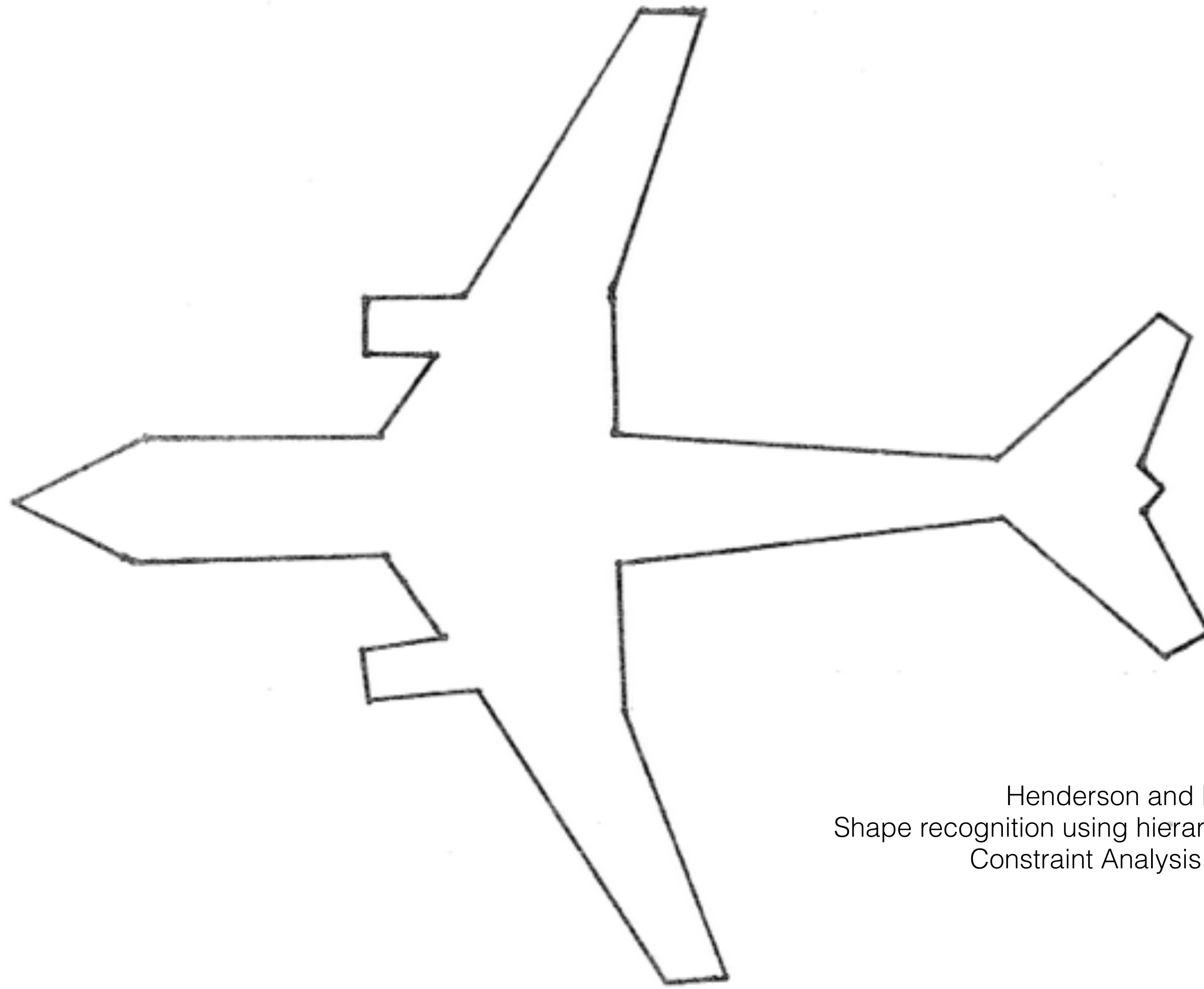


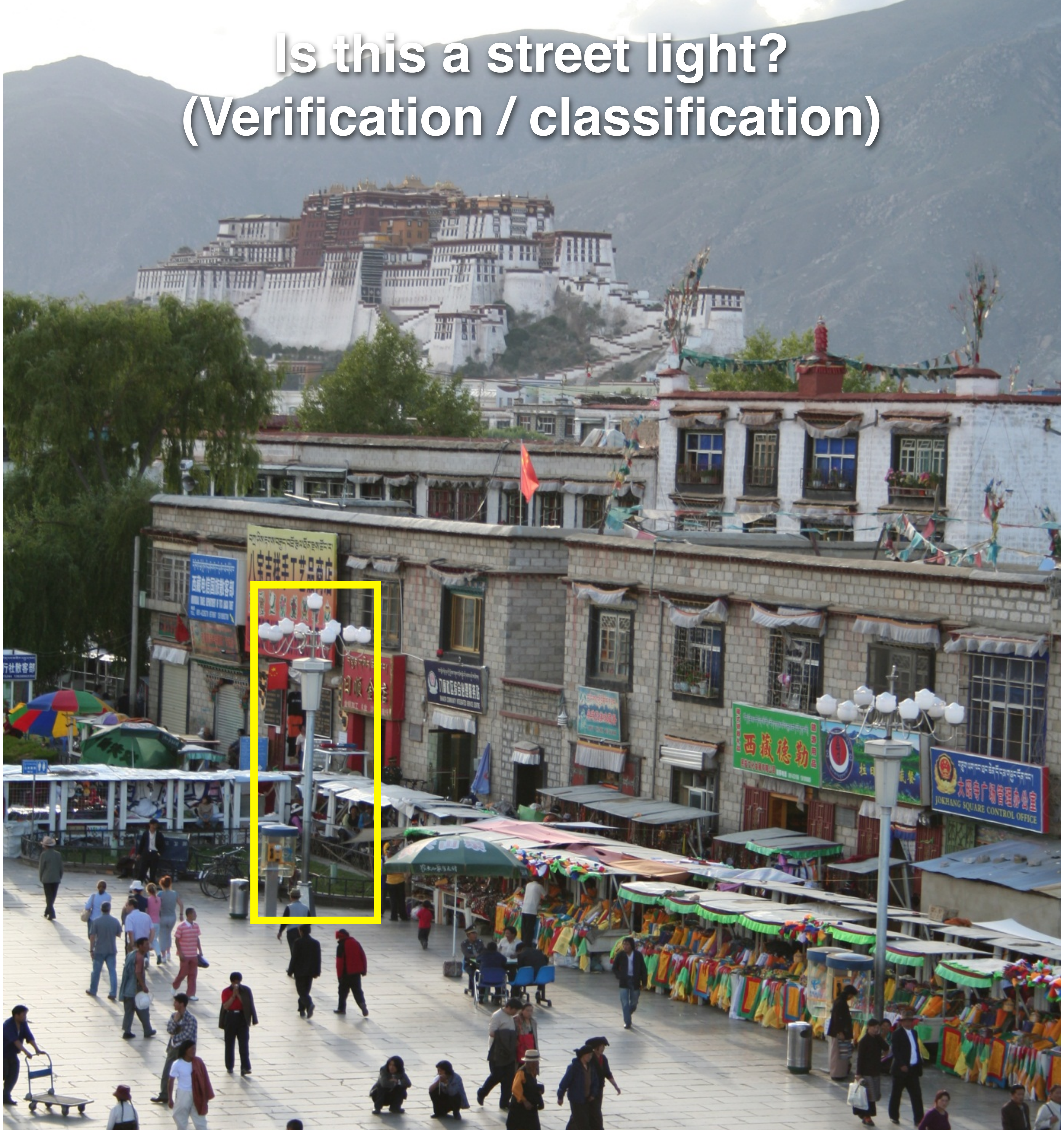
Henderson and Davis.
Shape recognition using hierarchical
Constraint Analysis. 1979

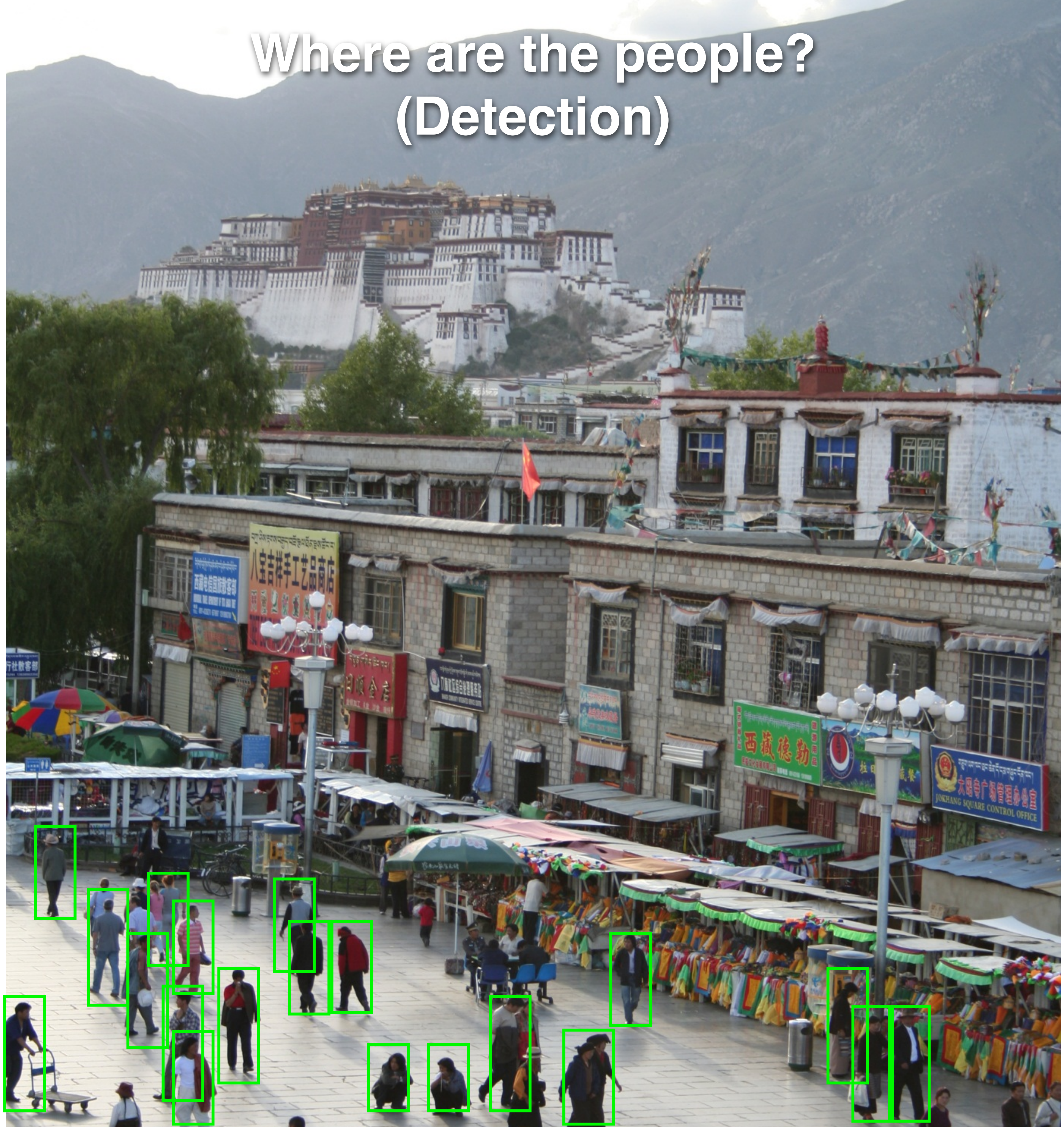
Object Recognition

16-385 Computer Vision
Carnegie Mellon University (Kris Kitani)

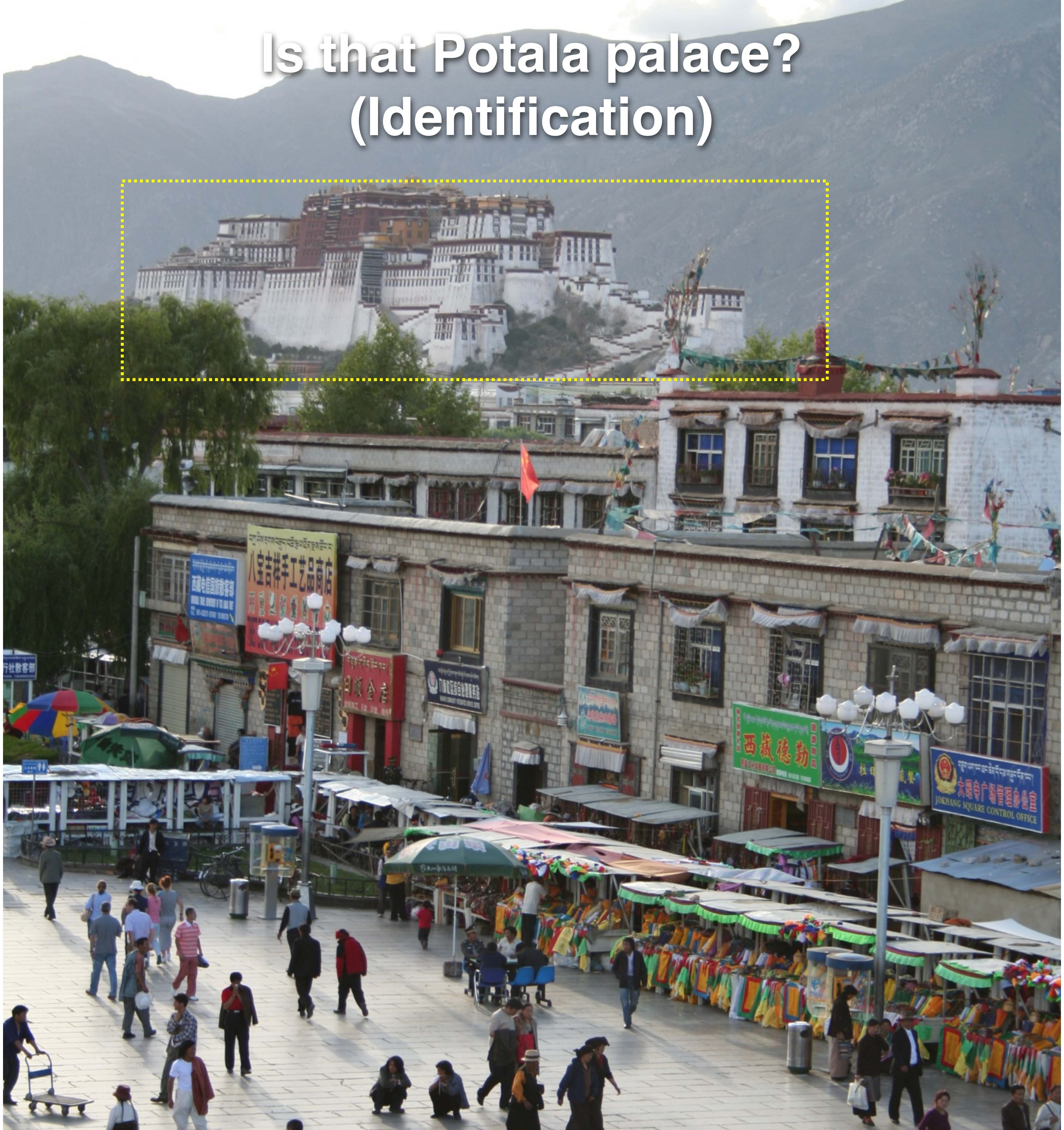
What do we mean by
‘object recognition’?

Is this a street light?
(Verification / classification)

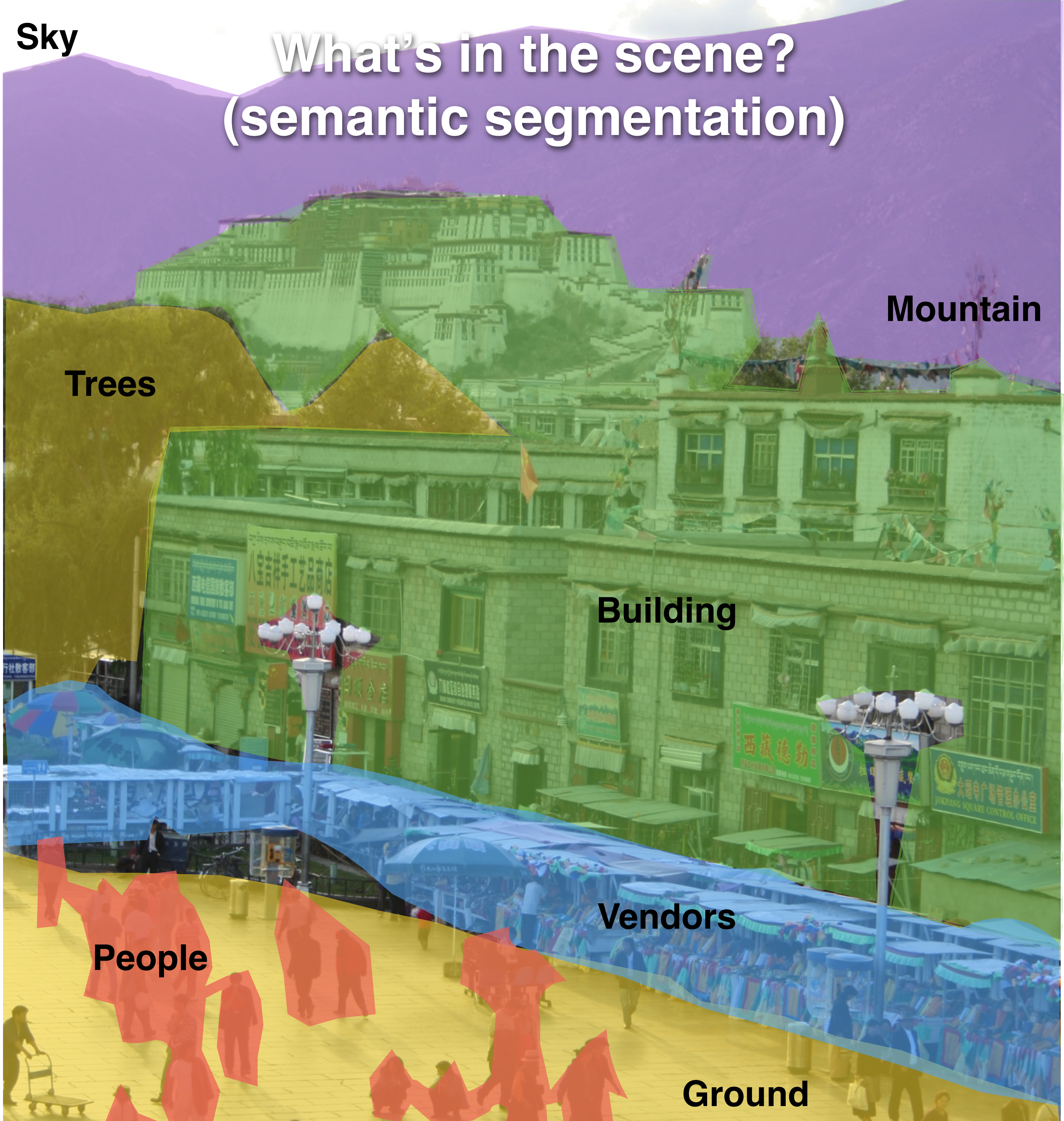


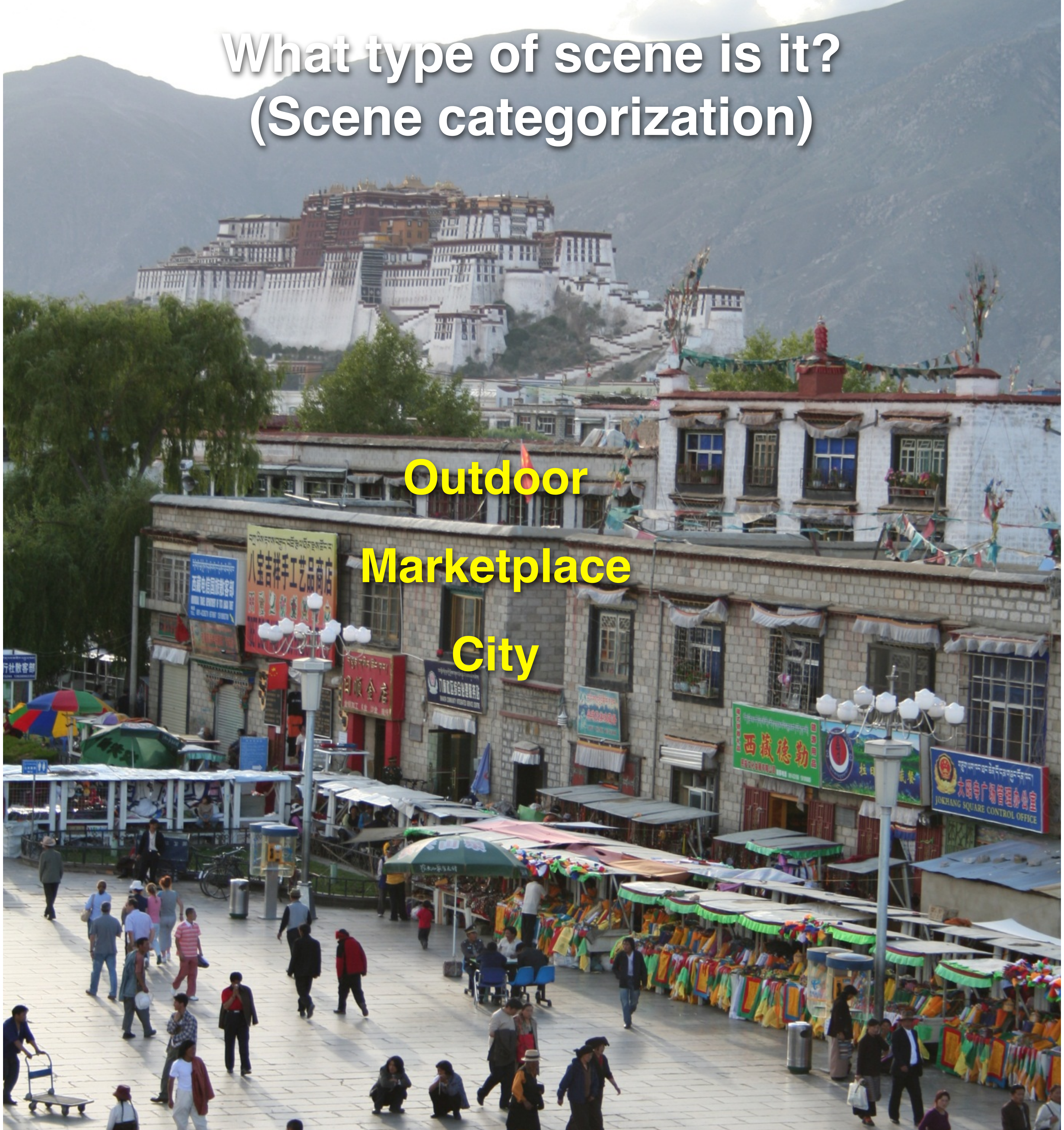

Where are the people?
(Detection)
八宝吉祥手工艺品商店
西藏德勒
大昭寺广场管理办公室
JOKHANG SQUARE CONTROL OFFICE

Is that Potala palace? (Identification)



What's in the scene? (semantic segmentation)




What type of scene is it?
(Scene categorization)
Outdoor
Marketplace
City

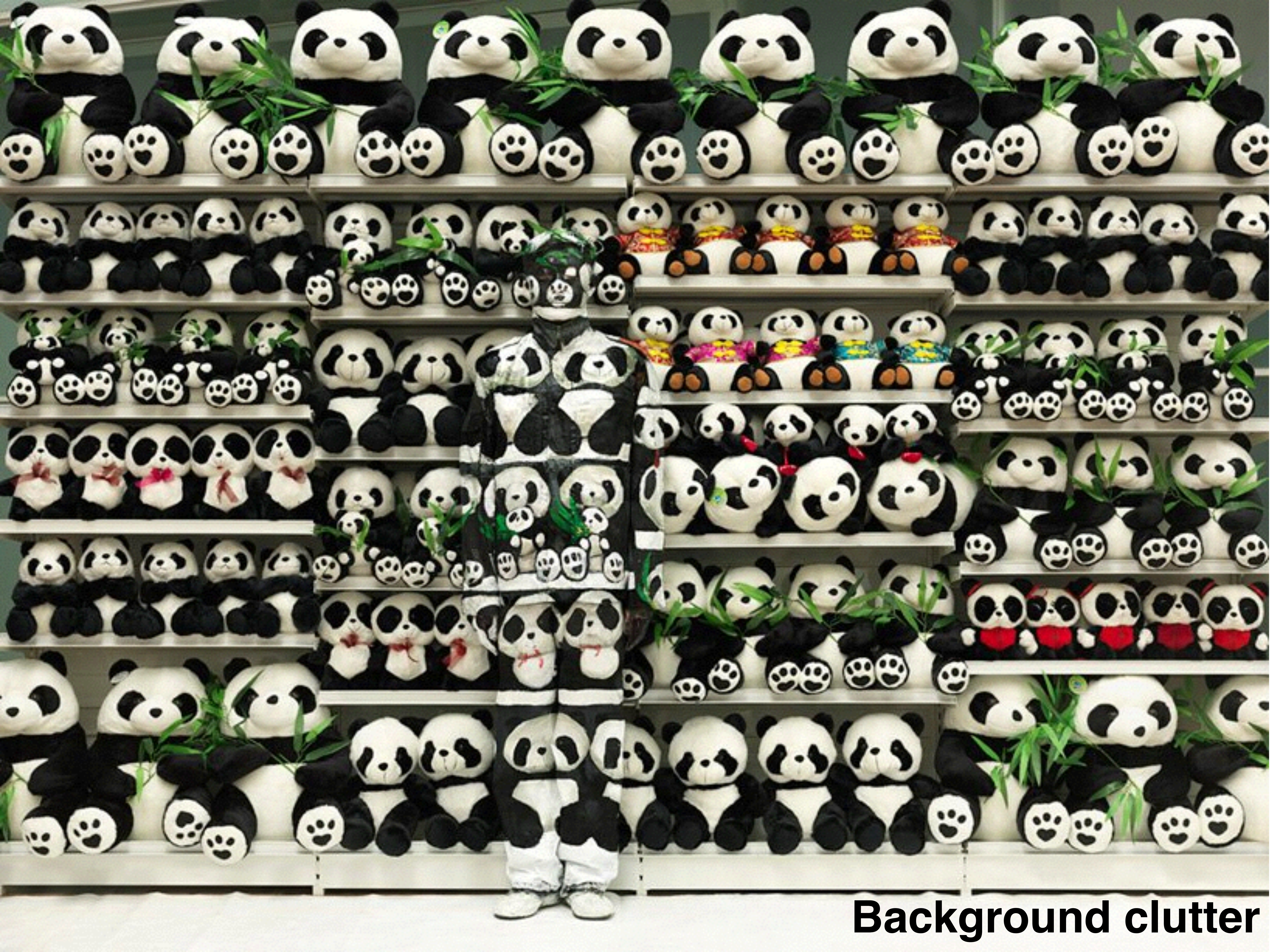
Challenges

(Object Recognition)

Viewpoint variation

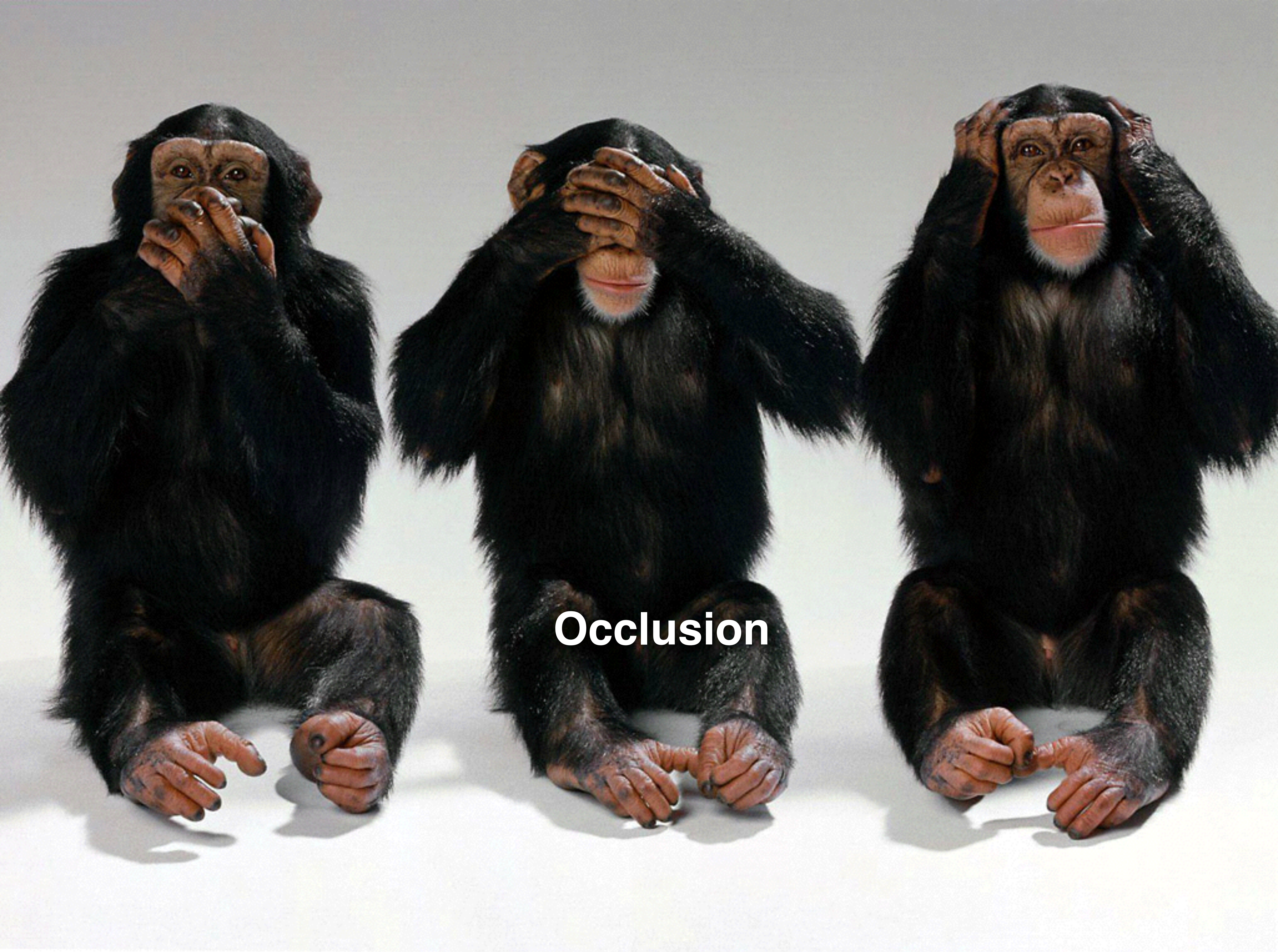

Illumination variation

Scale variation



Background clutter

Deformation


Occlusion

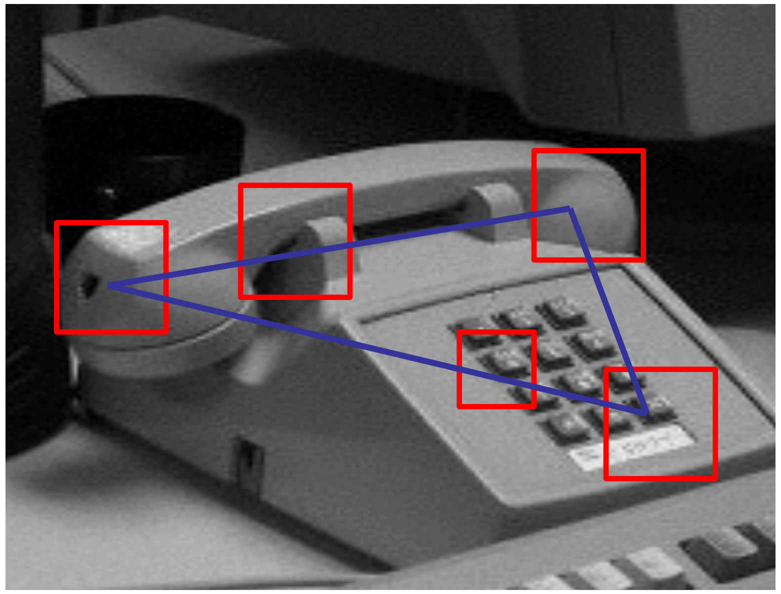




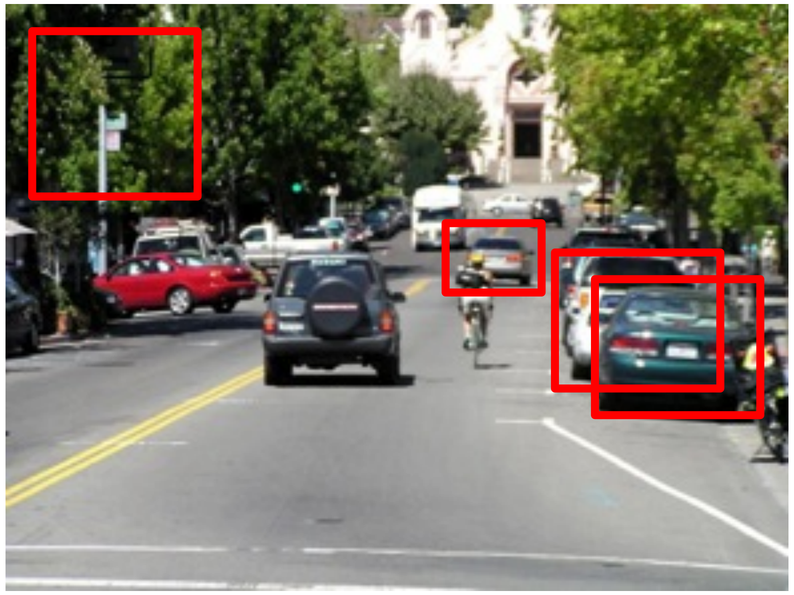
Intra-class variation

Common approaches

Feature
Matching



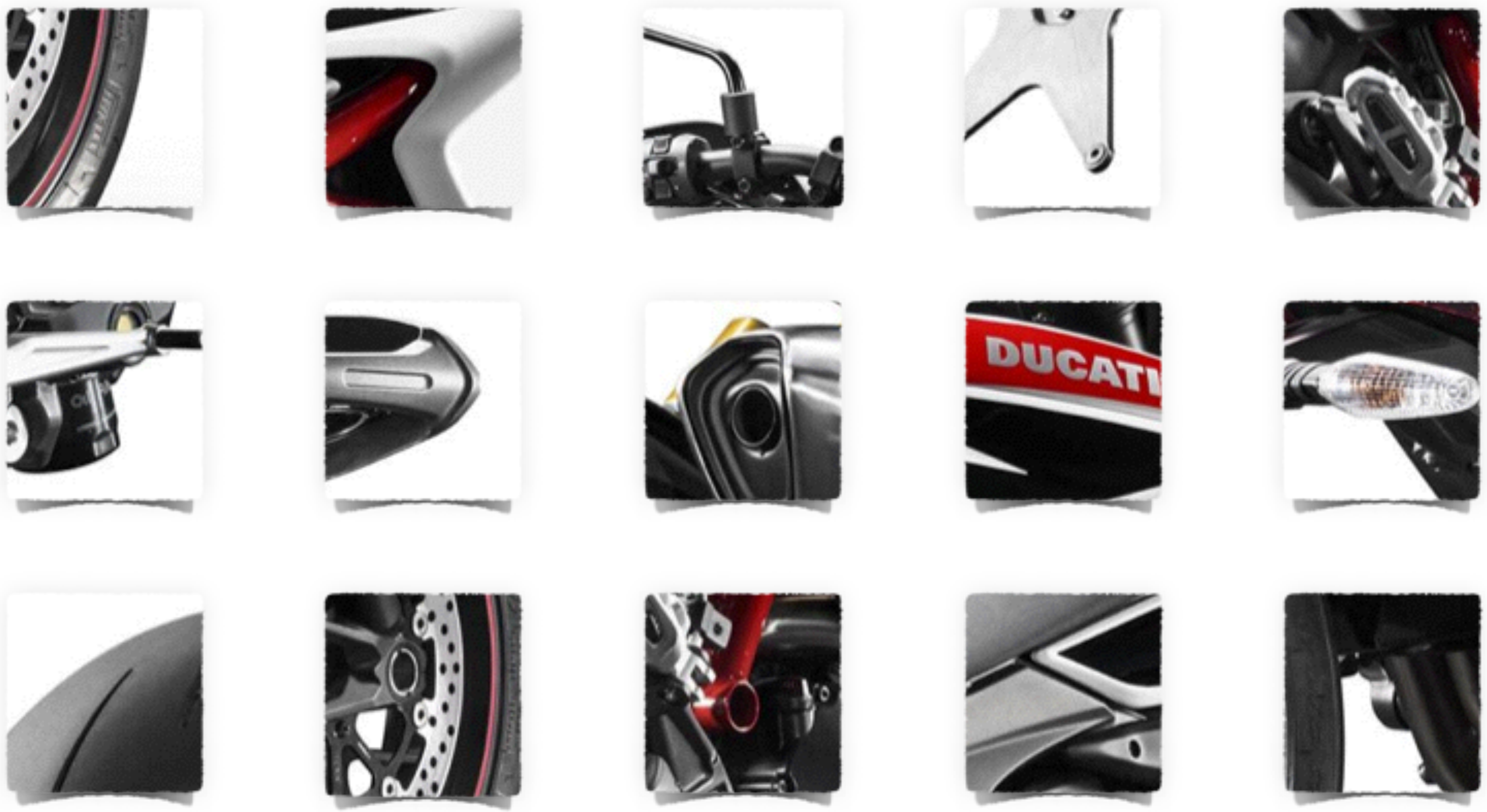
Spatial
reasoning



Window
classification

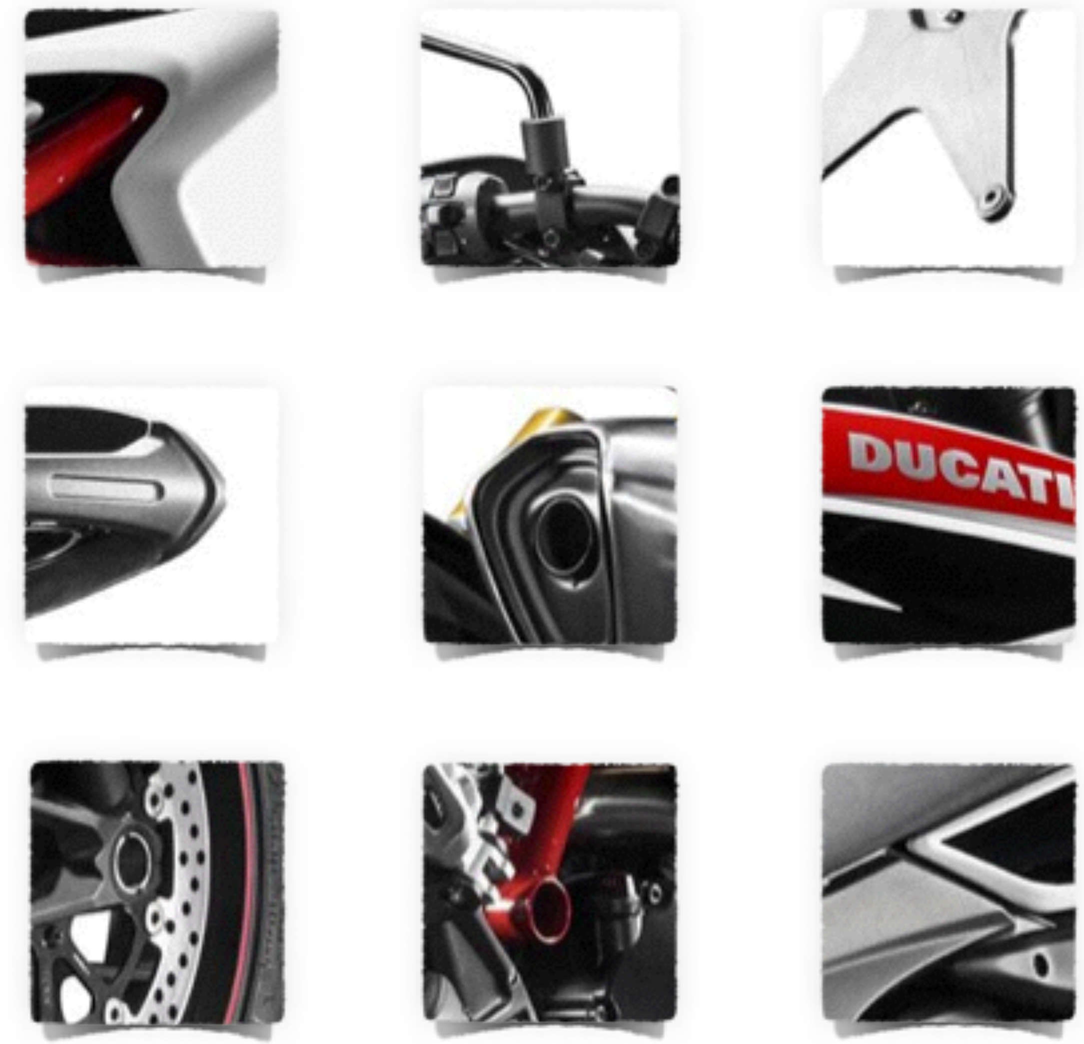
Feature matching

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?

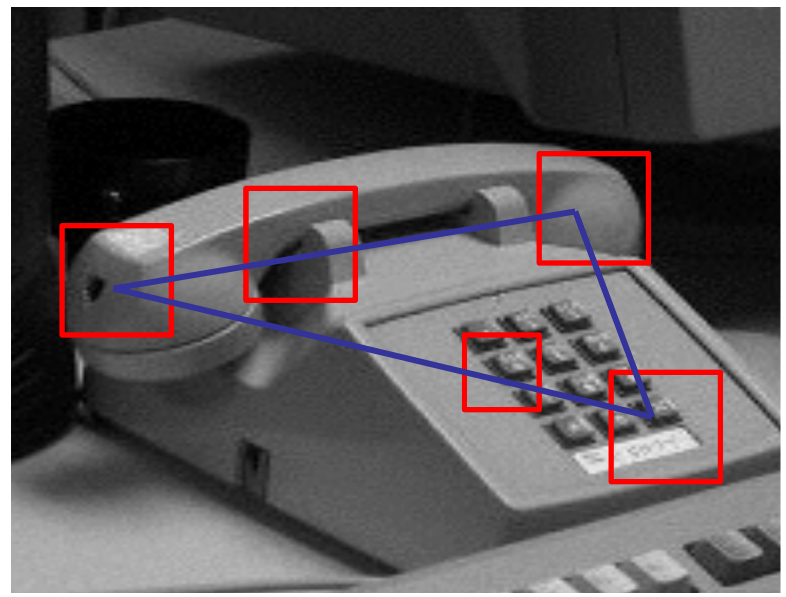
Pros

- Simple
- Efficient algorithms
- Robust to deformations

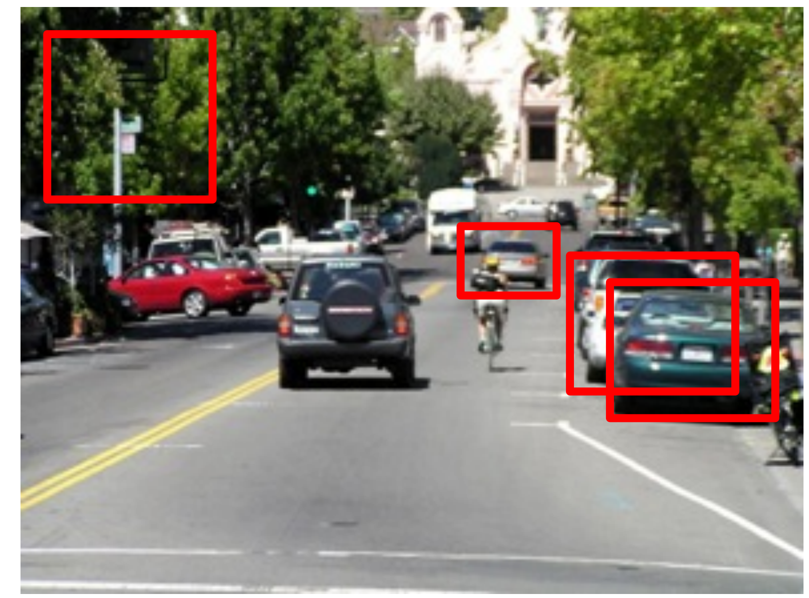
Cons

- No spatial reasoning

Feature Matching



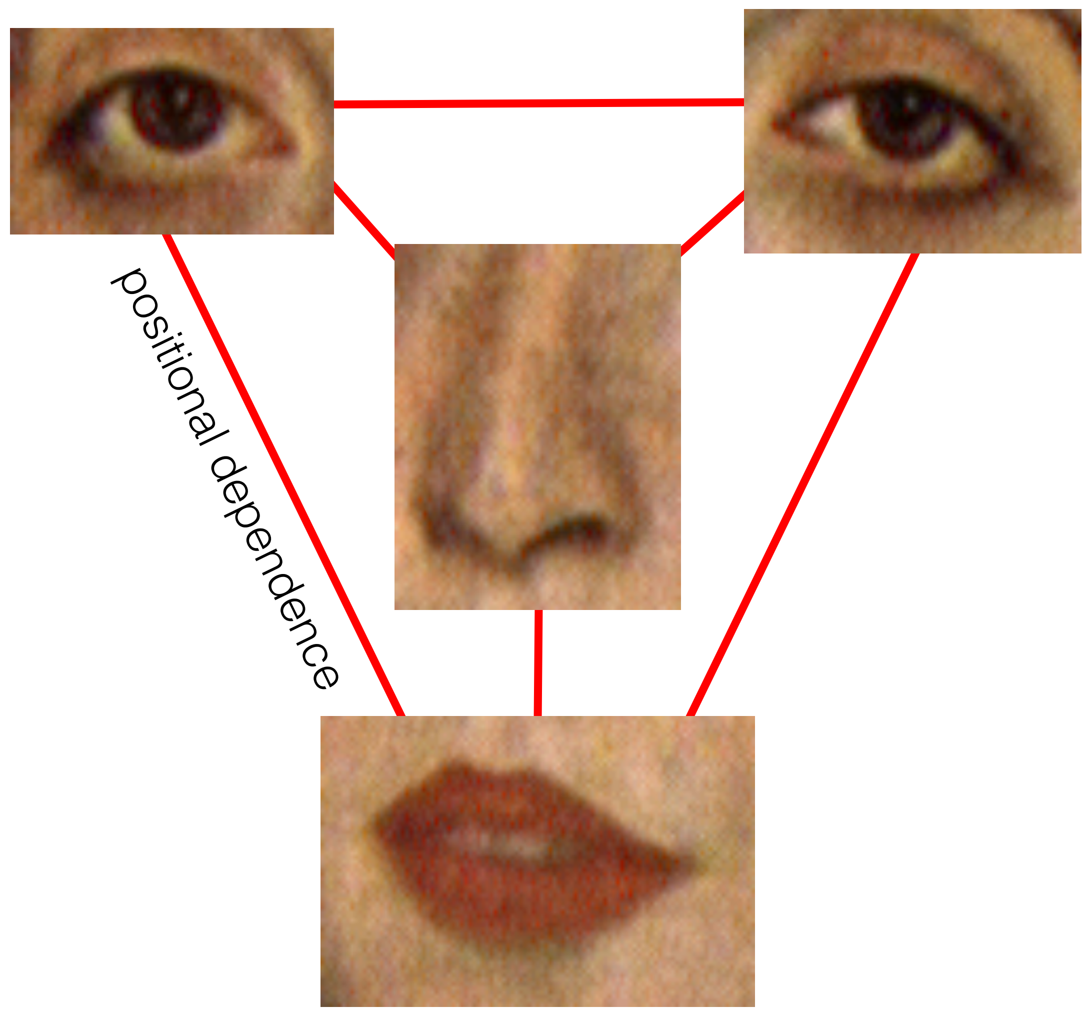
Spatial reasoning



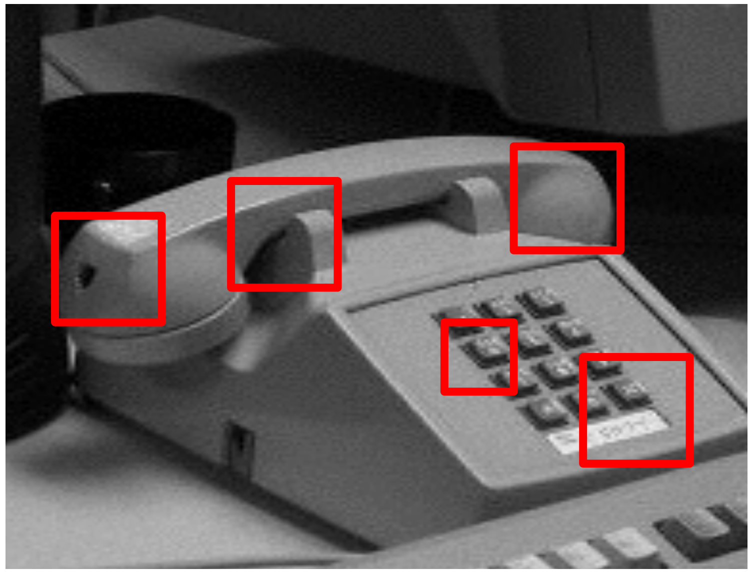
Window classification

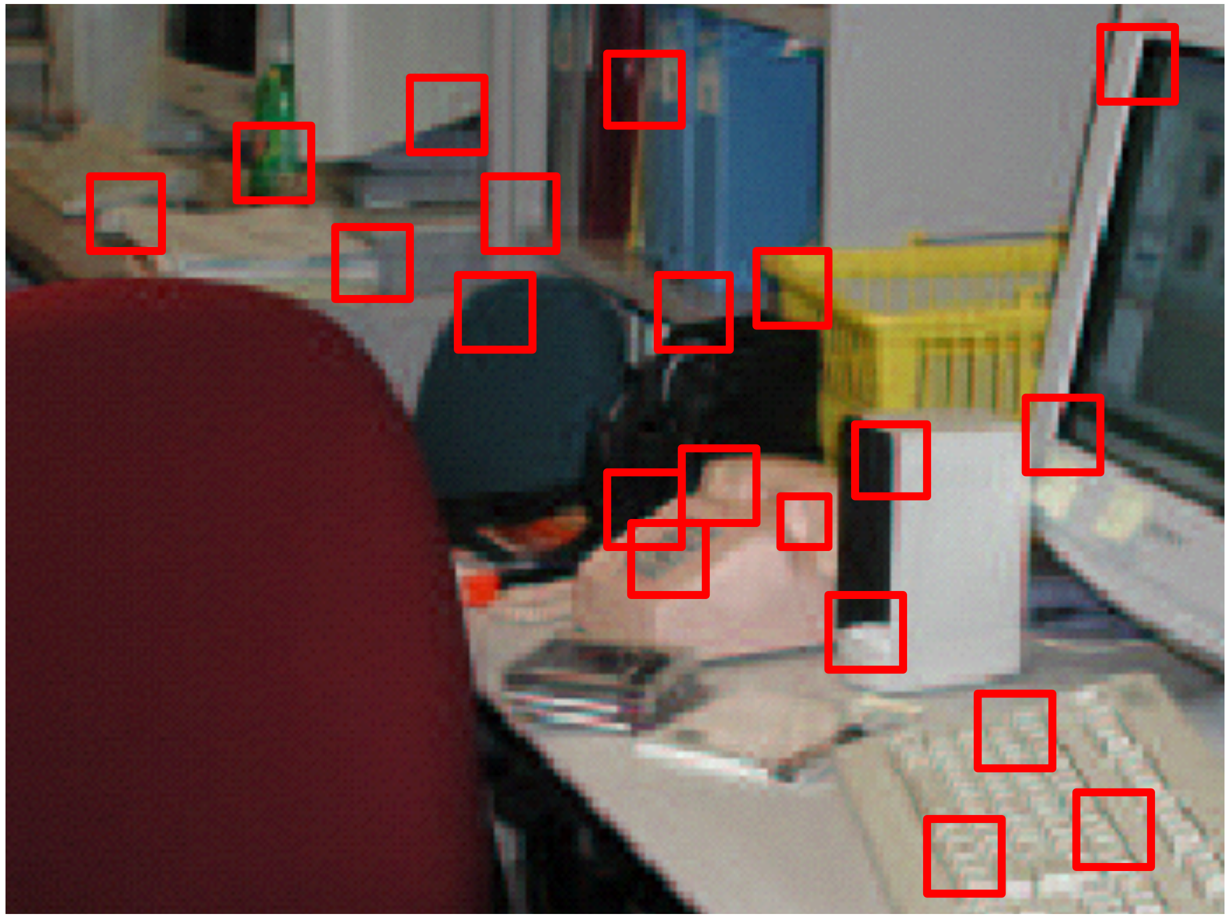
Spatial reasoning

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

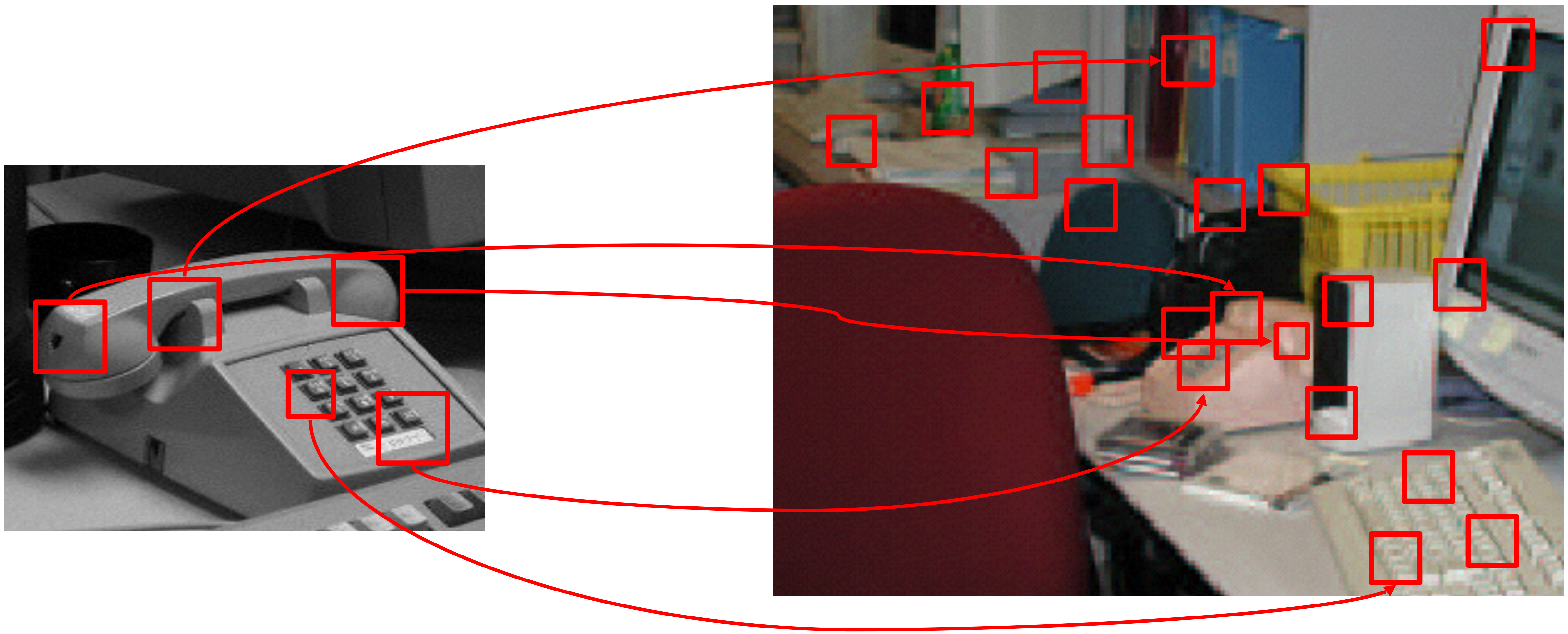


Many parts, many dependencies!

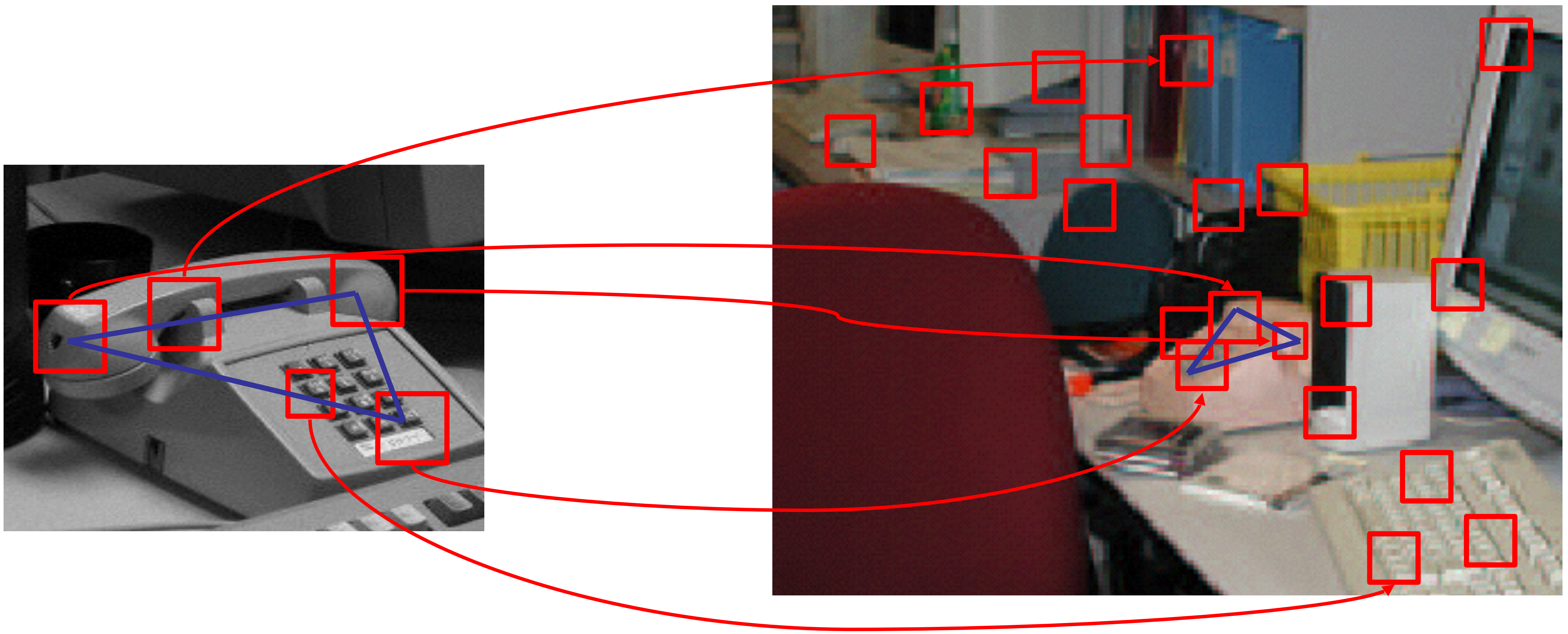




1. Extract features 2. Match features 3. Spatial verification



1. Extract features 2. Match features 3. Spatial verification

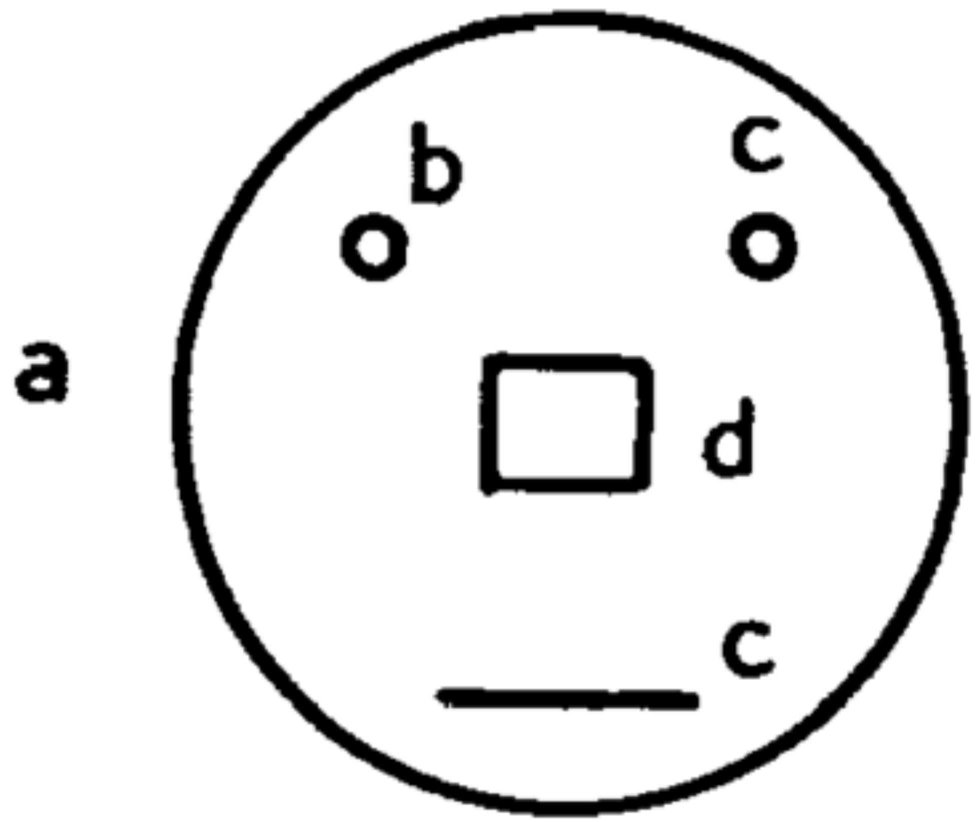


1. Extract features

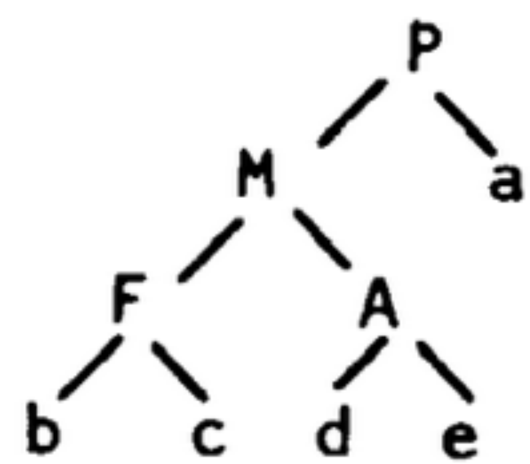
2. Match features

3. Spatial verification

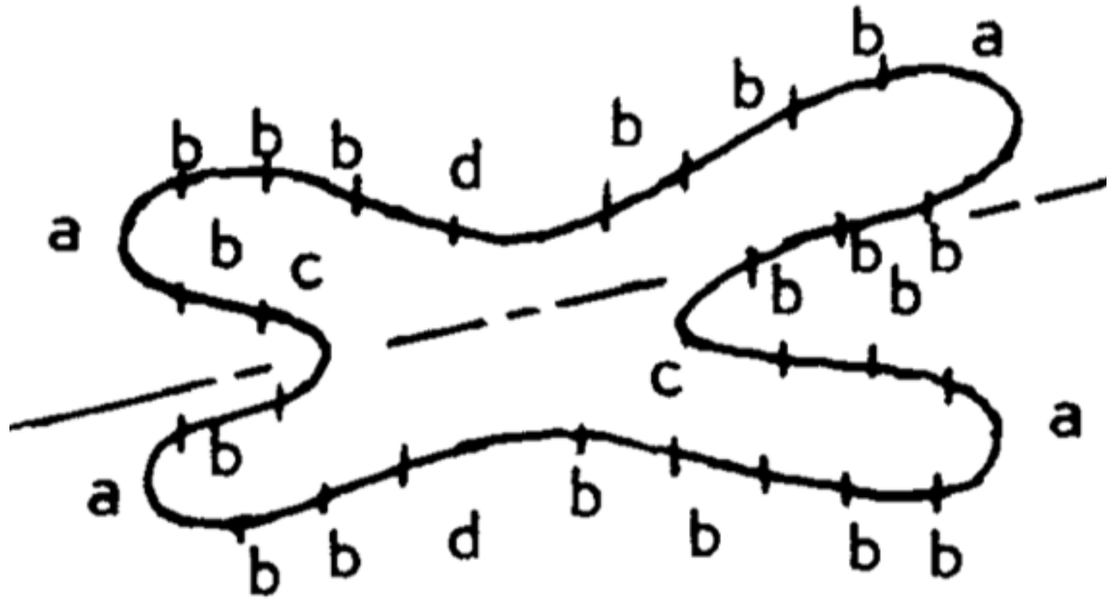
Fu and Booth. Grammatical Inference. 1975



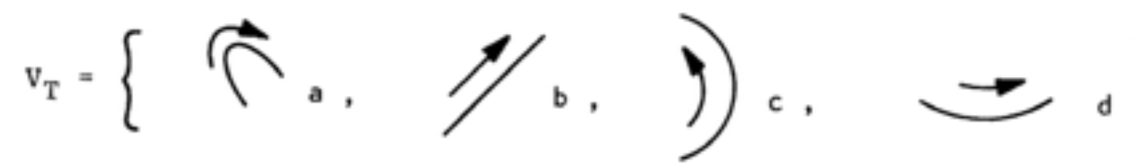
Scene



Structural (grammatical) description

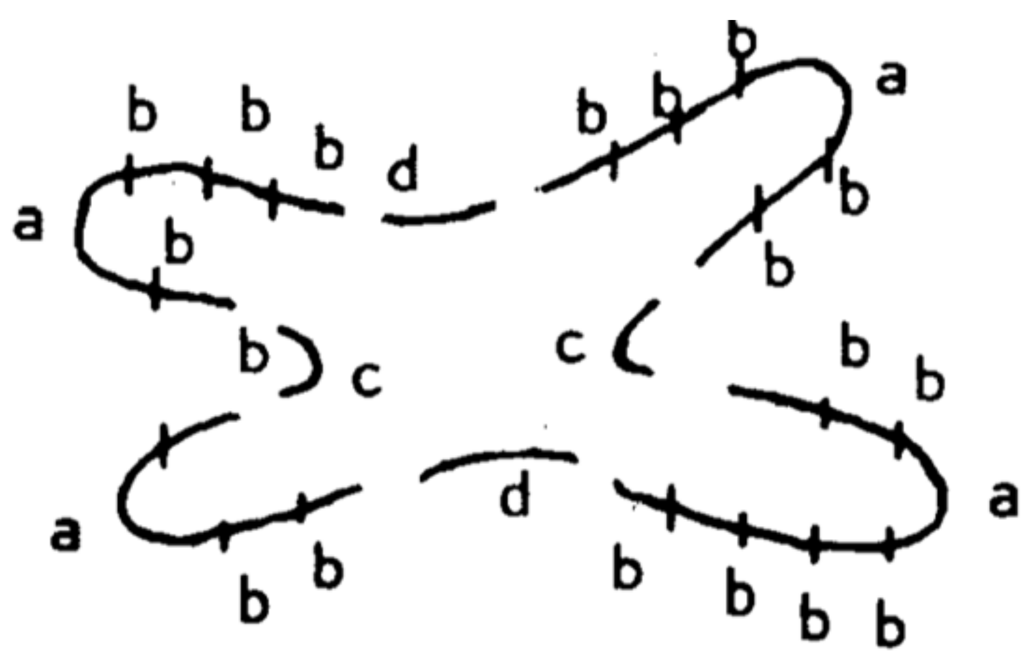


Coded Chromosome



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

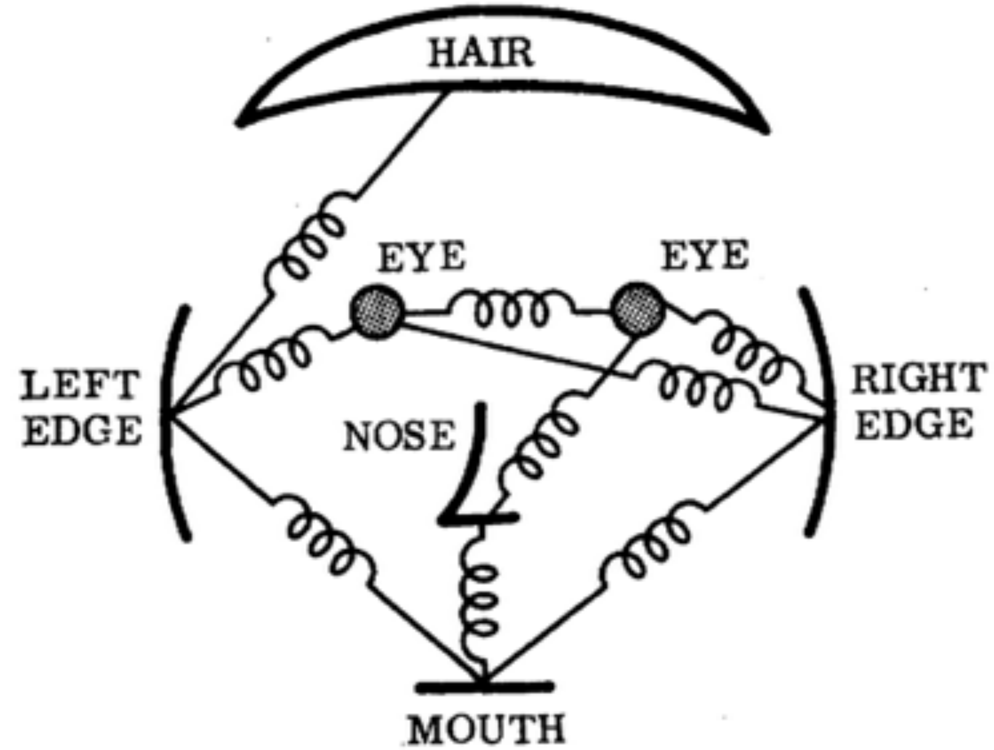
MARTIN A. FISCHLER AND ROBERT A. ELSCHLAGER

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.

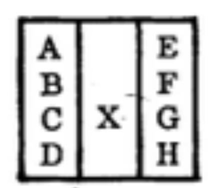
We offer a combined descriptive scheme and decision metric which is general, intuitively satisfying, and which has led to promising experimental results. We also present an algorithm which takes the above descriptions, together with a matrix representing the intensities of the actual photograph, and 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



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.

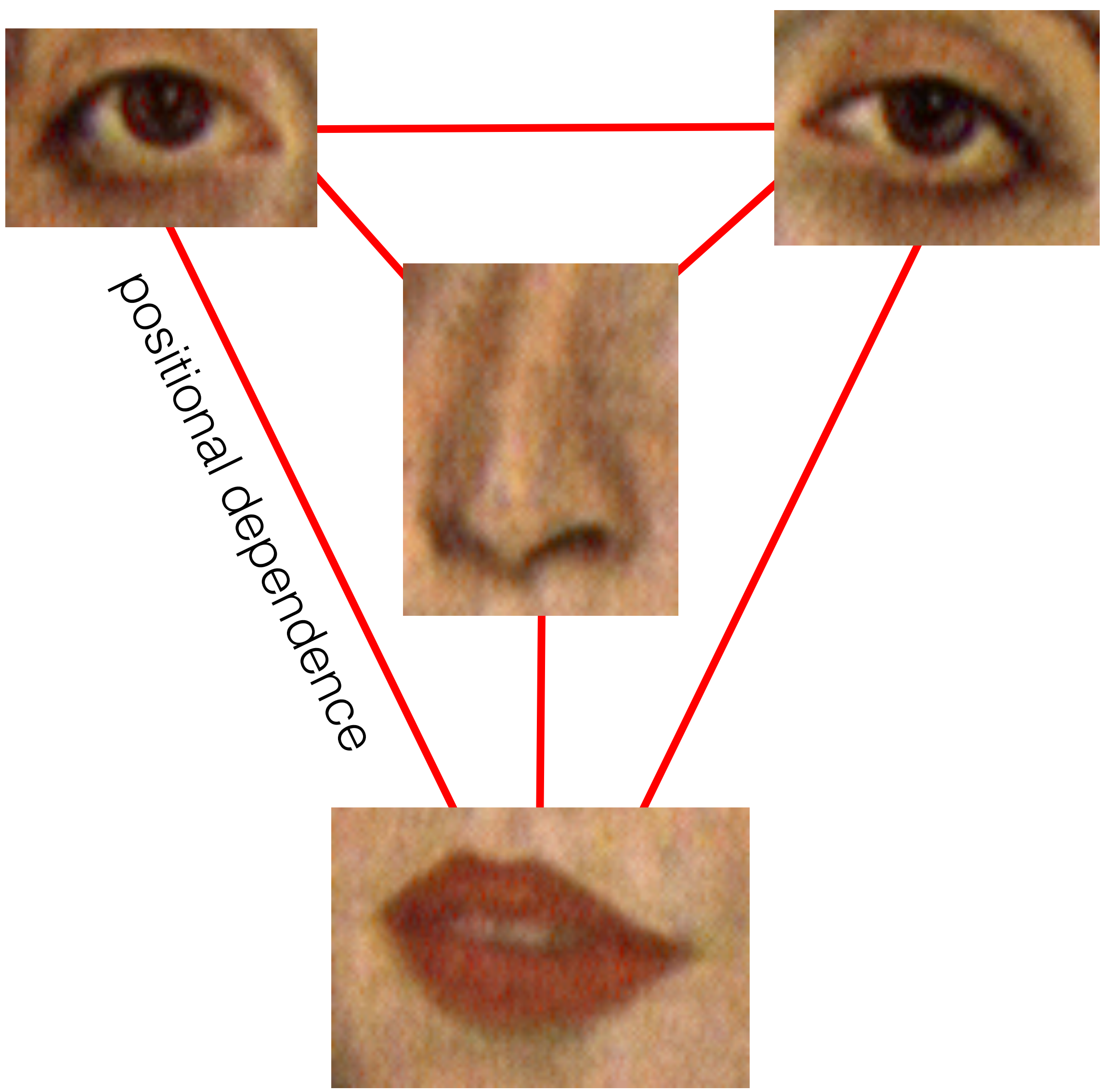
A more probabilistic approach

- L $= \{l_1,..,l_M\}$ N^M possible combinations of locations
- Most likely location L is found by maximizing:

$$P(L|I) \propto P(I|L)P(L)$$

- P(I|L): How likely is it to observe image I given that the P parts are at locations L
- Evaluated by comparing the model of each part a_i with the image content at l_i (locations are unknown)
- P(L): spatial prior controls the geometric configuration of the parts. How to represent P(L)??

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
Intractable for moderate number of parts

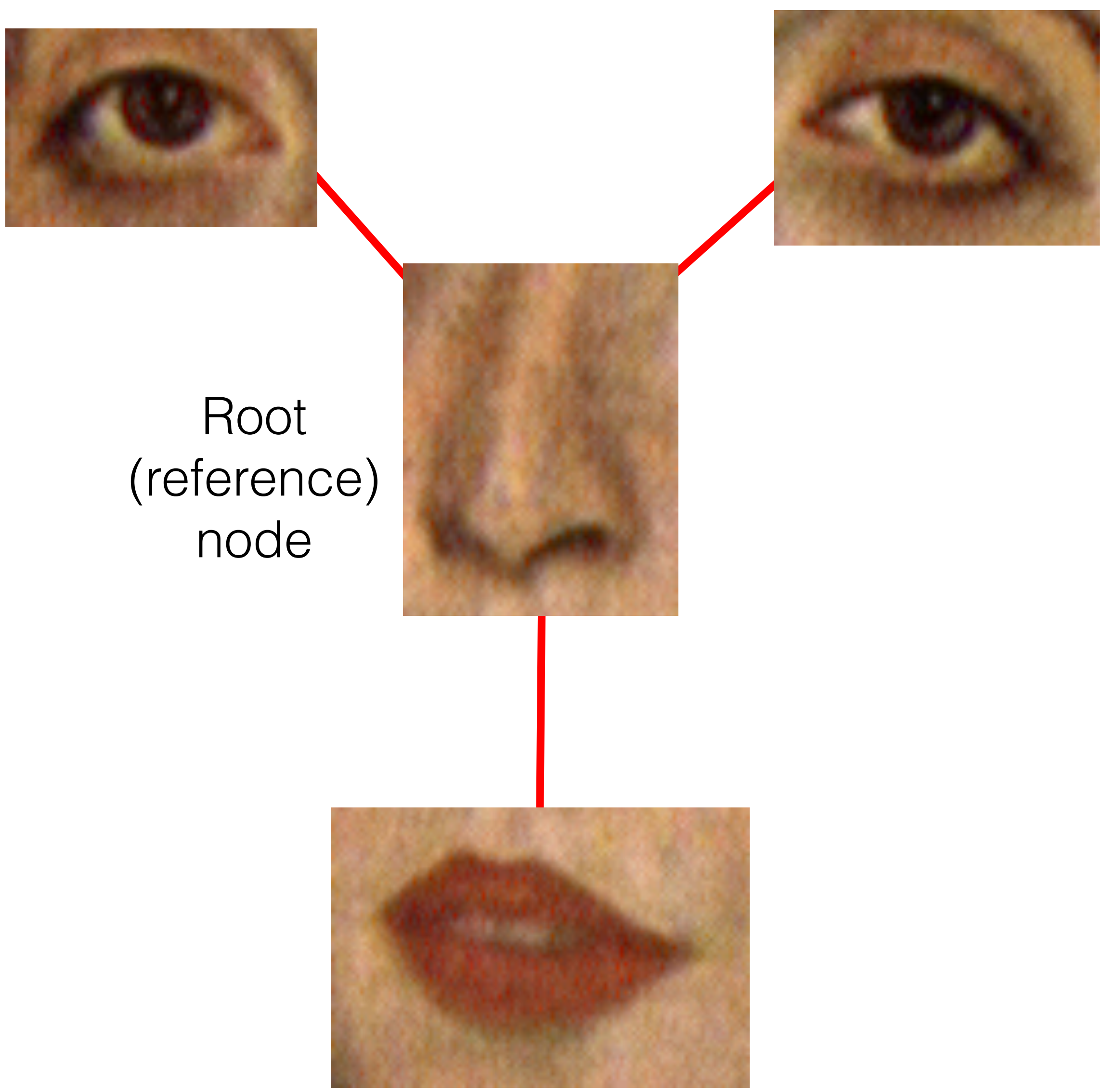
How can you constrain the number of configurations?

Factorize

$$p(L) = \prod_j p(L_j)$$

Break up the joint probability into smaller (independent) terms

Tree structure
(star model)

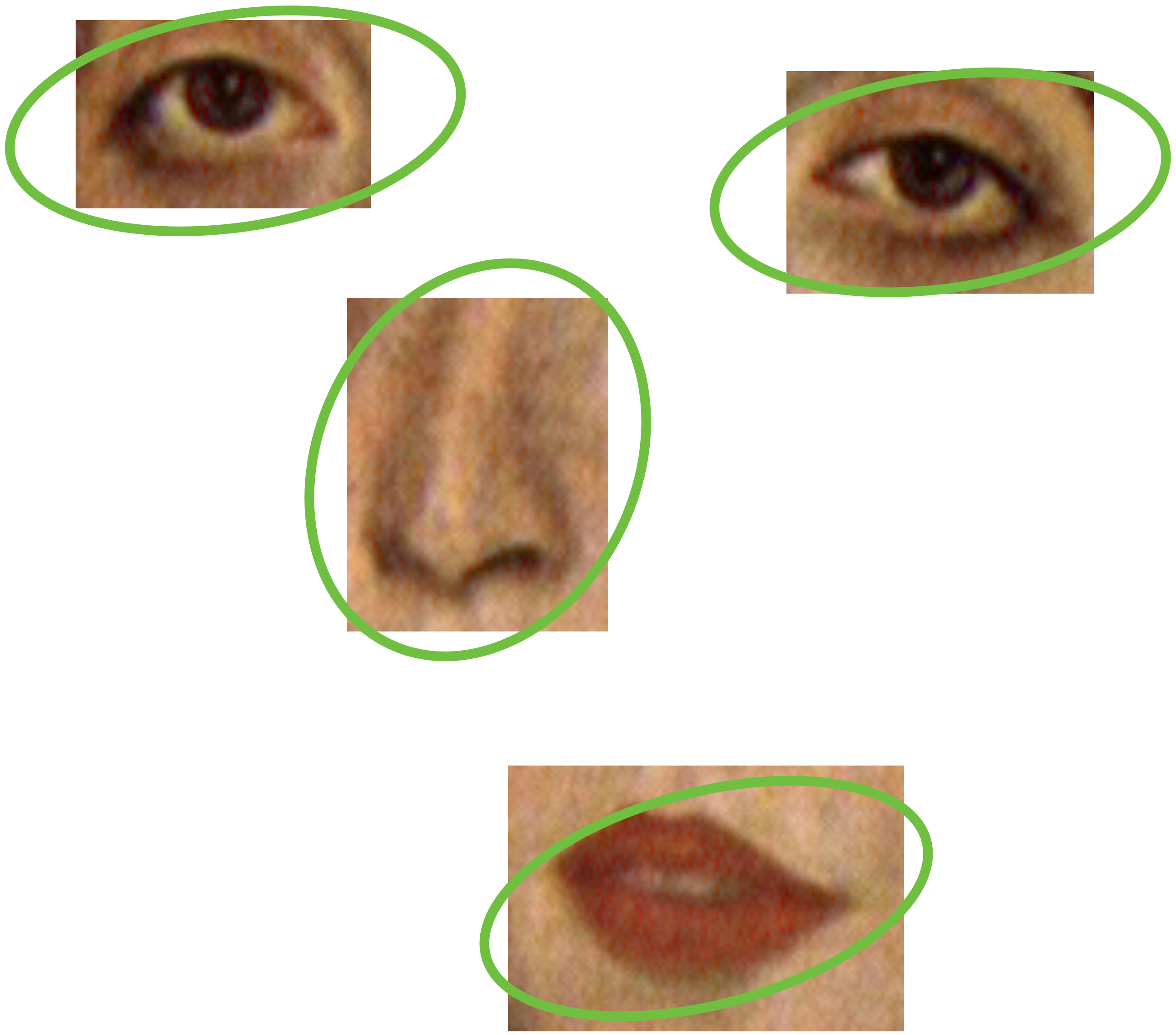


$$p(L) = p(l_r) \prod_{n=1}^{N-1} p(l_n | l_r)$$

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

<u>OK model:</u>
Assumes that one reference part is defined

Independent locations



$$p(L) = \prod_{n=1}^{N} p(l_n)$$

Each feature is allowed to move independently

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

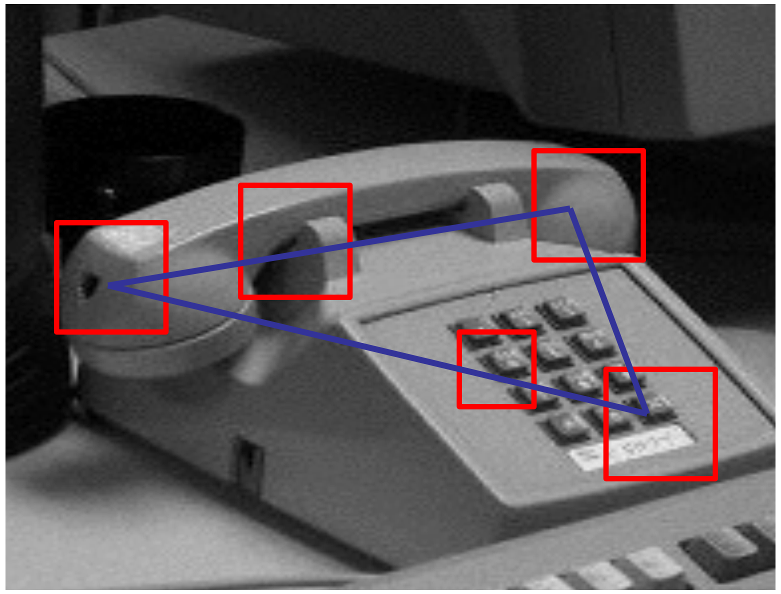
Pros

- Retains spatial constraints
- Robust to deformations

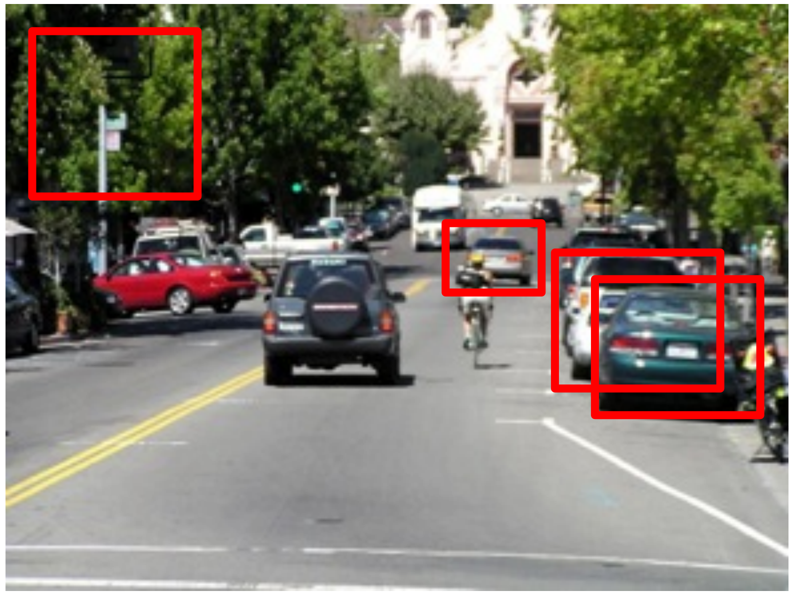
Cons

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

Feature
Matching



Spatial
reasoning

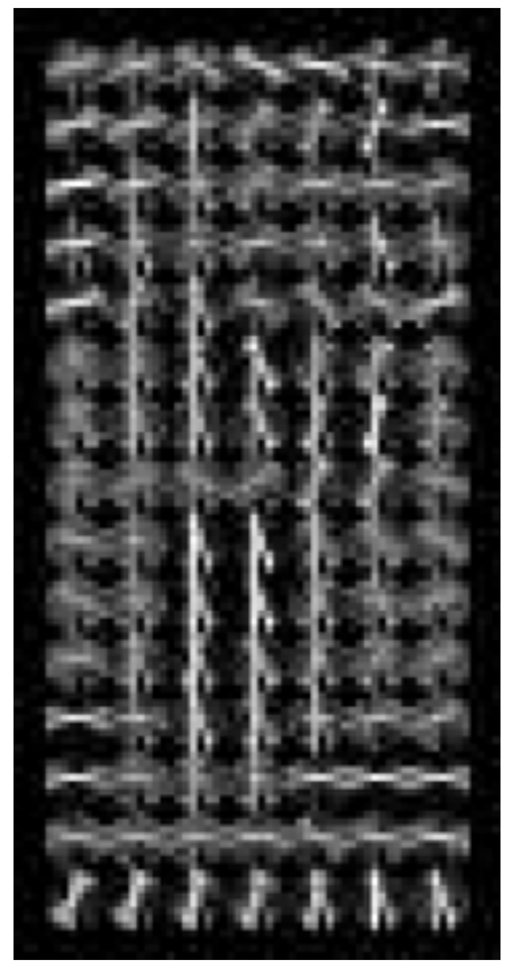


Window
classification

Window-based

Template Matching

1. get image window



2. compute features

3. compare to template

When does this work and when does it fail?

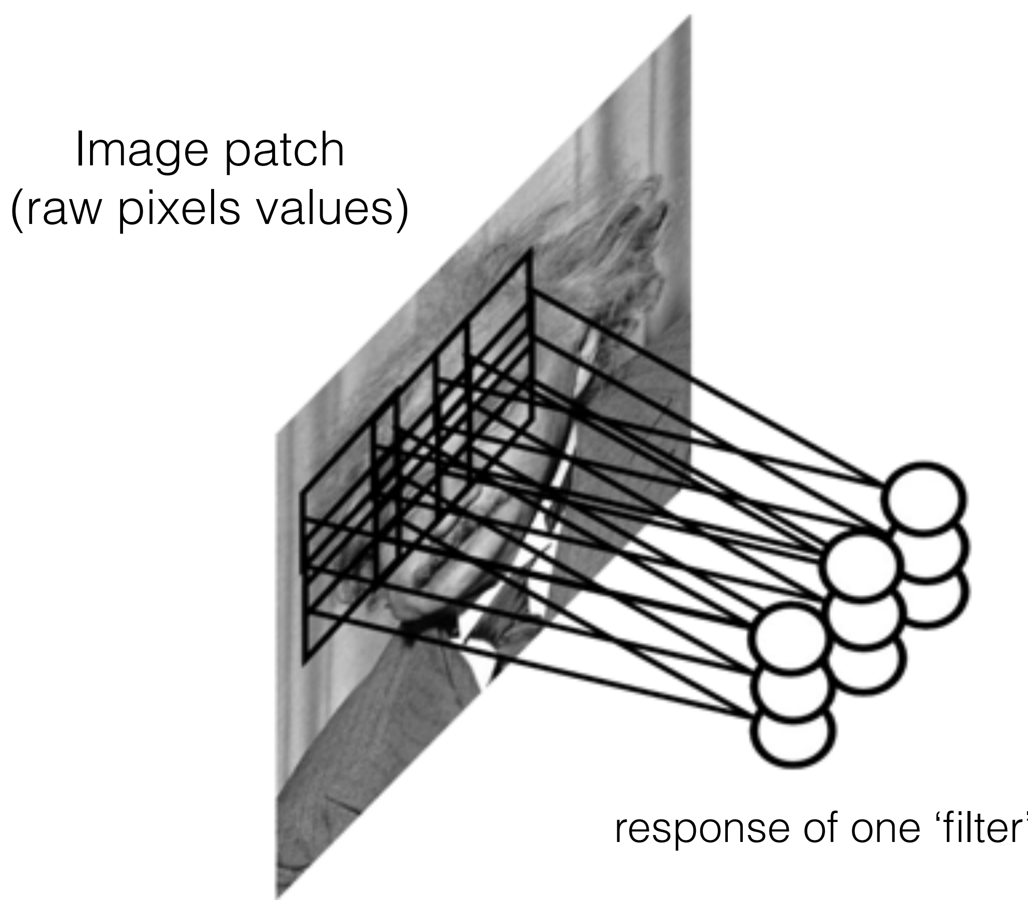
How many templates do you need?

Per-exemplar

find the ‘nearest’ exemplar, inherit its label

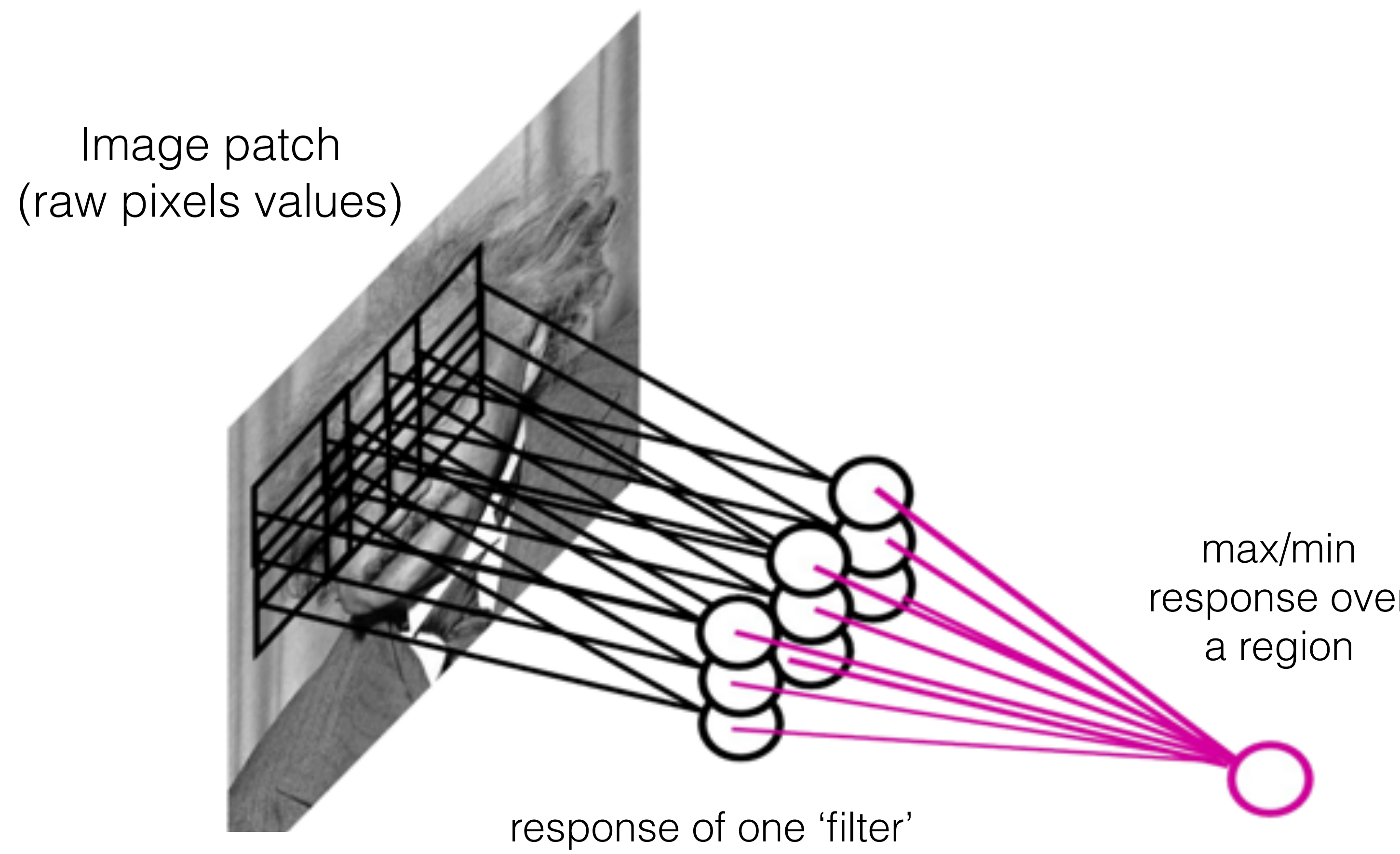
Deep networks

Convolution

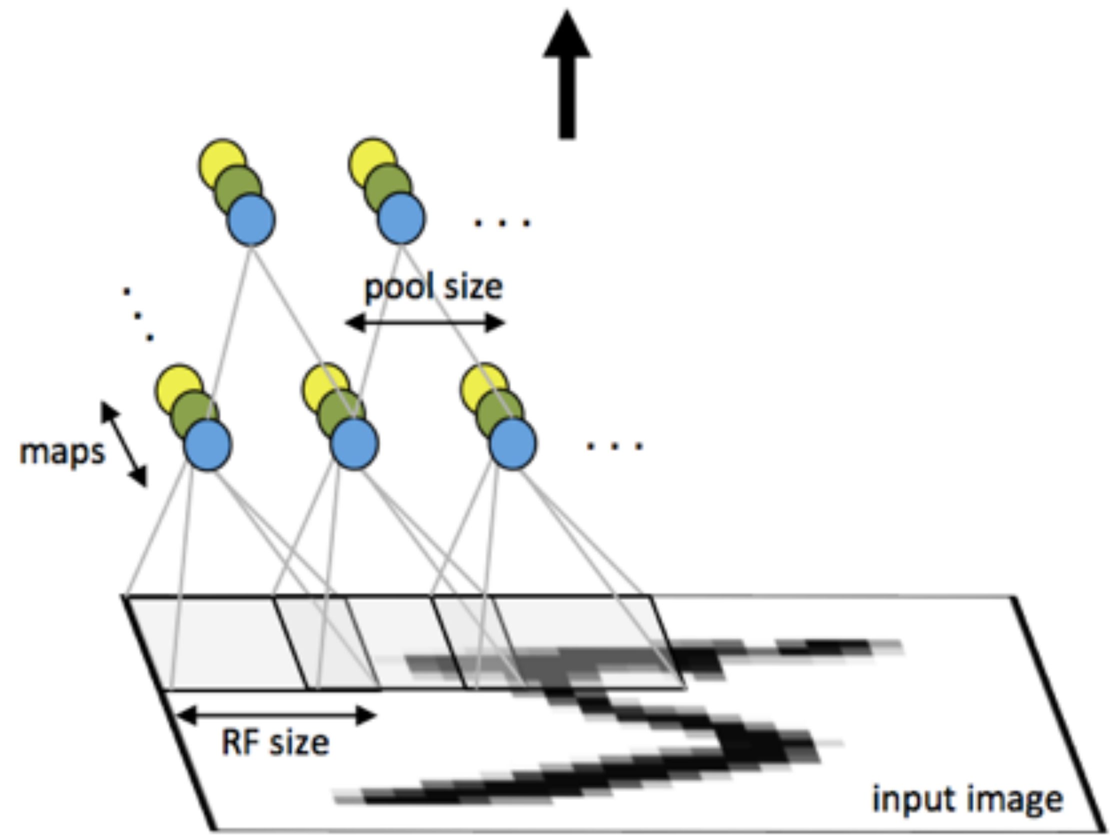


A 96 x 96 image convolved with 400 filters (features) of size 8 x 8 generates about 3 million values (89^2x400)

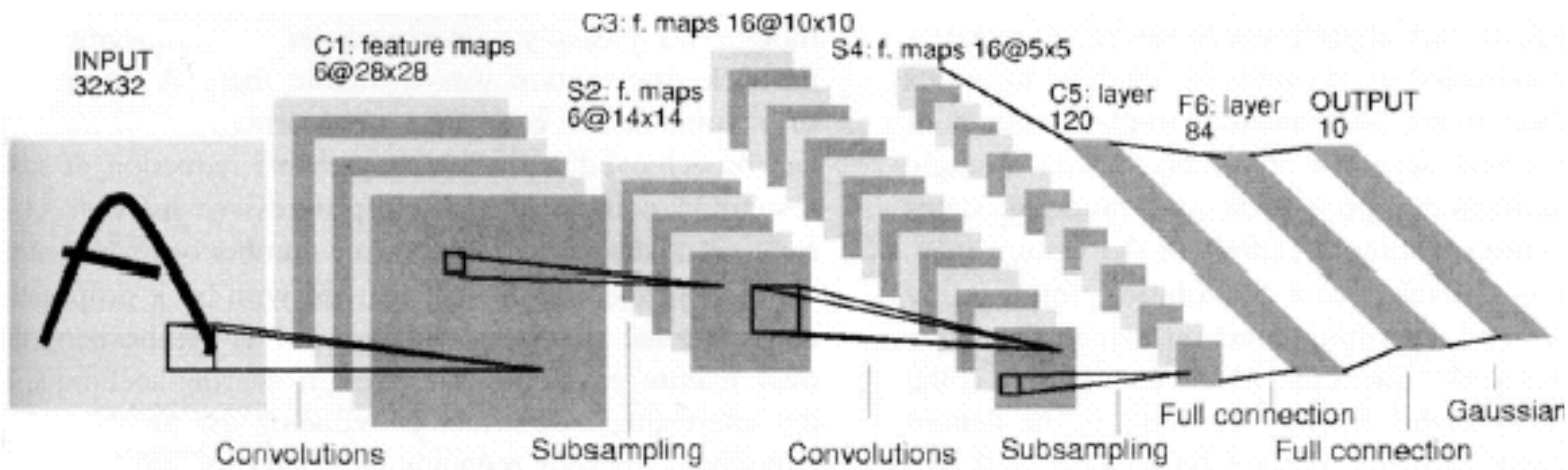
Pooling



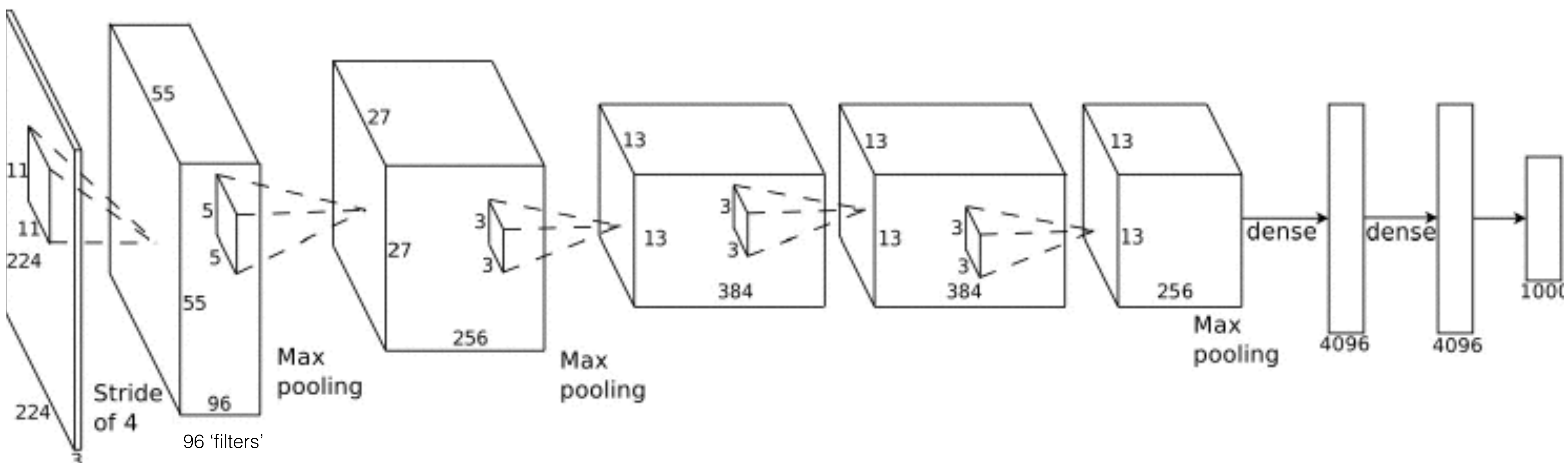
Pooling aggregates statistics and lowers the dimension of convolution



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
Gaussian



96 ‘filters’

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

- Requires large amounts of data
- Sometimes (very) slow to train