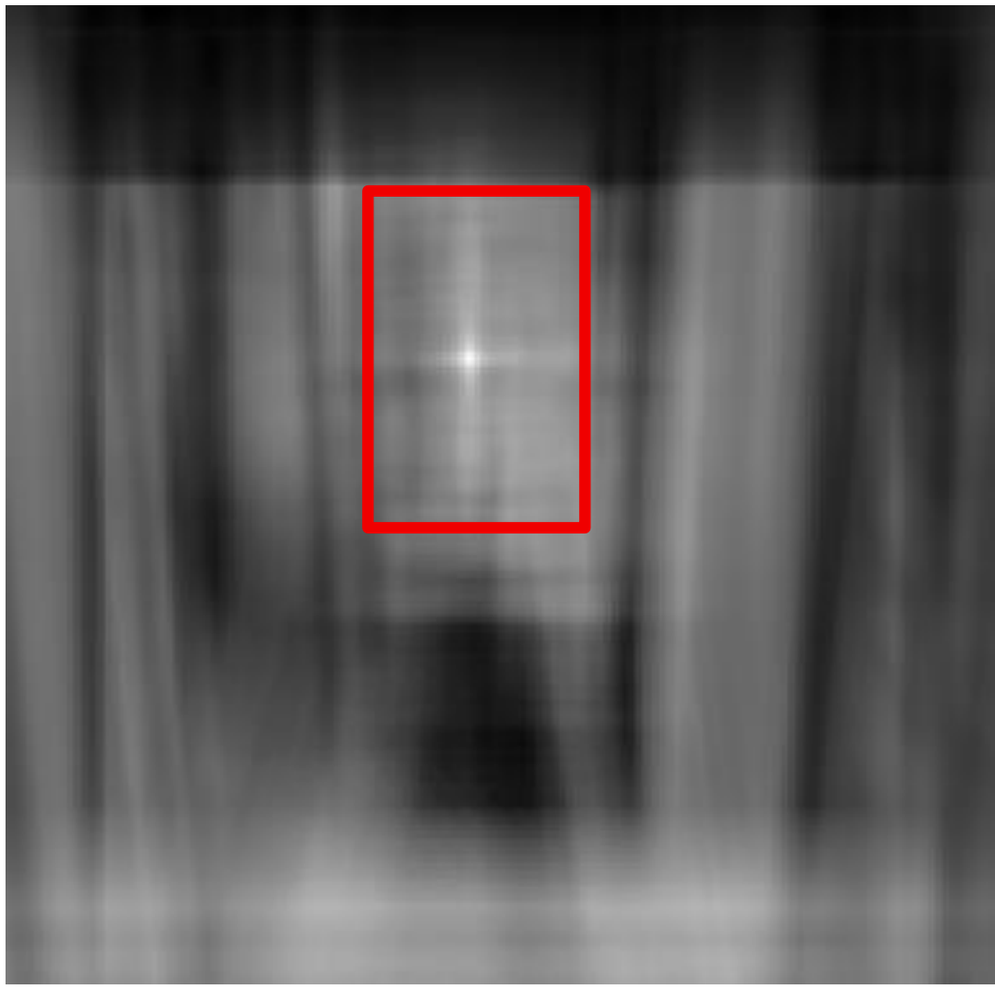


Object recognition
Is it really so hard?

This is a chair

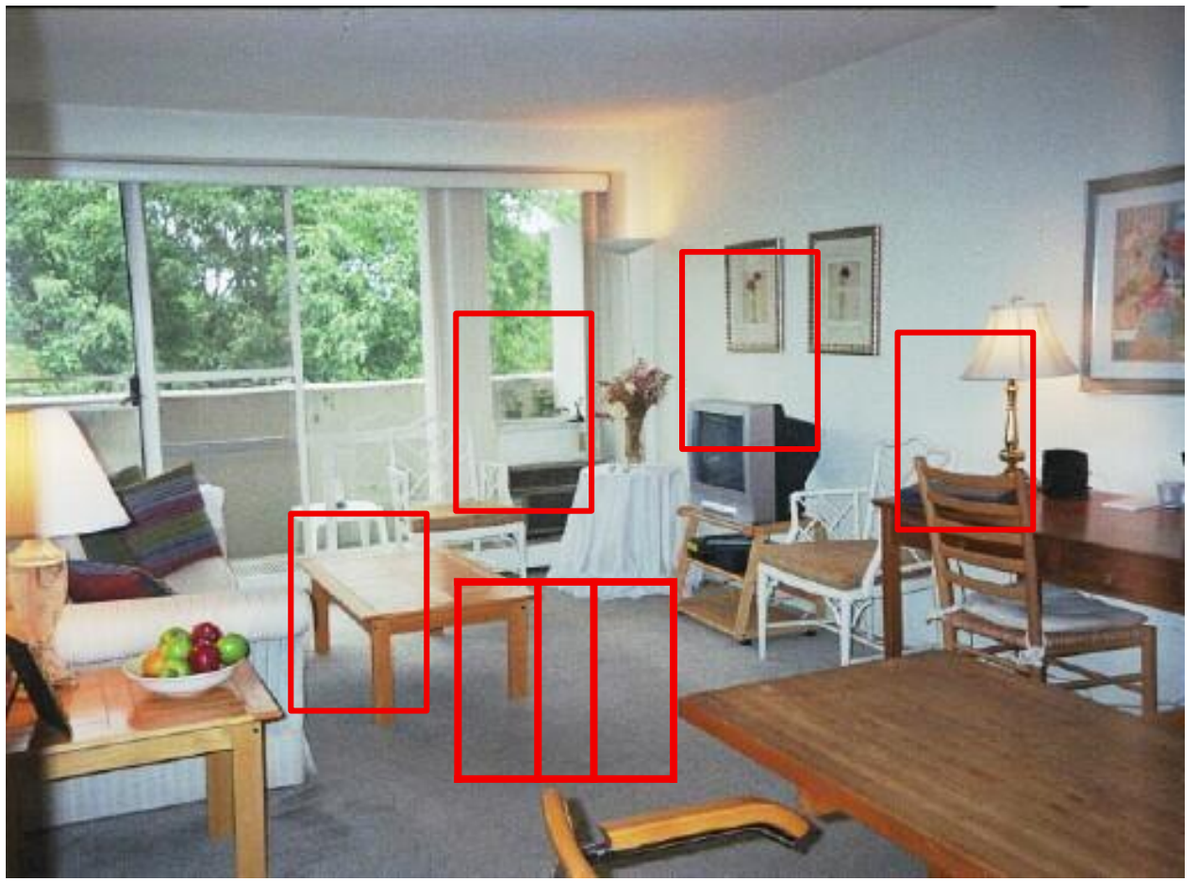
Find the chair in this image

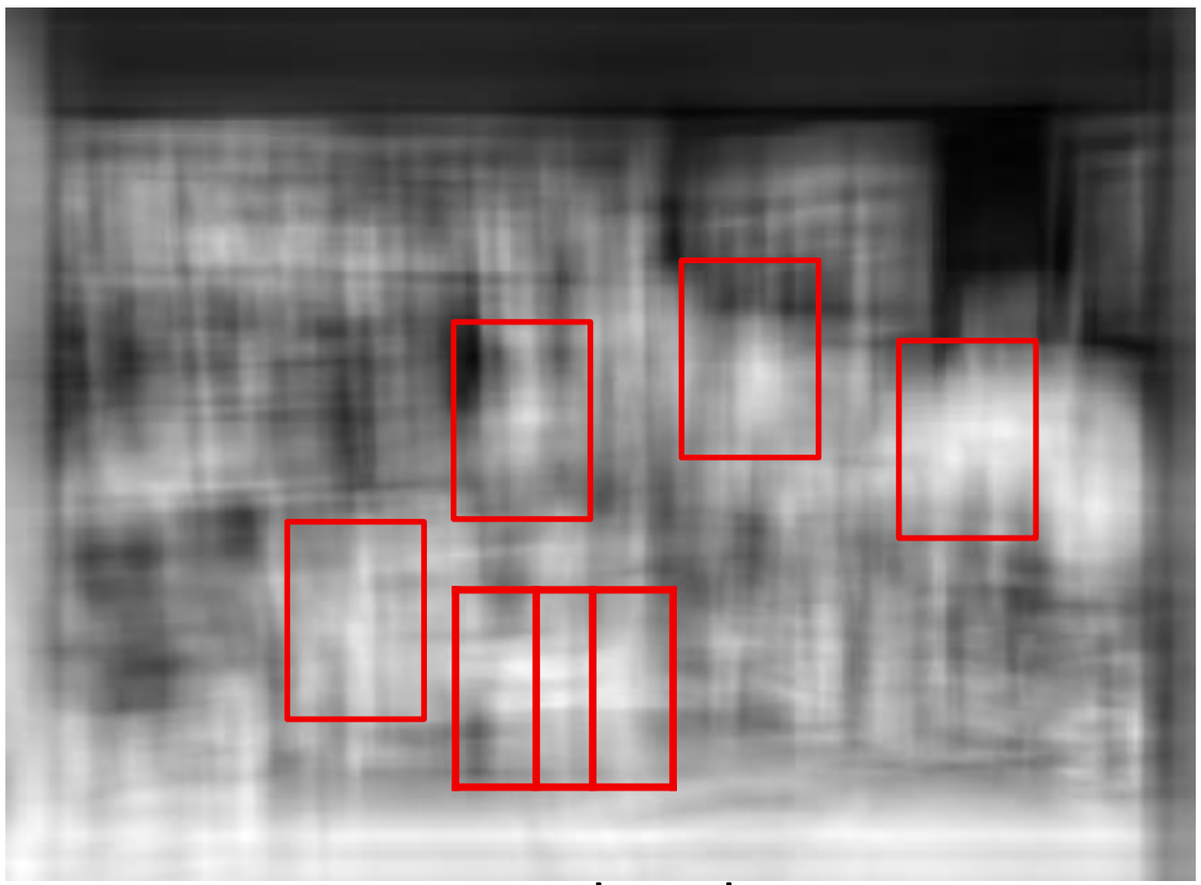
Output of normalized correlation



Object recognition
Is it really so hard?

Find the chair in this image





Pretty much garbage

Simple template matching is not going to make it

A “popular method is that of template matching, by point to point correlation of a model pattern with the image pattern. These techniques are inadequate for three-dimensional scene analysis for many reasons, such as occlusion, changes in viewing angle, and articulation of parts.” Nivatia & Binford, 1977.

And it can get a lot harder

Brady, M. J., & Kersten, D. (2003). Bootstrapped learning of novel objects. J Vis, 3(6), 413-422

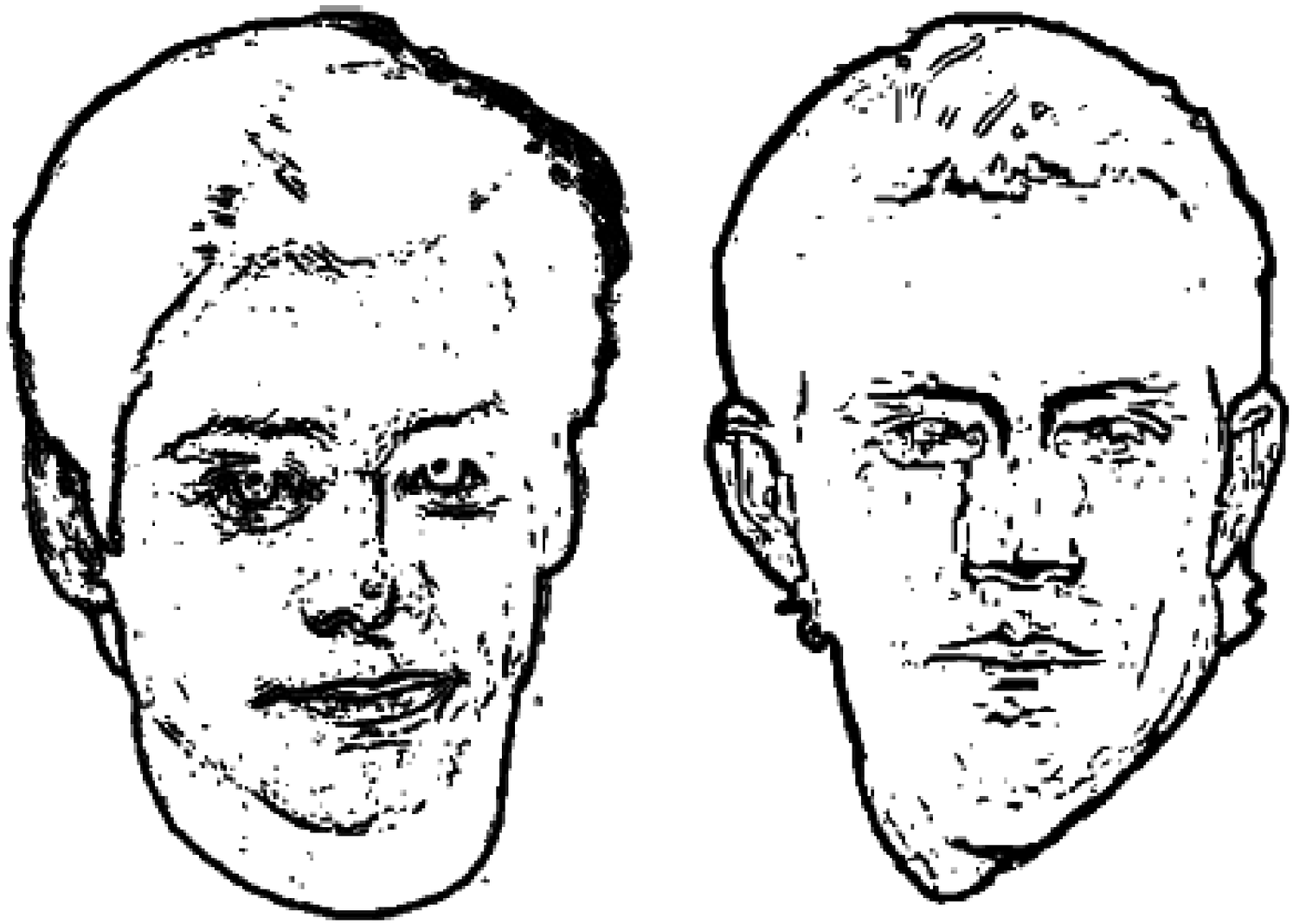
How do humans do recognition?

- We don’t completely know yet
- But we have some experimental observations.

Observation 1

- We can recognize familiar faces even in low-resolution images

Observation 2:

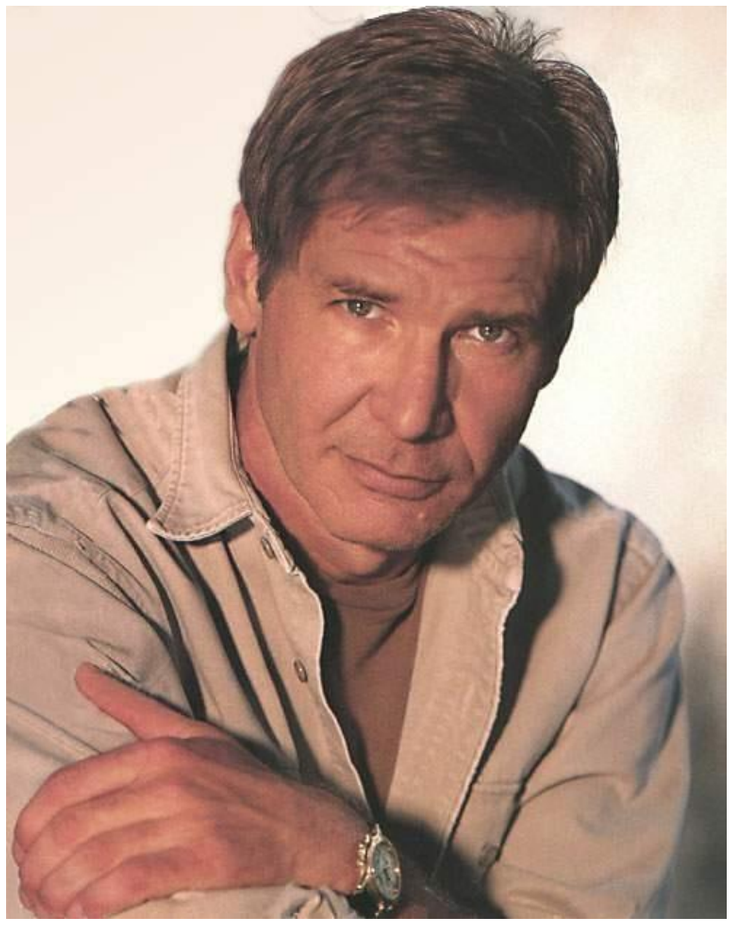


Jim Carrey

Kevin Costner

- High frequency information is not enough

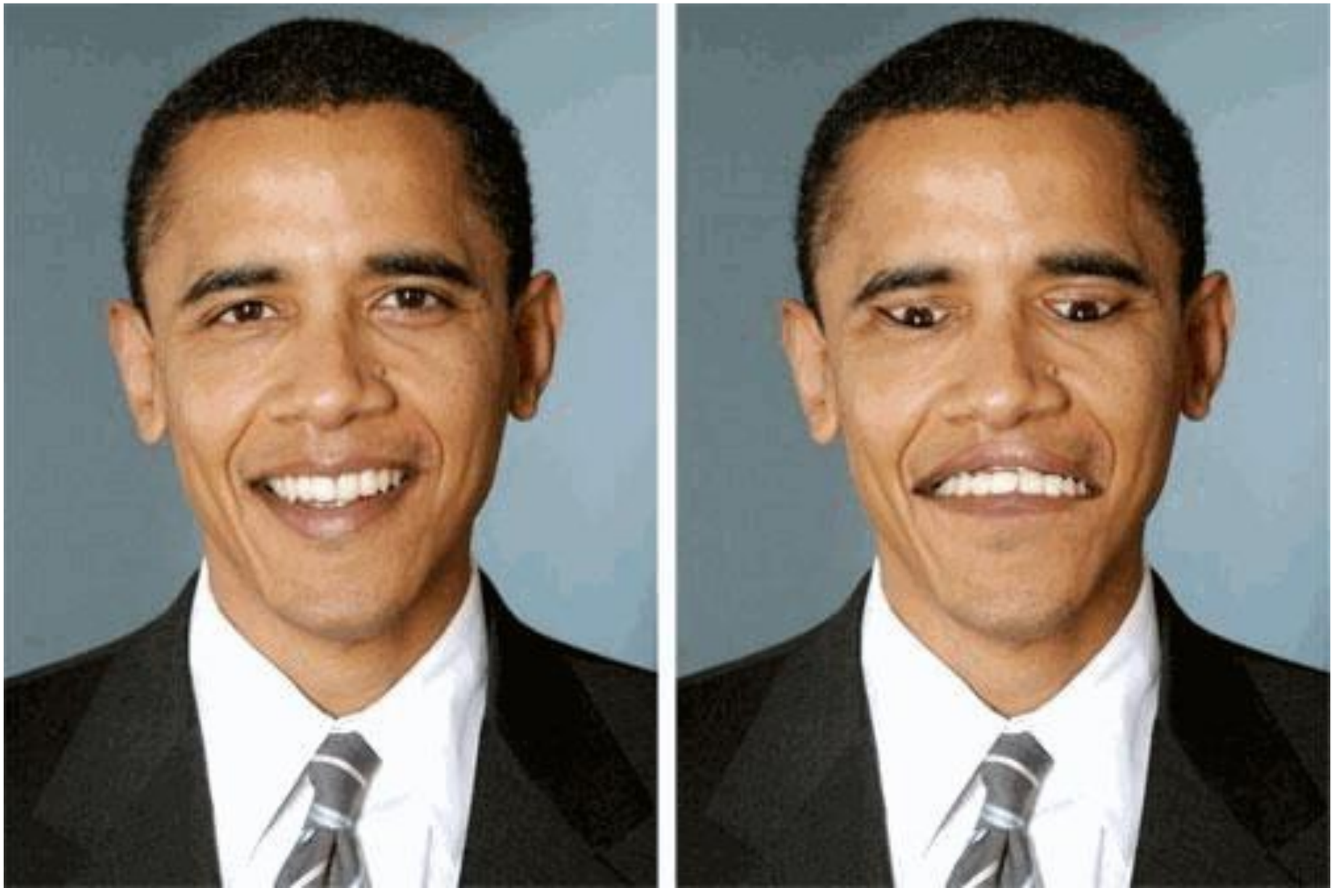
What is the single most important facial features for recognition?



What is the single most important facial features for recognition?

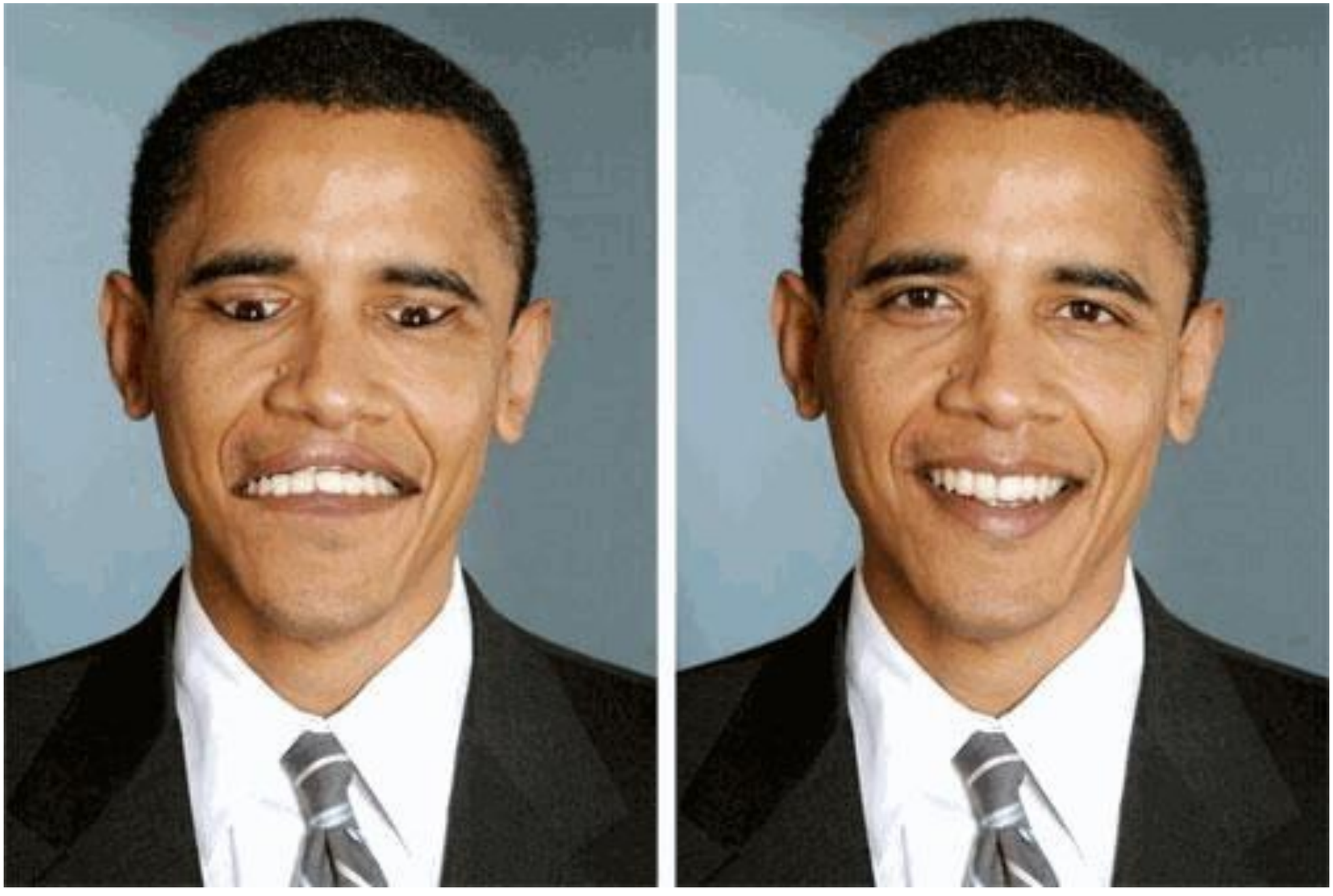
Observation 4:

- Image Warping is OK



Spatial configuration matters too

Spatial configuration matters too

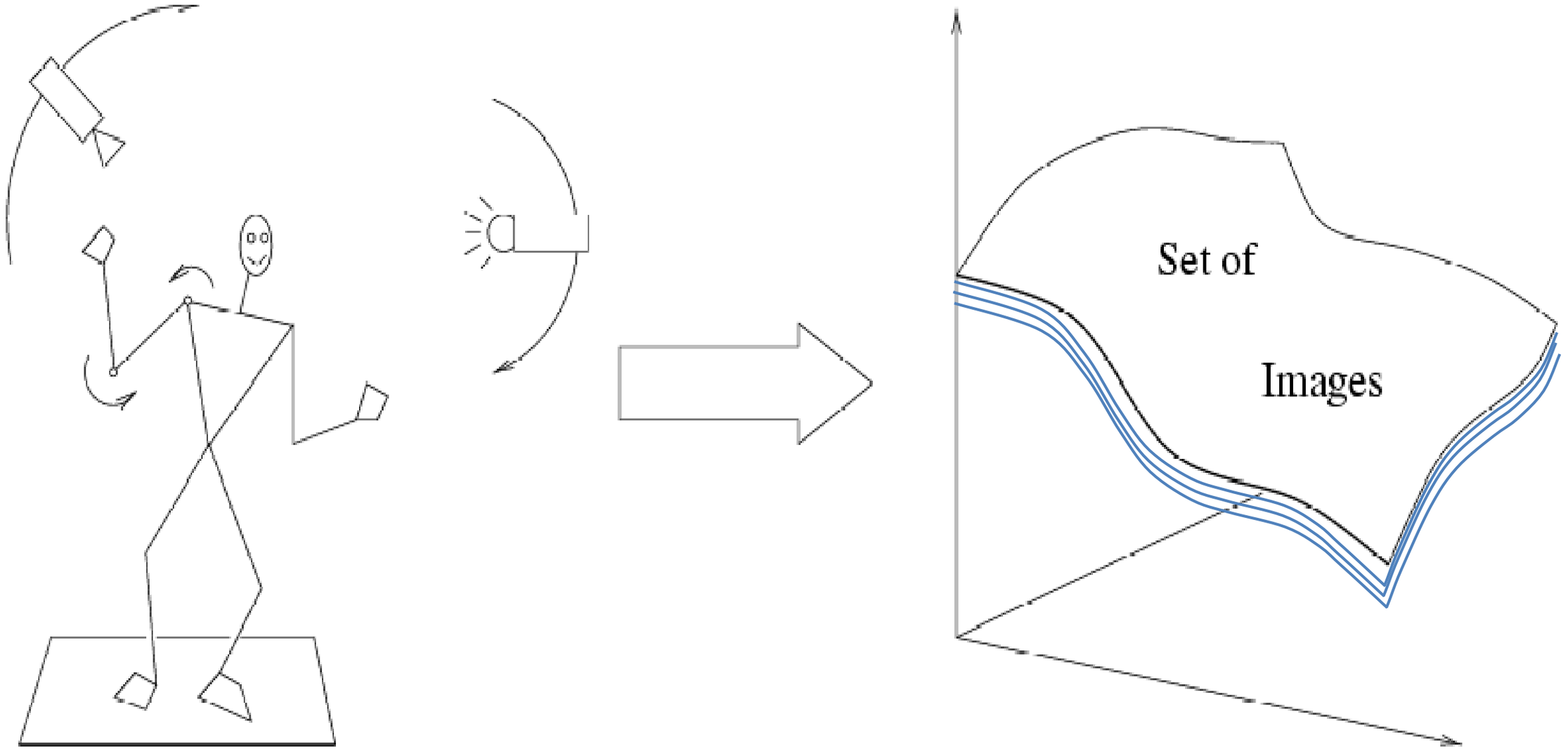


The list goes on

Face Recognition by Humans: Nineteen Results All Computer Vision Researchers Should Know About

- http://web.mit.edu/bcs/sinha/papers/19results_sinha_etal.pdf

Why is this hard?



Variability: Camera position
Illumination
Shape parameters

How many object categories are there?

~10,000 to 30,000

Challenge: variable viewpoint

Michelangelo 1475-1564

Challenge: variable illumination

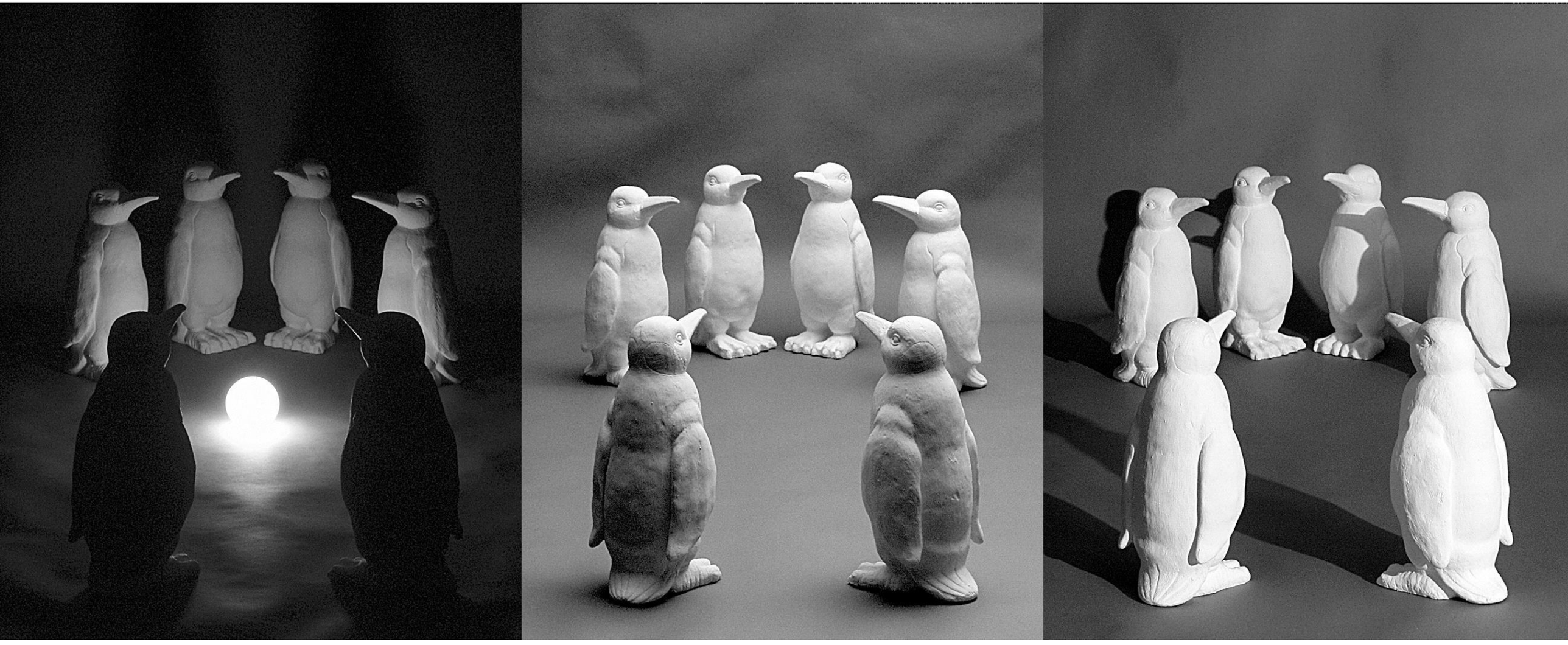


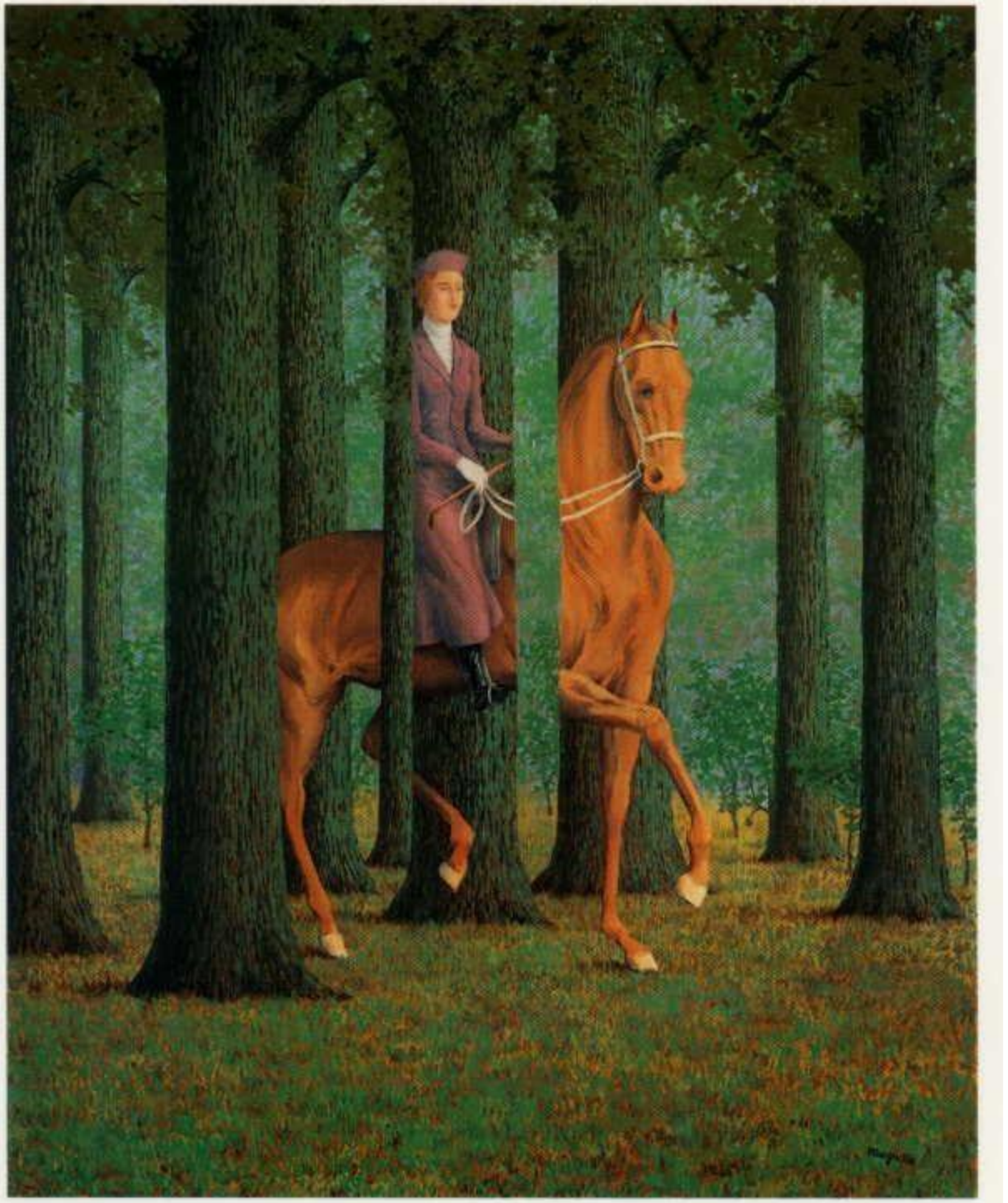
image credit: J. Koenderink

Challenge: scale

Challenge: deformation

Deformation

Challenge: Occlusion



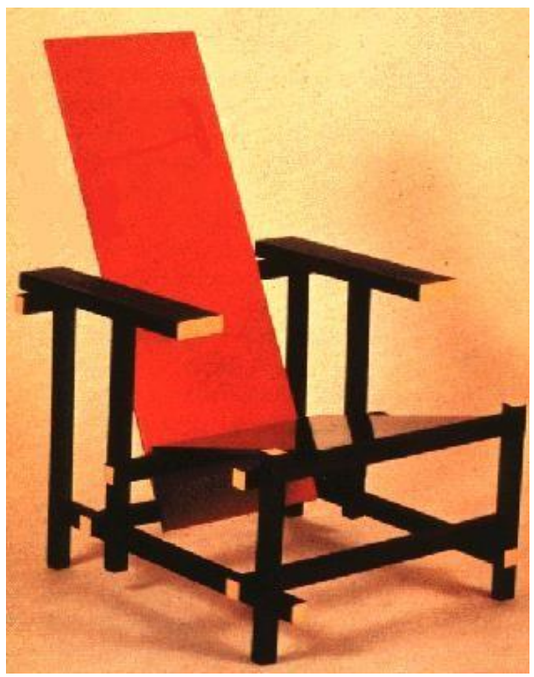
Magritte, 1957

Occlusion

Challenge: background clutter

Kilmeny Niland. 1995

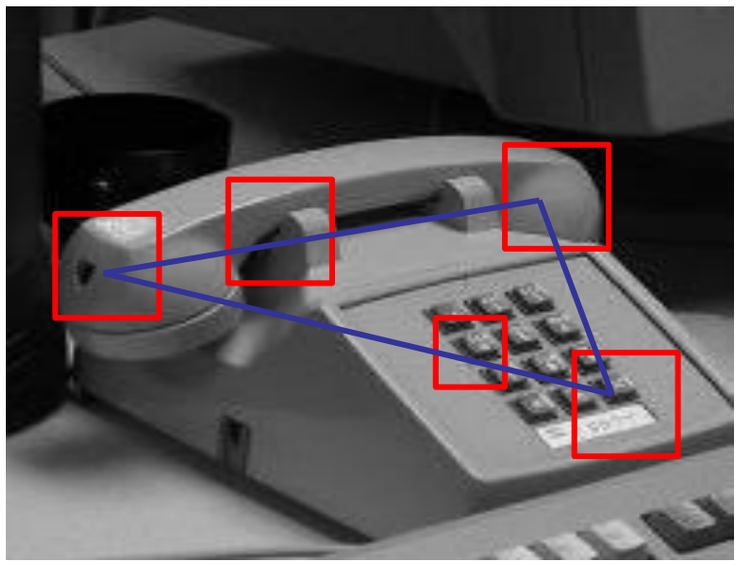
Challenge: intra-class variations



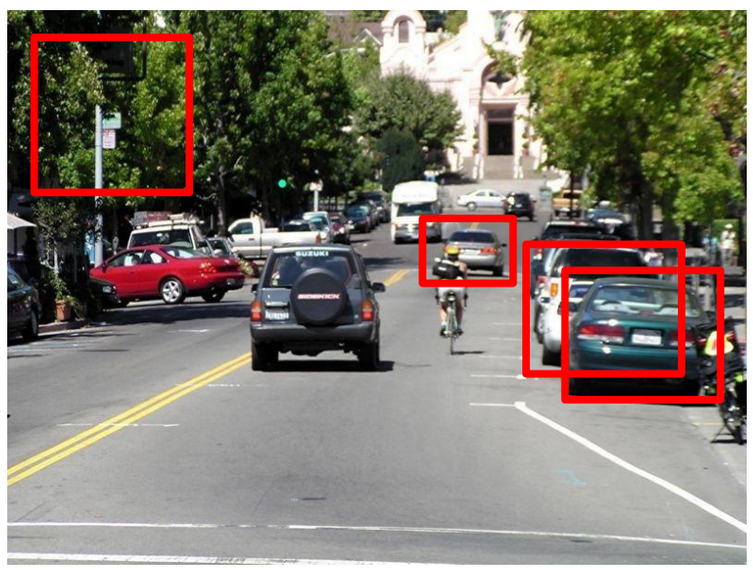
Common approaches

Common approaches: object recognition

Feature Matching



Spatial reasoning



Window classification

Feature matching

What object do these parts belong to?





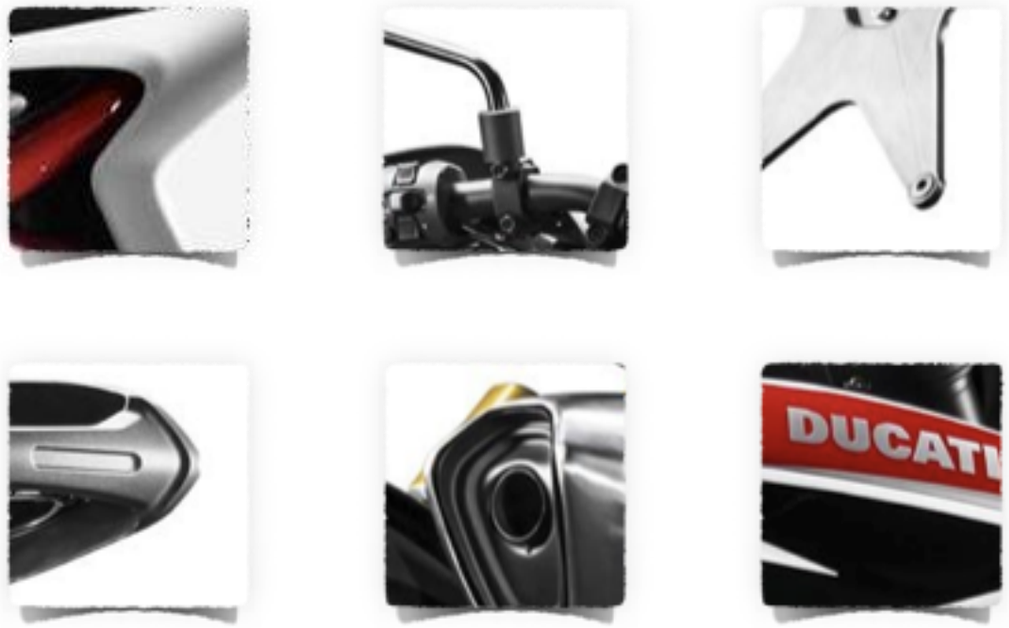









Some local feature are very informative



An object as

a collection of local features

(bag-of-features)

- deals well with occlusion
- scale invariant
- rotation invariant

Are the positions of the parts important?

Why not use SIFT matching for everything?

- Works well for object *instances*

- Not great for generic object *categories*

Pros

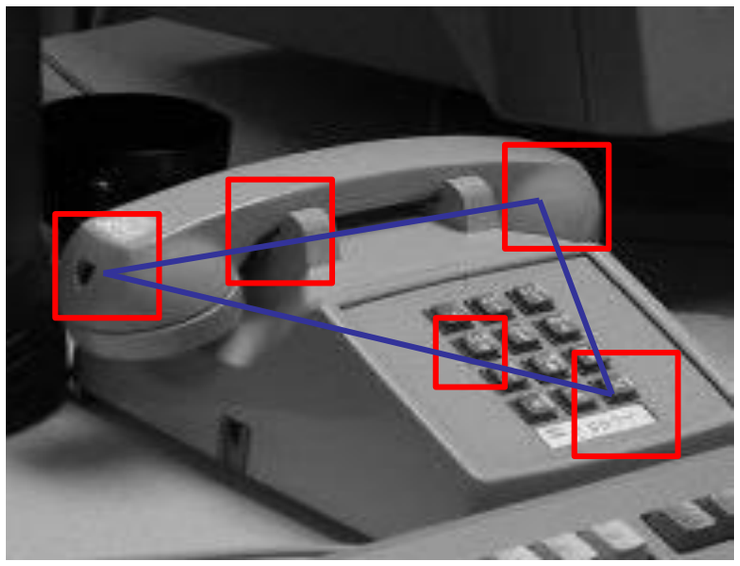
- Simple
- Efficient algorithms
- Robust to deformations

Cons

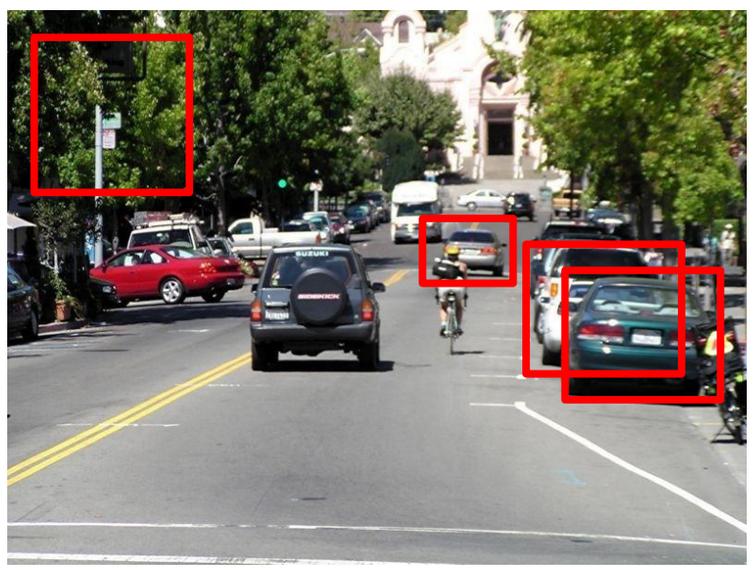
- No spatial reasoning

Common approaches: object recognition

Feature Matching



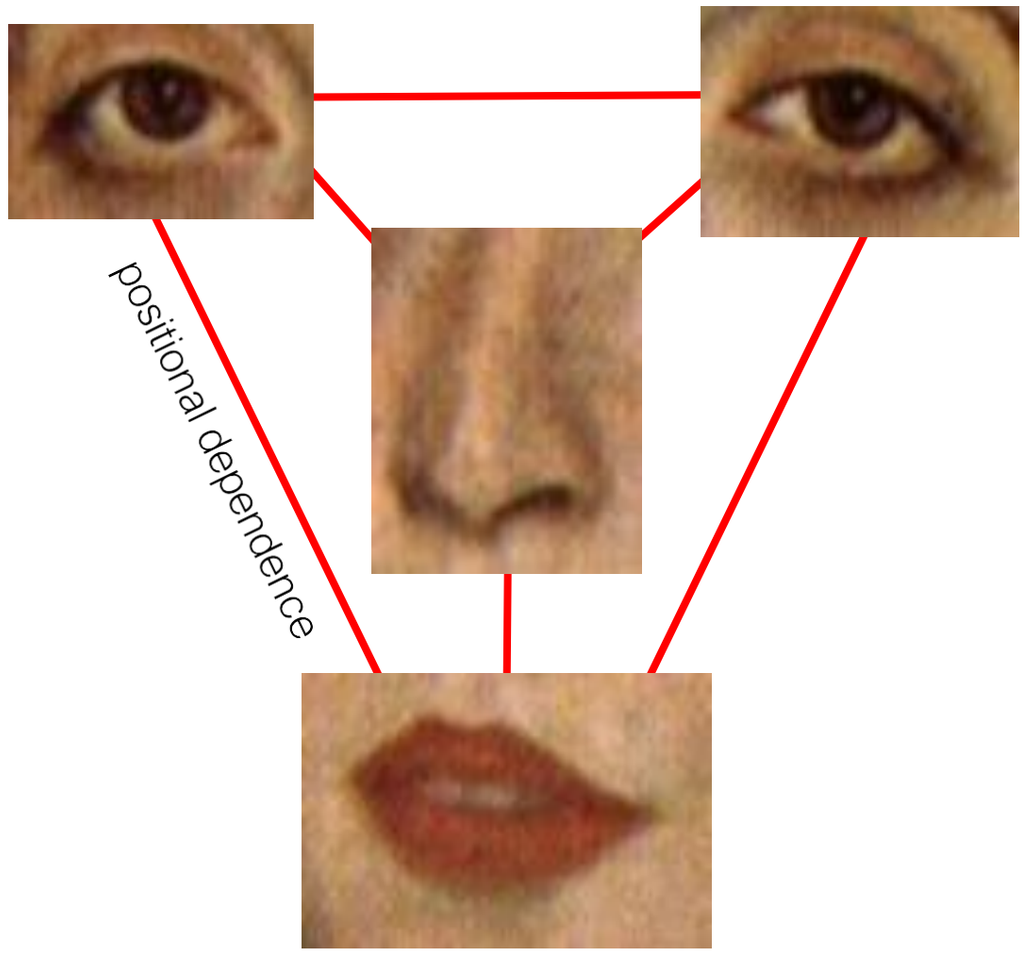
Spatial reasoning



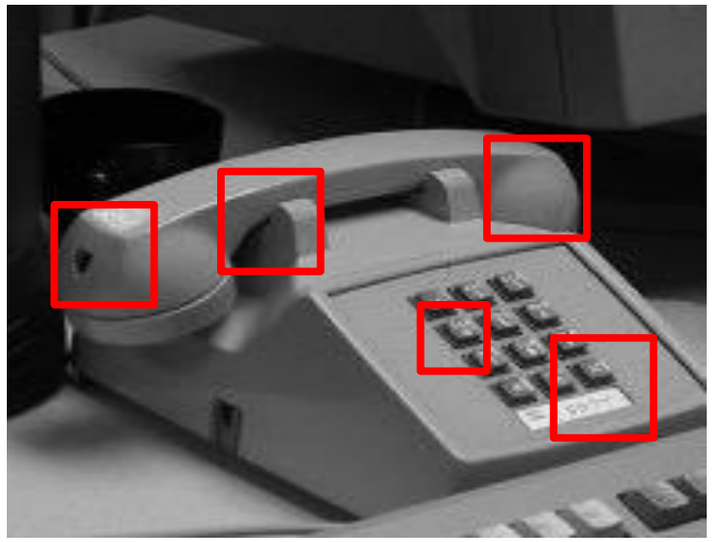
Window classification

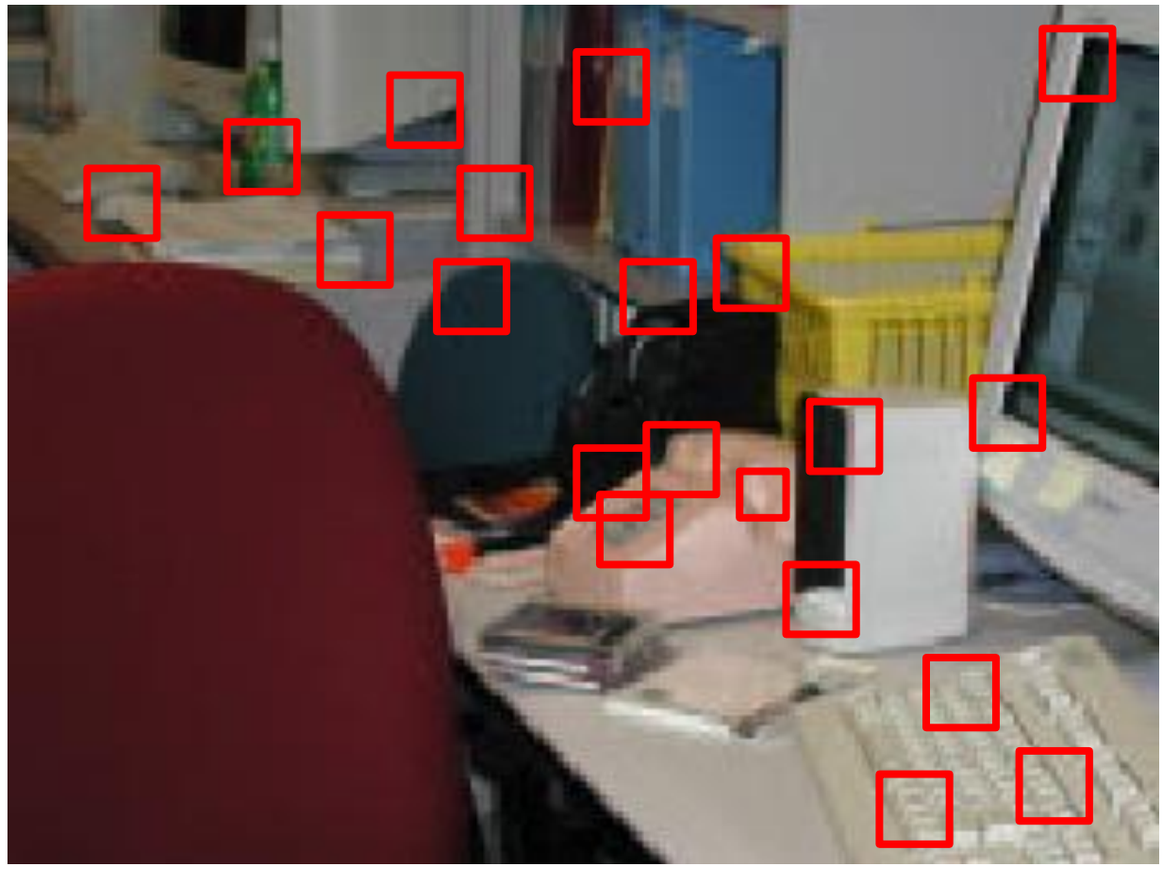
Spatial reasoning

The position of every part depends on the positions of all the other parts



Many parts, many dependencies!

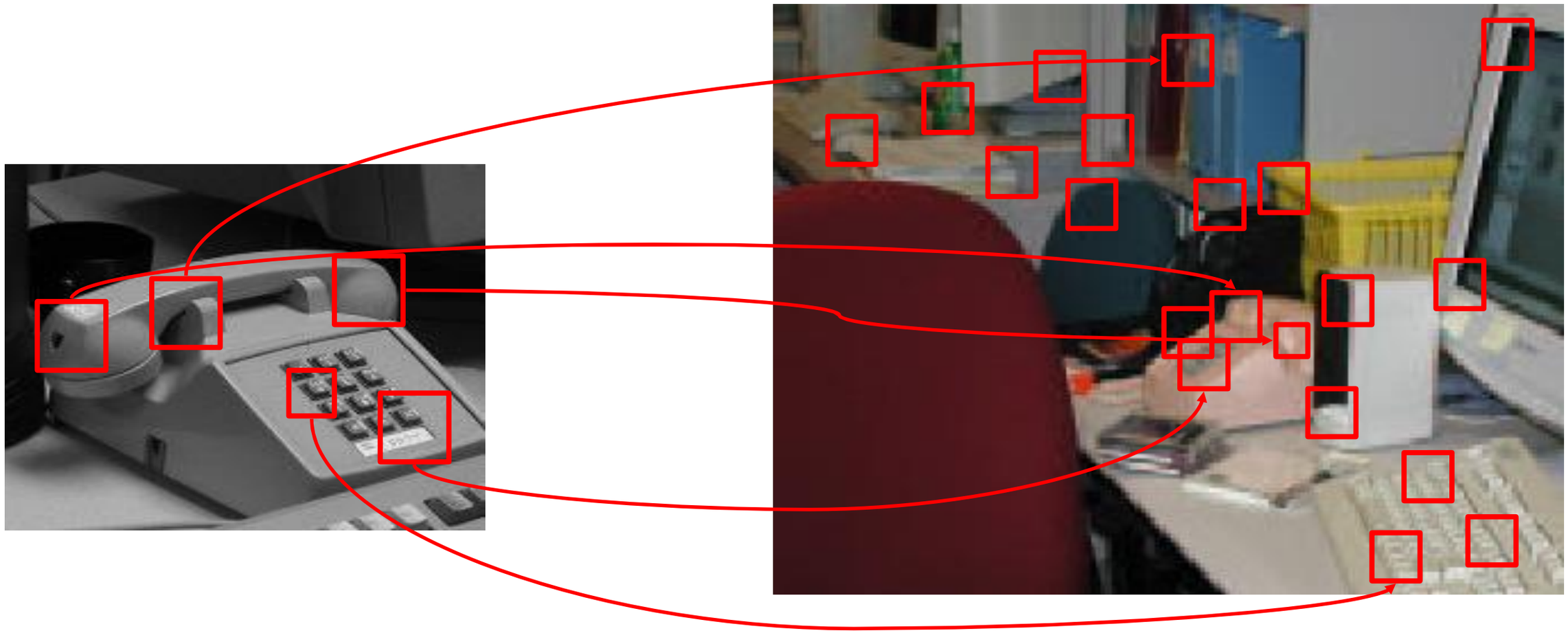




1. Extract features

2. Match features

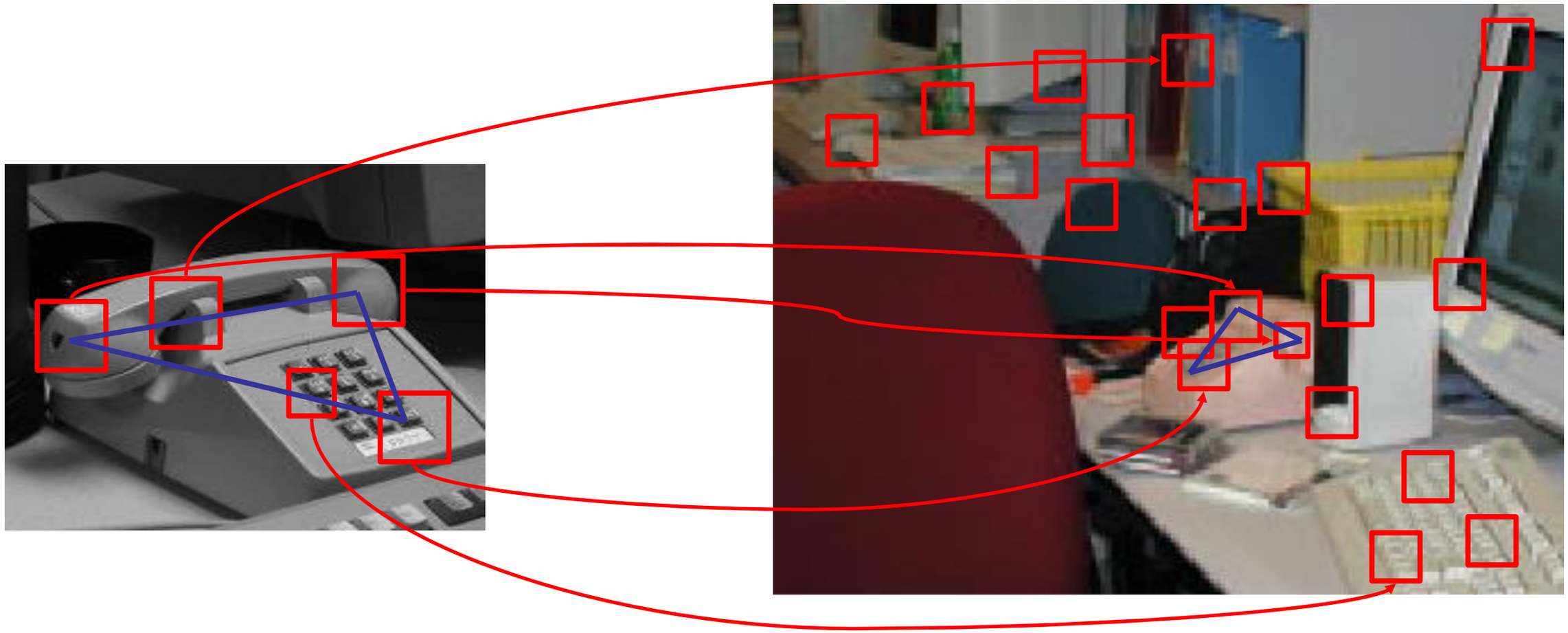
3. Spatial verification



1. Extract features

2. Match features

3. Spatial verification

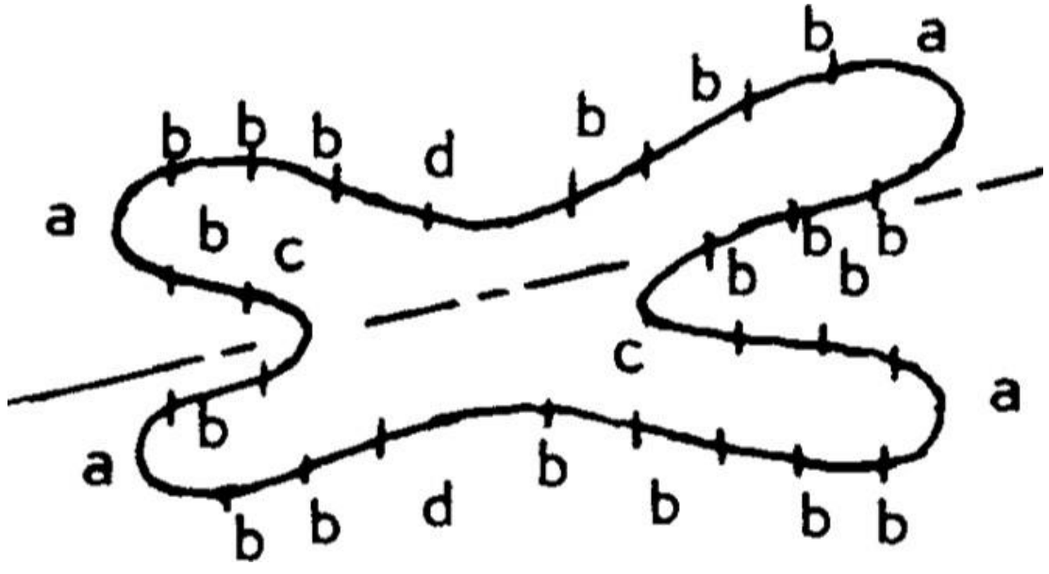


1. Extract features

2. Match features

3. Spatial verification

an old idea…

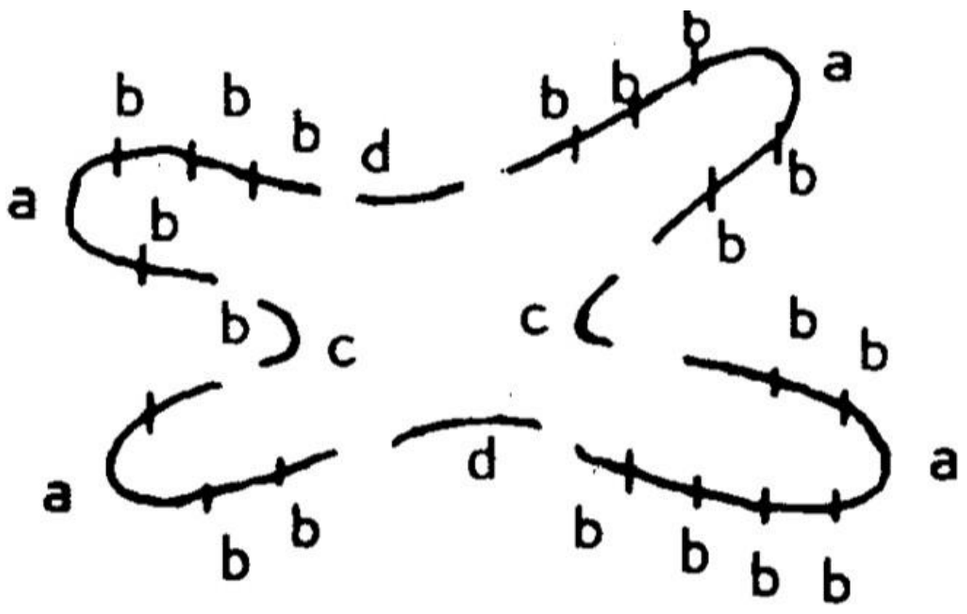


Coded Chromosome

$V_T = \{$  $\}$

x = cdabbbdbbbabbcbbabbbbdbbabb

Substructures of Coded Chromosome



S_1 = {[b[[[a]b]b]b]; [b[b[b[a]]b]b];
[b[b[[[a]b]b]b]b]; [b[b[a]]b]}

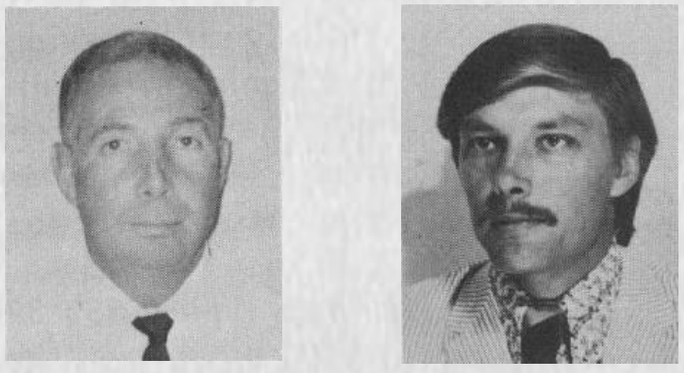
The Representation and Matching of Pictorial Structures

MARTIN A. FISCHLER AND ROBERT A. ELSCHLAGER

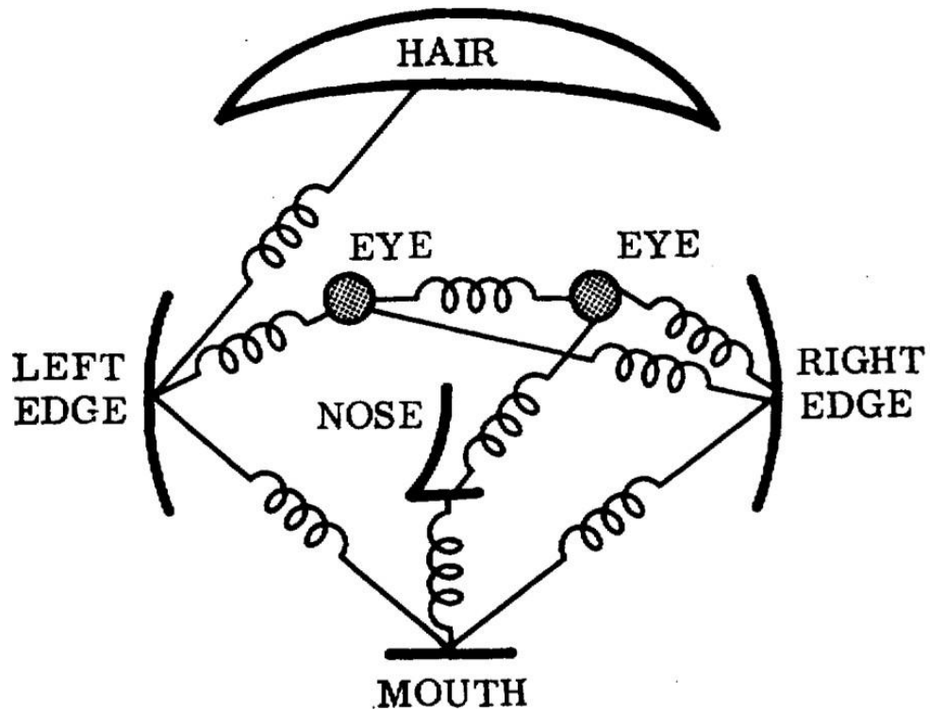
Abstract—The primary problem dealt with in this paper is the following. Given some description of a visual object, find that object in an actual photograph. Part of the solution to this problem is the specification of a descriptive scheme, and a metric on which to base the decision of "goodness" of matching or detection.

We offer a combined descriptive scheme and decision metric which is general, intuitively satisfying, and which has led to promising experimental results. We also present an algorithm which takes the above descriptions, together with a matrix representing the intensities of the actual photograph, and then finds the described object in the matrix. The algorithm uses a procedure similar to dynamic programming in order to cut down on the vast amount of computation otherwise necessary.

One desirable feature of the approach is its generality. A new programming system does not need to be written for every new description; instead, one just specifies descriptions in terms of a certain set of primitives and parameters.



1972



Description for left edge of face

A B C D	X	E F G H

VALUE(X)=(E+F+G+H)-(A+B+C+D)

Note: VALUE(X) is the value assigned to the L(EV)A corresponding to the location X as a function of the intensities of locations A through H in the sensed scene.

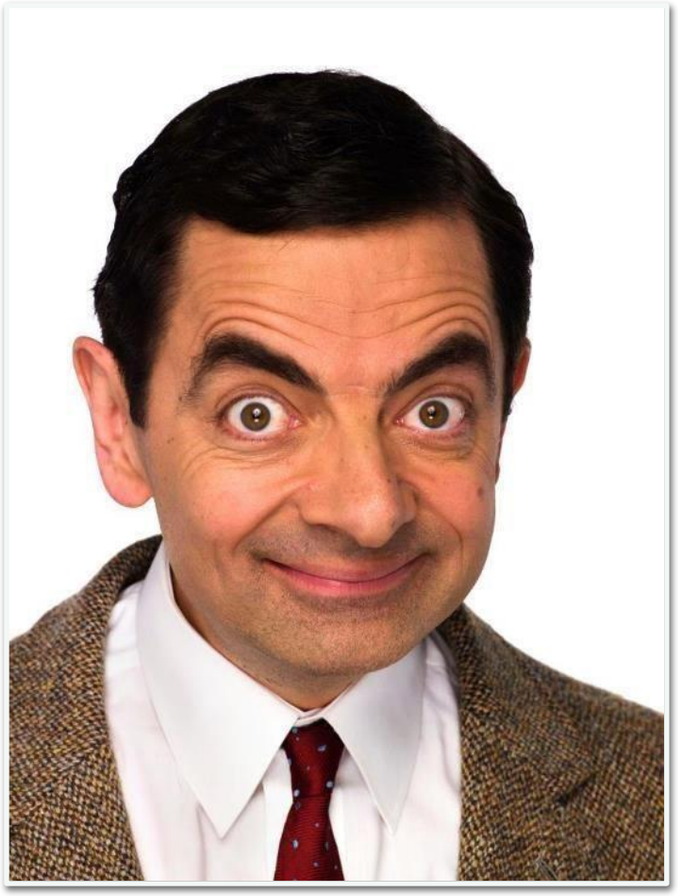
A more probabilistic approach…

think of locations as random variables (RV)

vector of RVs:
set of part locations

$$\boldsymbol{L} = \{\overset{\text{RV}}{L_1}, \overset{\text{RV}}{L_2}, \ldots, \overset{\text{RV}}{L_M}\}$$

Given any collection of selfie images,
where would you expect the nose to be?



*What would be an appropriate **prior**?*

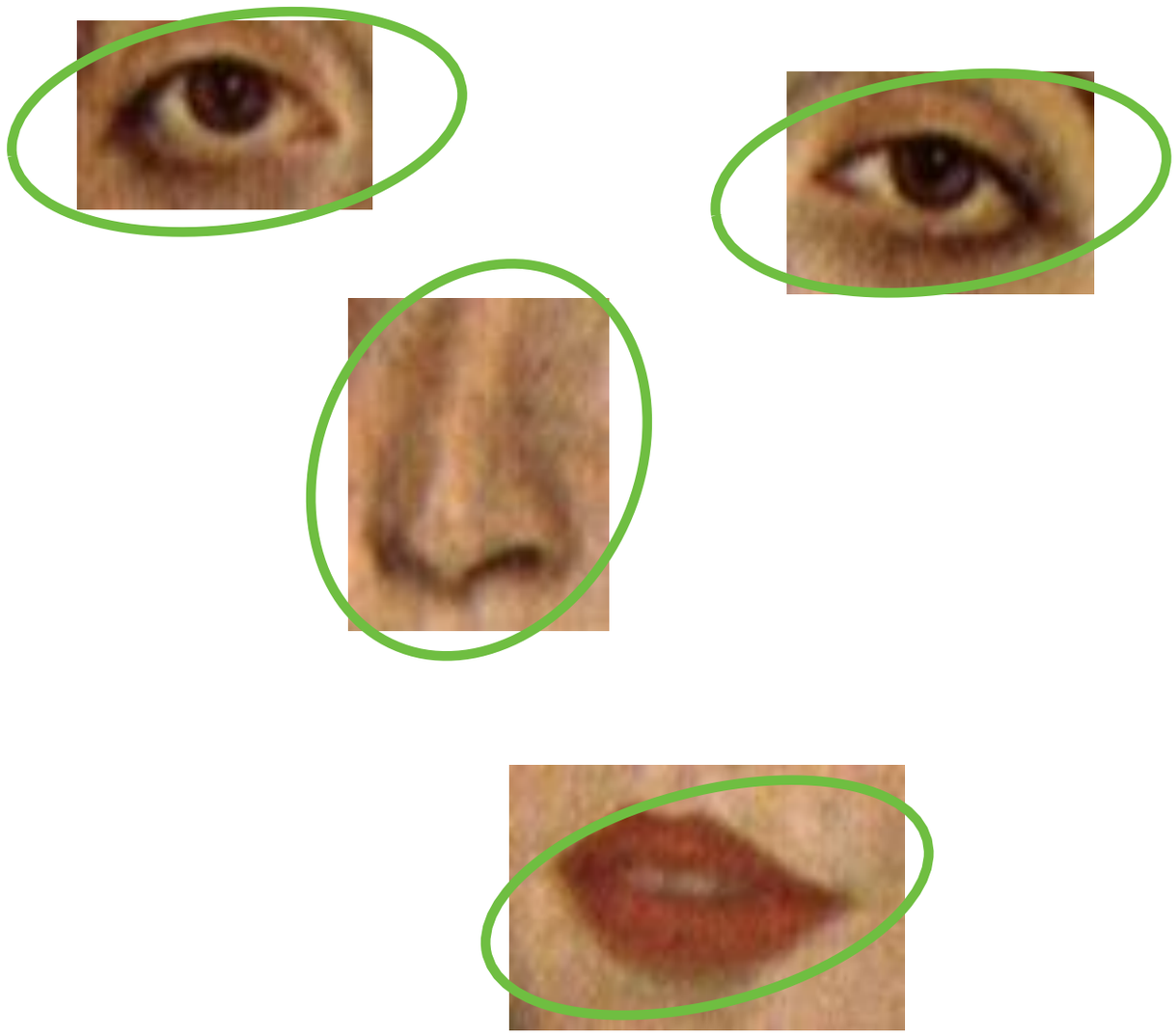
$$P(L_{\text{nose}}) = ?$$

A simple factorized model

$$p(\boldsymbol{L}) = \prod_{m} p(L_m)$$

Break up the joint probability into smaller (independent) terms

Independent locations

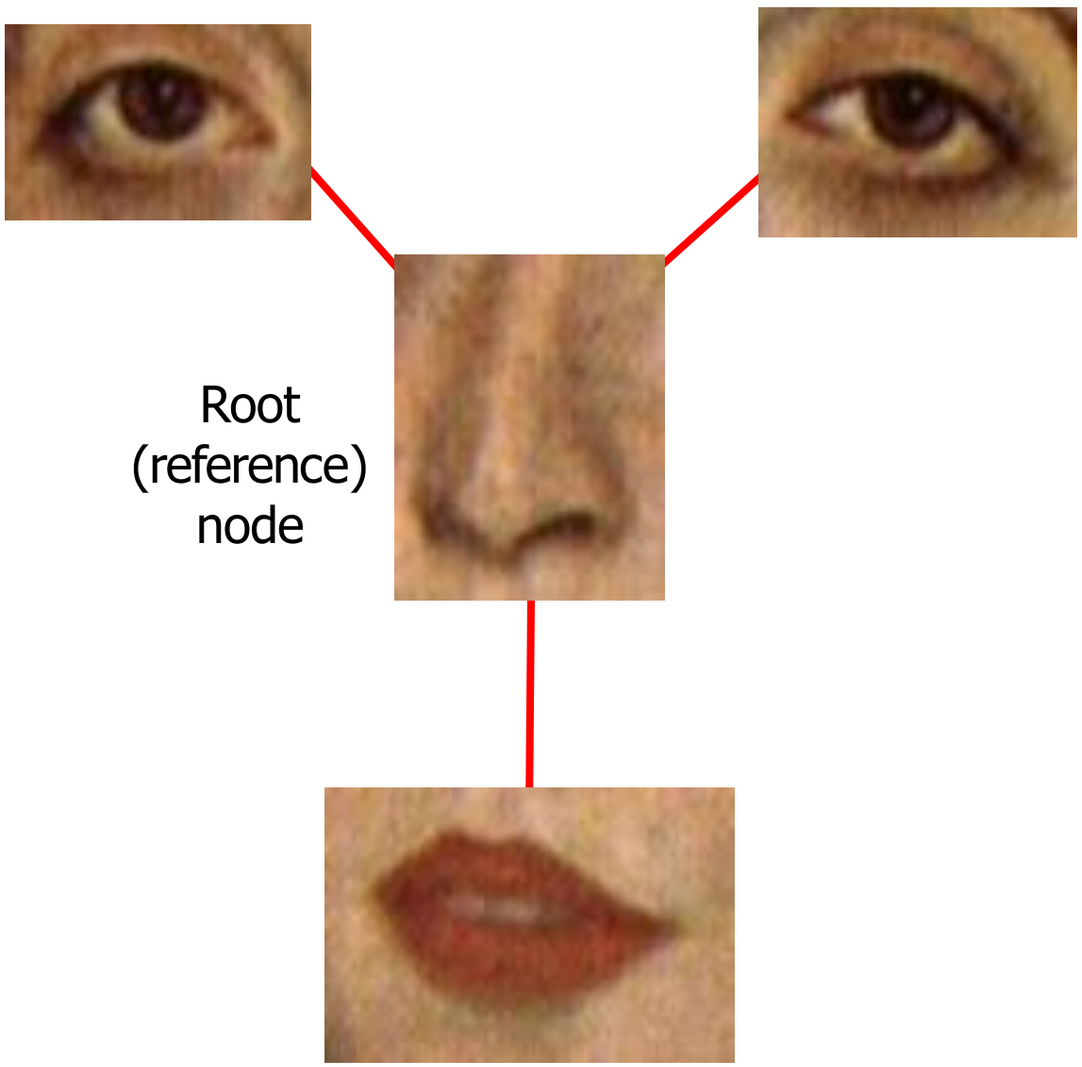


$$p(\boldsymbol{L}) = \prod_m p(L_m)$$

Each feature is allowed to move independently

Does not model the **relative** location of parts at all

Tree structure
(star model)

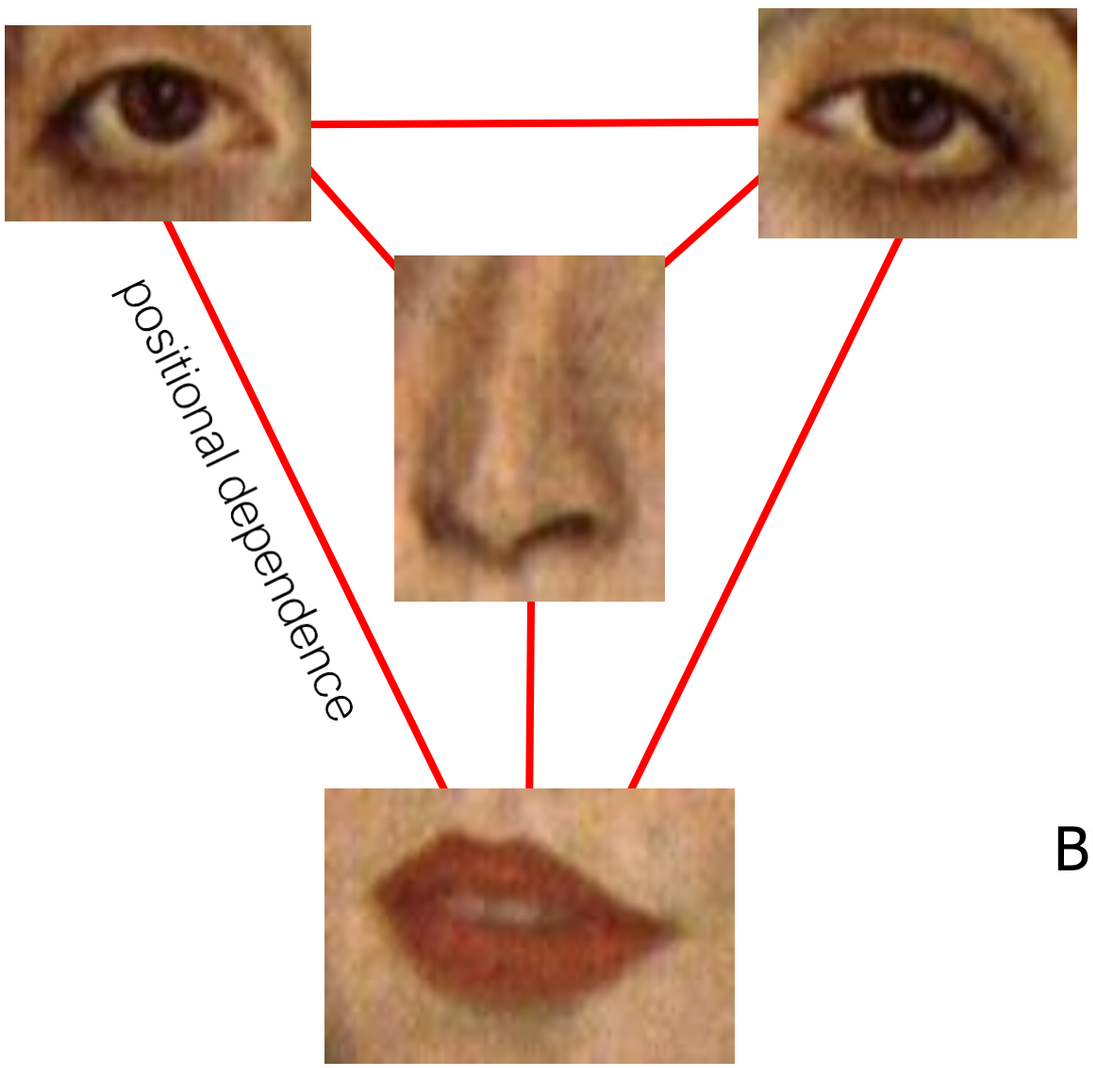


$$p(\boldsymbol{L}) = p(L_{\text{root}}) \prod_{m=1}^{M-1} p(L_m | L_{\text{root}})$$

Represent the location of all the parts relative to a single reference part

Assumes that one reference part is defined (who will decide this?)

Fully connected
(constellation model)



$$p(L) = p(l_1, \ldots, l_N)$$

Explicitly represents the
joint distribution of locations

<u>Good model:</u>
Models relative location of parts
BUT Intractable for moderate number of parts

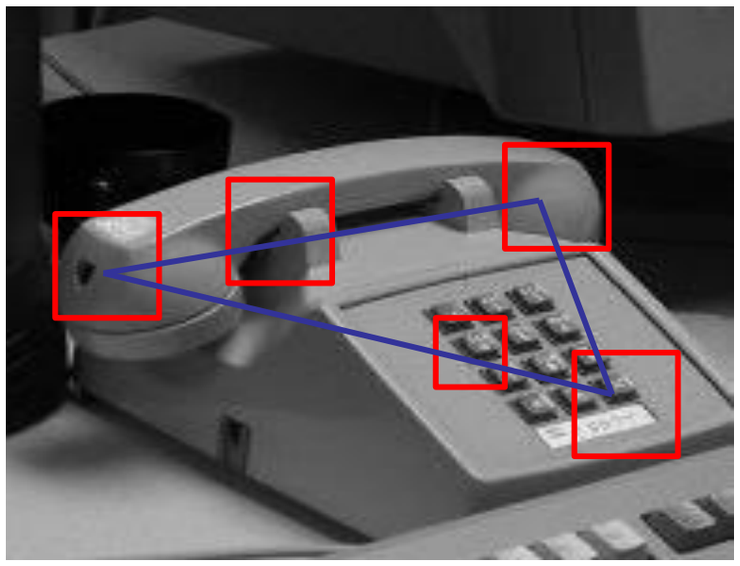
Pros

- Retains spatial constraints
- Robust to deformations

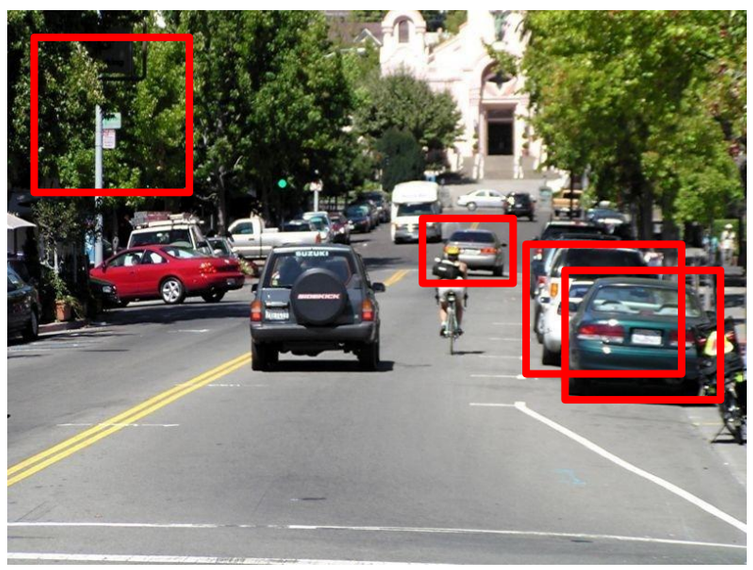
Cons

- Computationally expensive
- Generalization to **large** inter-class variation (e.g., modeling chairs)

Feature Matching



Spatial reasoning

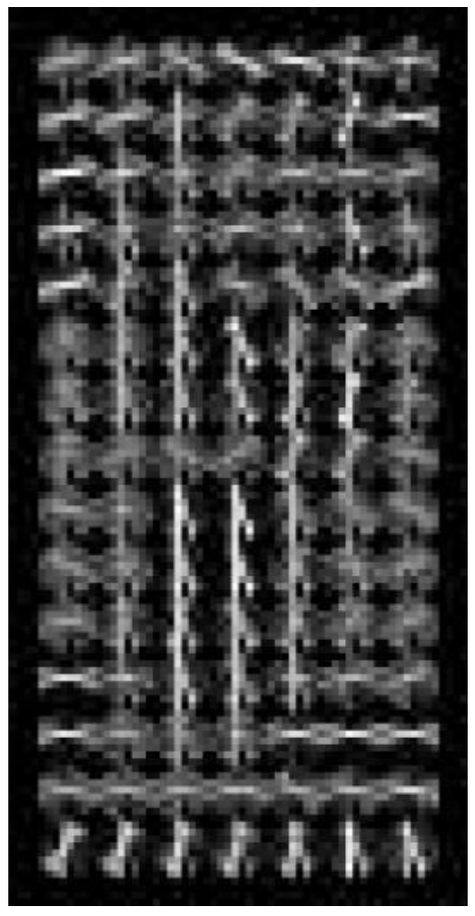


Window classification

Window-based

Template Matching

1. get image window



2. extract features

3. classify

When does this work and when does it fail?

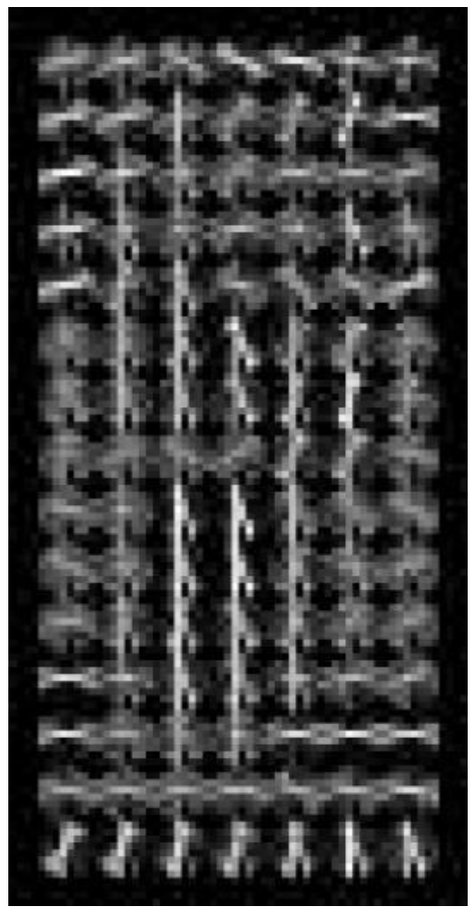
How many templates do you need?

Per-exemplar

find the ‘nearest’ exemplar, inherit its label

Template Matching

1. get image window
(or region proposals)



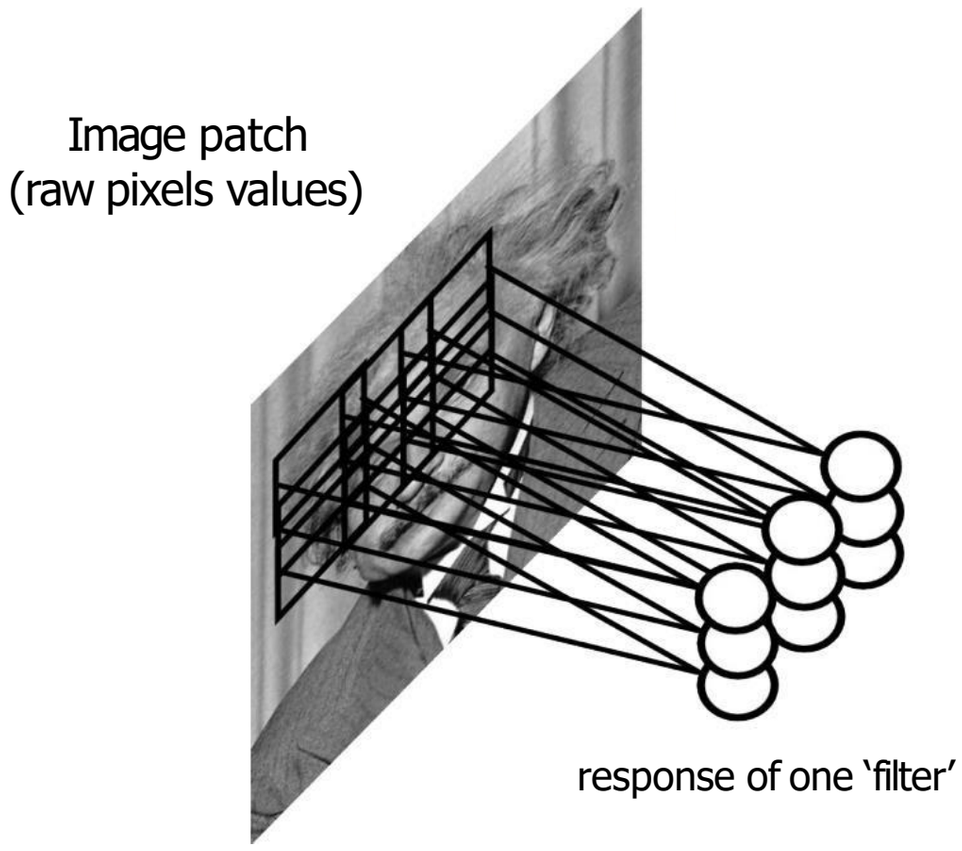
2. extract features

3. compare to template

Do this part with one big classifier
'end to end learning'

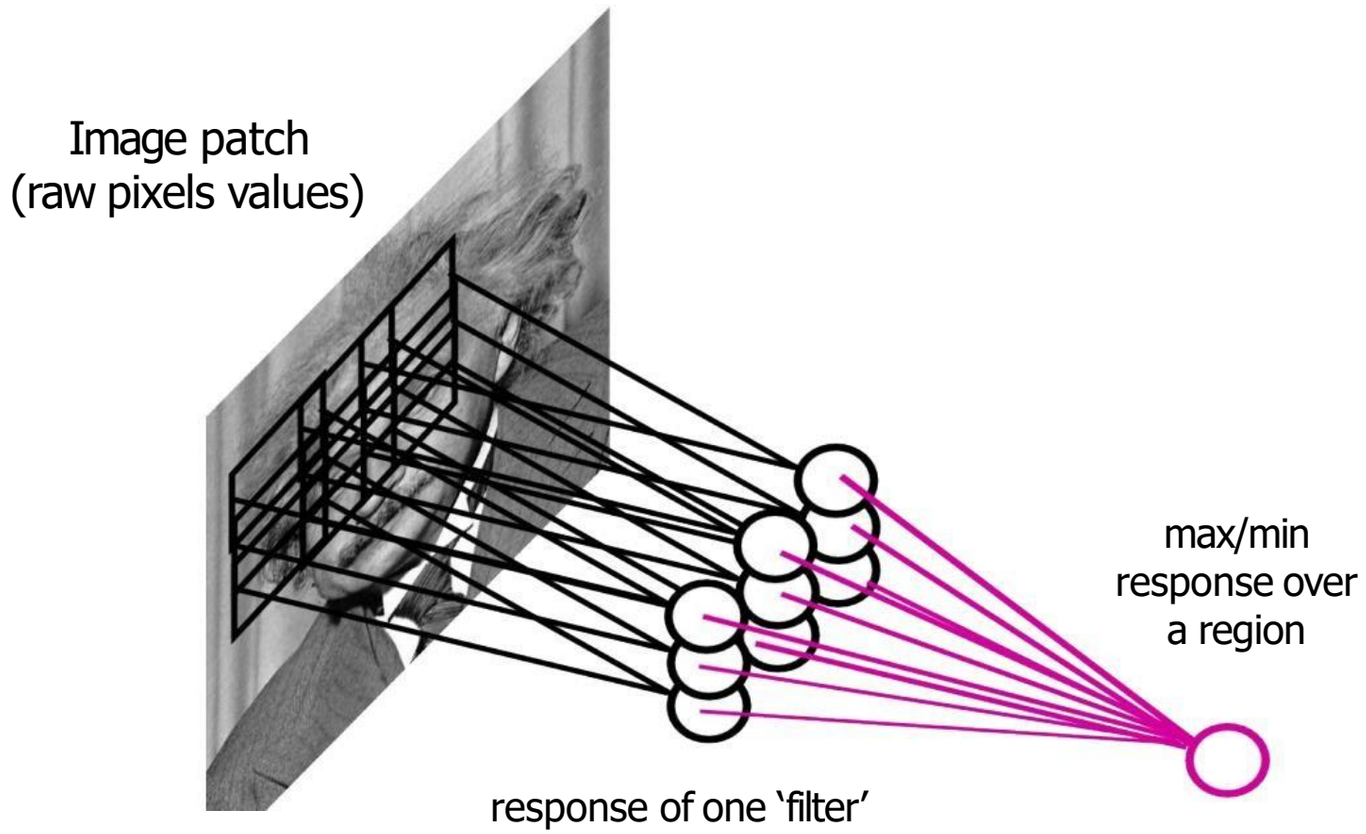
Convolutional Neural Networks

Convolution



A 96 x 96 image convolved with 400 filters (features) of size 8 x 8 generates about 3 million values (89^2x400)

Pooling



Pooling aggregates statistics and lowers the dimension of convolution

Step-by-step Calculation

1.Image Dimensions:

The input image is 96×9696 \times 9696×96 pixels.

2.Filter Size:

Each filter has dimensions 8×8.

3.Output Size After Convolution:

When a convolution operation is performed, the output size is determined by the formula for valid convolutions (no padding):

$$\text{Output Dimension} = \text{Input Dimension} - \text{Filter Dimension} + 1$$

Applying this to the 96×96 image with 8×8 filters:

$$\text{Output Width} = 96 - 8 + 1 = 89$$

$$\text{Output Height} = 96 - 8 + 1 = 89$$

So, the output of each filter is an 89×89 feature map.

4. **Number of Filters**:

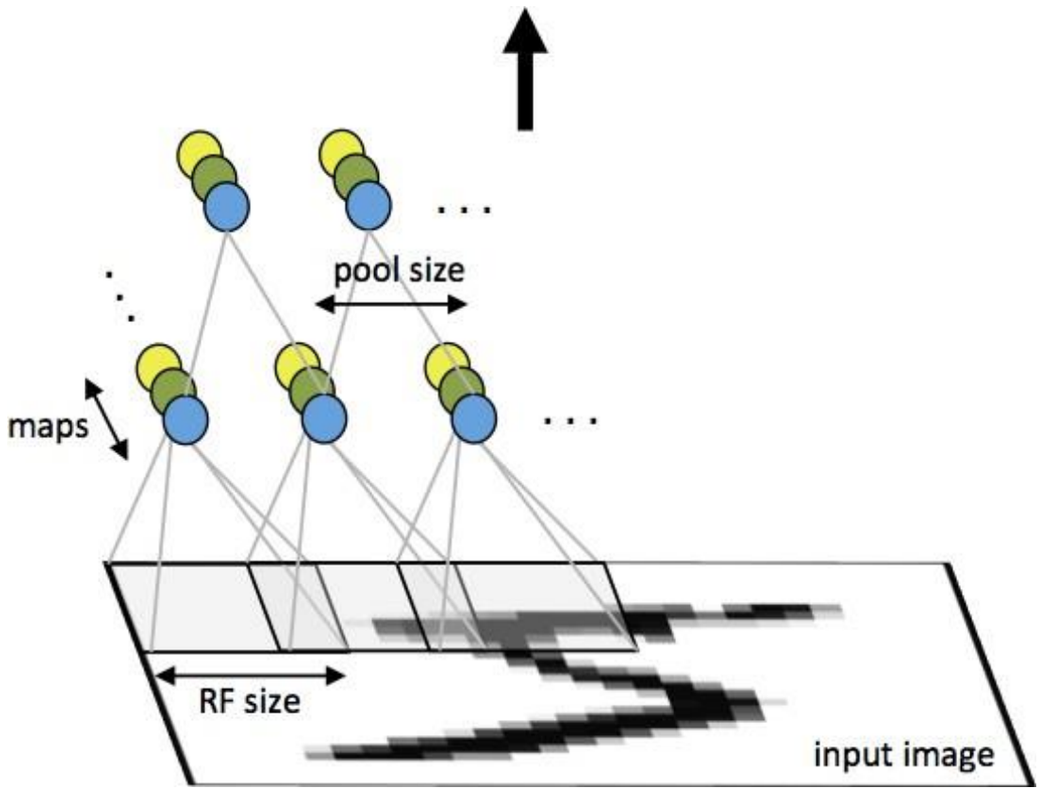
   There are 400 filters applied to the image. This means each filter produces one feature map, resulting in 400 feature maps.

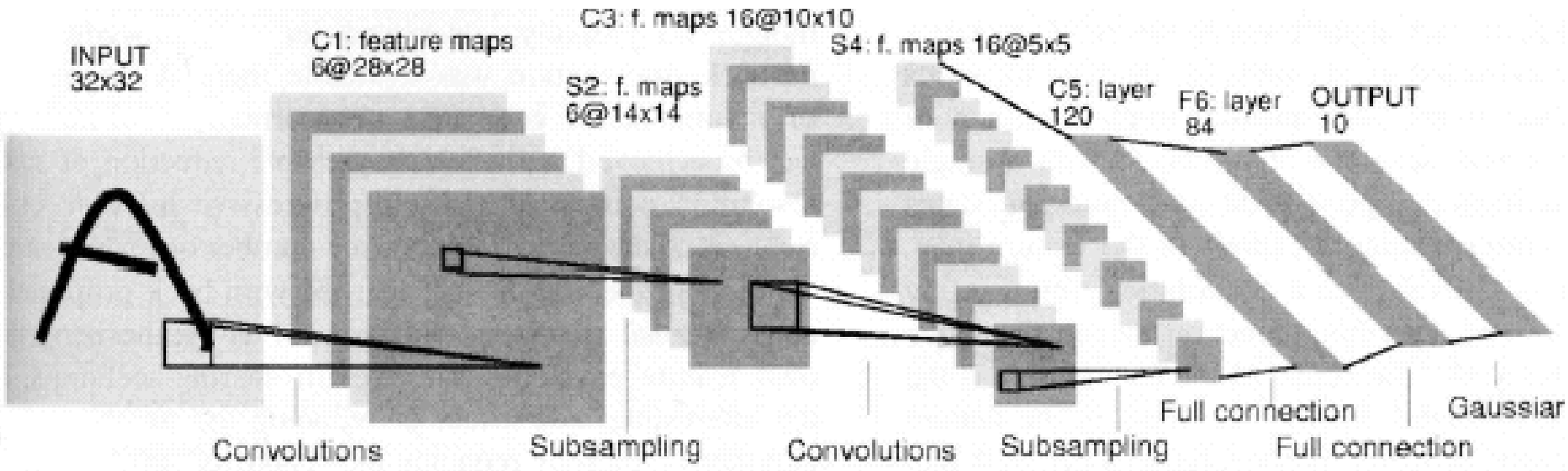
5. **Total Number of Values**:

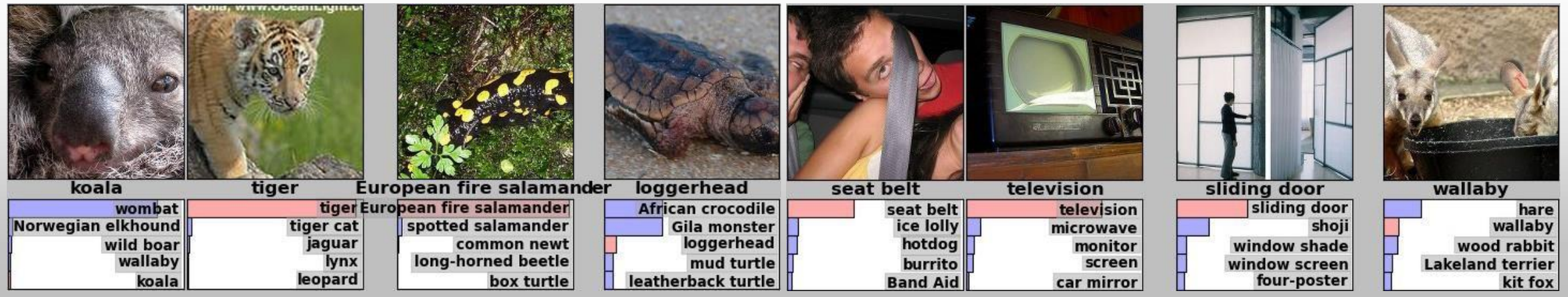
   Each feature map has $89 \times 89 = 7,921$ values. With 400 feature maps, the total number of values is:

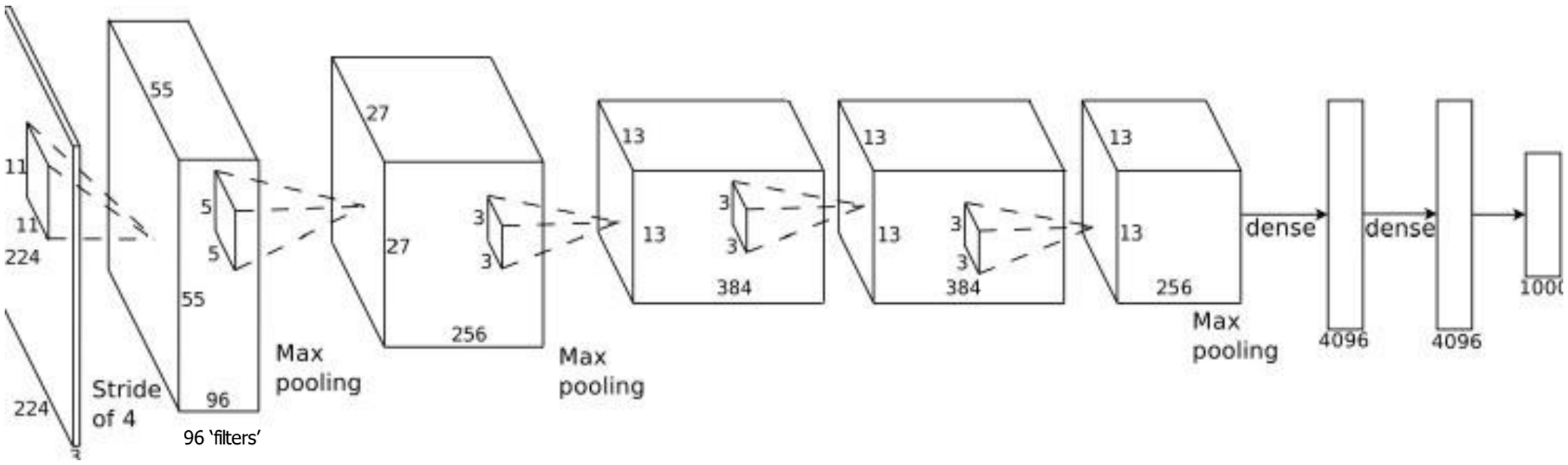
$$\text{Total Values} = 7,921 \times 400 = 3,168,400$$

This is approximately **3 million values**.


pool size
maps
RF size
input image


INPUT
32x32
C1: feature maps
6@28x28
S2: f. maps
6@14x14
C3: f. maps 16@10x10
S4: f. maps 16@5x5
C5: layer
120
F6: layer
84
OUTPUT
10
Convolutions
Subsampling
Convolutions
Subsampling
Full connection
Full connection
Gaussiar





224/4=56

630 million connections
60 millions parameters to learn

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

Pros

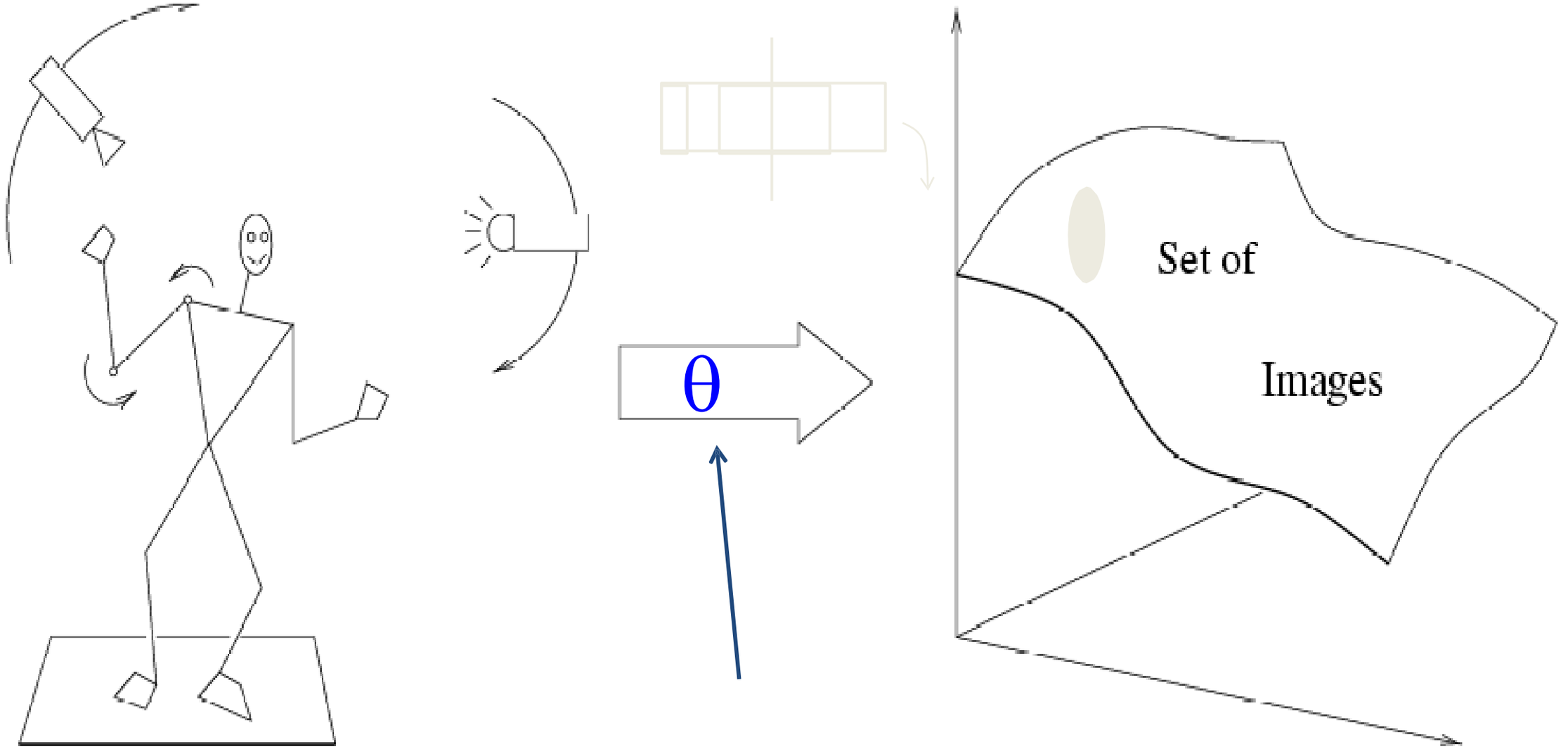
- Retains spatial constraints
- Efficient test time performance

Cons

- Many many possible windows to evaluate
- Requires large amounts of data
- Sometimes (very) slow to train

History of ideas in recognition

- 1960s – early 1990s: the geometric era



Variability: θ Camera position
Illumination

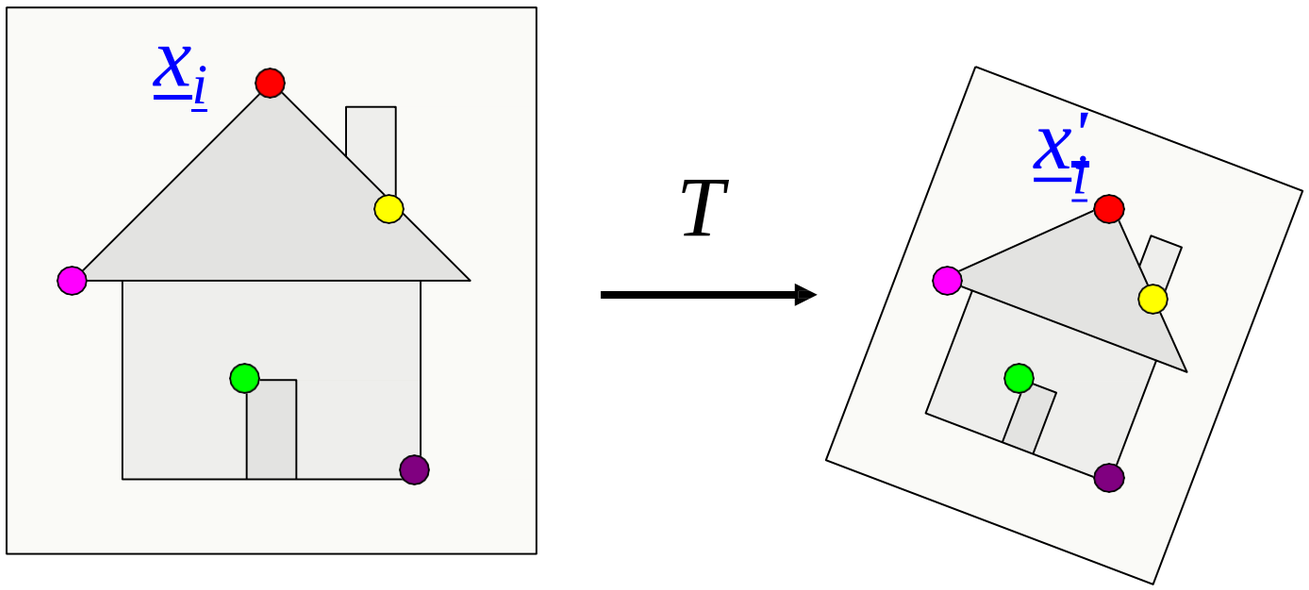
Alignment

Shape: assumed known

Roberts (1965); Lowe (1987); Faugeras & Hebert (1986); Grimson & Lozano-Perez (1986); Huttenlocher & Ullman (1987)

Instance Recognition

- Alignment: fitting a model to a transformation between pairs of features (*matches*) in two images



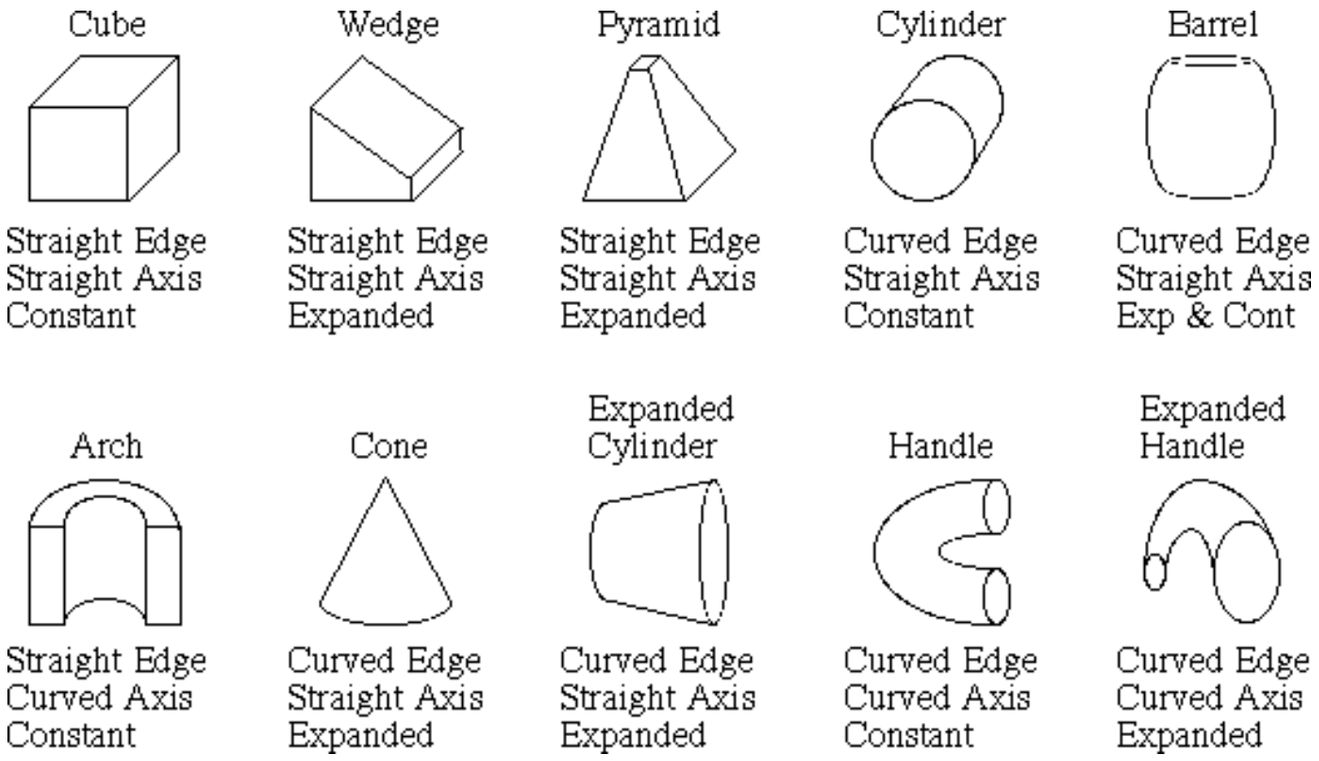
Find transformation T that minimizes

$$\sum_i \text{residual}\quad (T(x_i), x'_i)$$

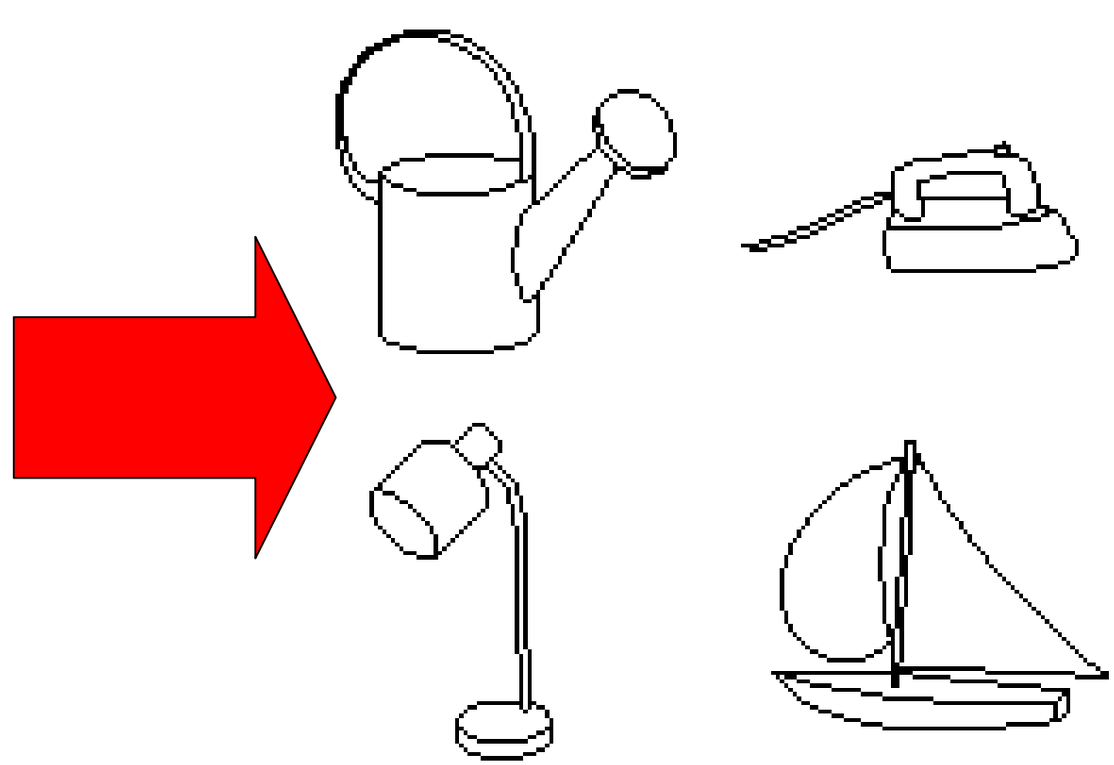
Recognition by components

Biederman (1987)

Primitives (geons)



Objects

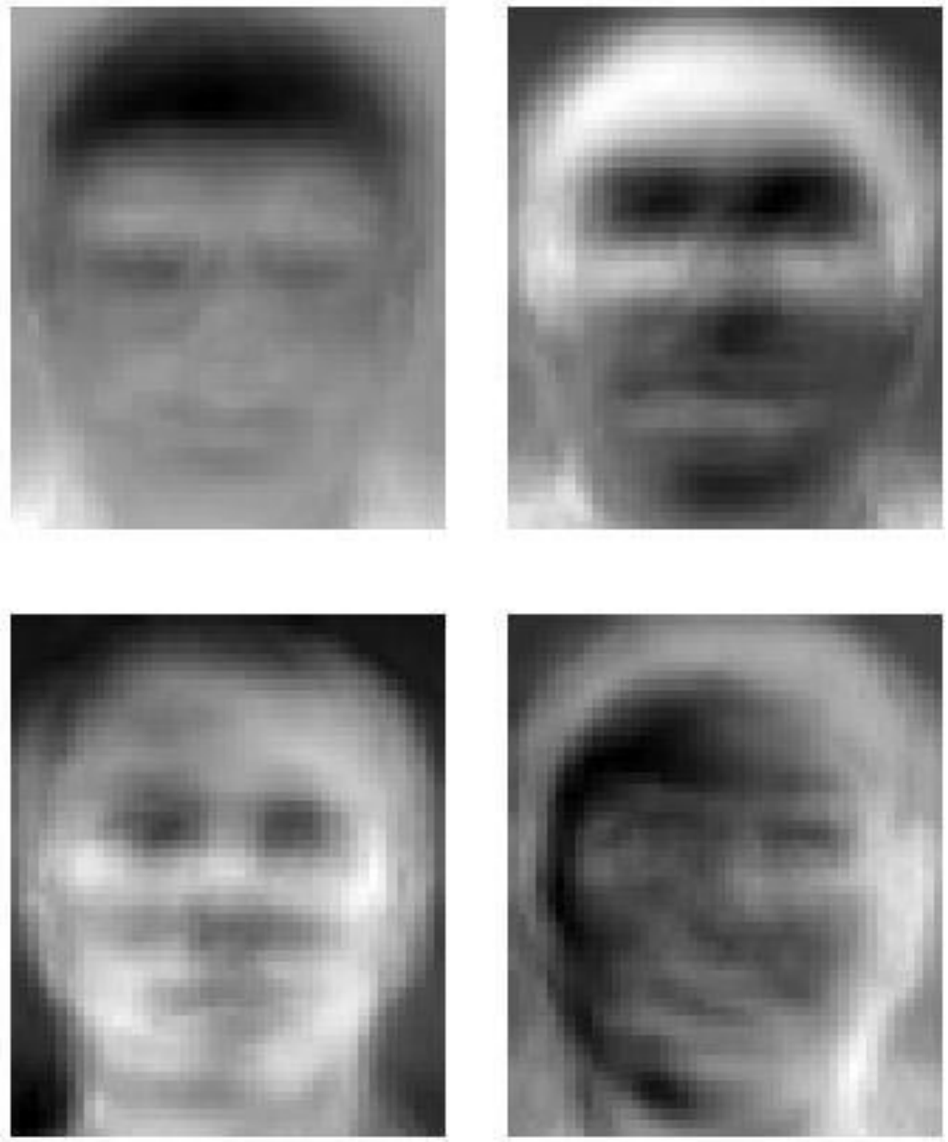


http://en.wikipedia.org/wiki/Recognition_by_Components_Theory

History of ideas in recognition

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models

Eigenfaces (Turk & Pentland, 1991)



Experimental	Correct/Unknown Recognition Percentage		
Condition	Lighting	Orientation	Scale
Forced classification	96/0	85/0	64/0
Forced 100% accuracy	100/19	100/39	100/60
Forced 20% unknown rate	100/20	94/20	74/20

Svetlana Lazebnik

Limitations of global appearance models

- Requires global registration of patterns
- Not robust to clutter, occlusion, geometric transformations

History of ideas in recognition

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- 1990s – present: sliding window approaches

Sliding window approaches

Sliding window approaches

- Turk and Pentland, 1991
- Belhumeur, Hespanha, & Kriegman, 1997
- Schneiderman & Kanade 2004
- Viola and Jones, 2000

- Schneiderman & Kanade, 2004
- Agrawal and Roth, 2002
- Poggio et al. 1993

History of ideas in recognition

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- Mid-1990s: sliding window approaches
- Late 1990s: local features

Local features for object instance recognition

D. Lowe (1999, 2004)

History of ideas in recognition

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- Mid-1990s: sliding window approaches
- Late 1990s: local features
- Early 2000s: parts-and-shape models

Parts-and-shape models

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

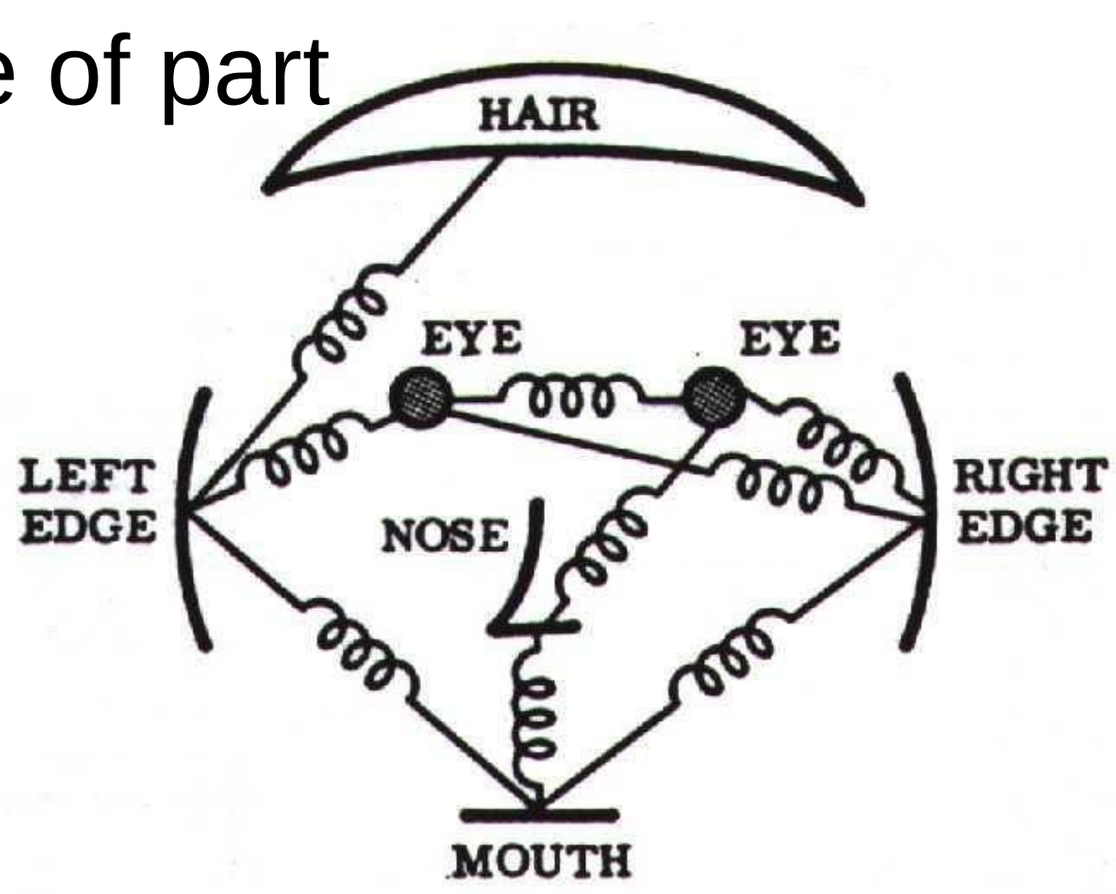
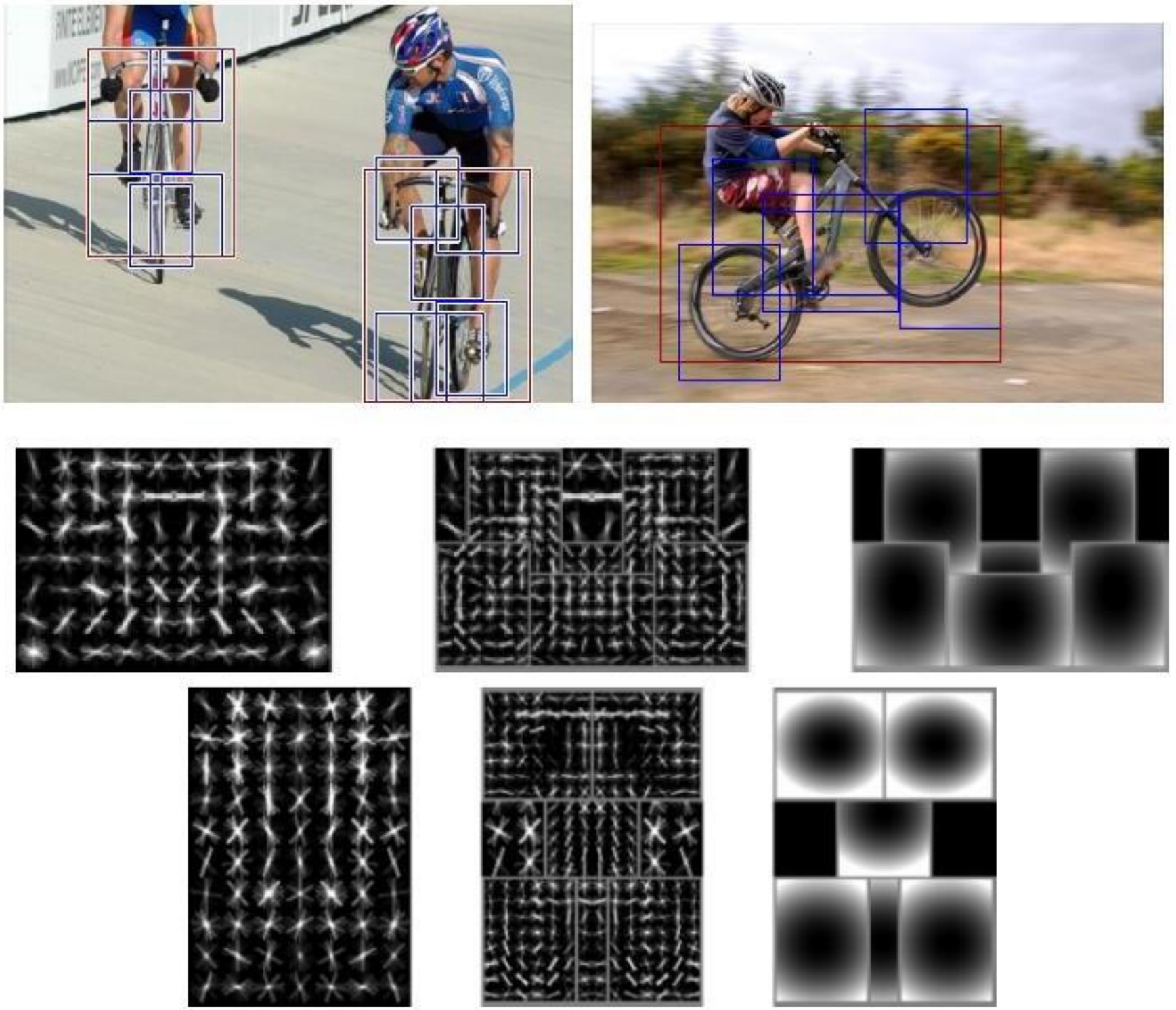


Figure from [Fischler & Elschlager 73]

Discriminatively trained part-based models



P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, **"Object Detection with Discriminatively Trained Part-Based Models,"** PAMI 2009

History of ideas in recognition

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- Mid-1990s: sliding window approaches
- Late 1990s: local features
- Early 2000s: parts-and-shape models
- Mid-2000s: bags of features

Bag-of-features models

Bag-of-features models

Object

Bag of ‘words’

History of ideas in recognition

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- Mid-1990s: sliding window approaches
- Late 1990s: local features
- Early 2000s: parts-and-shape models
- Mid-2000s: bags of features
- Present trends: data-driven methods, **deep learning**

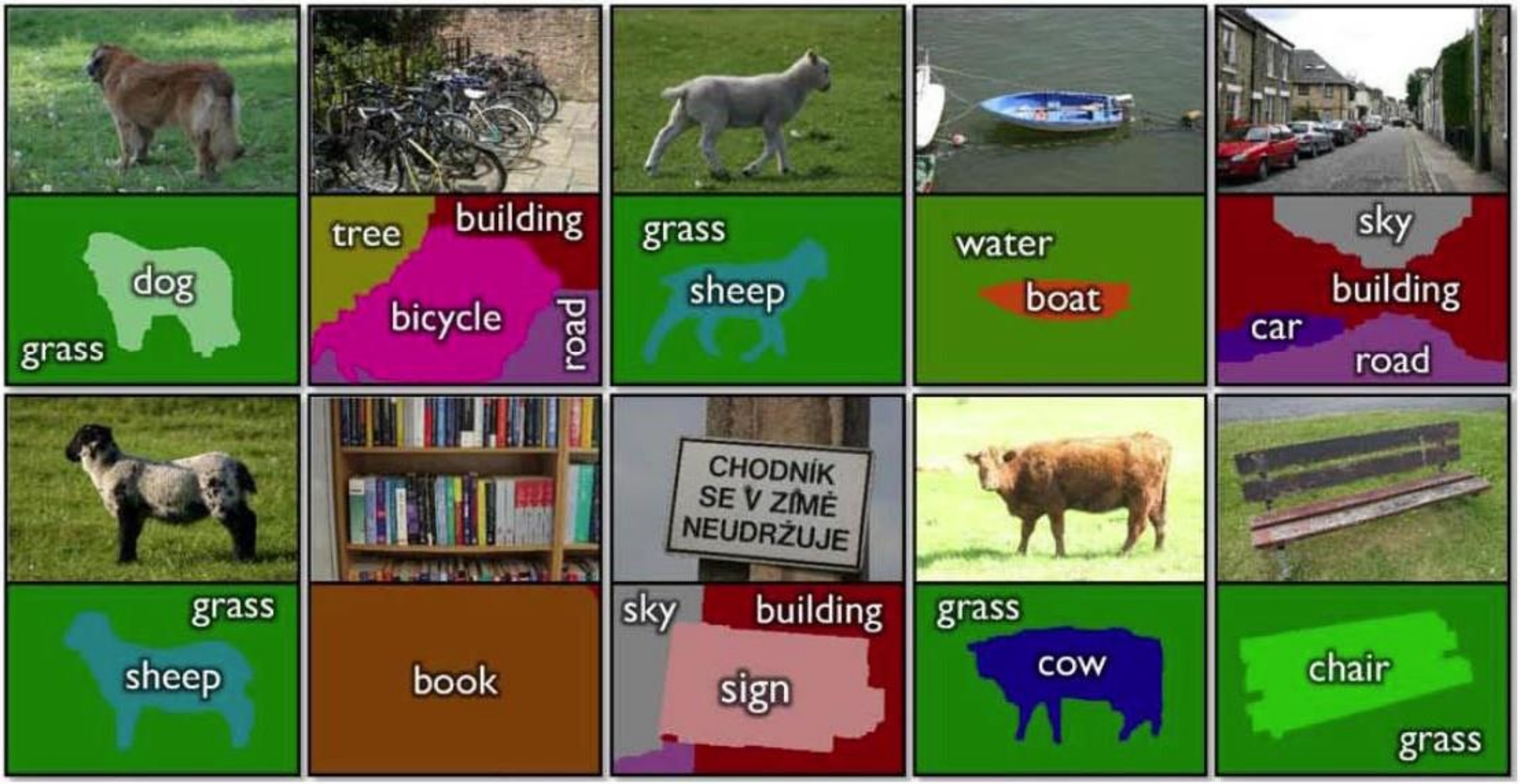
What Matters in Recognition?

- Learning Techniques
  - E.g. choice of classifier or inference method
- Representation
  - Low level: SIFT, HoG, GIST, edges
  - Mid level: Bag of words, sliding window, deformable model
  - High level: Contextual dependence
  - Deep features
- Data
  - More is always better
  - Annotation is the hard part

Types of Recognition

- Instance recognition
  - Recognizing a known object but in a new viewpoint, with clutter and occlusion
  - Location/Landmark Recognition
    - Recognize Paris, Rome, ... in photographs
    - Ideas from information retrieval
- Category recognition
  - Harder problem, even for humans
  - Bag of words, part-based, recognition and segmentation

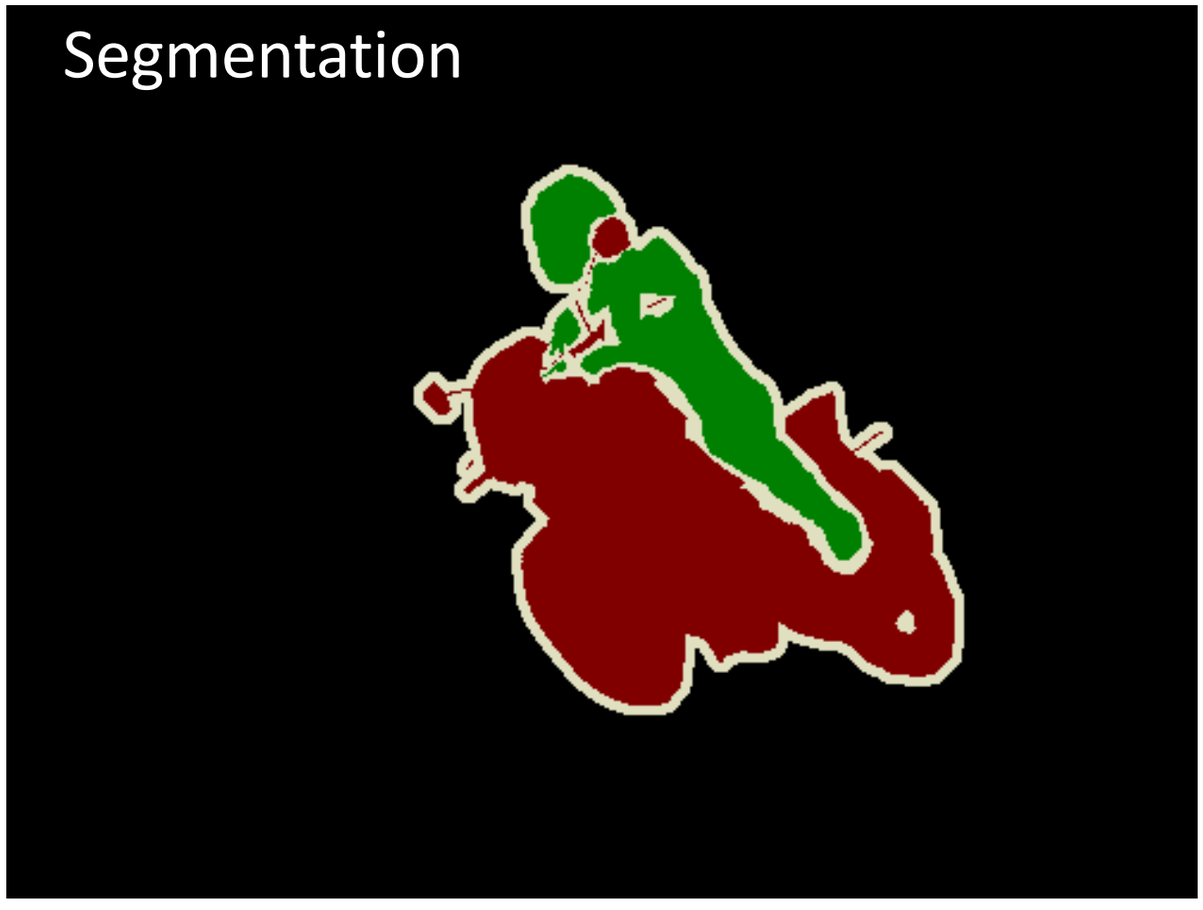
Simultaneous recognition and detection



PASCAL VOC 2005-2012

20 object classes **22,591 images**

Classification: person, motorcycle

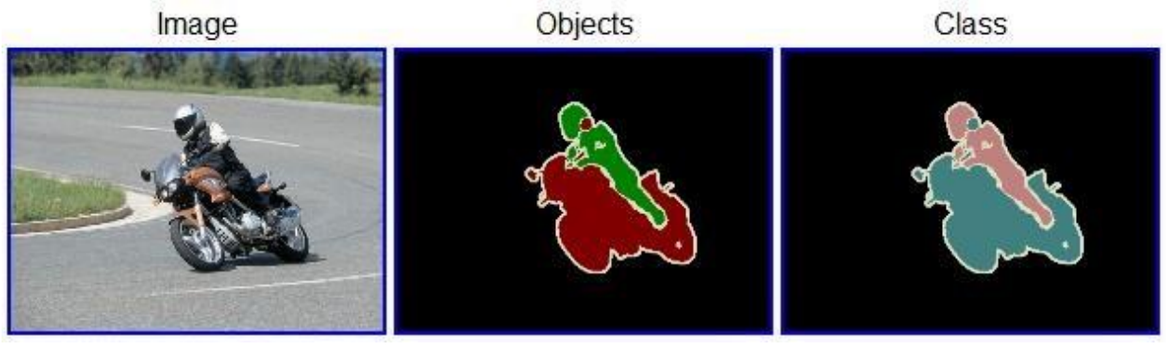


Action: riding bicycle

Everingham, Van Gool, Williams, Winn and Zisserman.
The PASCAL Visual Object Classes (VOC) Challenge. IJCV 2010.

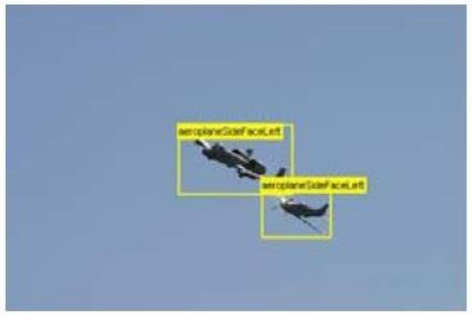
The PASCAL Visual Object Classes Challenge 2009 (VOC2009)

- 20 object categories (aeroplane to TV/monitor)
- Three (+2) challenges:
  - Classification challenge (is there an X in this image?)
  - Detection challenge (draw a box around every X)
  - Segmentation challenge (which class is each pixel?)



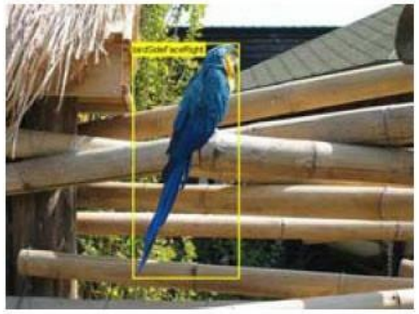
Examples

Aeroplane

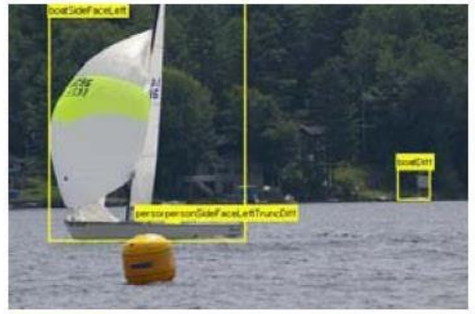


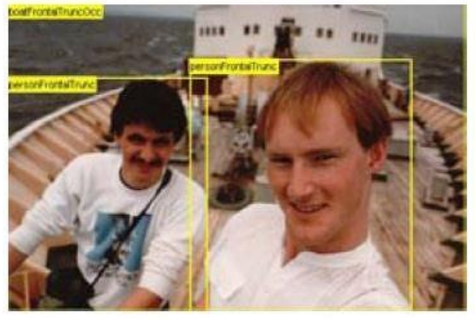
Bicycle

Bird



Boat





Bottle

Bus

Car

Cat

Chair

Cow

Classification Challenge

- Predict whether at least one object of a given class is present in an image

is there a cat?

Detection Challenge

- Predict the bounding boxes of all objects of a given class in an image (if any)

True Positives - Person

UoCTTI_LSVM-MDPM

MIZZOU_DEF-HOG-LBP

NECUIUC_CLS-DTCT

False Positives - Person

UoCTTI_LSVM-MDPM



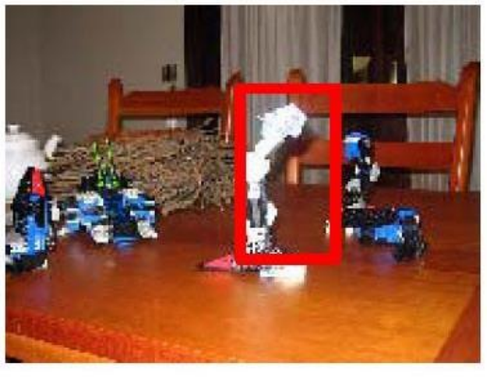


MIZZOU_DEF-HOG-LBP





NECUIUC_CLS-DTCT


“Near Misses” - Person

UoCTTI_LSVM-MDPM

MIZZOU_DEF-HOG-LBP

NECUIUC_CLS-DTCT

True Positives - Bicycle

UoCTTI_LSVM-MDPM





OXFORD_MKL





NECUIUC_CLS-DTCT





False Positives - Bicycle

UoCTTI_LSVM-MDPM





OXFORD_MKL





NECUIUC_CLS-DTCT





Where to from here?

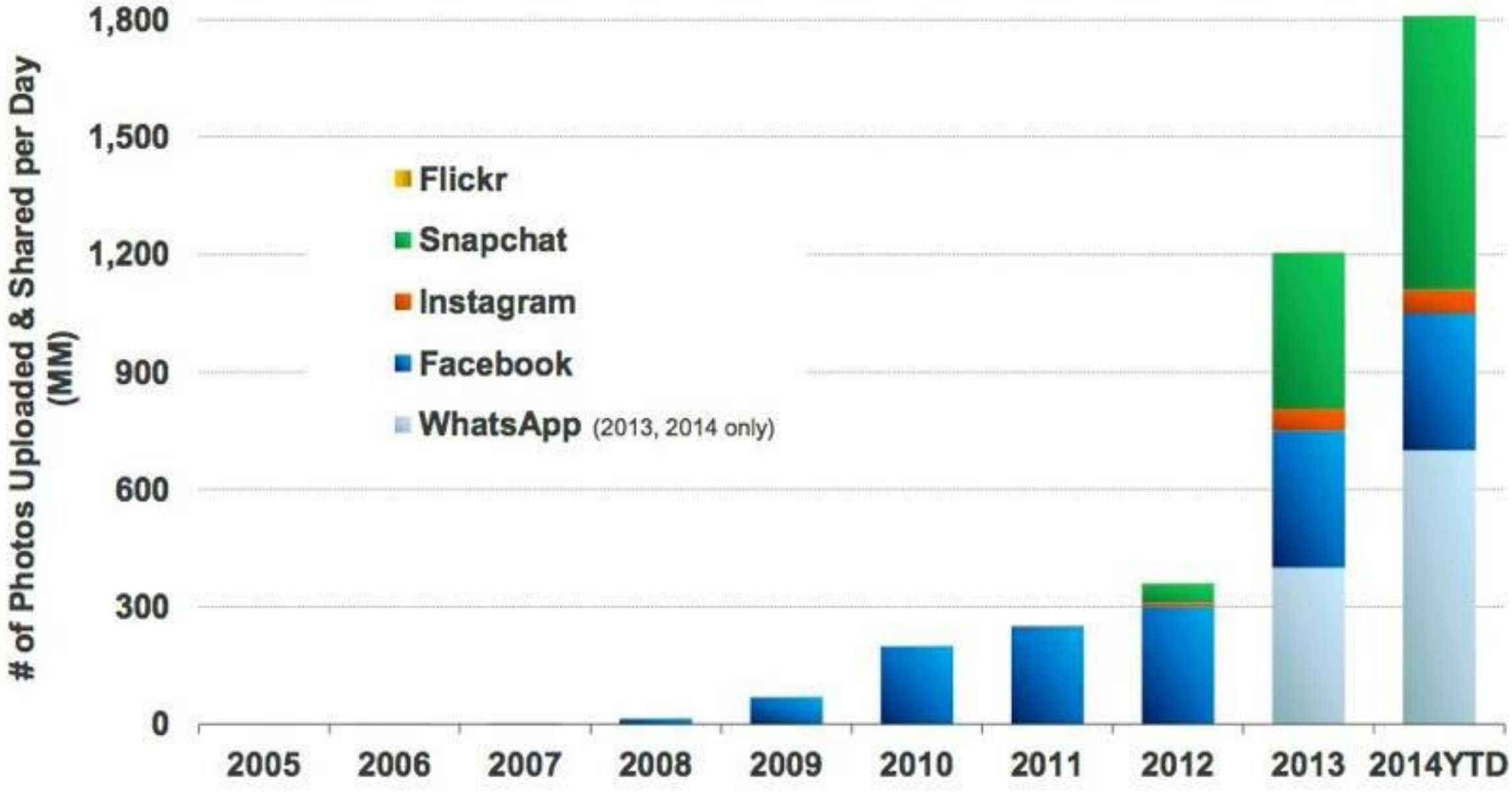
- Scene Understanding
    - Big data – lots of images
    - Crowd-sourcing – lots of people
    - Deep Learning – lots of compute

24 Hrs in Photos

http://www.kesselskramer.com/exhibitions/24-hrs-of-photos

installation by Erik Kessels

Daily Number of Photos Uploaded & Shared on Select Platforms, 2005 – 2014YTD



@KPCB

Source: KPCB estimates based on publicly disclosed company data, 2014 YTD data per latest as of 5/14.

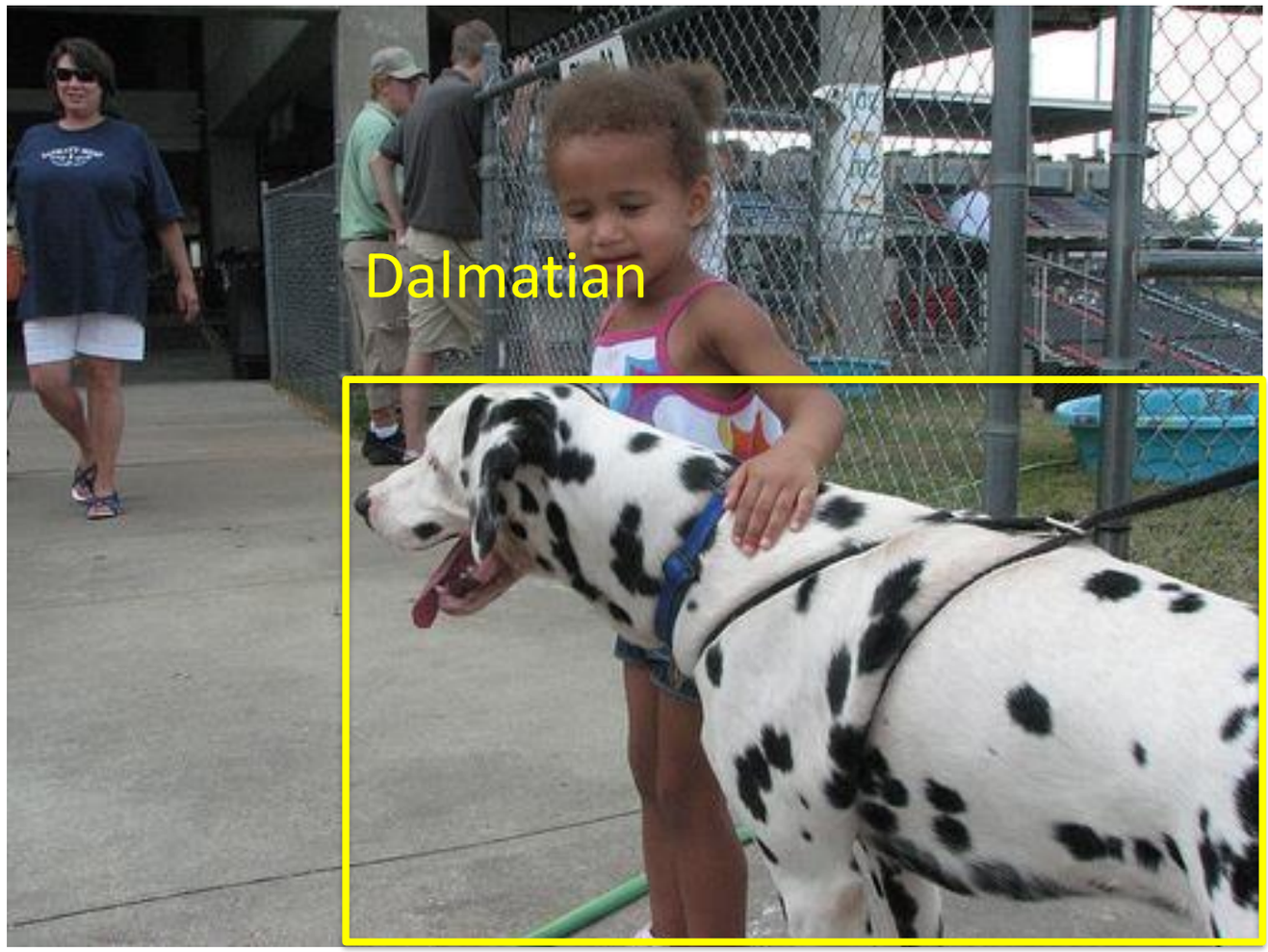
Data Sets

- ImageNet
  - Huge, Crowdsourced, Hierarchical, *Iconic* objects
- PASCAL VOC
  - *Not* Crowdsourced, bounding boxes, 20 categories
- SUN Scene Database, Places
  - *Not* Crowdsourced, 397 (or 720) scene categories
- LabelMe (Overlaps with SUN)
  - Sort of Crowdsourced, Segmentations, Open ended
- SUN *Attribute* database (Overlaps with SUN)
  - Crowdsourced, 102 attributes for every scene
- OpenSurfaces
  - Crowdsourced, materials
- Microsoft COCO
  - Crowdsourced, large-scale objects

IMAGENET Large Scale Visual Recognition Challenge (ILSVRC) 2010-2012

~~20 object classes~~ ~~22,591 images~~

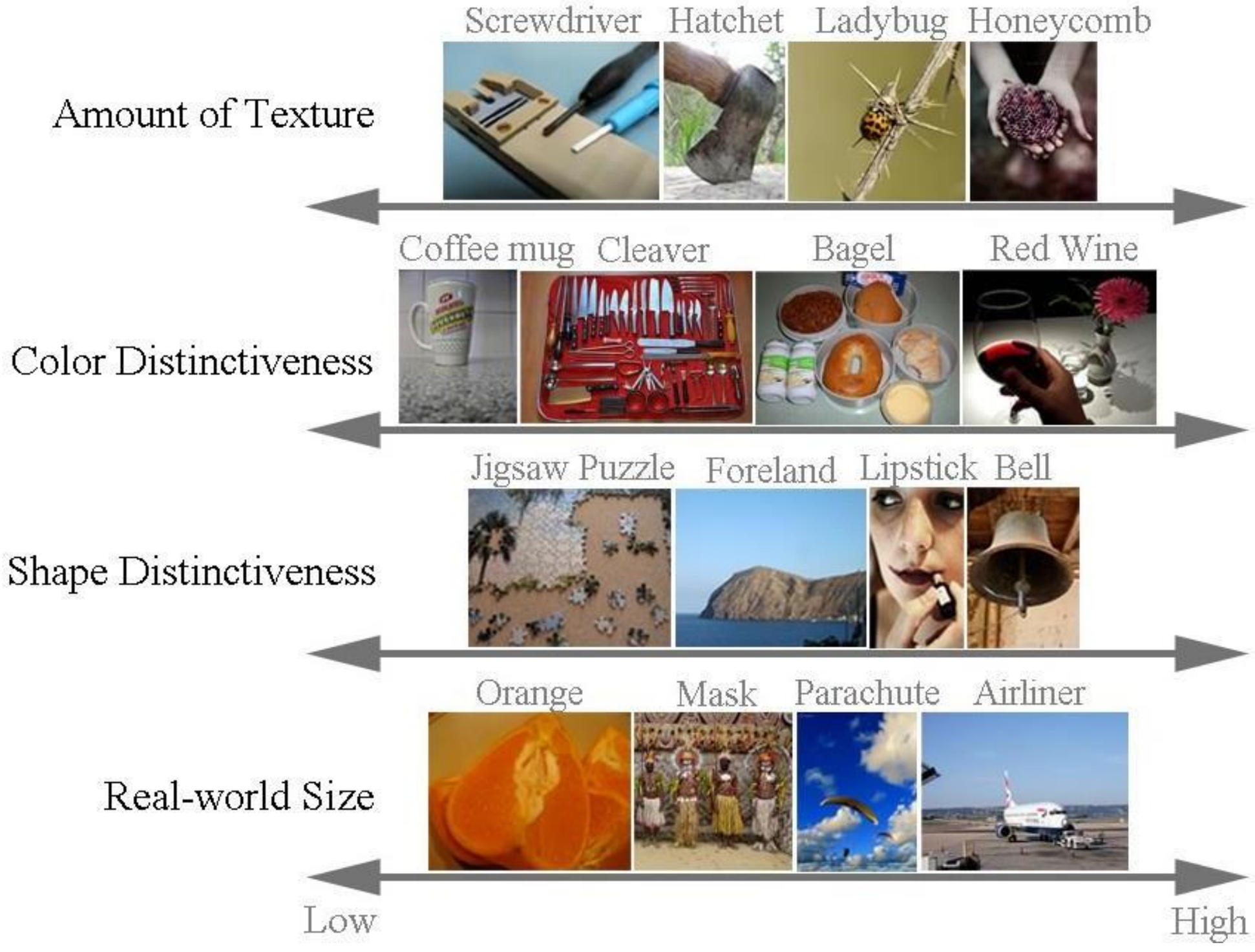
1000 object classes **1,431,167 images**



http://image-net.org/challenges/LSVRC/{2010,2011,2012}

Variety of object classes in ILSVRC

Variety of object classes in ILSVRC




7 APR 2007

Next step: Infer object properties

Can I **poke with it**?

Is it **alive**?

What **shape** is it?

Does it have a **tail**?

Can I **put stuff in it**?

Is it **soft**?

Will it **blend**?

What are attributes?

What do we want to know about this object?

Object recognition expert: “Dog”

What are attributes?

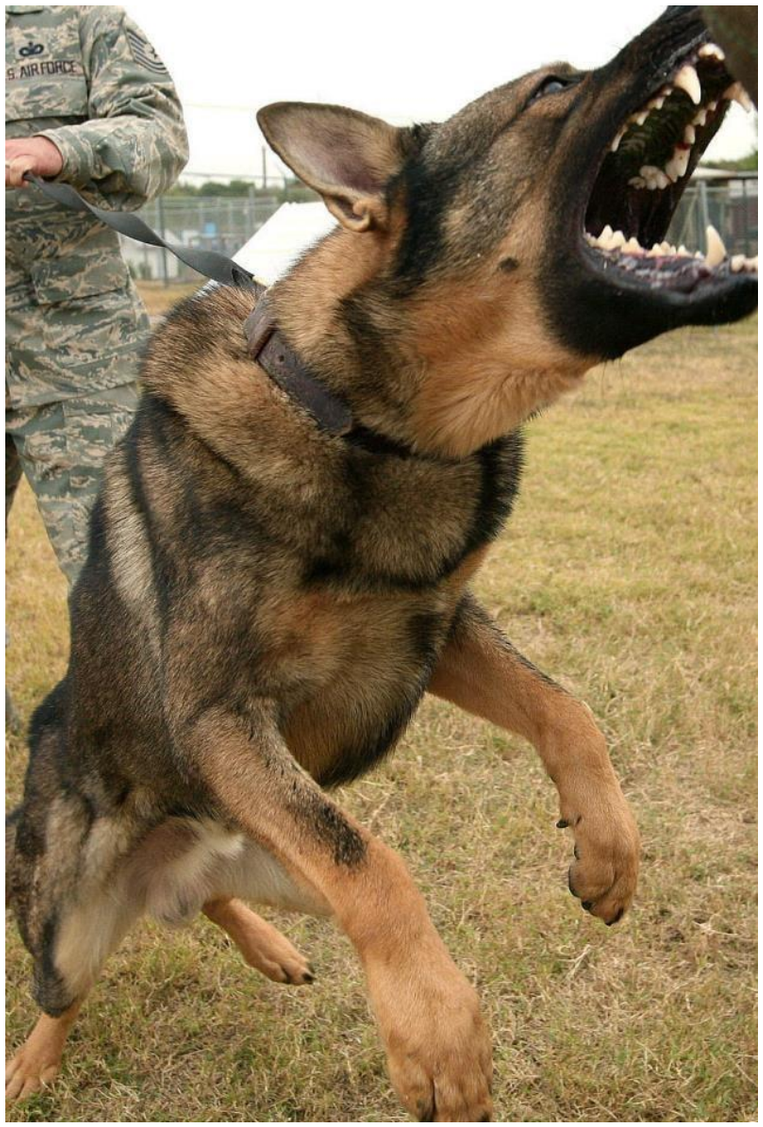
What do we want to know about this object?

Object recognition expert:
“Dog”

Person in the Scene:
“Big pointy teeth”, “Can move fast”, “Looks angry”

Why infer properties

1. We want detailed information about objects



"Dog"
vs.
"Large, angry animal with pointy teeth"

Why infer properties

2. We want to be able to infer something about unfamiliar objects

Familiar Objects

New Object

Why infer properties

2. We want to be able to infer something about unfamiliar objects

If we can infer properties…

Familiar Objects

Has Stripes
Has Ears
Has Eyes
….

Has Four Legs
Has Mane
Has Tail
Has Snout
….

Brown
Muscular
Has Snout
….

New Object

Has Stripes (like cat)
Has Mane and Tail (like horse)
Has Snout (like horse and dog)

Why infer properties

3. We want to make comparisons between objects or categories

What is unusual about this dog?

What is the difference between horses and zebras?

References

Basic reading:

- Szeliski, Chapter 14.