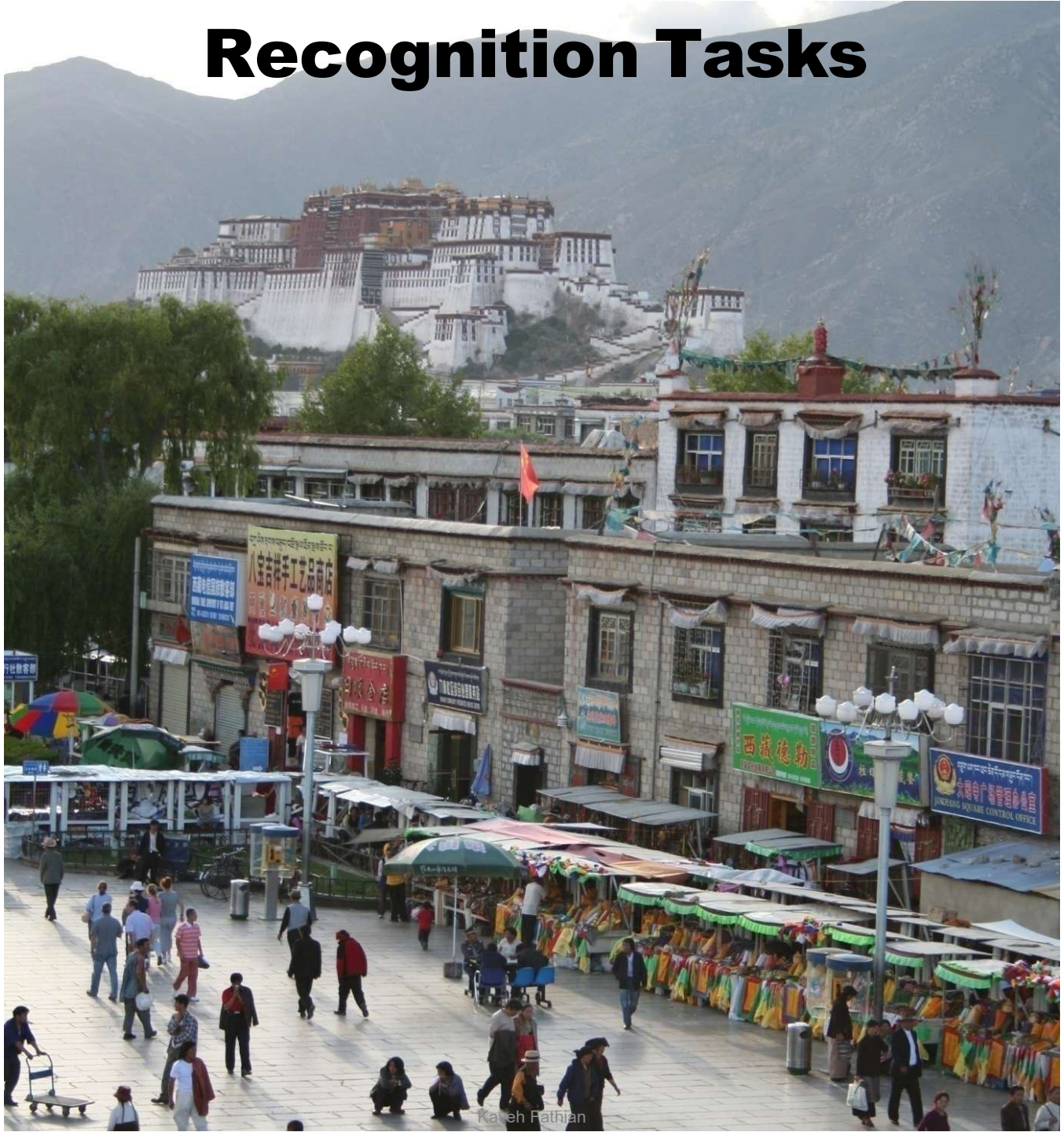
Introduction to Computer Vision

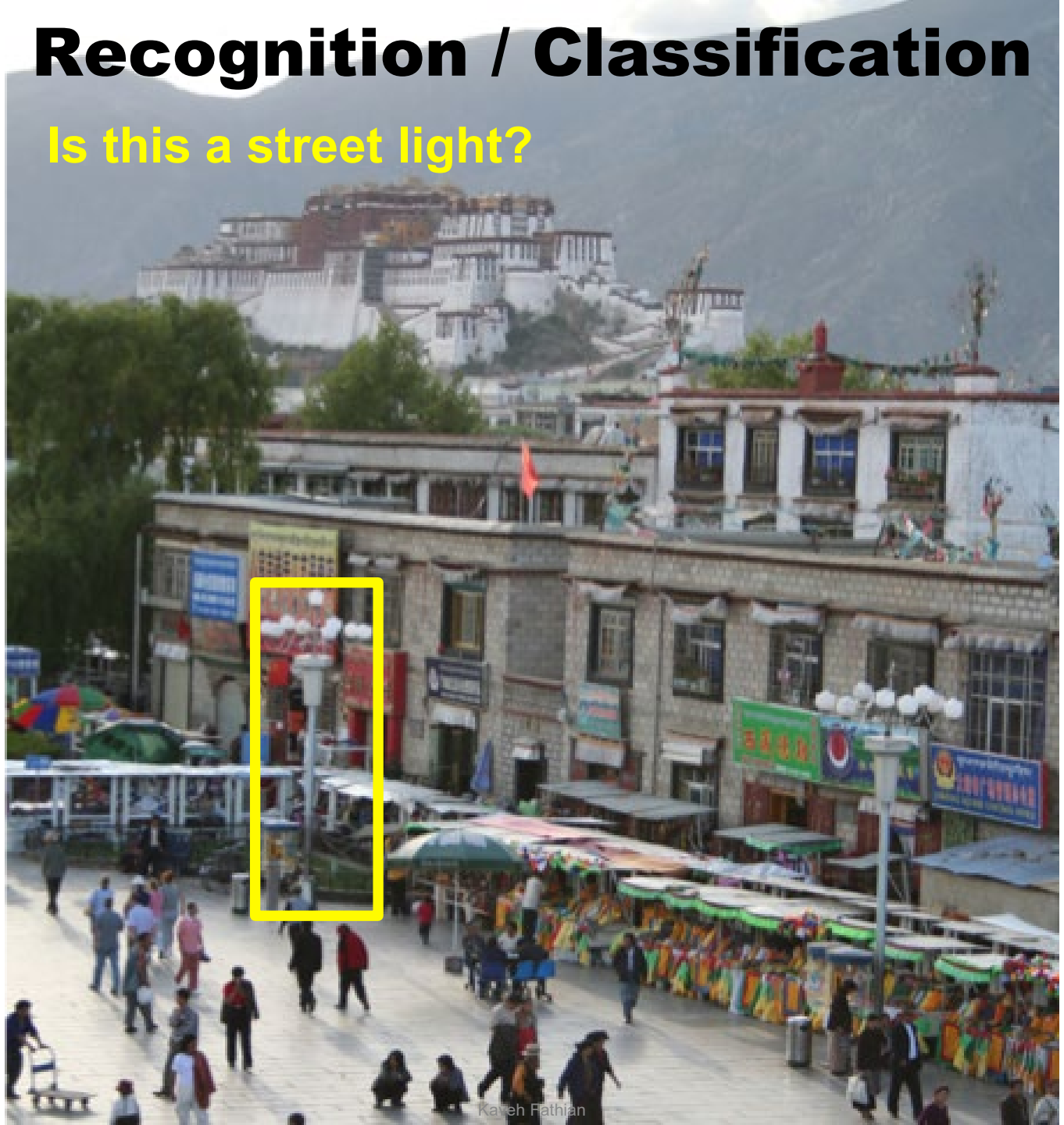
Kaveh Fathian
Assistant Professor
Computer Science Department
Colorado School of Mines

Lecture 15

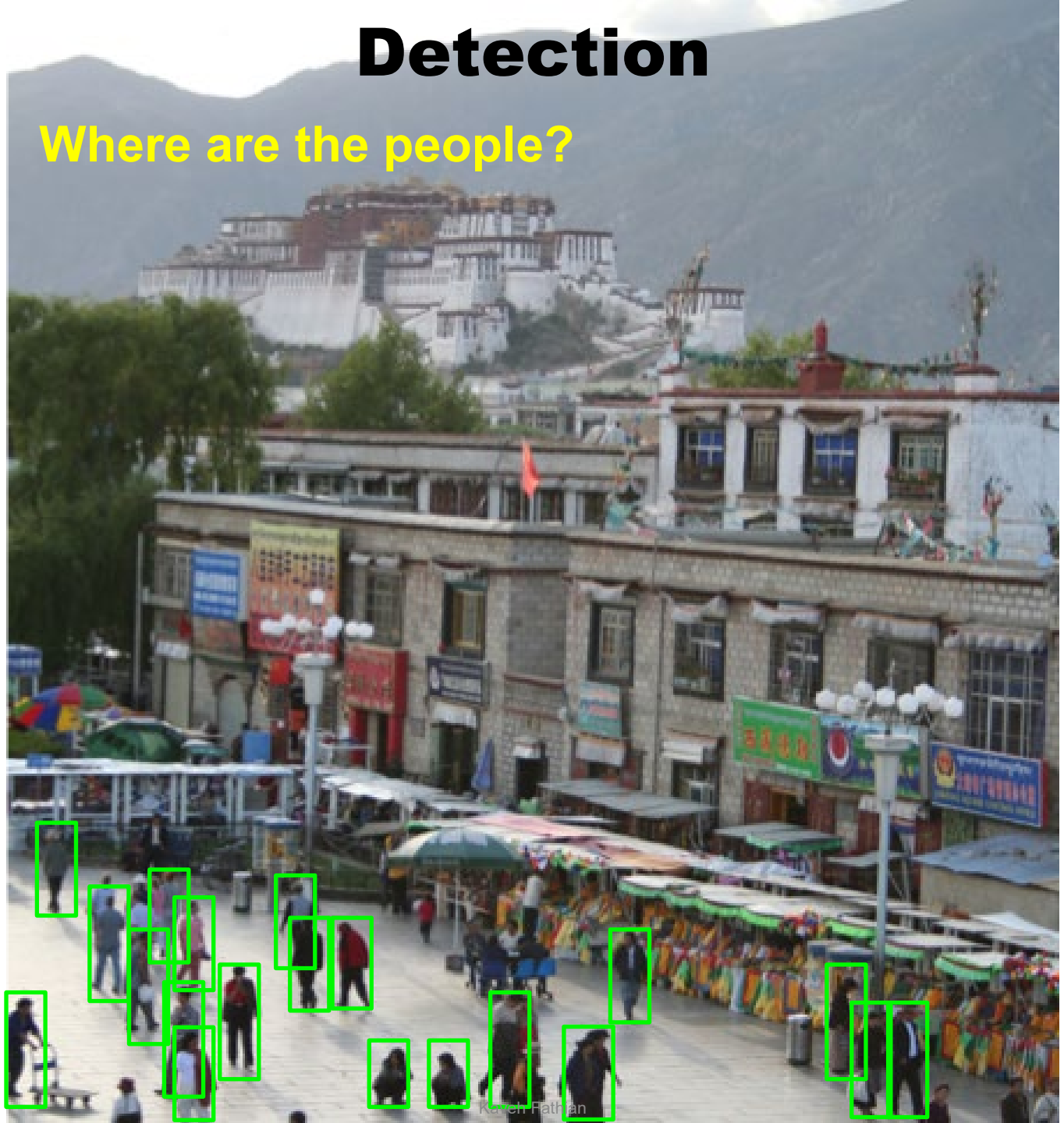
Recognition Tasks



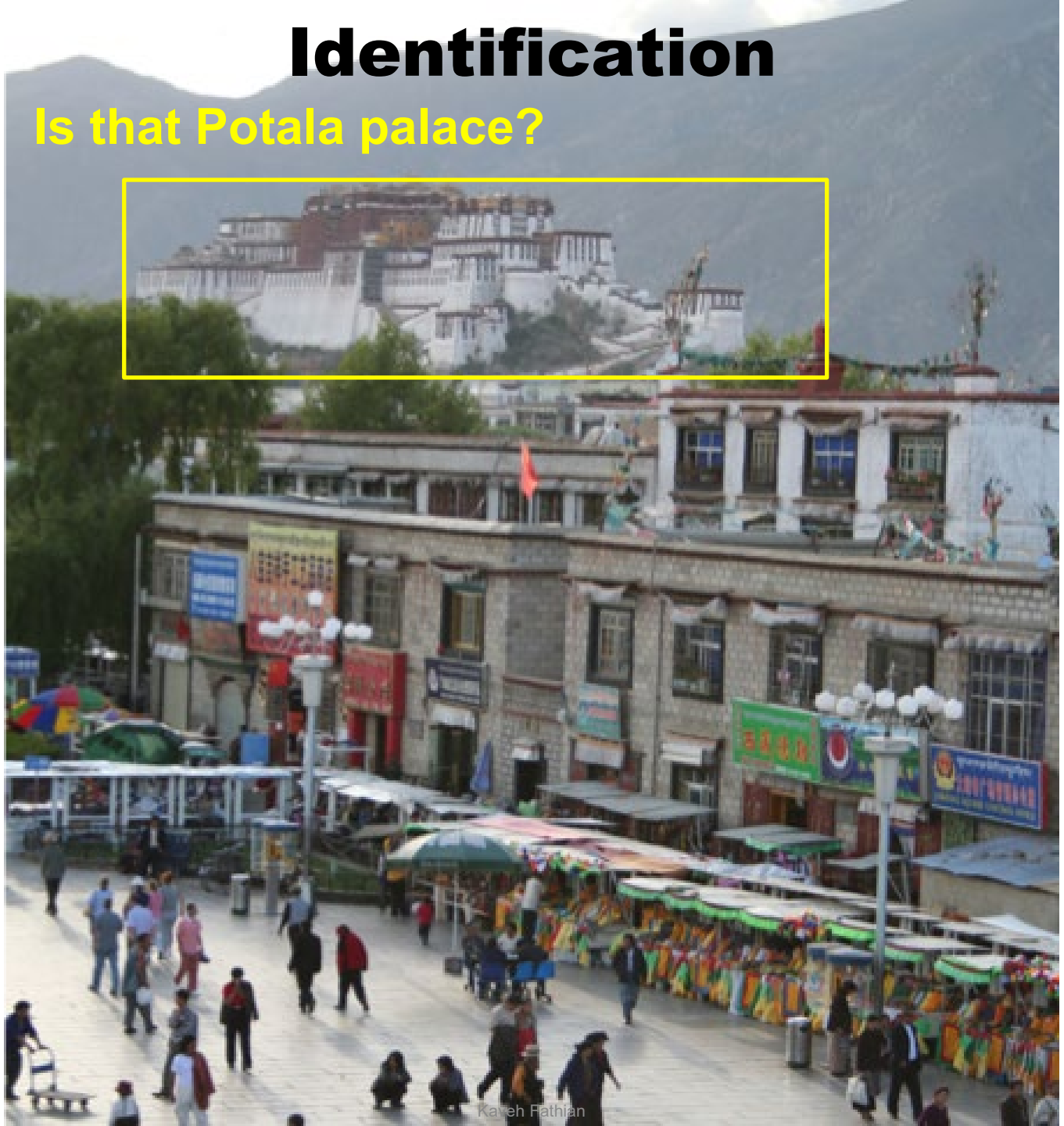
Recognition / Classification



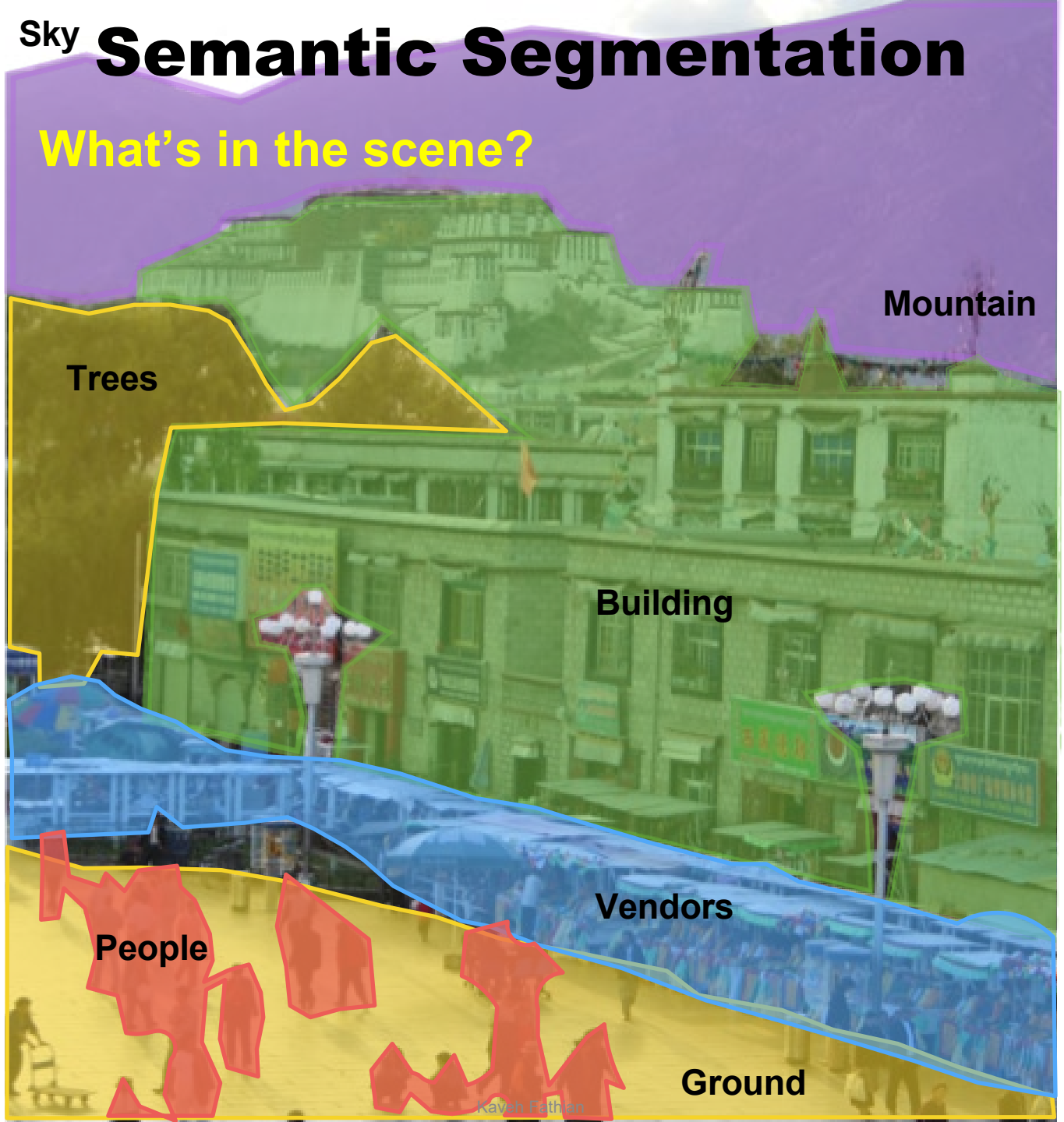
Detection



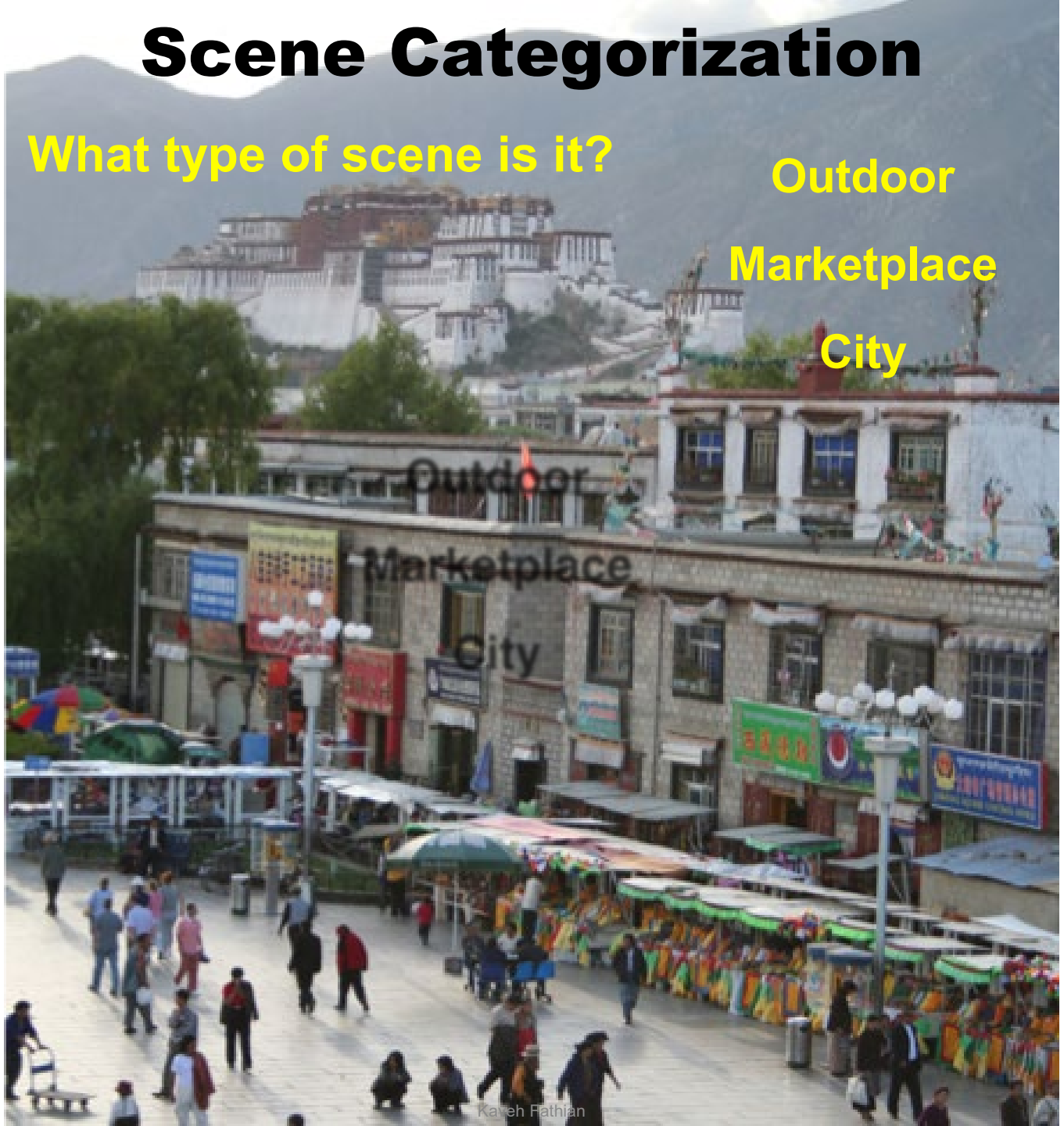
Identification



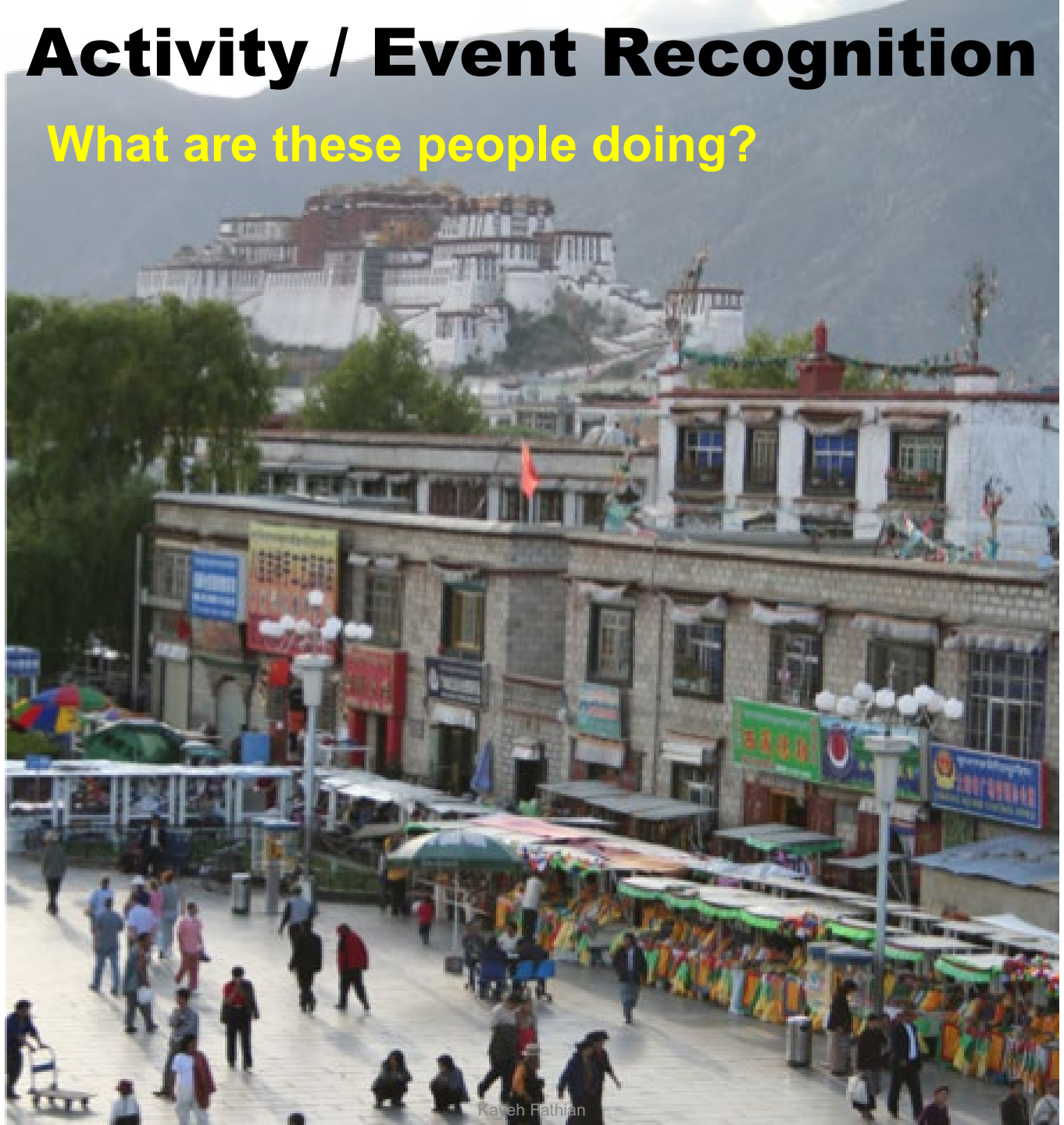
Semantic Segmentation



Scene Categorization



Activity / Event Recognition

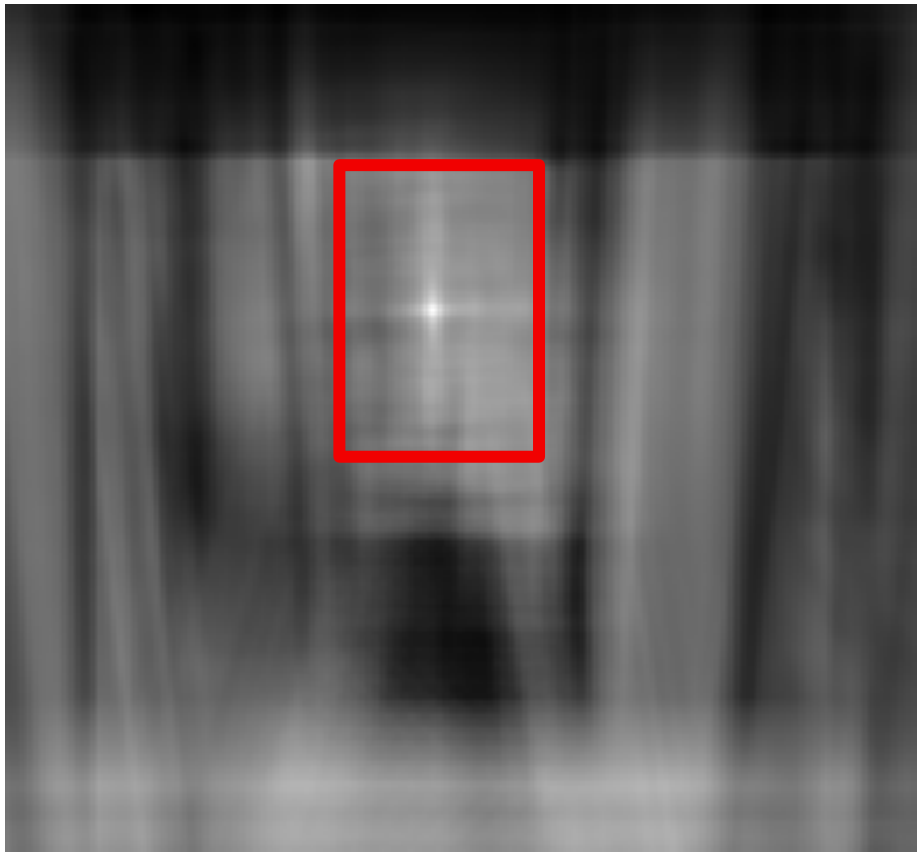


Object Recognition: Is it really so hard?

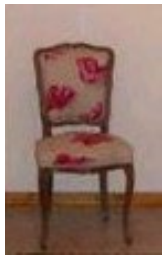
This is a chair

Find the chair in this image

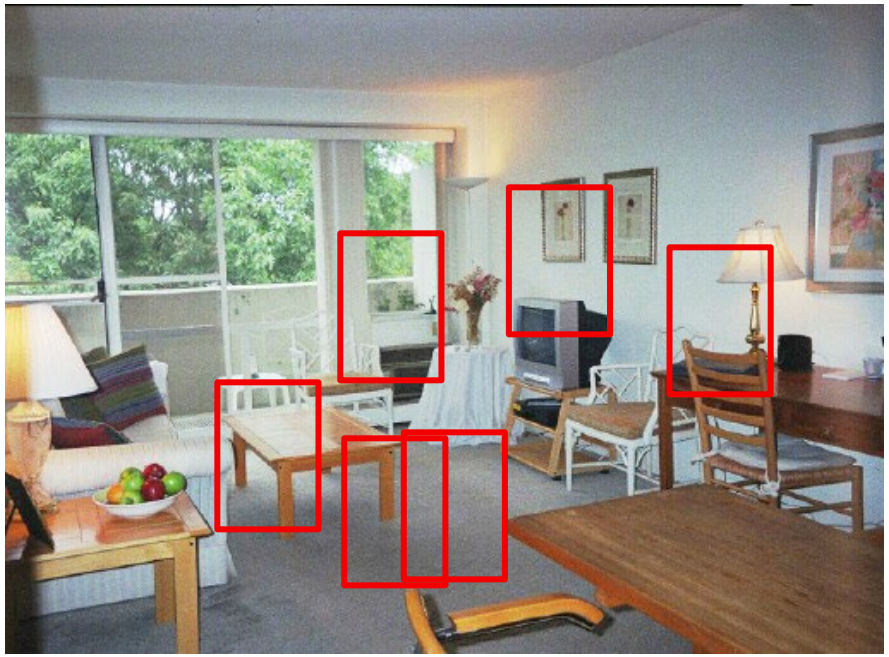
Output of normalized correlation

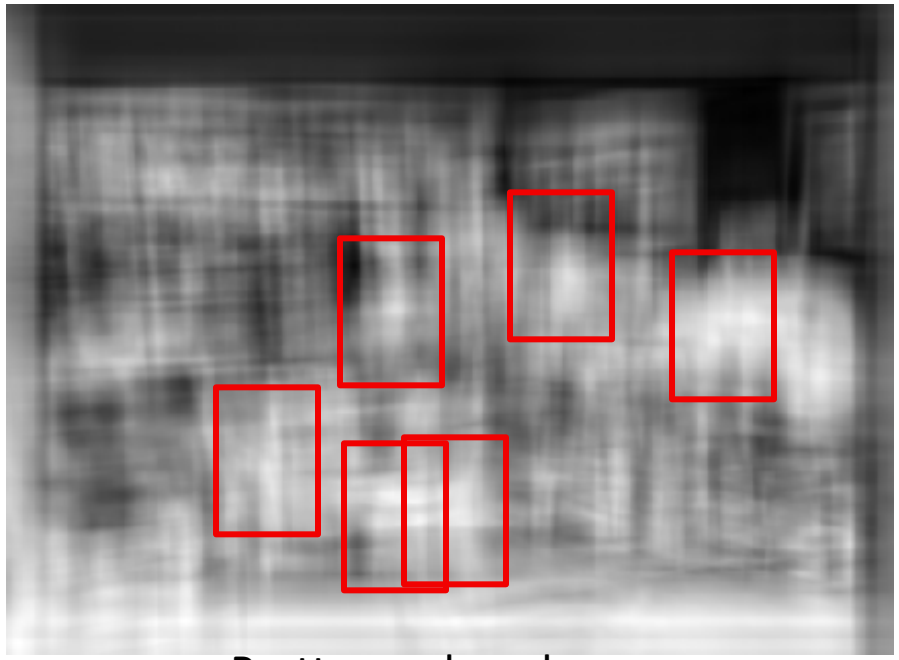


Object Recognition: Is it really so hard?



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

Let’s ask ChatGPT…

ChatGPT

Explain to me what is in this image?

This image shows the trunk of a tree, but it contains a well-camouflaged animal blending almost perfectly with the bark. If you look closely at the center of the image, slightly toward the left, you can spot a **lizard** — specifically, a **mossy leaf-tailed gecko (Uroplatus sikorae)**.

This species is known for its remarkable camouflage that mimics tree bark, moss, and lichen. Its body shape, coloration, and texture make it almost indistinguishable from the tree surface, which helps it avoid predators and ambush prey in its natural habitat (typically Madagascar).

Challenge: Variable Viewpoint

Michelangelo 1475-1564

Challenge: Variable Illumination

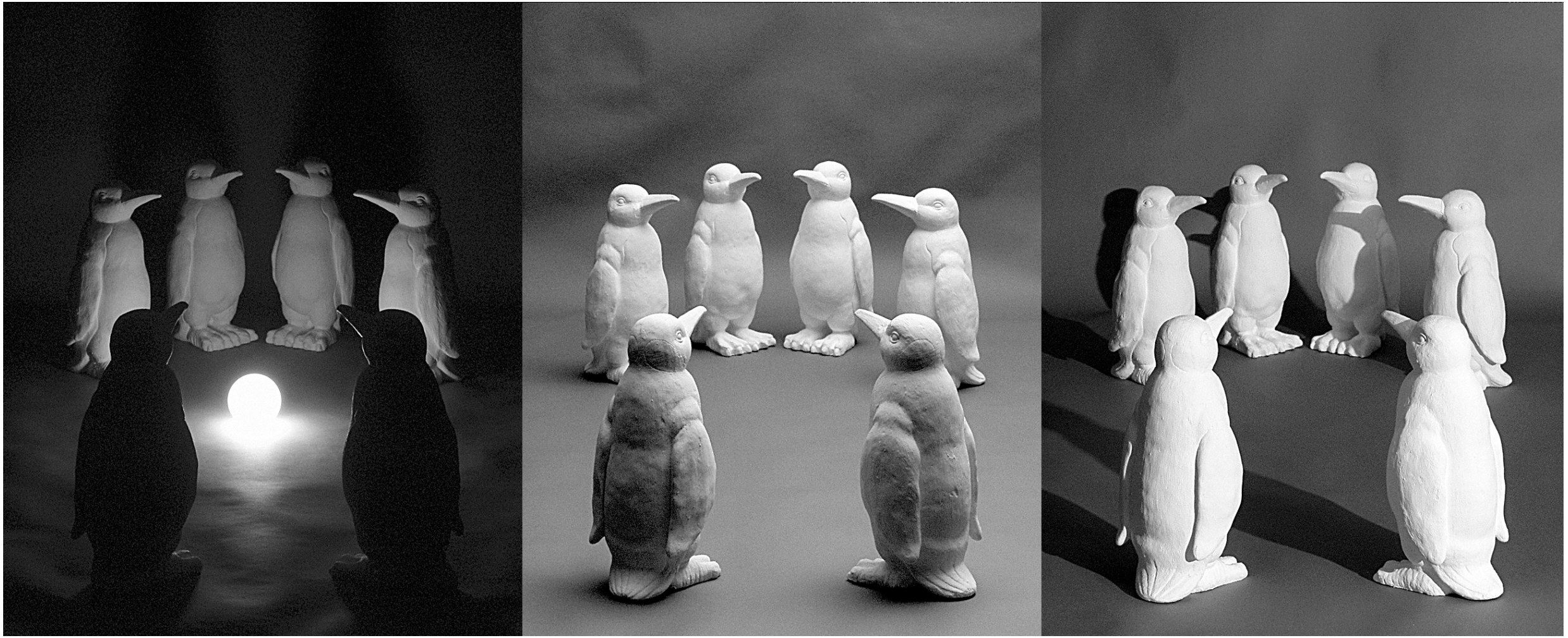
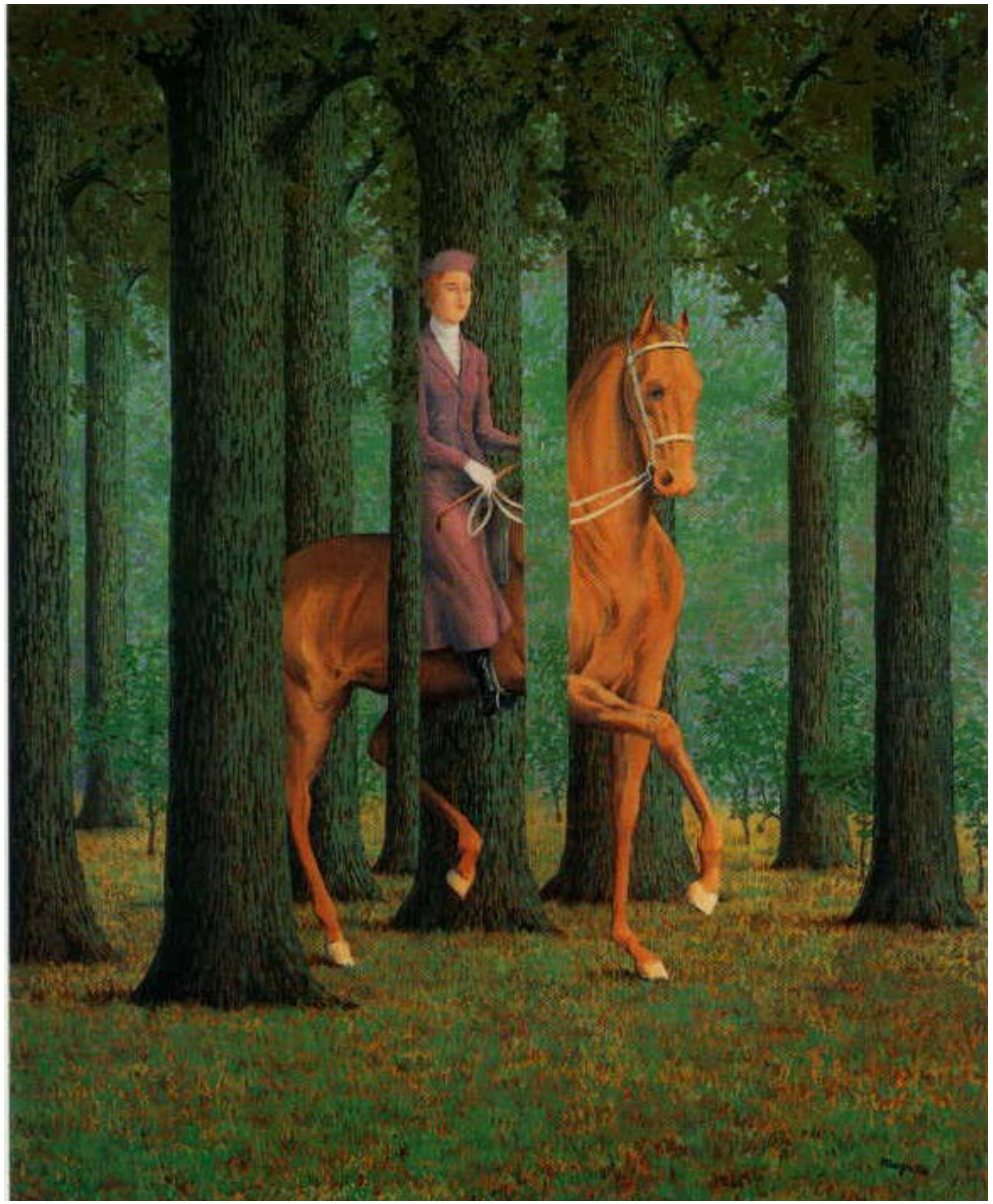


image credit: J. Koenderink

Challenge: Scale

Challenge: Deformation

Challenge: Occlusion



Magritte, 1957

Challenge: Background Clutter

Challenge: Intra-Class Variations

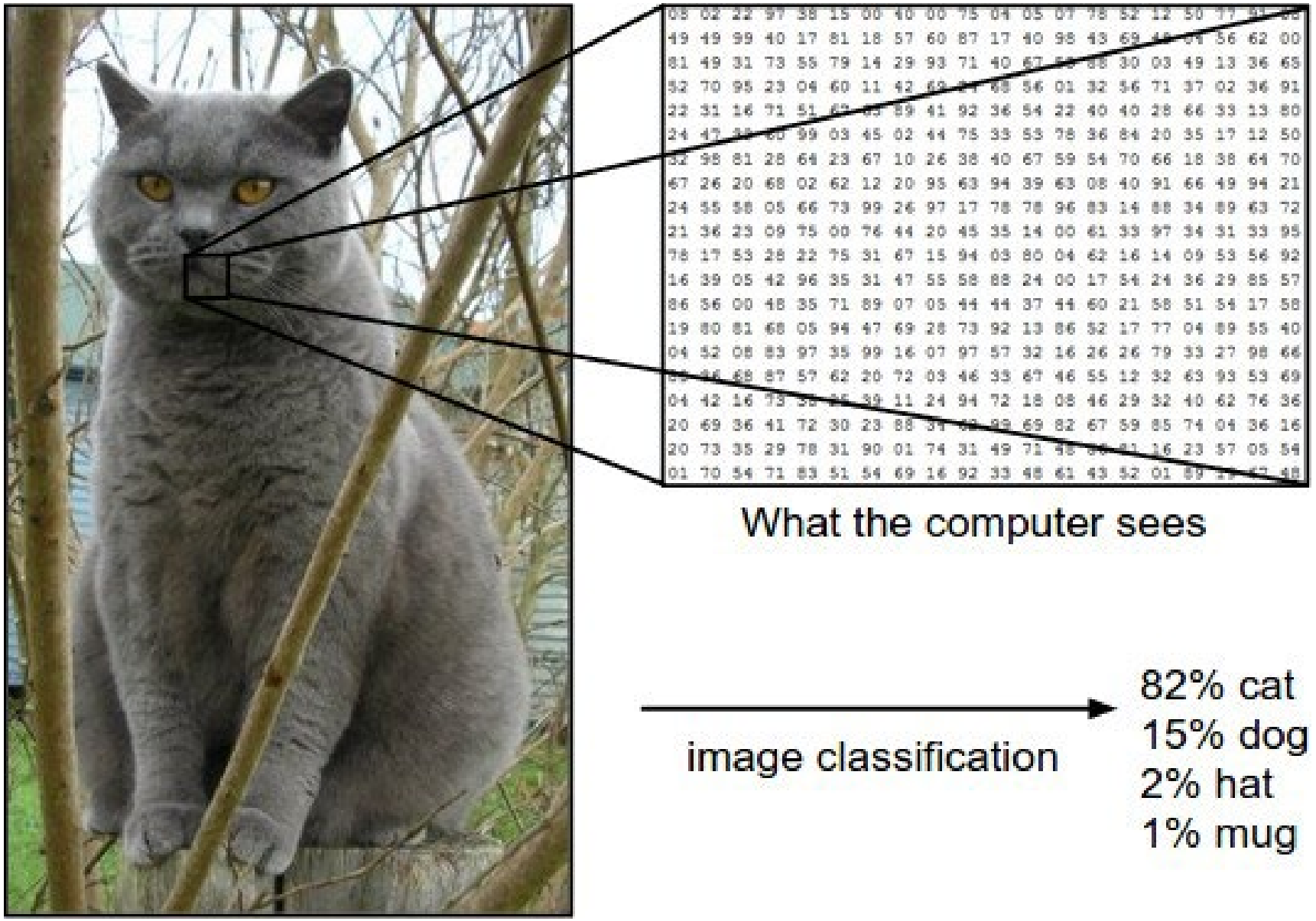
How many object categories are there?

Image Classification

(assume given set of discrete labels)
{dog, cat, truck, plane, ...}

cat

Image Classification: Problem

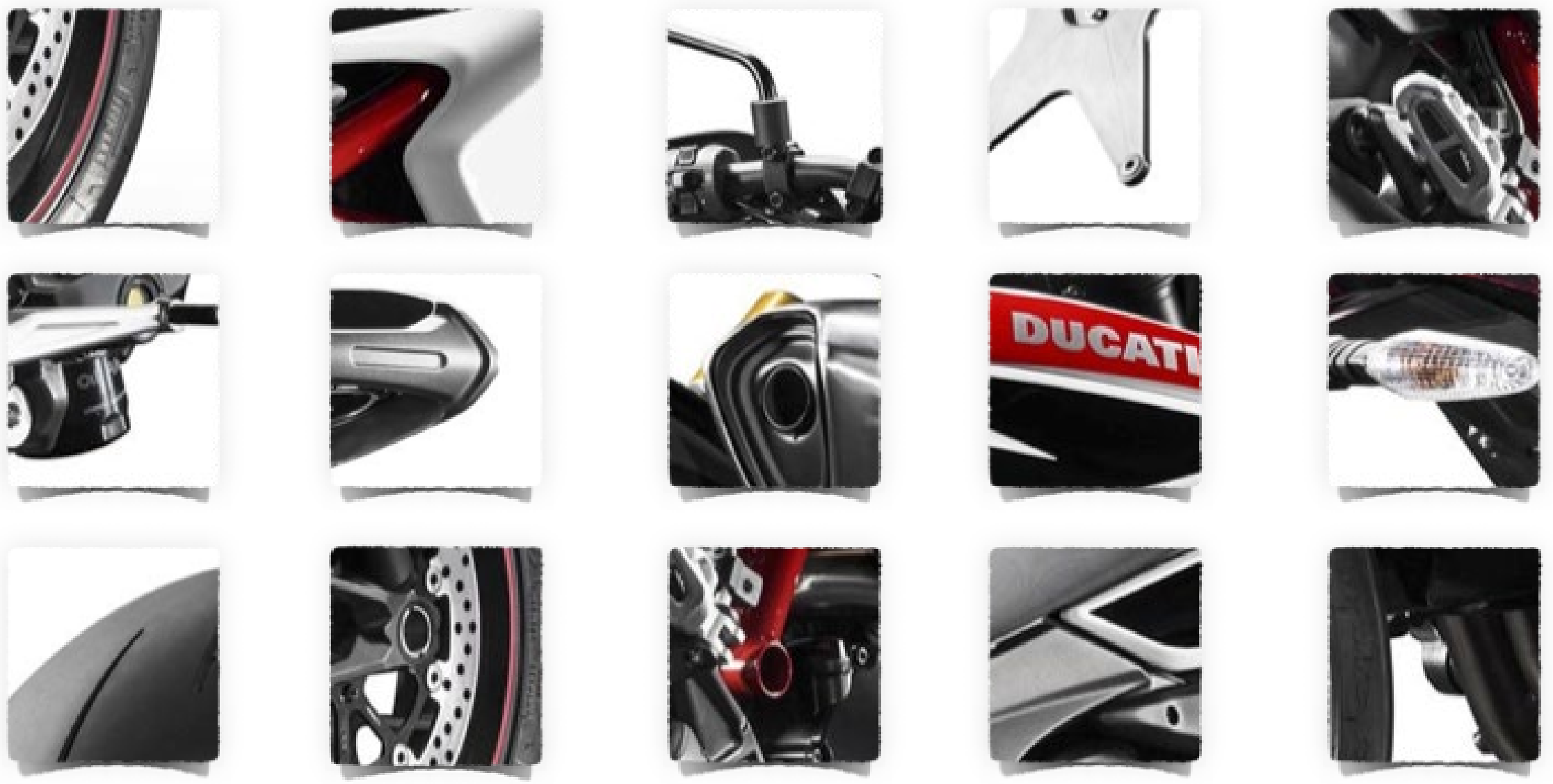


Data-Driven Approach

- Collect a database of images with labels
- Use ML to train an image classifier
- Evaluate the classifier on test images

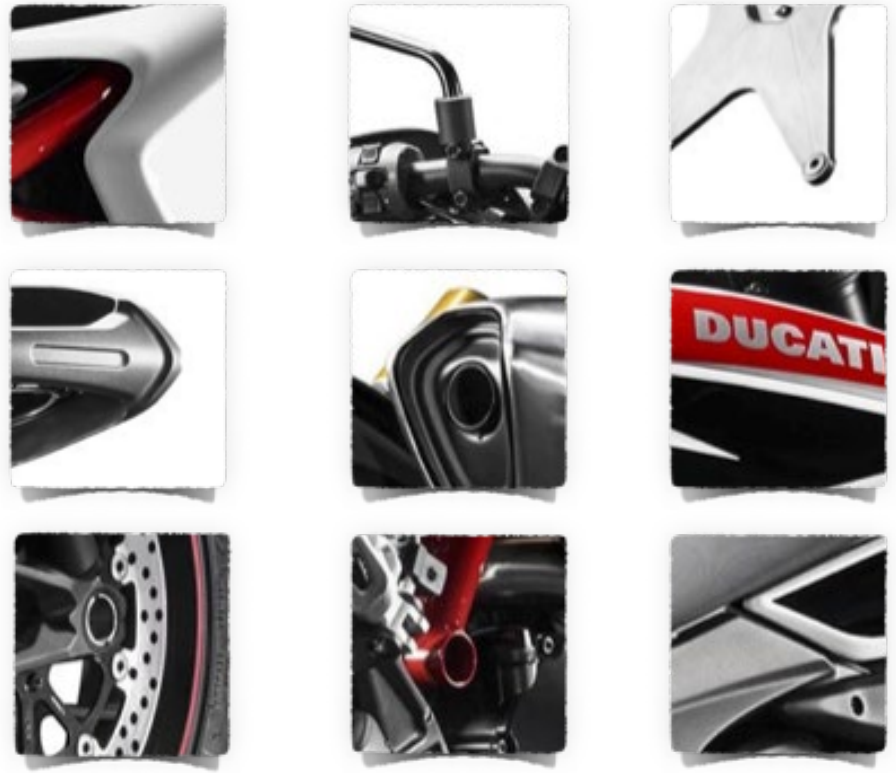
Bag of Words (BoW)

What object do these parts belong to?



Some local feature are very informative

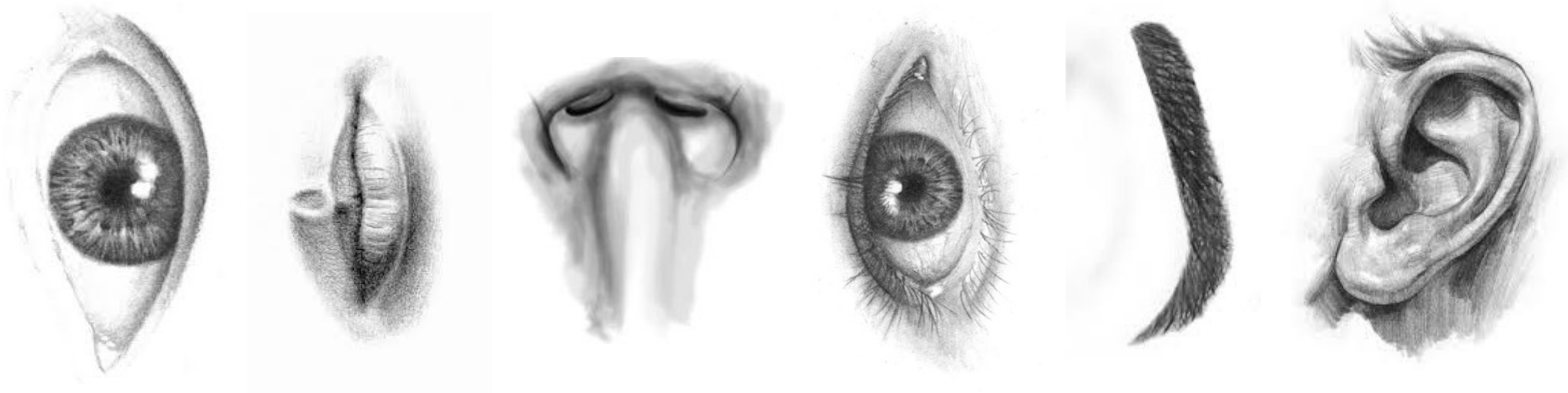
An object as



A collection of local features (bag-of-features)

- Deals well with occlusion
- Scale invariant
- Rotation invariant

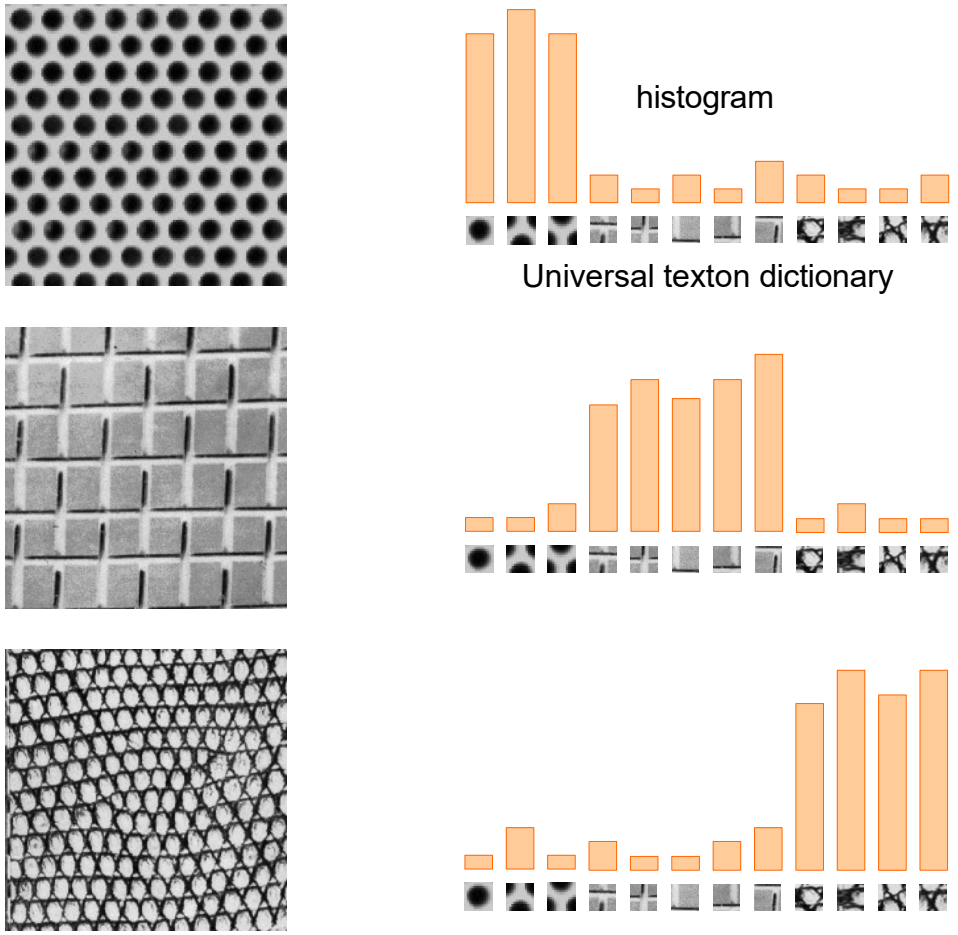
Not So Crazy Assumption!



Spatial information of local features
can be ignored for object recognition

Bag-of-Features

- Represent a data item (document, texture, image) as a histogram over features
- An old idea (e.g., texture recognition and information retrieval)



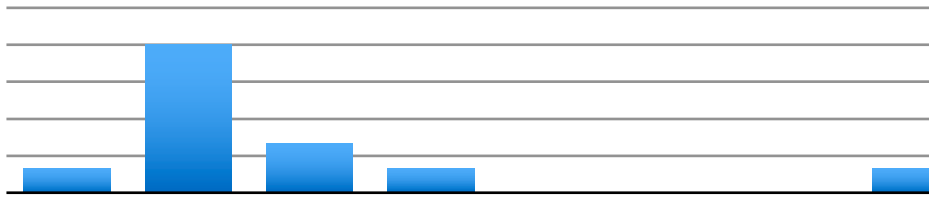
Vector Space Model

G. Salton. 'Mathematics and Information Retrieval' Journal of Documentation,1979

Sunday, December 22, 2013

DARPA Selects Carnegie Me

The Tartan Rescue Team from Carnegie Mellon University's National Robotics Engineering Center ranked third among teams competing in the Defense Advanced Research Projects Agency (DARPA) Robotics Challenge Trials this weekend in Homestead, Fla., and was selected by the agency as one of eight teams eligible for DARPA funding to prepare for next December's finals. The team's four-limbed CMU Highly Intelligent Mobile Platform, or CHIMP, robot scored 18 out of a possible 32 points during the two-day trials. It demonstrated its ability to perform such tasks as removing debris, cutting a hole through a wall and closing a series of valves.

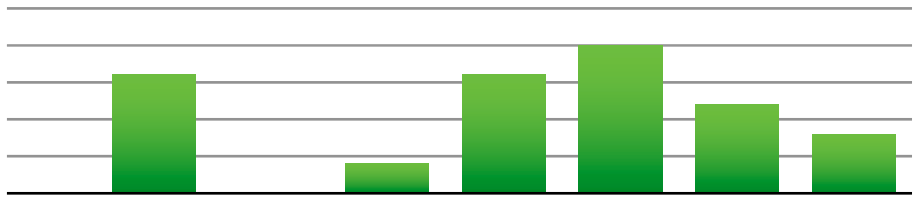


1	6	2	1	0	0	0	1
Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor

Monday, January 20, 2014

Bio-Inspired Robotic Device

PITTSBURGH—A soft, wearable device that mimics the muscles, tendons and ligaments of the lower leg could aid in the rehabilitation of patients with ankle-foot disorders such as drop foot, said Yong-Lae Park, an assistant professor of robotics at Carnegie Mellon University. Park, working with collaborators at Harvard University, the University of Southern California, MIT and BioSensics, developed an active orthotic device using soft plastics and composite materials, instead of a rigid exoskeleton. The soft materials, combined with pneumatic artificial muscles (PAMs), lightweight sensors and advanced control software, made it possible for the robotic device to achieve natural motions in the ankle.



0	4	0	1	4	5	3	2
Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor

http://www.fodey.com/generators/newspaper/snippet.asp

A document (datapoint) is a vector of counts over each word (feature)

$$\boldsymbol{v}_d = [n(w_{1,d}) \quad n(w_{2,d}) \quad \cdots \quad n(w_{T,d})]$$

just a histogram over words

$n(\cdot)$ counts the number of occurrences

What is the similarity between two documents?

A document (datapoint) is a vector of counts over each word (feature)

$$\boldsymbol{v}_d = [n(w_{1,d}) \quad n(w_{2,d}) \quad \cdots \quad n(w_{T,d})]$$

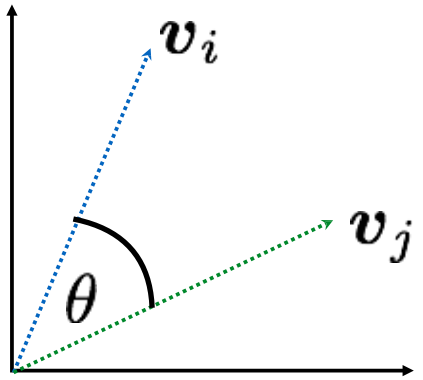
$n(\cdot)$ counts the number of occurrences

just a histogram over words

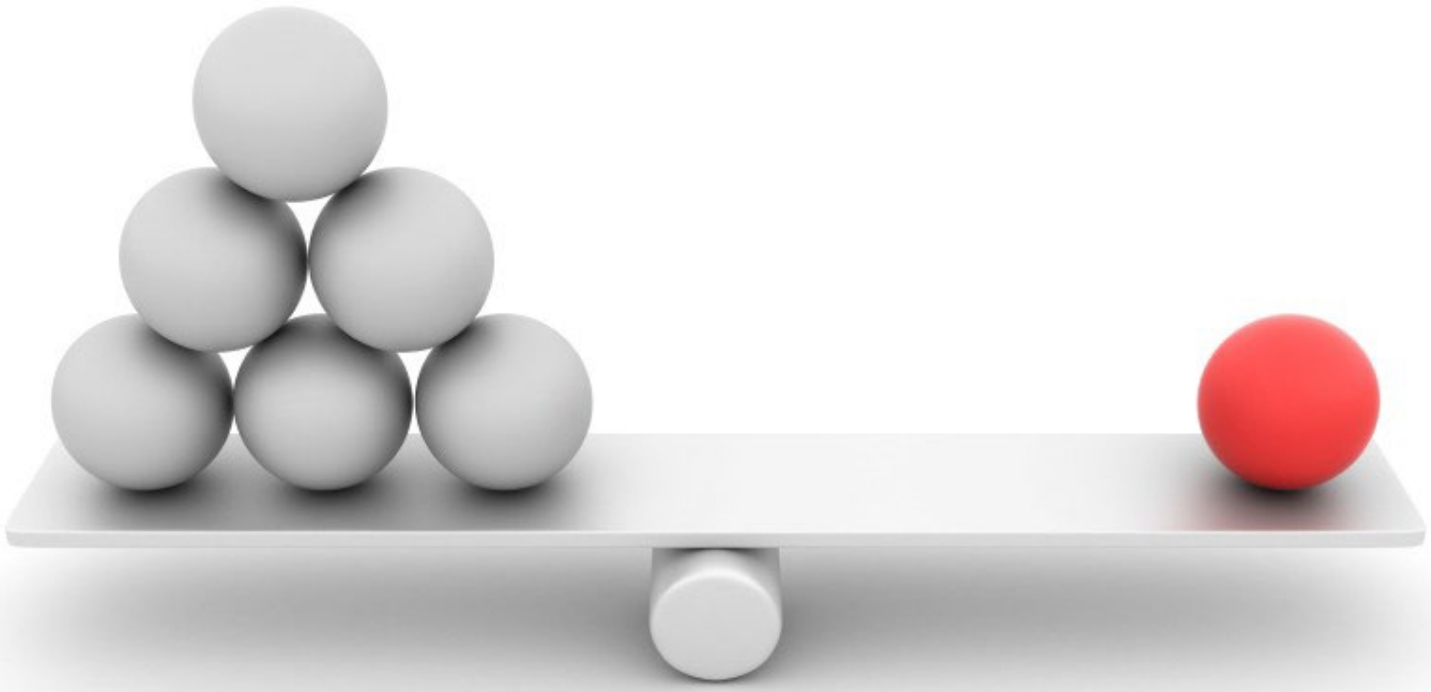
What is the similarity between two documents?

Use any distance you want but the cosine distance is fast.

$$d(\boldsymbol{v}_i, \boldsymbol{v}_j) = \cos\theta$$
$$= \frac{\boldsymbol{v}_i \cdot \boldsymbol{v}_j}{\|\boldsymbol{v}_i\| \|\boldsymbol{v}_j\|}$$



But Not All Words Are Created Equal



TF-IDF

TF-IDF: **T**erm **F**requency **I**nverse **D**ocument **F**requency

$$\boldsymbol{v}_d = [n(w_{1,d}) \quad n(w_{2,d}) \quad \cdots \quad n(w_{T,d})]$$

weigh each word by a heuristic

$$\boldsymbol{v}_d = [n(w_{1,d})\alpha_1 \quad n(w_{2,d})\alpha_2 \quad \cdots \quad n(w_{T,d})\alpha_T]$$

term frequency

inverse document frequency

$$n(w_{i,d})\alpha_i = n(w_{i,d}) \log \left\{ \frac{D}{\sum_{d'} \mathbf{1}[w_i \in d']} \right\}$$

(down-weights **common** terms)

Standard BOW Pipeline
(for image classification)

Standard BOW Pipeline

Dictionary Learning:
Learn Visual Words using clustering

Encode:
Build Bags-of-Words (BOW) vectors
for each image

Classify:
Train and test data using BOWs

Standard BOW Pipeline

Dictionary Learning:
Learn Visual Words using clustering

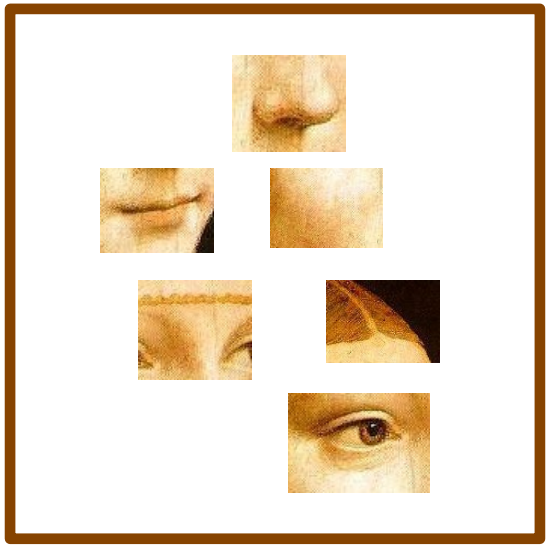
Encode:
Build Bags-of-Words (BOW) vectors for each image

Classify:
Train and test data using BOWs

Standard BOW Pipeline

Dictionary Learning:
Learn Visual Words using clustering

1. Extract features (e.g., SIFT) from images



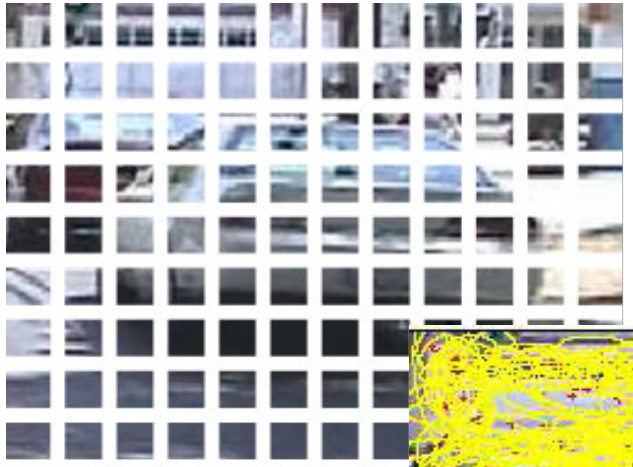
Standard BOW Pipeline

Dictionary Learning:
Learn Visual Words using clustering

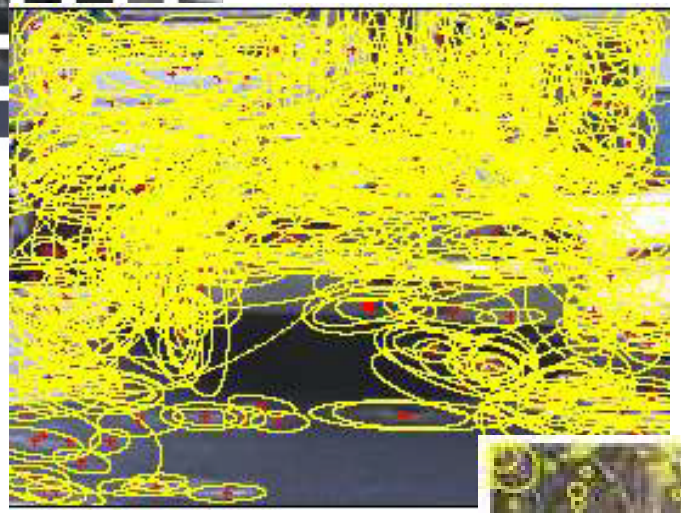
2. Learn visual dictionary (e.g., K-means clustering)

What Kinds of Features Can We Extract?

- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005
- Interest point detector
  - Csurka et al. 2004
  - Fei-Fei & Perona, 2005
  - Sivic et al. 2005
- Other methods
  - Random sampling (Vidal-Naquet & Ullman, 2002)
  - Segmentation-based patches (Barnard et al. 2003)
  - SIFT features



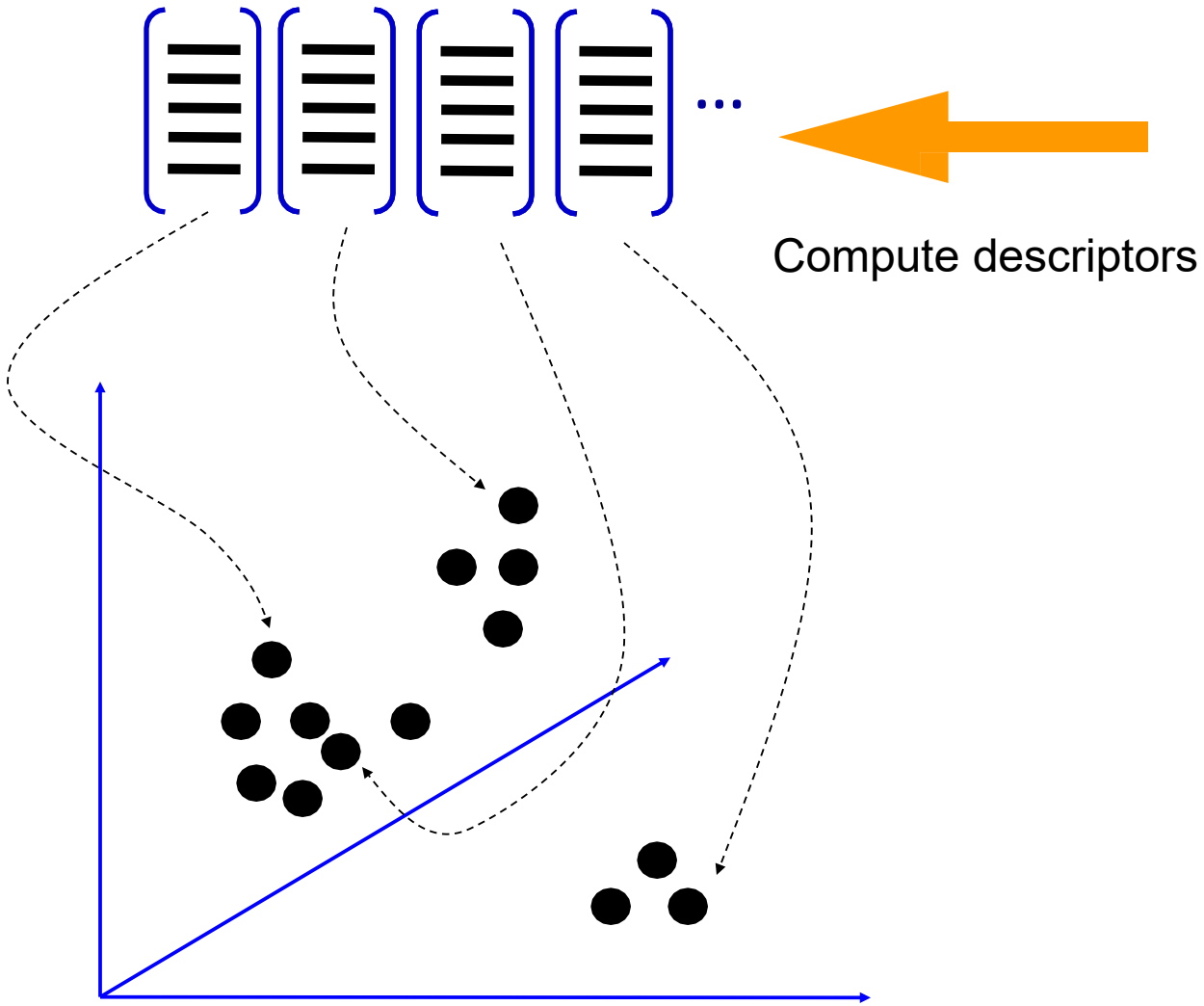
Dense, regular grid



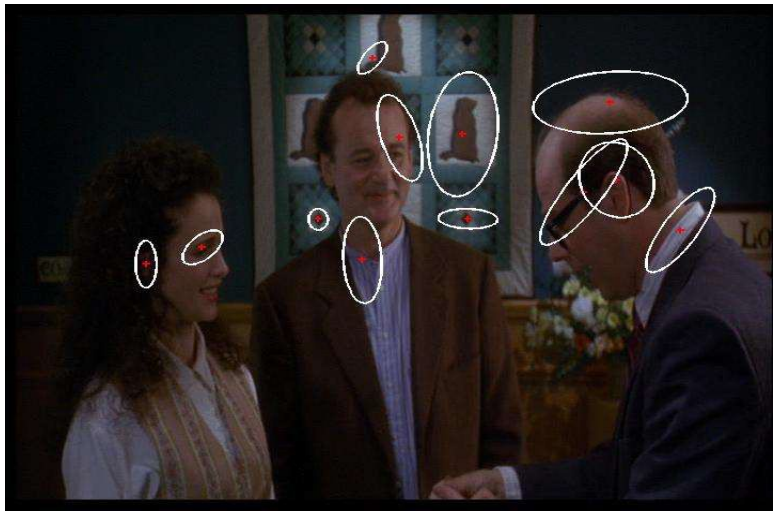
Sparse at interest points

Randomly

How Do We Learn the Dictionary?

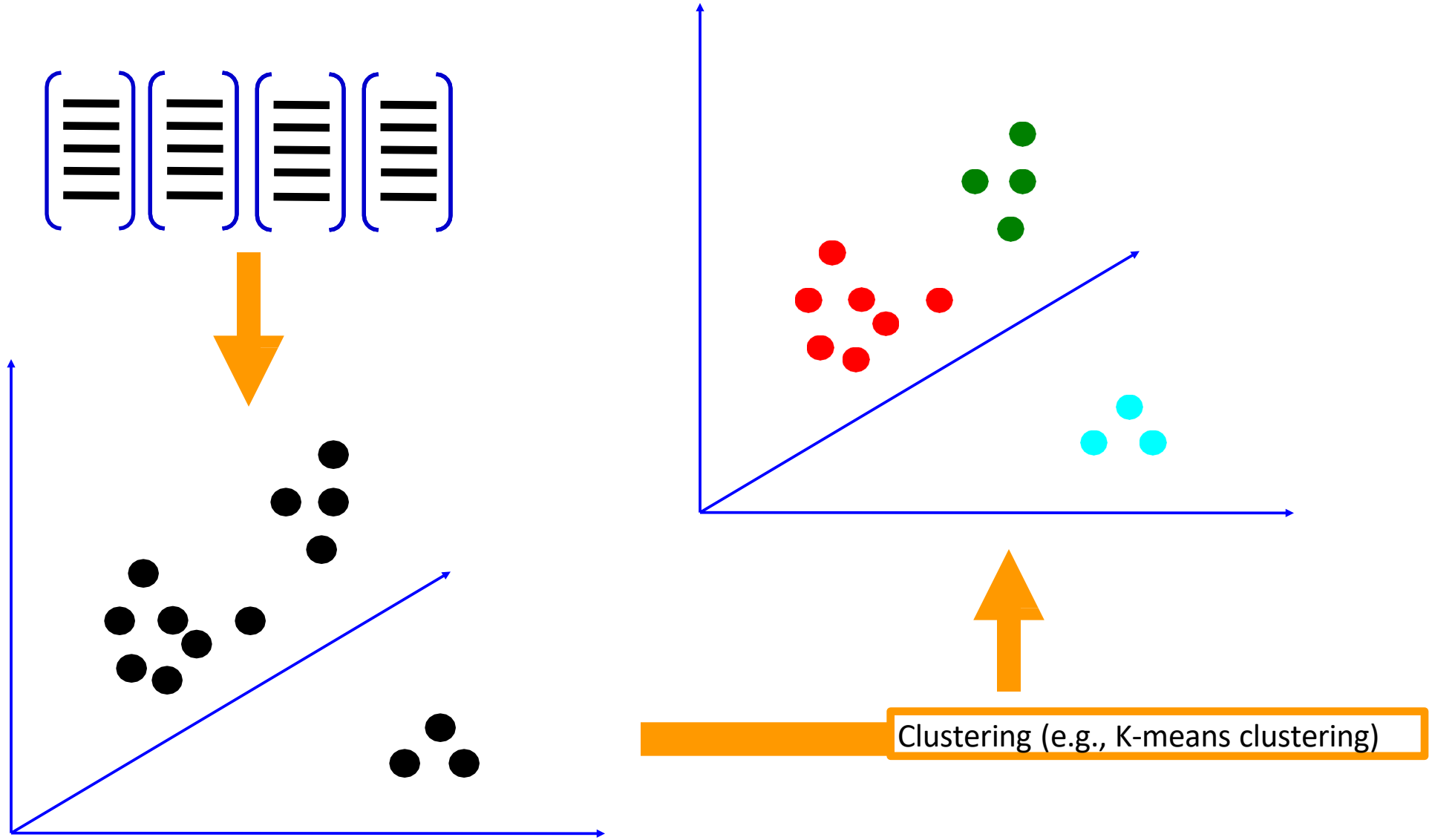


Compute descriptors

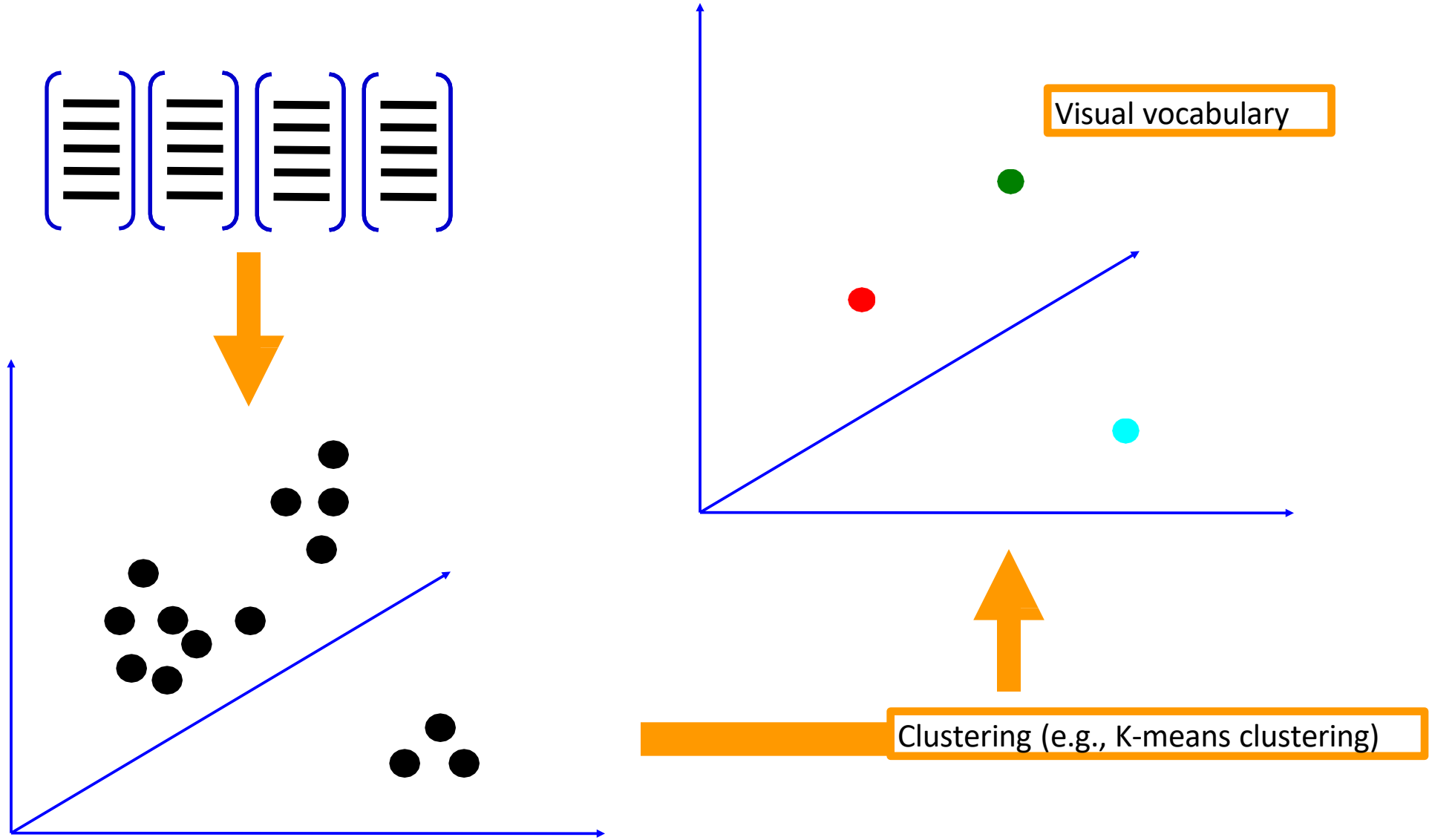


Detect features

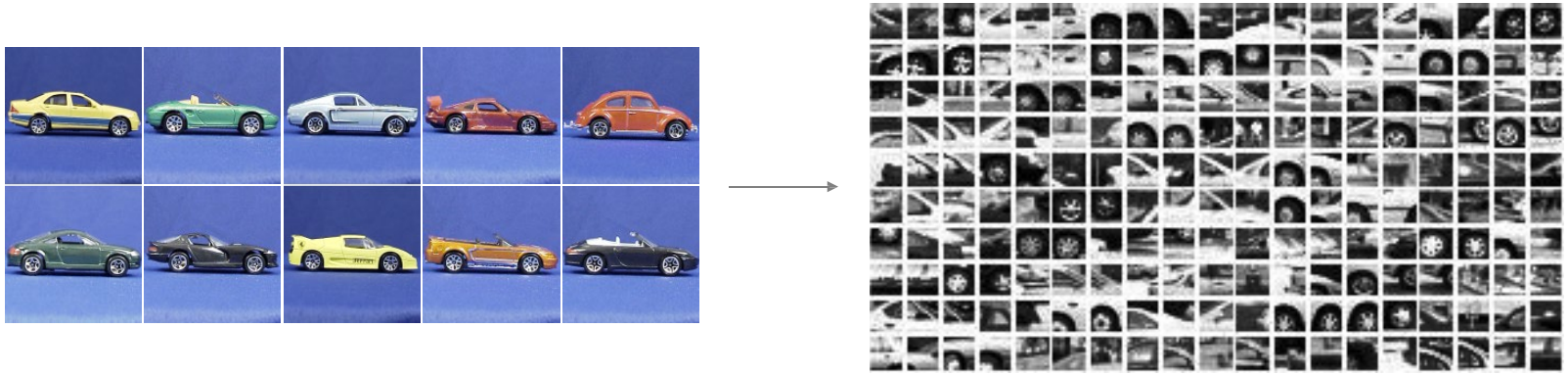
How Do We Learn the Dictionary?



How Do We Learn the Dictionary?

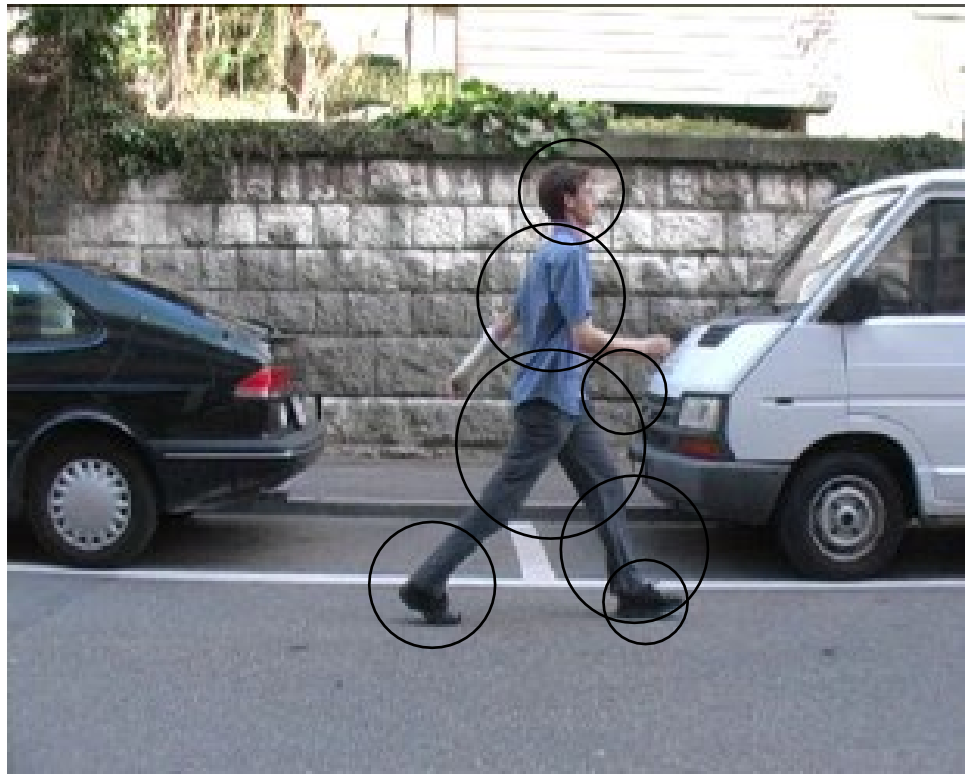


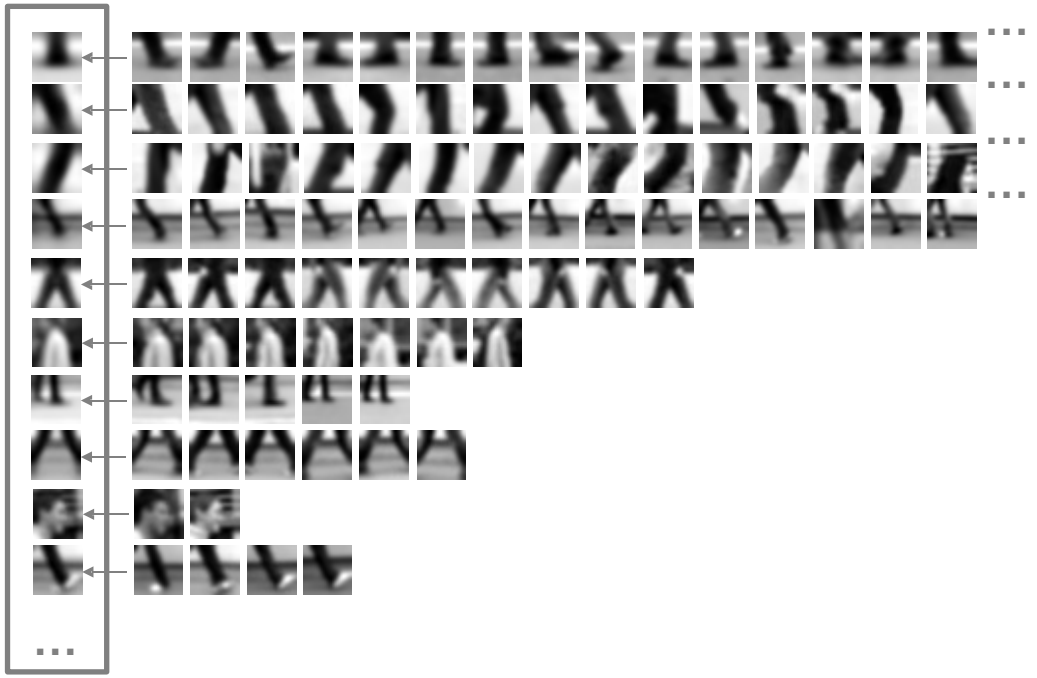
Example of Visual Dictionary



Source: B. Leibe

Another Dictionary Example





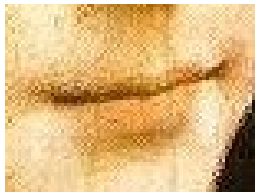
Standard BOW Pipeline

Dictionary Learning:
Learn Visual Words using clustering

Encode:
Build Bags-of-Words (BOW) vectors for each image

Classify:
Train and test data using BOWs

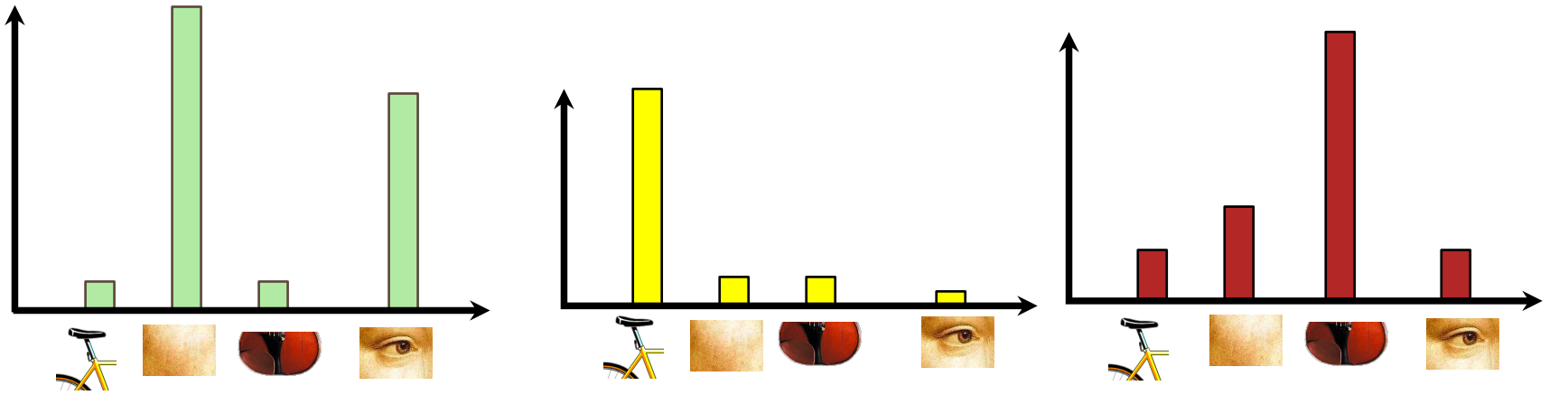
Encode:
Build Bags-of-Words (BOW) vectors for each image

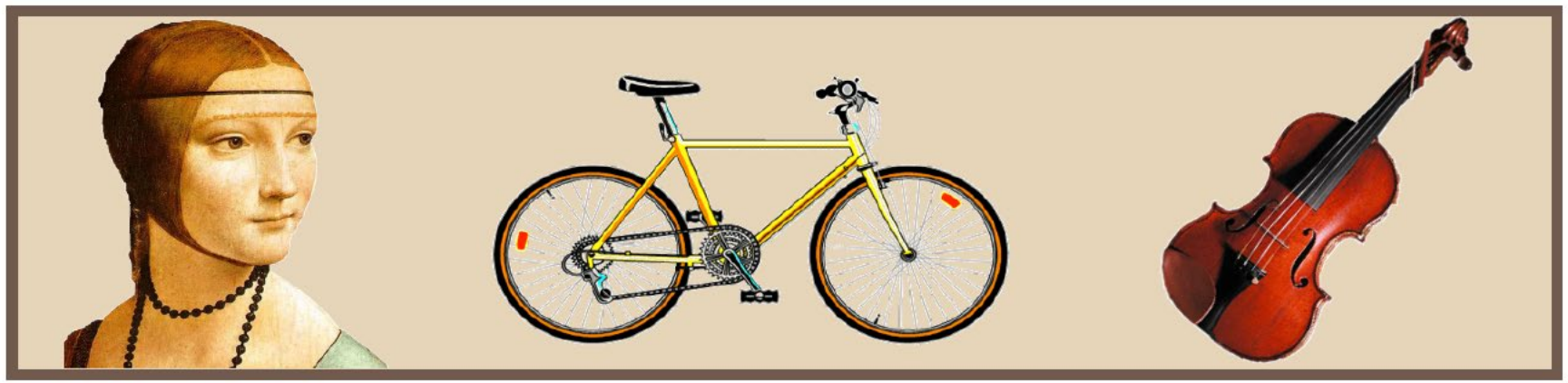


1. Quantization: image features gets associated to a visual word (nearest cluster center)

Encode:
Build Bags-of-Words (BOW) vectors for each image

2. Histogram: count the number of visual word occurrences





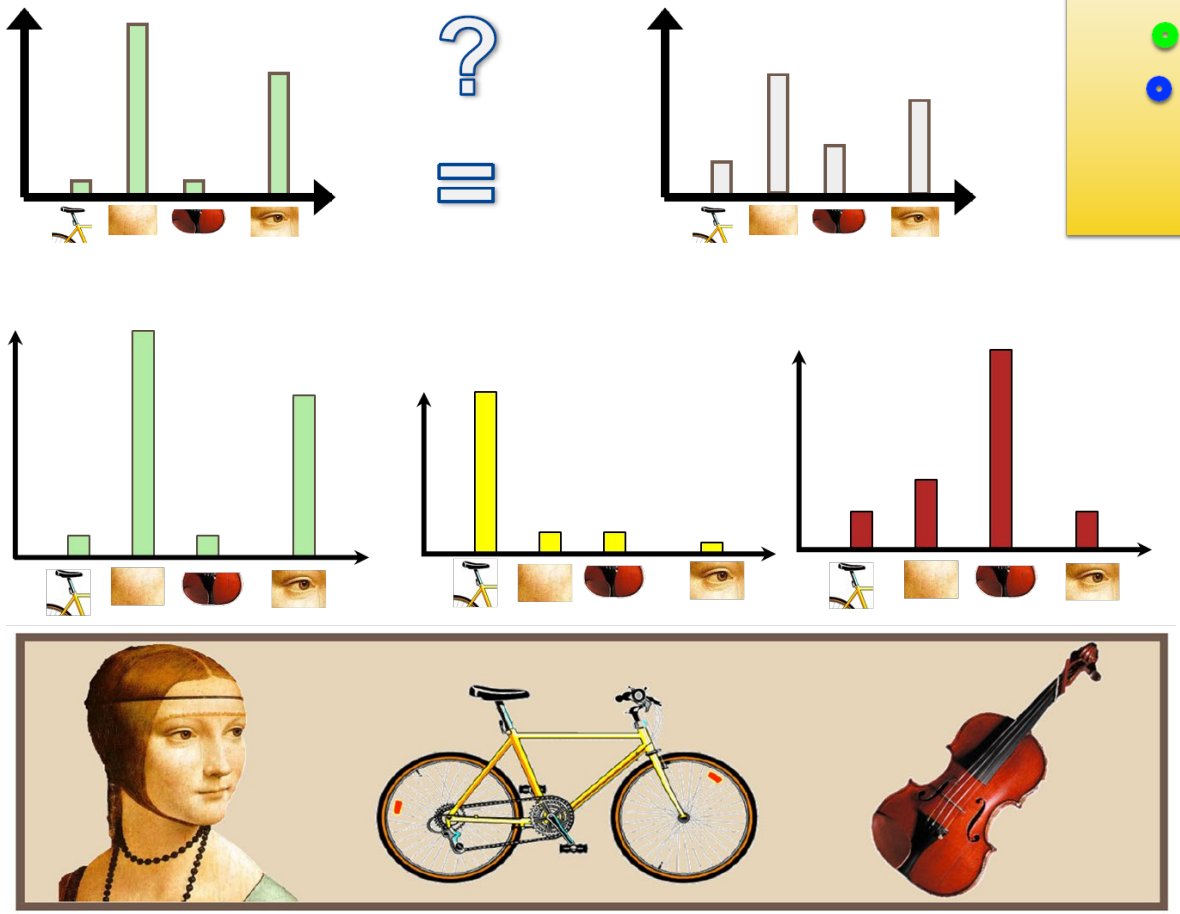
Standard BOW Pipeline

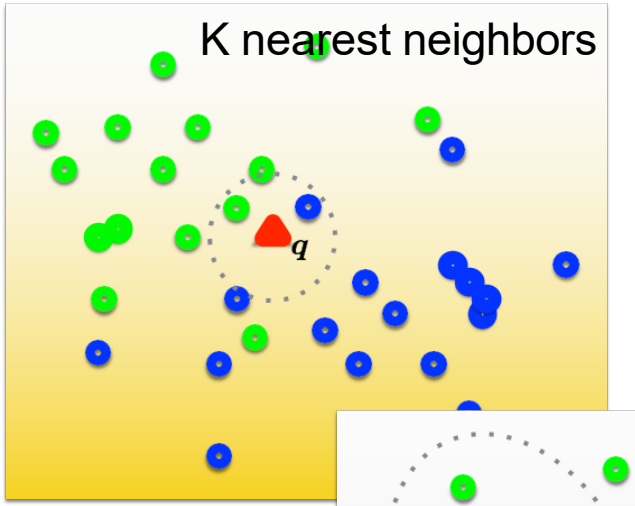
Dictionary Learning:
Learn Visual Words using clustering

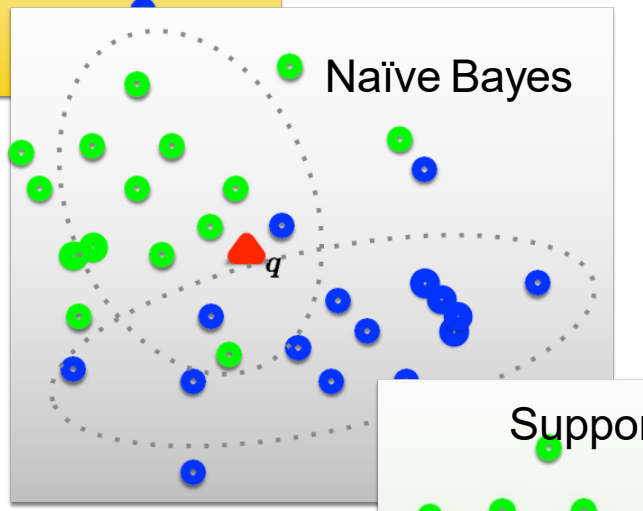
Encode:
Build Bags-of-Words (BOW) vectors for each image

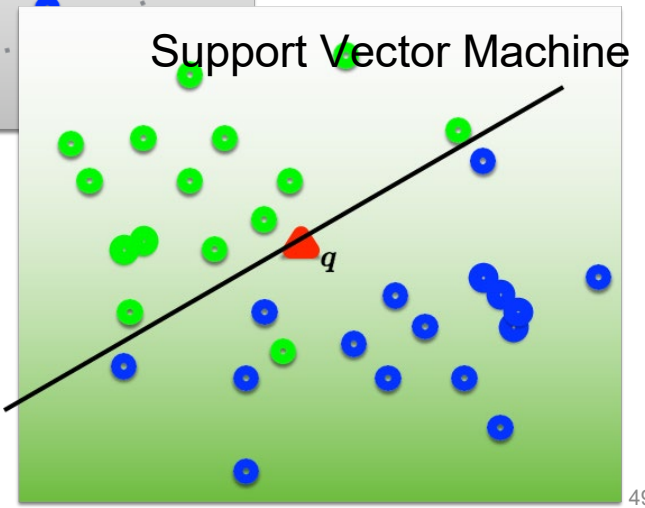
Classify:
Train and test data using BOWs

Classification Methods For BOW



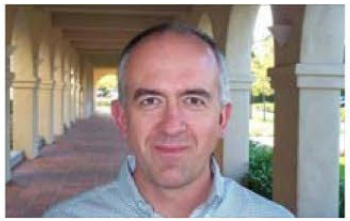






CalTech-256 Dataset



class	bag of features	bag of features	Parts-and-shape model
	Zhang et al. (2005)	Willamowski et al. (2004)	Fergus et al. (2003)
airplanes	**98.8**	97.1	90.2
cars (rear)	98.3	**98.6**	90.3
cars (side)	**95.0**	87.3	88.5
faces	**100**	99.3	96.4
motorbikes	**98.5**	98.0	92.5
spotted cats	**97.0**	—	90.0

- Caltech-256 object recognition dataset contains 30,607 real-world images
- Spans 257 classes (256 objects and a clutter class)
- Bag of features works pretty well for image-level classification!

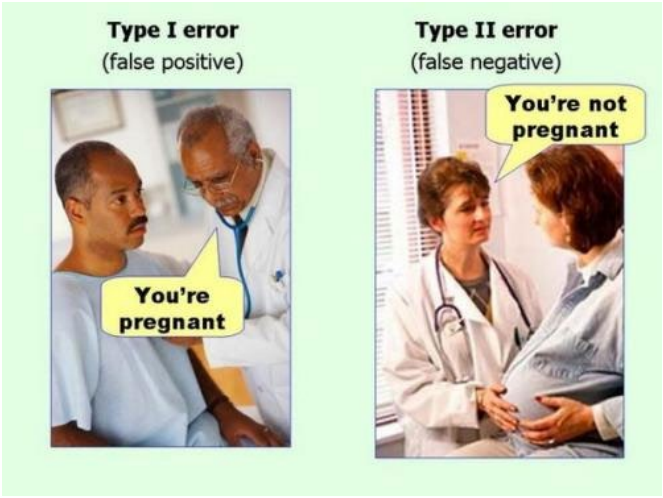
Csurka et al. (2004), Willamowski et al. (2005), Grauman & Darrell (2005), Sivic et al. (2003, 2005)

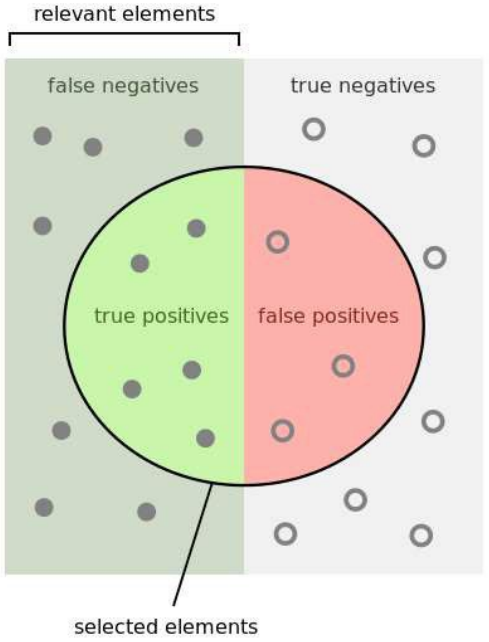
How to score the retrieval results?

Precision and Recall

True positive (tp) – correct attribution
True negative (tn) – correct rejection

False positive (fp) – incorrect attribution
False negative (fn) – incorrect rejection



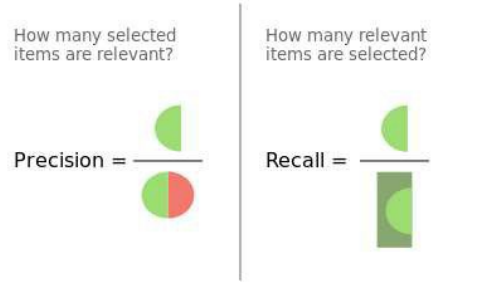


$$\text{Precision} = \frac{tp}{tp + fp}$$

Precision =
#relevant / #returned

$$\text{Recall} = \frac{tp}{tp + fn}$$

Recall =
#relevant / #total relevant

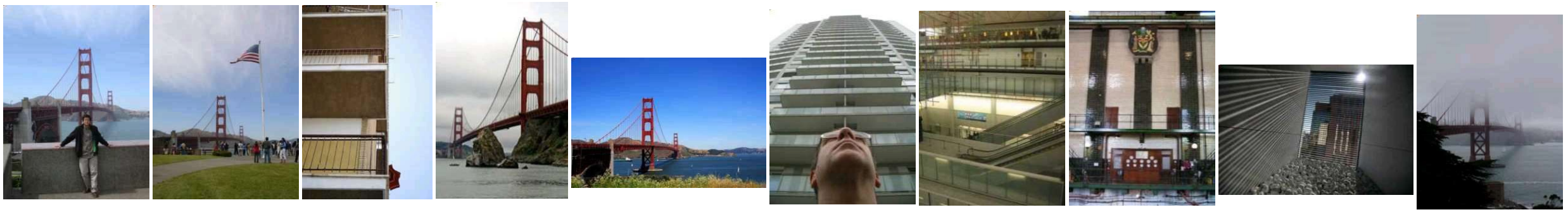


https://commons.wikimedia.org/w/index.php?curid=36926283

Scoring Retrieval Quality Example

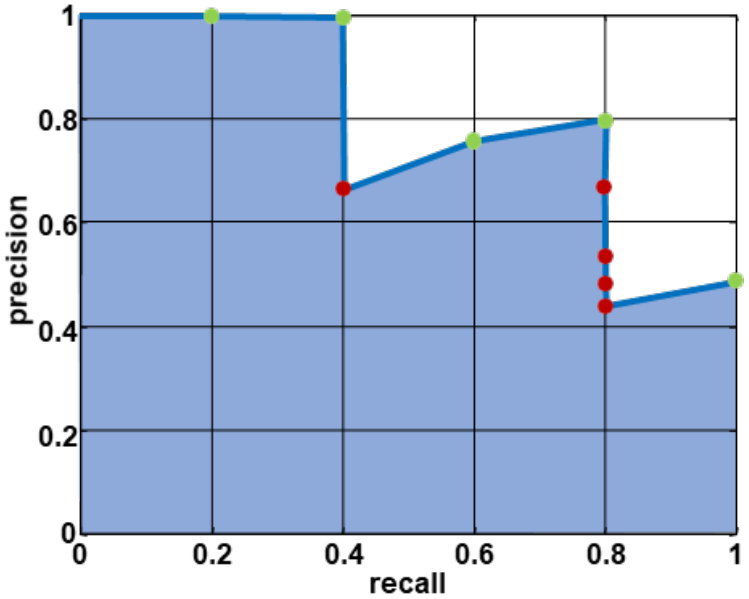
Query

Results (ordered):

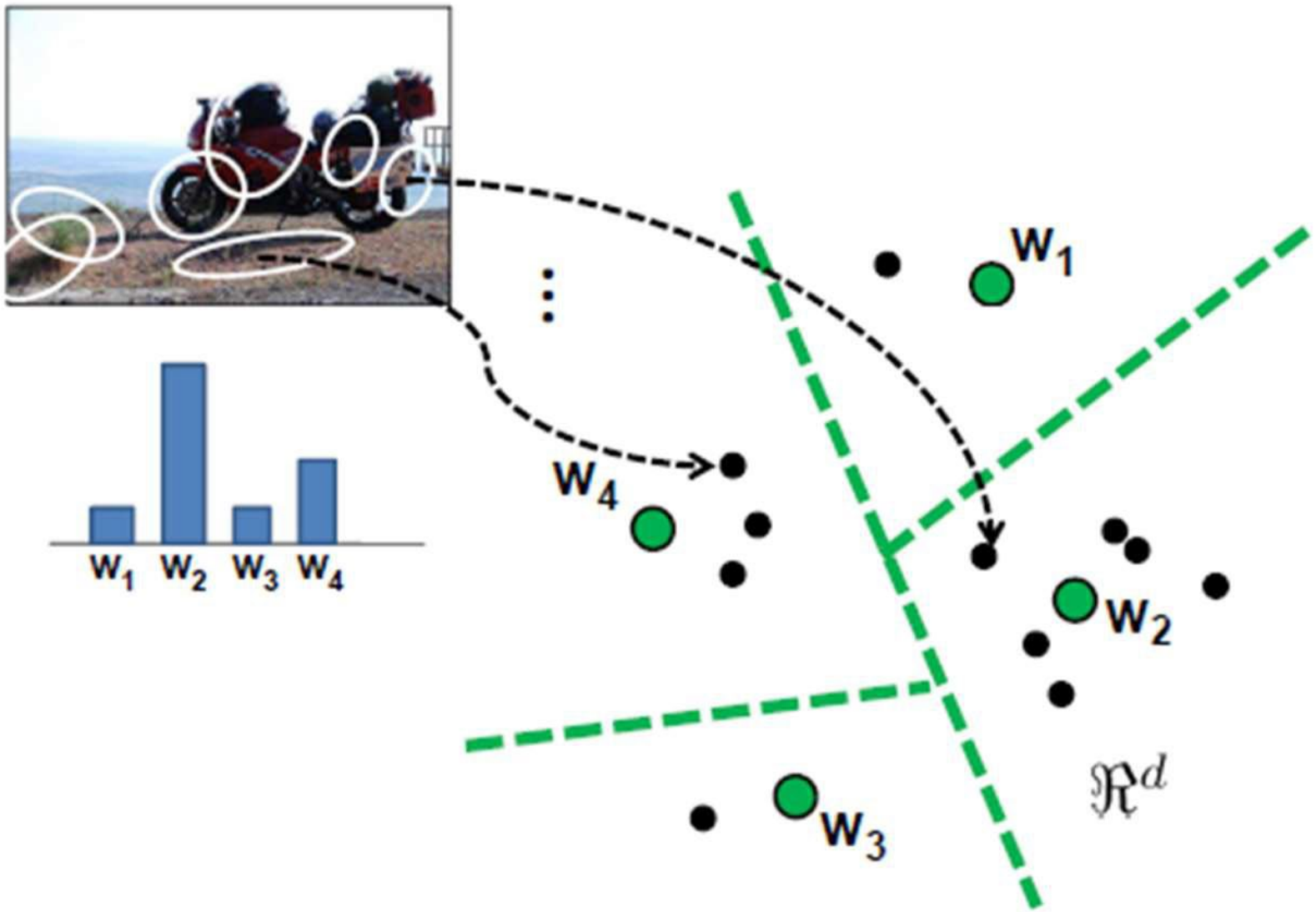


Database size: 10 images
Relevant (total): 5 images

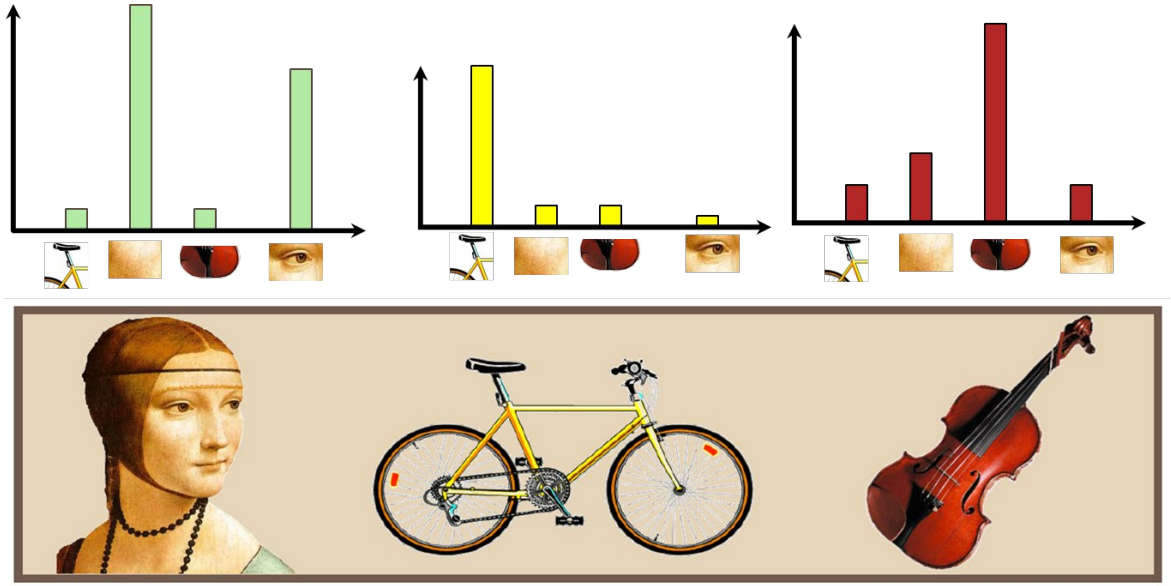
Precision = #relevant / #returned
Recall = #relevant / #total relevant



Standard K-means Bag of Words



Better Bags of Visual Features



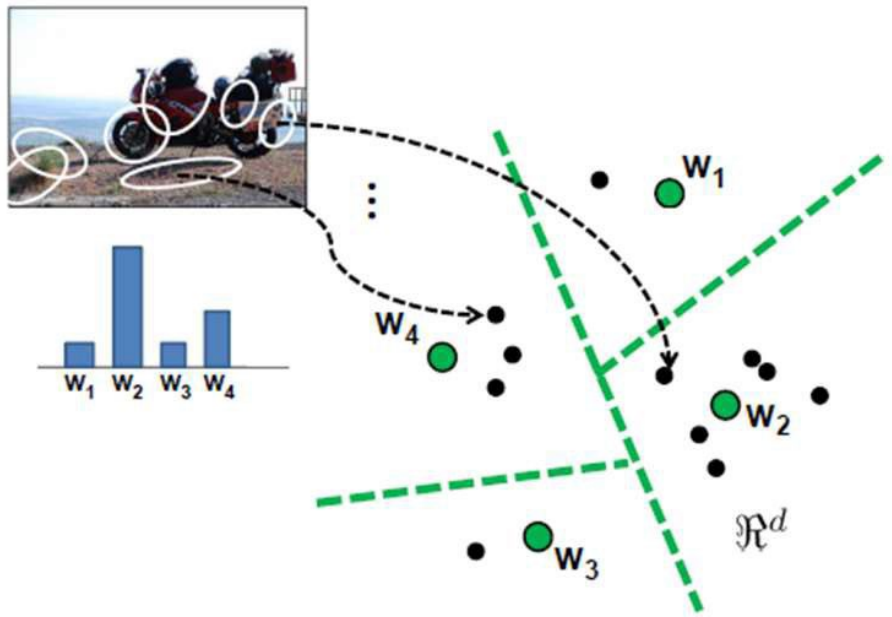
- More advanced quantization / encoding methods
  - Mixtures of Gaussians
  - Soft assignment (a.k.a. Kernel Codebook)
  - VLAD – Vectors of Locally-Aggregated Descriptors

- Deep learning has taken attention away from these methods…

Motivation

- Bag of Visual Words is only about **counting** the number of local descriptors assigned to each Voronoi region

- Why not including **other statistics**?

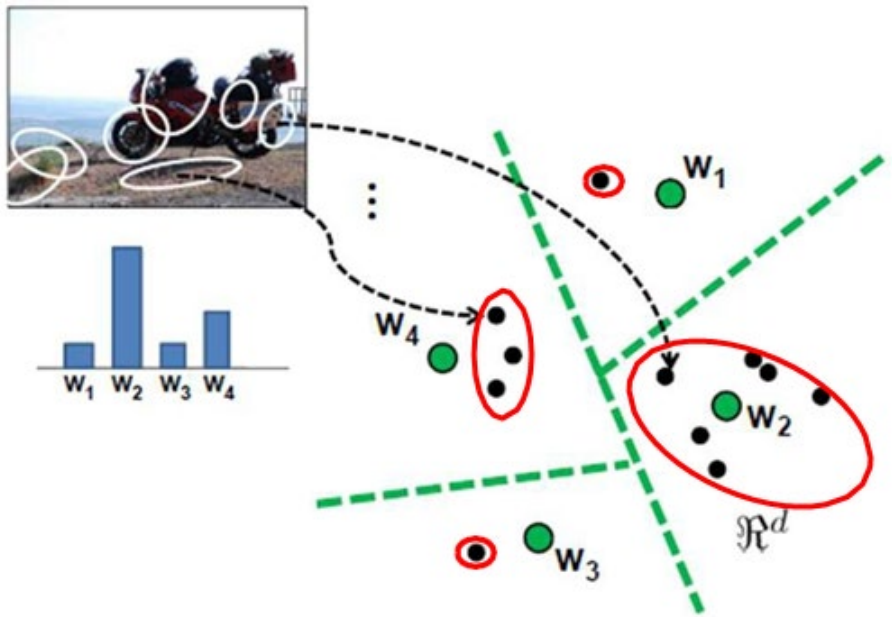


Motivation

- Bag of Visual Words is only about **counting** the number of local descriptors assigned to each Voronoi region

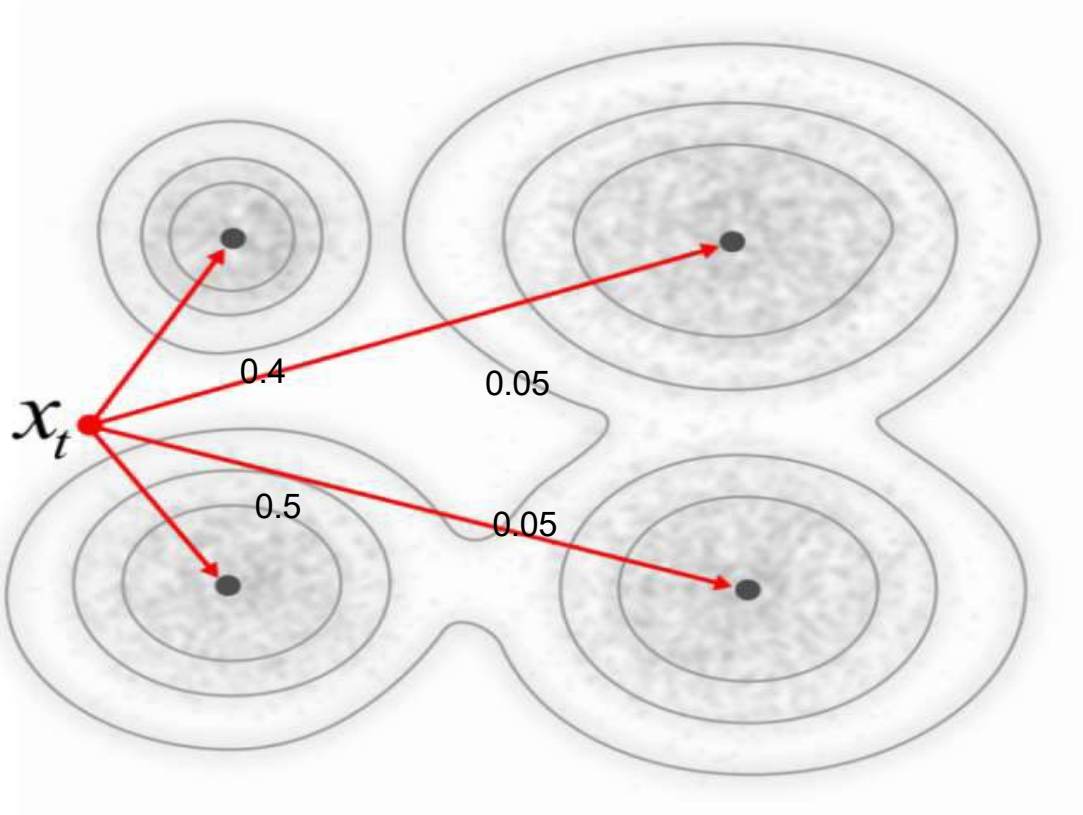
- Why not including **other statistics**?

- For instance:
  - Mean of local descriptors
  - (Co)variance of local descriptors



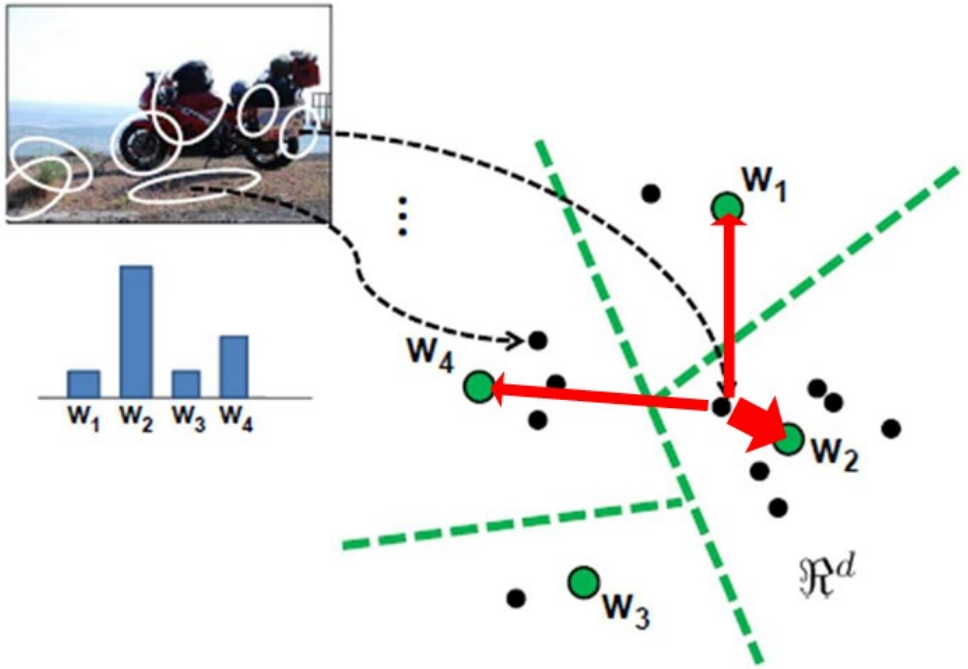
Gaussian Mixture Model (GMM)

- GMM can be thought of as “soft” k-means.
- Each component has a mean and a standard deviation along each direction (or full covariance)
- Can easily represent non-circular distributions



Simple case: Soft Assignment

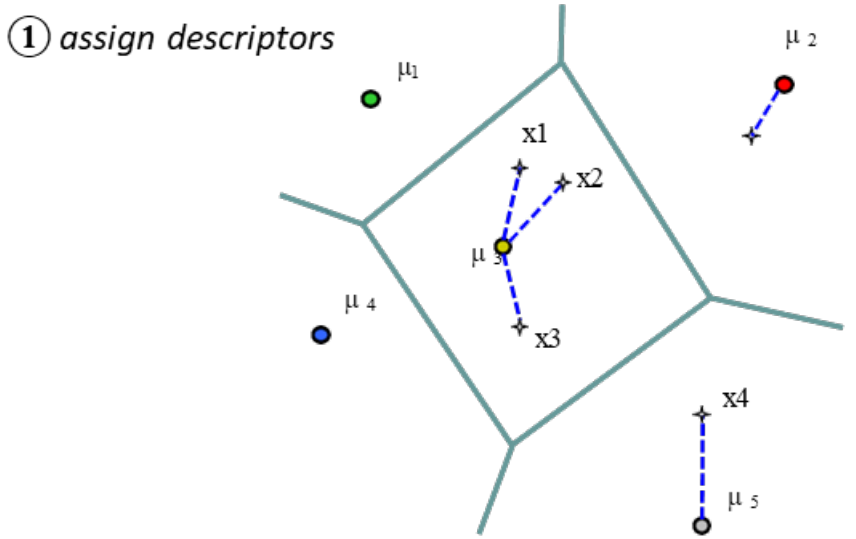
- “Kernel codebook encoding” by Chatfield et al. 2011
- Cast a set of proportional votes (weights) to *n* most similar clusters, rather than a single ‘hard’ vote
- This is fast and easy to implement, but it makes an inverted file index *less sparse*



VLAD – Vectors of Locally-Aggregated Descriptors

Given a codebook $\{\mu_i, i = 1 \dots N\}$, e.g. learned with K-means, and a set of local descriptors $X = \{x_t, t = 1 \dots T\}$:

① assign: $\text{NN}(x_t) = \arg\min_{\mu_i} ||x_t - \mu_i||$



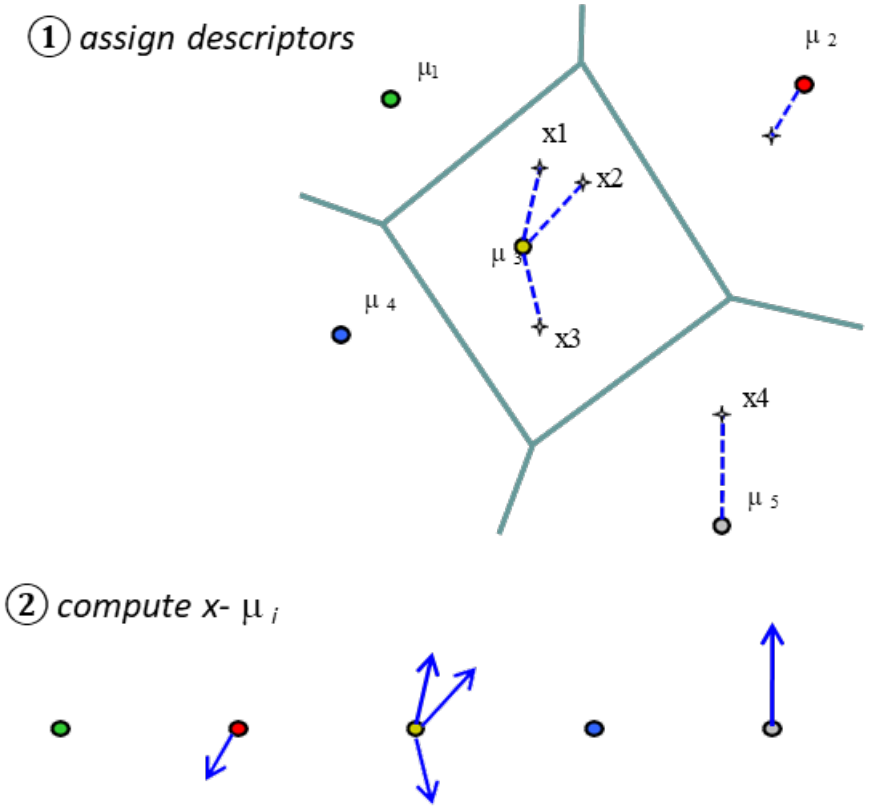
Jégou, Douze, Schmid and Pérez. "Aggregating local descriptors into a compact image representation", CVPR'10.

VLAD – Vectors of Locally-Aggregated Descriptors

Given a codebook $\{\mu_i, i = 1 \dots N\}$, e.g. learned with K-means, and a set of local descriptors $X = \{x_t, t = 1 \dots T\}$:

① assign: $$\mathrm{NN}(x_t) = \arg\min_{\mu_i} ||x_t - \mu_i||$$

②③ compute: $$v_i = \sum_{x_t : \mathrm{NN}(x_t) = \mu_i} x_t - \mu_i$$



Jégou, Douze, Schmid and Pérez. "Aggregating local descriptors into a compact image representation", CVPR'10.

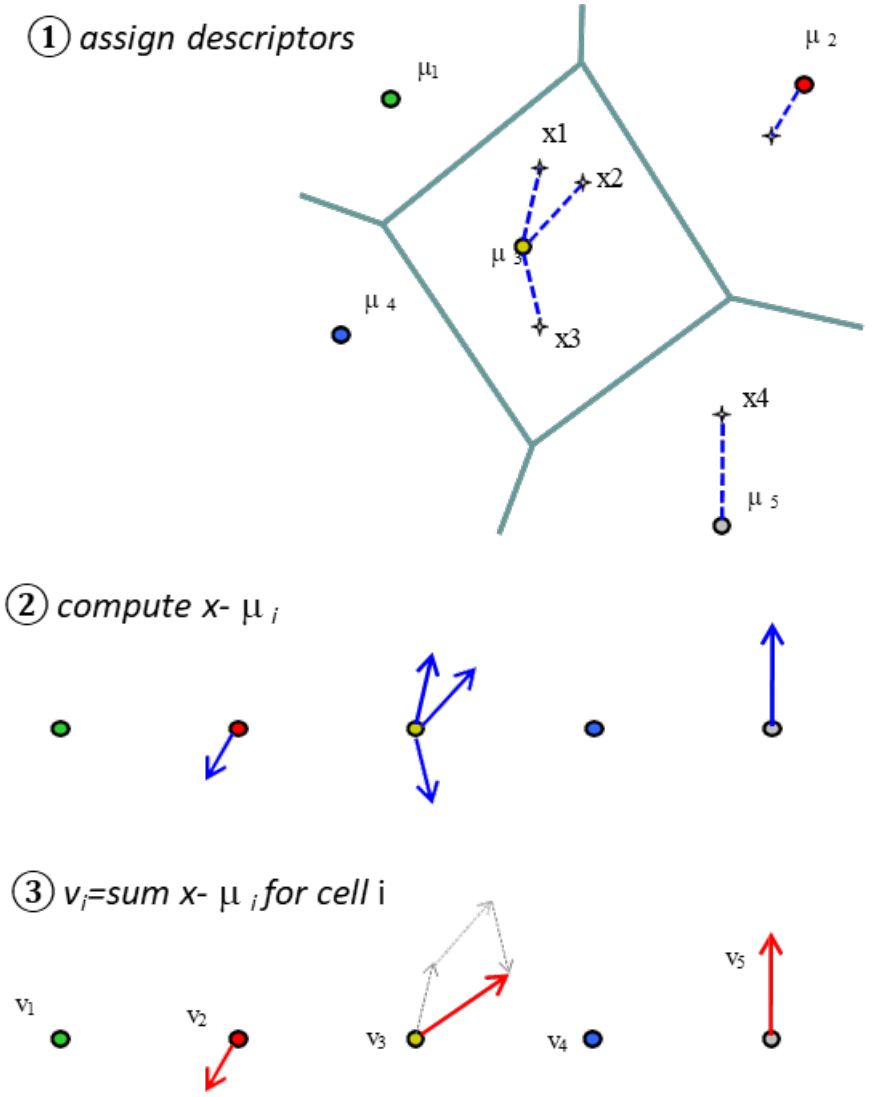
VLAD – Vectors of Locally-Aggregated Descriptors

Given a codebook $\{\mu_i, i = 1 \dots N\}$, e.g. learned with K-means, and a set of local descriptors $X = \{x_t, t = 1 \dots T\}$:

① assign: $\mathrm{NN}(x_t) = \arg\min_{\mu_i} ||x_t - \mu_i||$

②③ compute: $v_i = \sum_{x_t : \mathrm{NN}(x_t) = \mu_i} x_t - \mu_i$

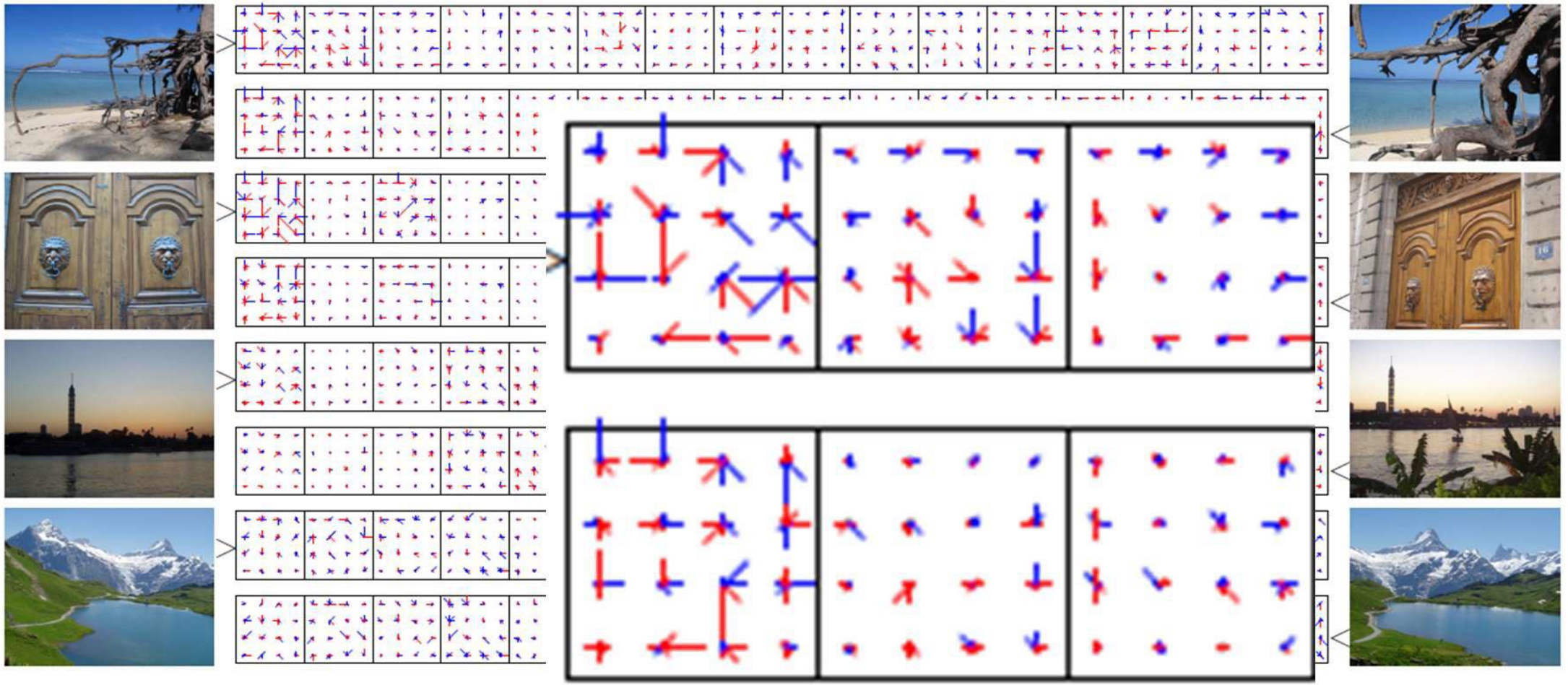
concatenate v_i's + ℓ_2 normalize



Jégou, Douze, Schmid and Pérez."Aggregating local descriptors into a compact image representation", CVPR'10.

Example

A graphical representation of $$v_i = \sum_{x_t : \mathrm{NN}(x_t) = \mu_i} x_t - \mu_i$$



Global Image Descriptors

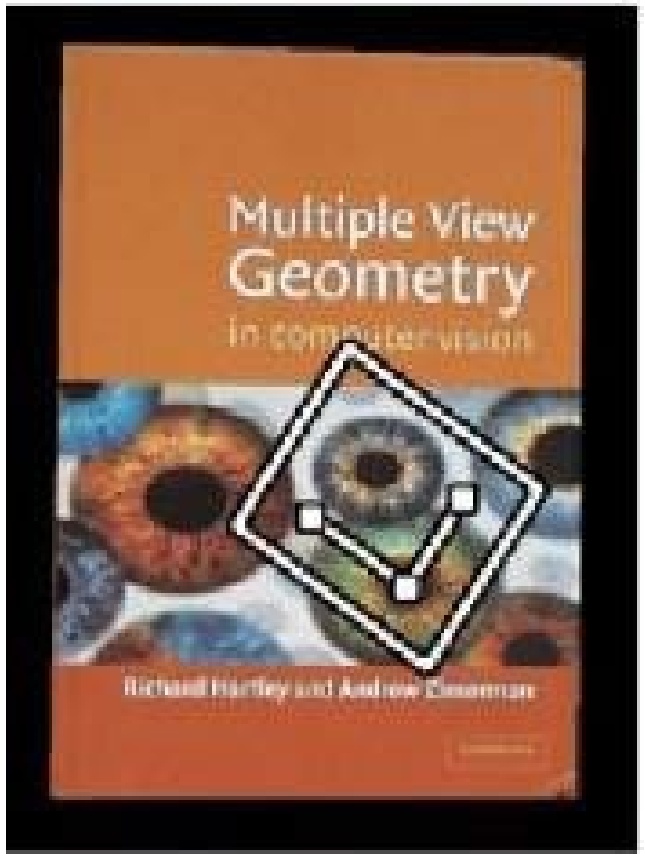
- Tiny images (Torralba et al, 2008)
- Color histograms
- Self-similarity (Shechtman and Irani, 2007)
- Geometric class layout (Hoiem et al, 2005)
- Geometry-specific histograms (Lalonde et al, 2007)
- Dense and Sparse SIFT histograms
- Berkeley texton histograms (Martin et al, 2001)
- HoG 2x2 spatial pyramids
- Gist scene descriptor (Oliva and Torralba, 2008)

Texture Features


How do we describe an image patch?
Patches with similar content should have similar descriptors.

Photometric transformations

Geometric transformations



objects will appear at different scales,
translation and rotation

What is the best descriptor for an image feature?

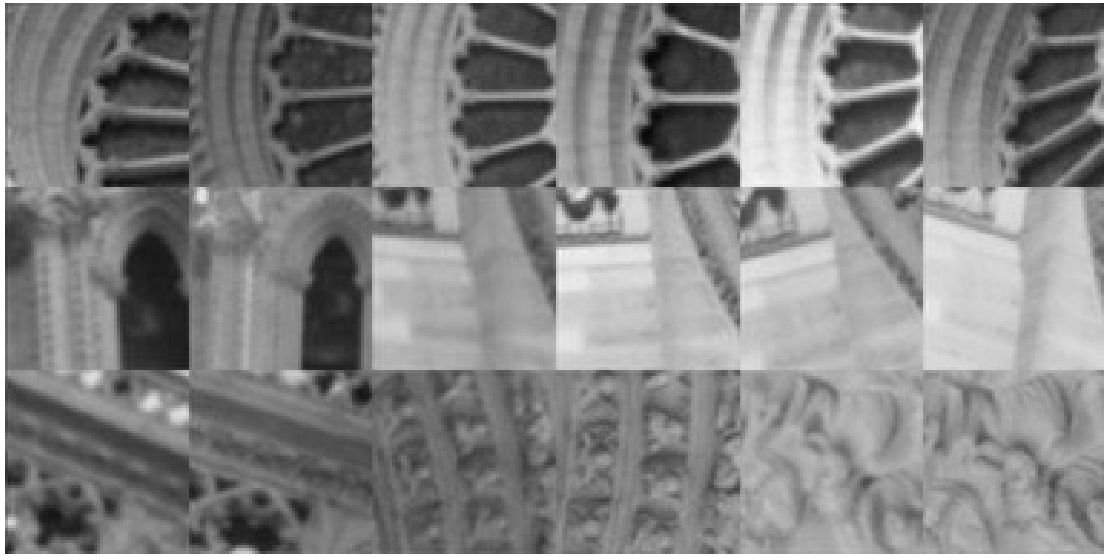


Image patch

Just use the pixel values of the patch!

Perfectly fine if geometry and appearance is unchanged

(a.k.a. template matching)

Tiny Images

Tiny Images

Just down-sample it!
Simple, fast, robust to small affine transforms.

Image patch

Just use the pixel values of the patch!

Perfectly fine if geometry and appearance is unchanged

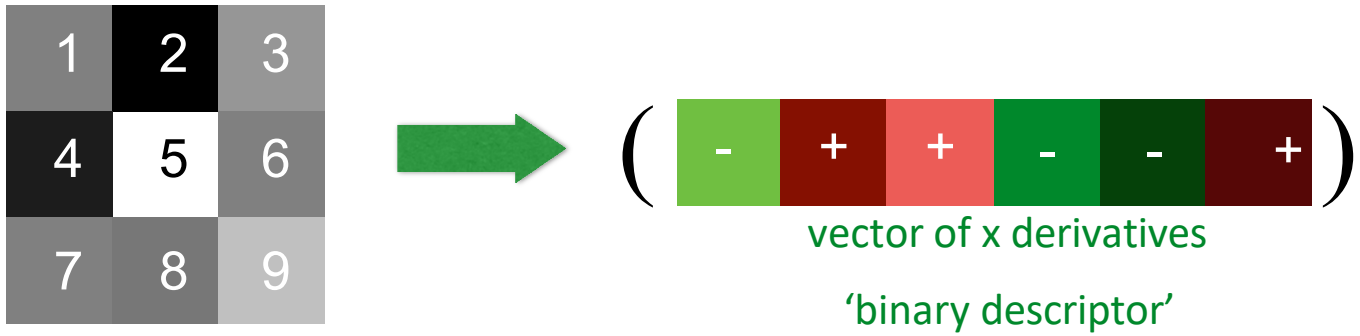
(a.k.a. template matching)

What are the problems?

How can you be less sensitive to absolute intensity values?

Image gradients

Use pixel differences



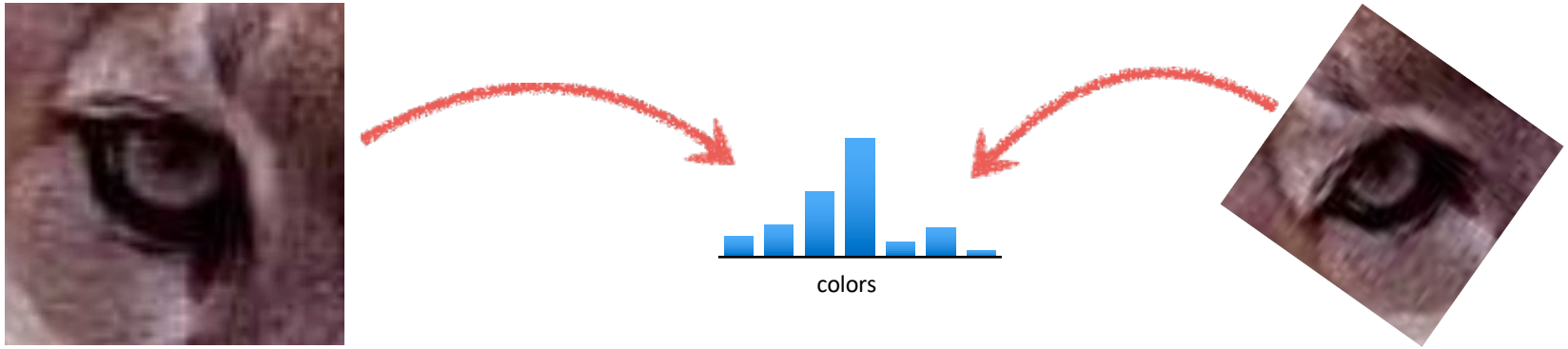
Feature is invariant to absolute intensity values

What are the problems?

How can you be less sensitive to deformations?

Color histogram

Count the colors in the image using a histogram

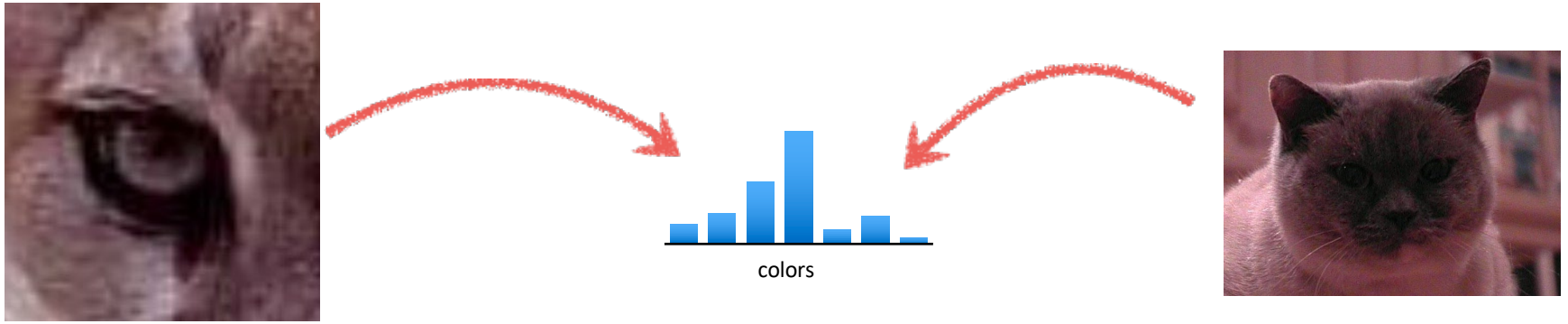


Invariant to changes in scale and rotation

What are the problems?

Color histogram

Count the colors in the image using a histogram



Invariant to changes in scale and rotation

What are the problems?

How can you be more sensitive to spatial layout?

GIST

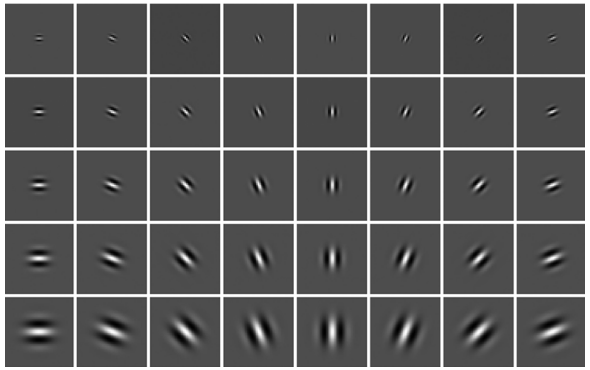
1. Compute filter responses (filter bank of Gabor filters)

2. Divide image patch into 4 x 4 cells

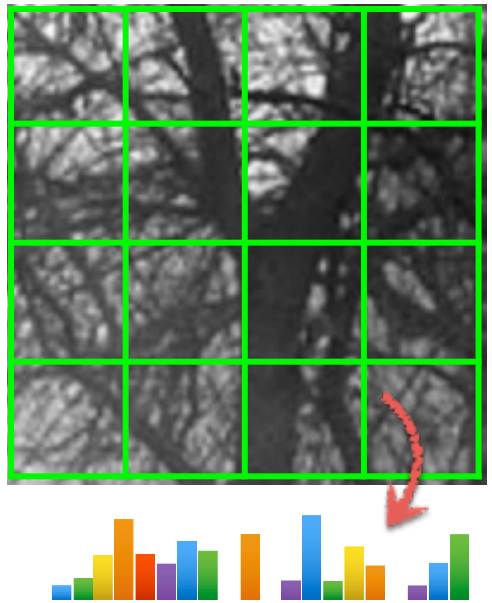
3. Compute filter response averages for each cell

4. Size of descriptor is 4 x 4 x N, where N is the size of the filter bank

Filter bank

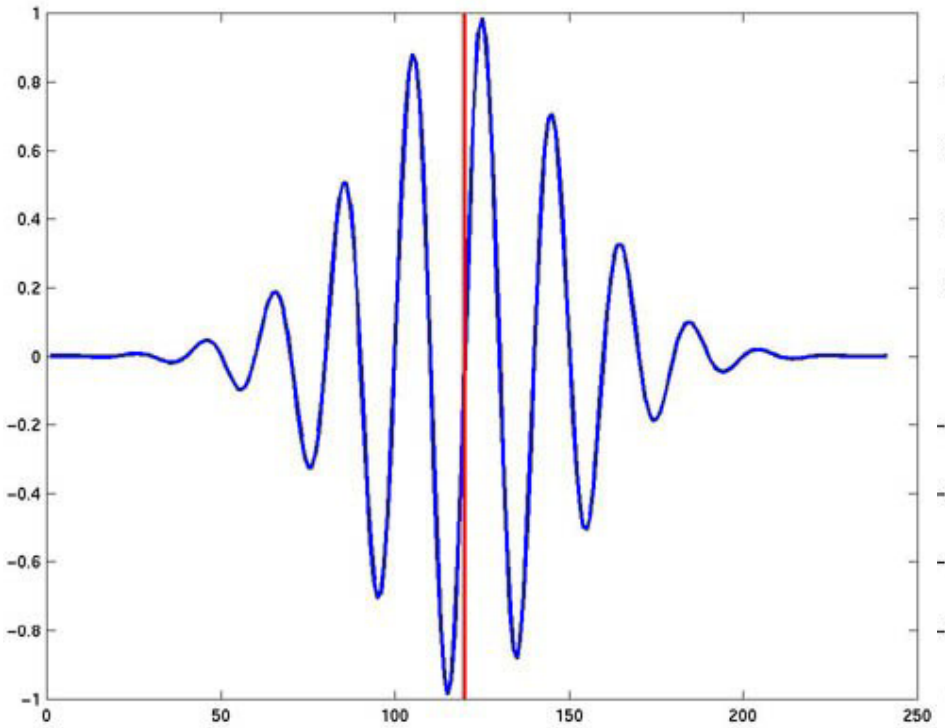


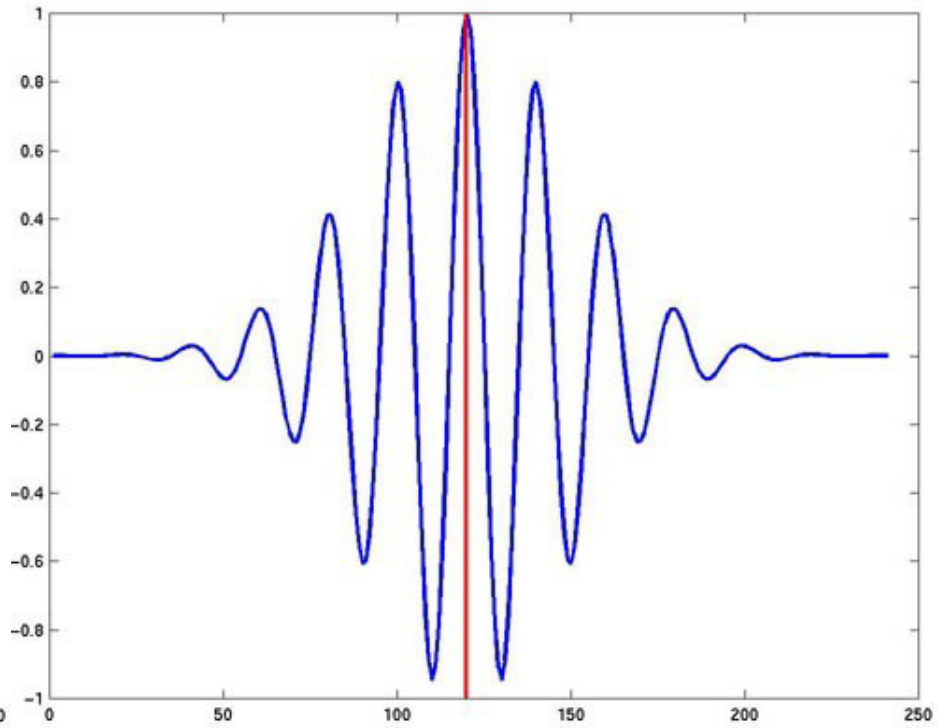
4 x 4 cell



averaged filter responses

Gabor Filters
(1D examples)

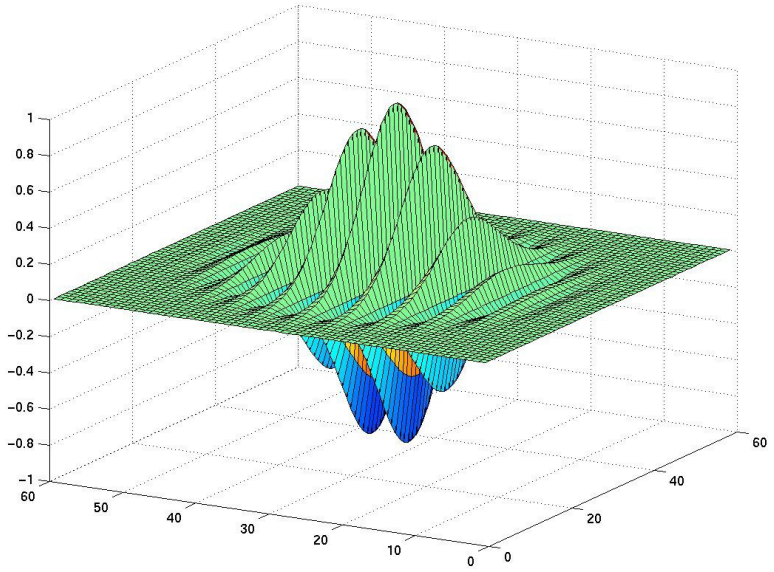


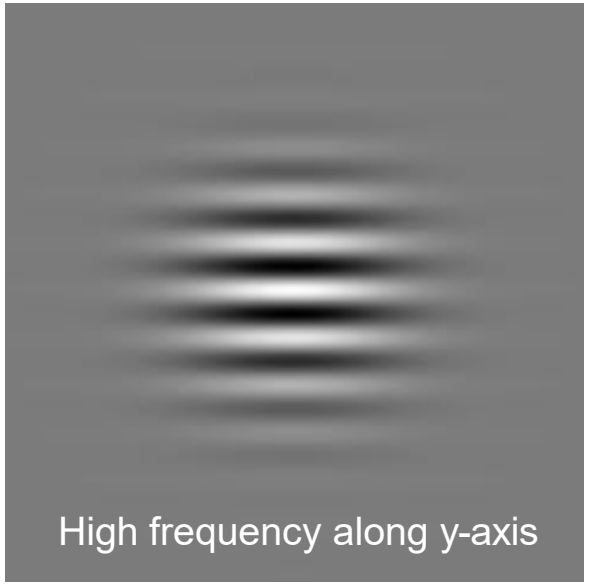


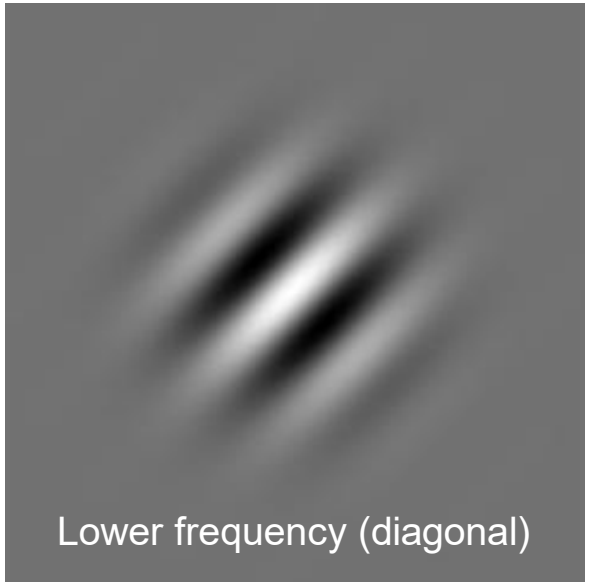
$$e^{-\frac{x^2}{2\sigma^2}} \sin(2\pi\omega x) \qquad e^{-\frac{x^2}{2\sigma^2}} \cos(2\pi\omega x)$$

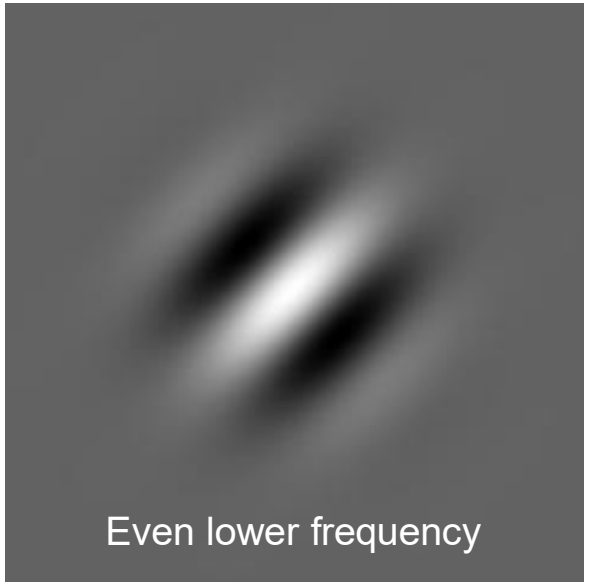
2D Gabor Filters

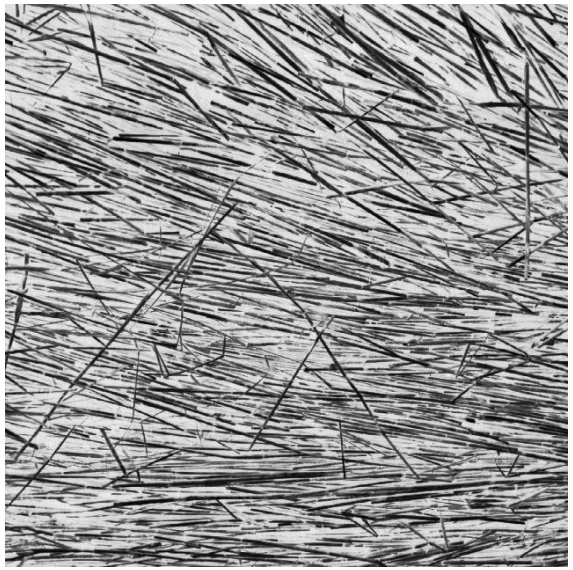
$$e^{-\frac{x^2+y^2}{2\sigma^2}} \cos\left(2\pi\left(k_x x + k_y y\right)\right)$$

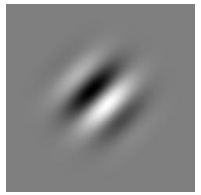


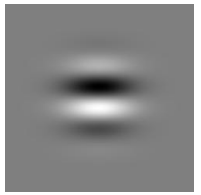

High frequency along y-axis

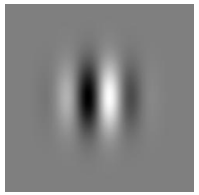

Lower frequency (diagonal)

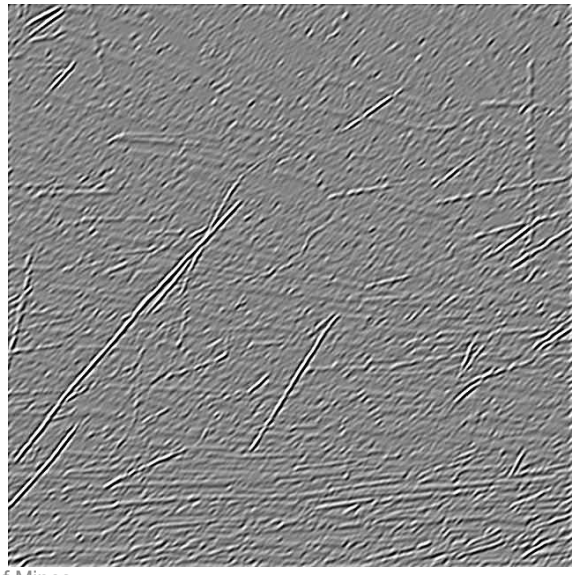

Even lower frequency

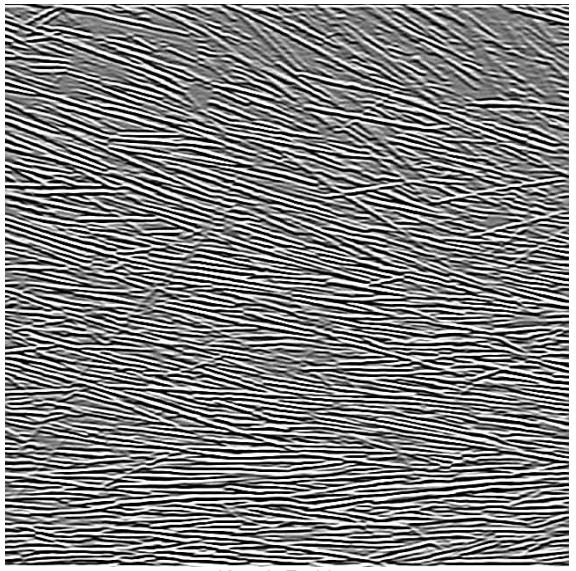


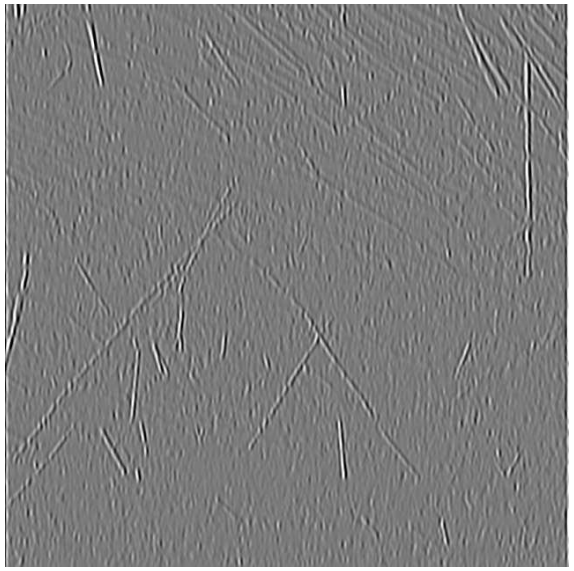




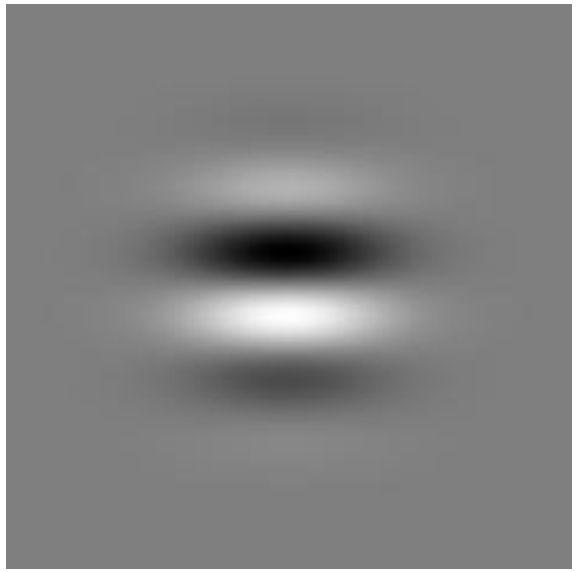


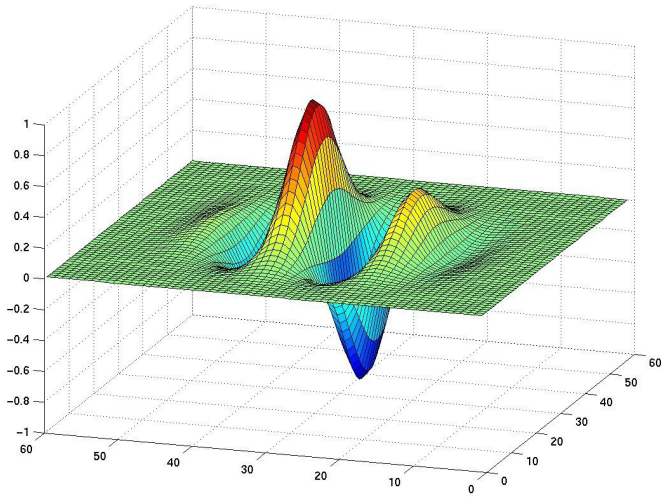




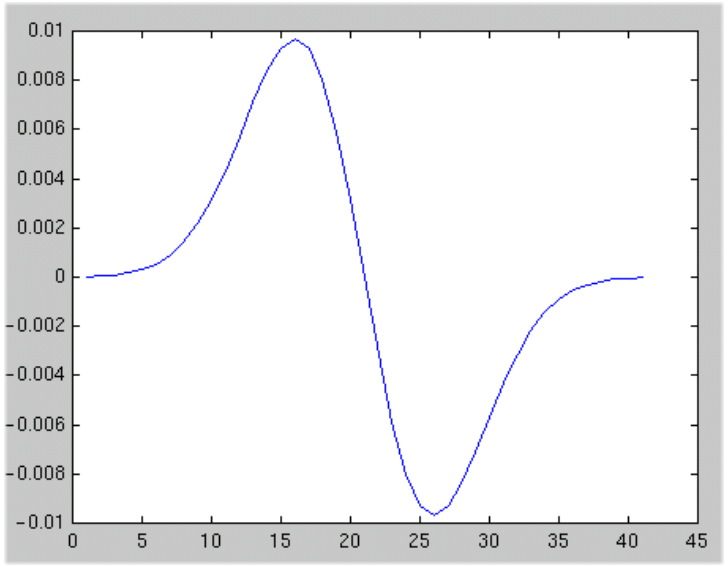


Odd Gabor filter

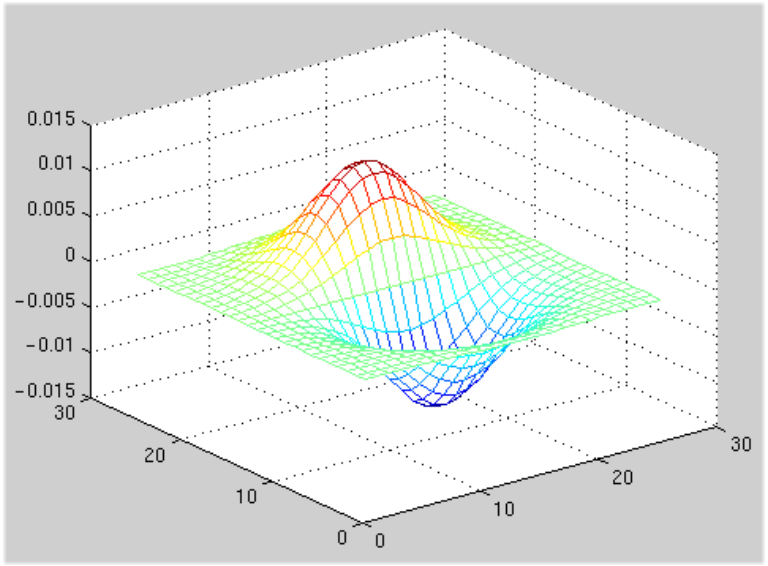




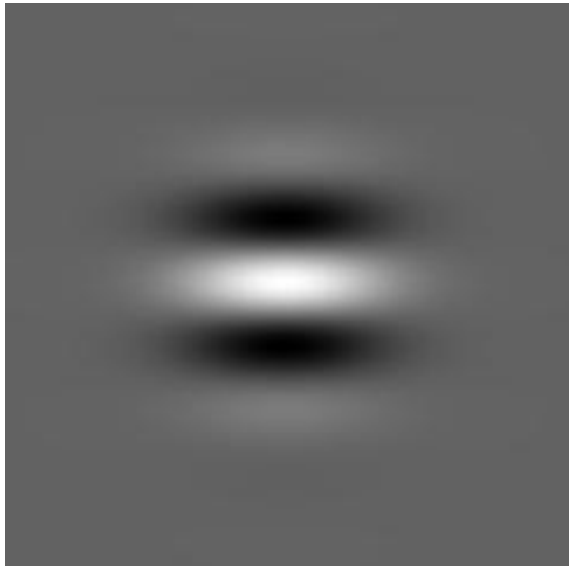
… looks a lot like…

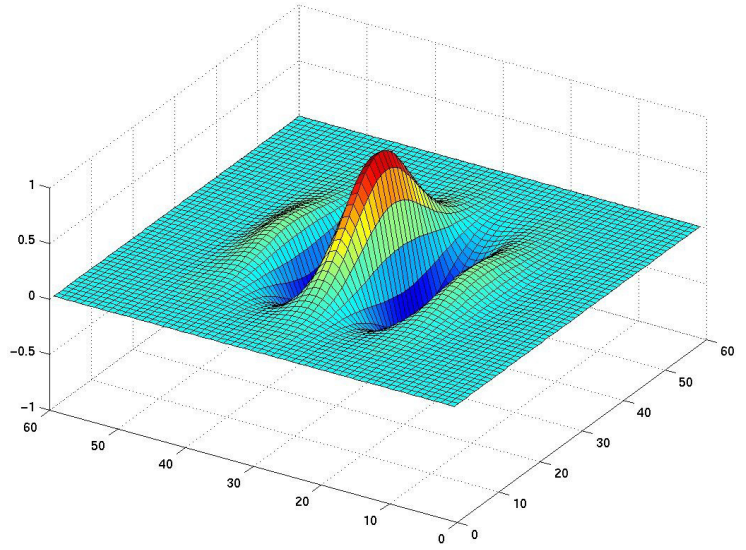


Gaussian Derivative

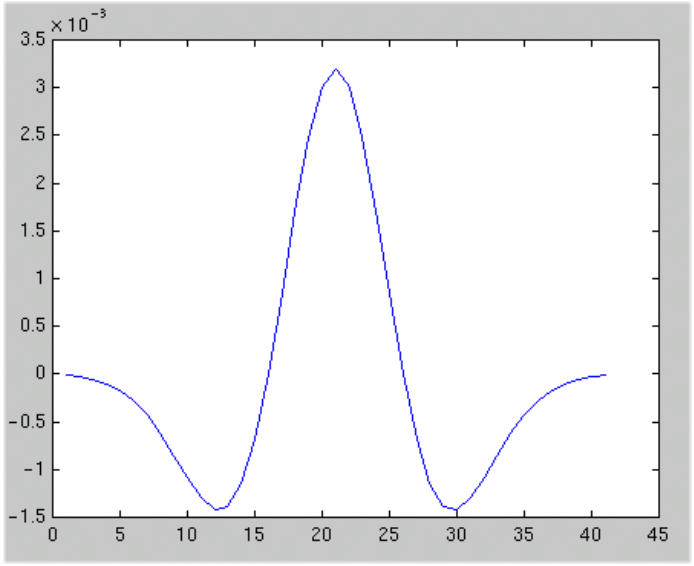


Even Gabor filter

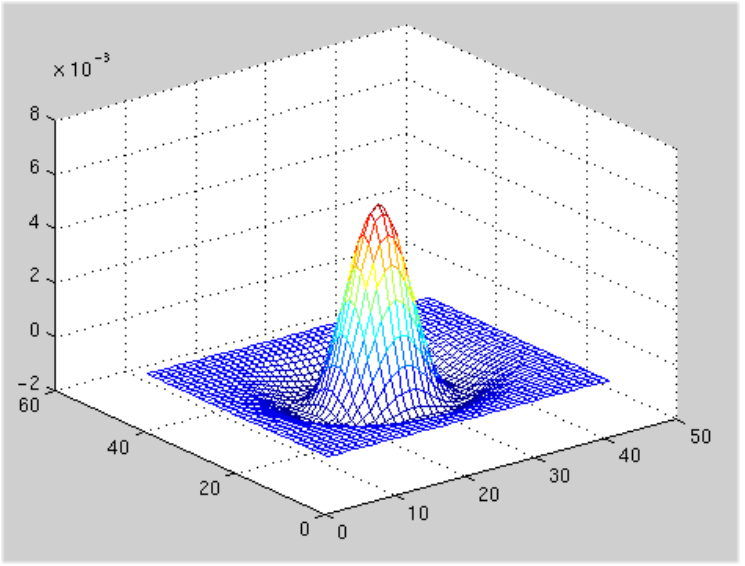




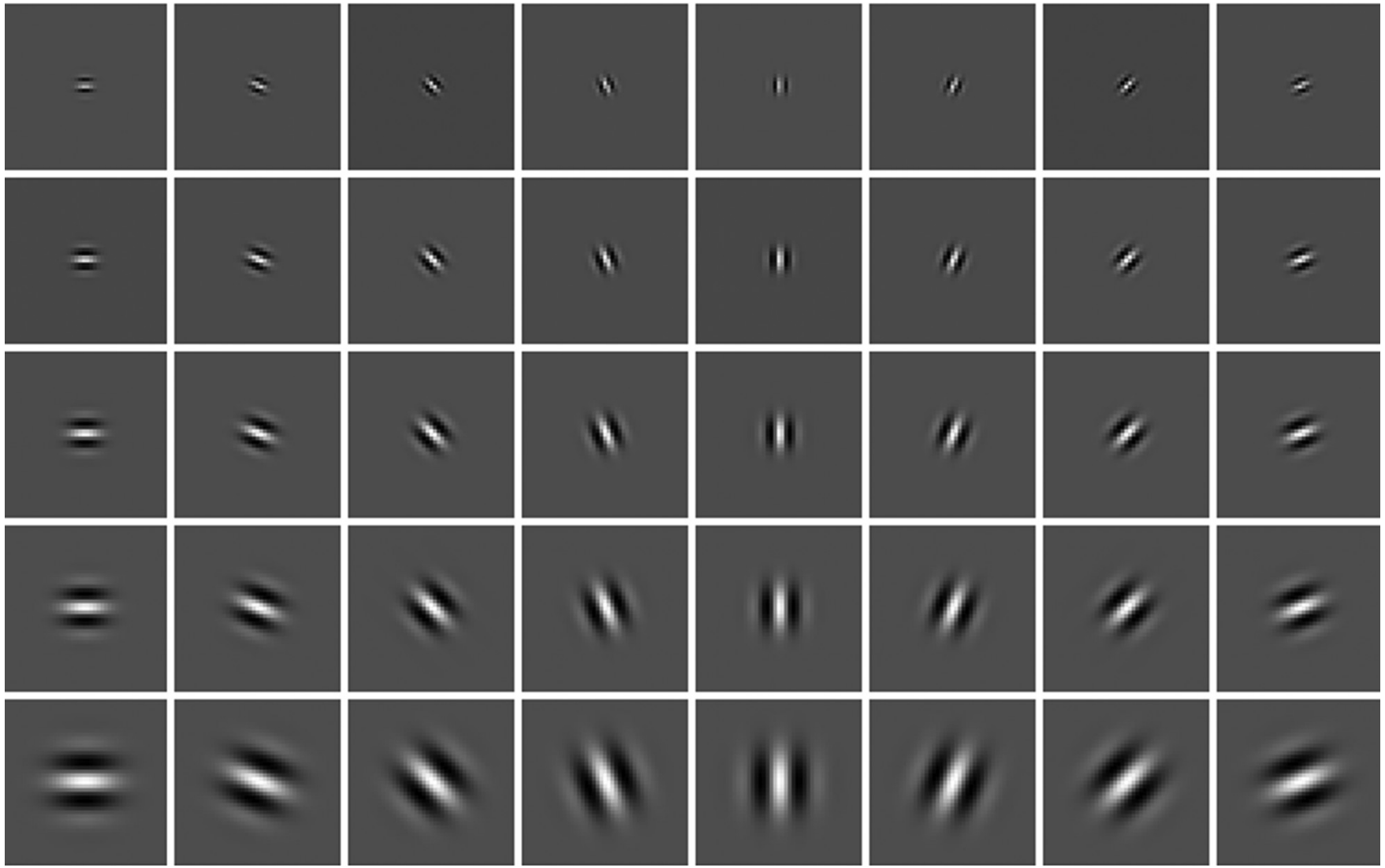
… looks a lot like…



Laplacian



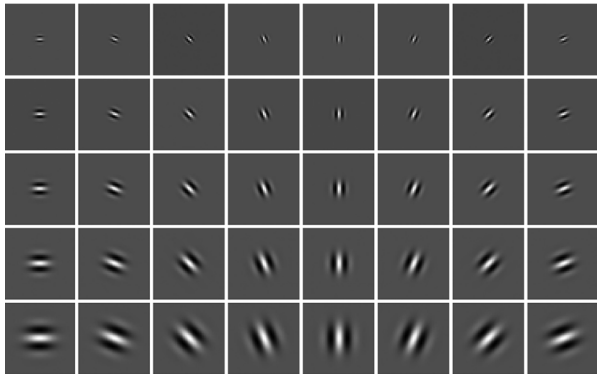
Directional edge detectors



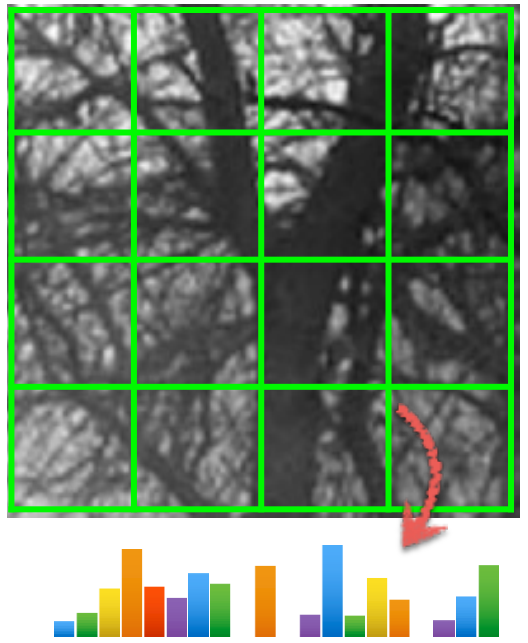
GIST

1. Compute filter responses (filter bank of Gabor filters)
2. Divide image patch into 4 x 4 cells
3. Compute filter response averages for each cell
4. Size of descriptor is 4 x 4 x N, where N is the size of the filter bank

Filter bank



4 x 4 cell



averaged filter responses

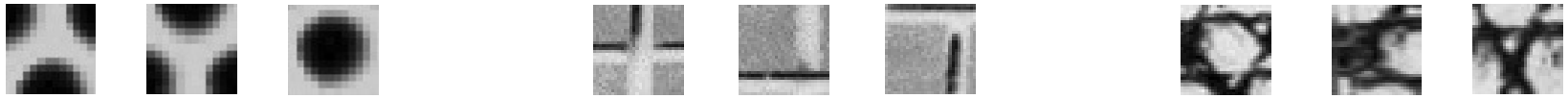
What is the GIST descriptor encoding?

Rough spatial distribution of image gradients

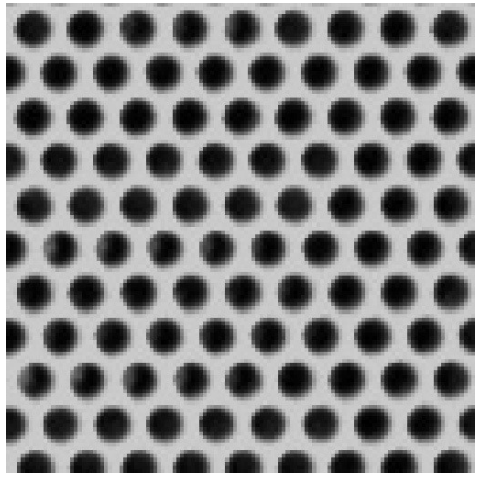
Textons

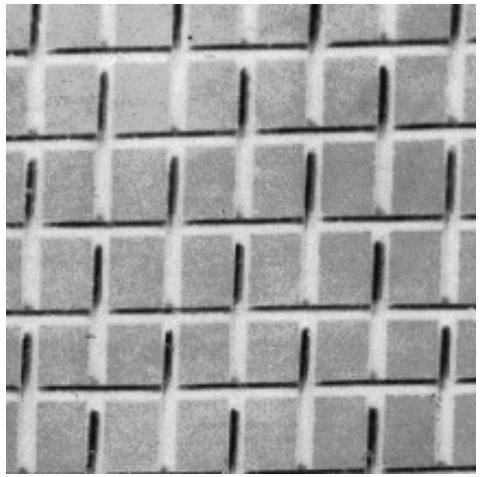
Julesz. Textons, the elements of texture perception, and their interactions. Nature 1981

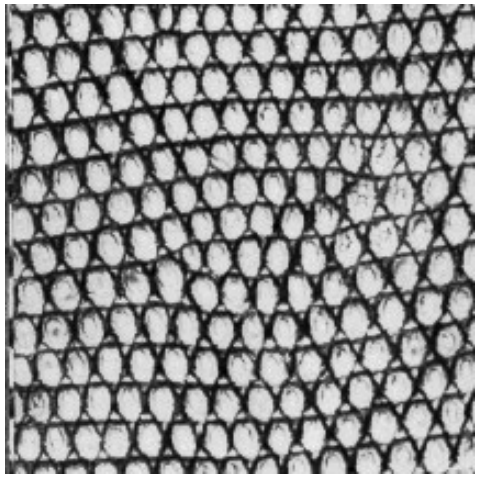
Texture is characterized by the repetition of basic elements or ***textons***



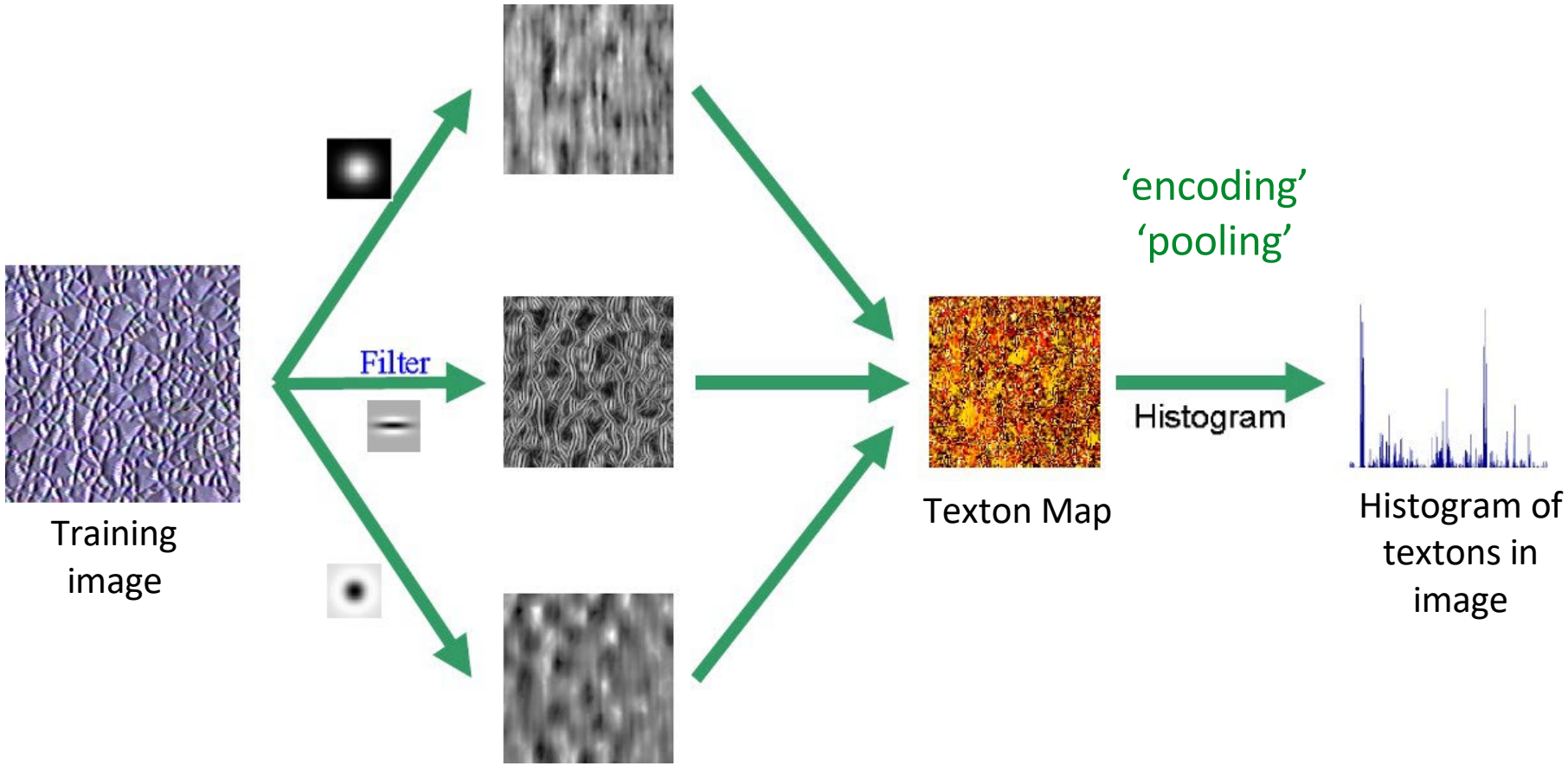
For stochastic textures, it is the identity of the ***textons***, not their spatial arrangement, that matters



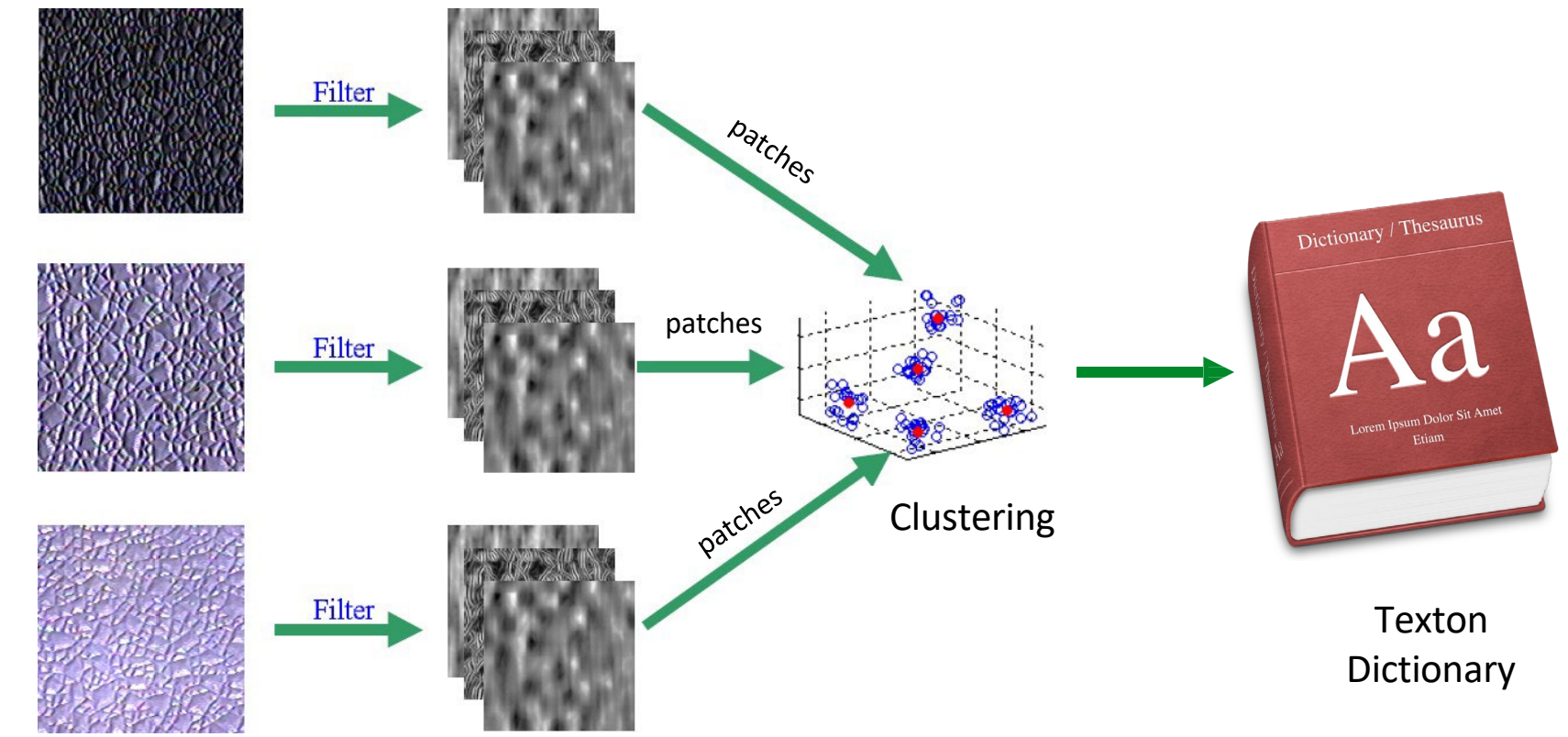




Histogram of Textons descriptor

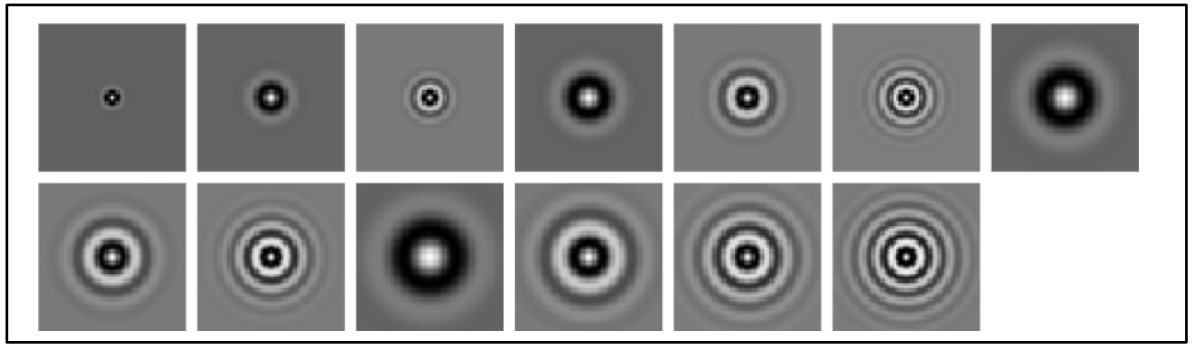


Learning Textons from data



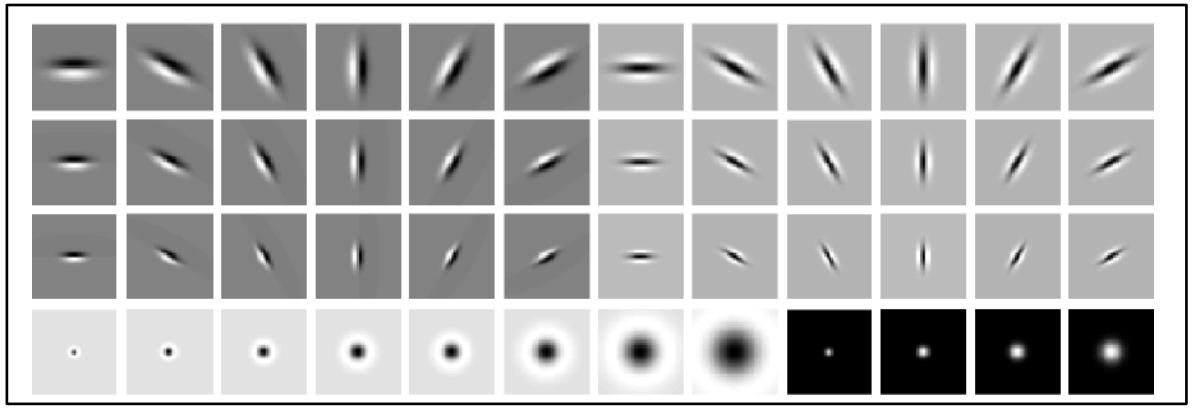
Example of Filter Banks

Isotropic Gabor

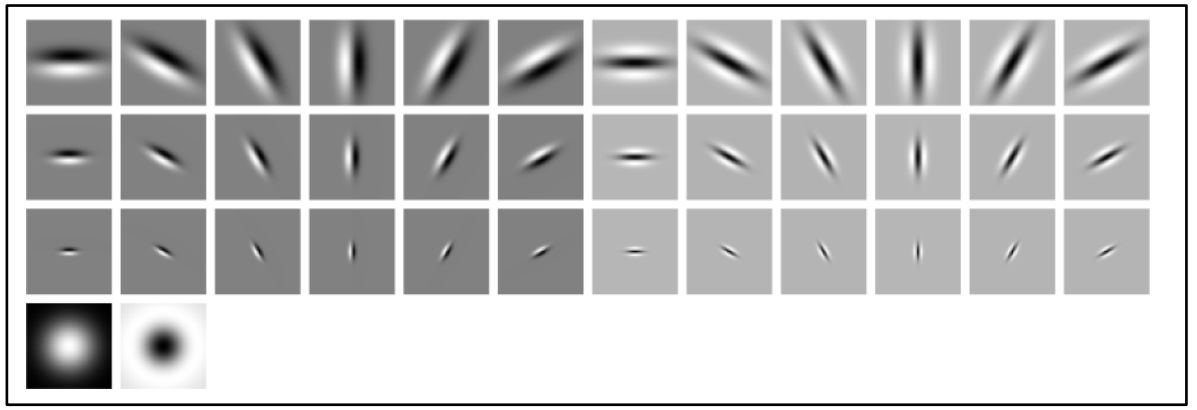


'S'

Gaussian derivatives at different scales and orientations

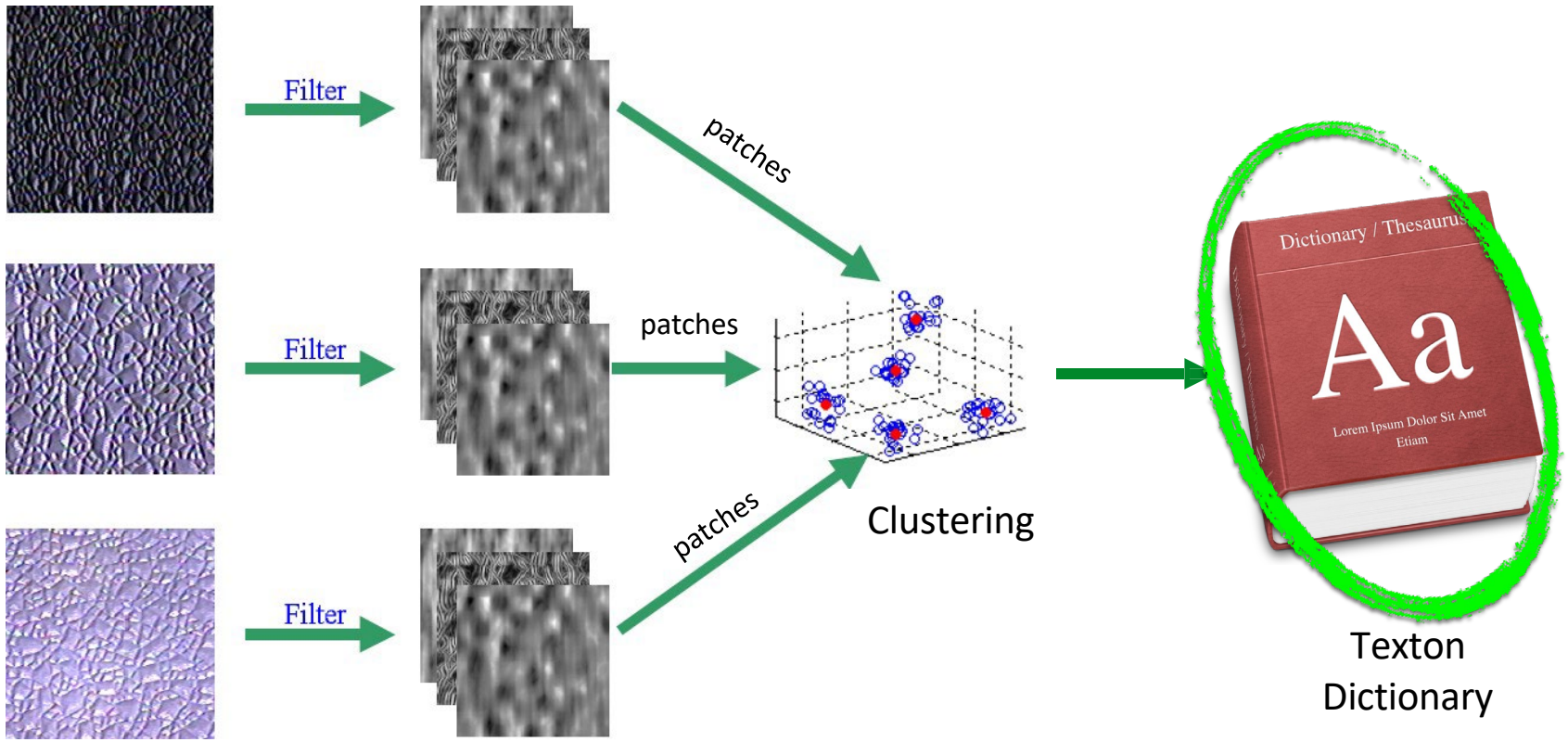


'LM'



'MR8'

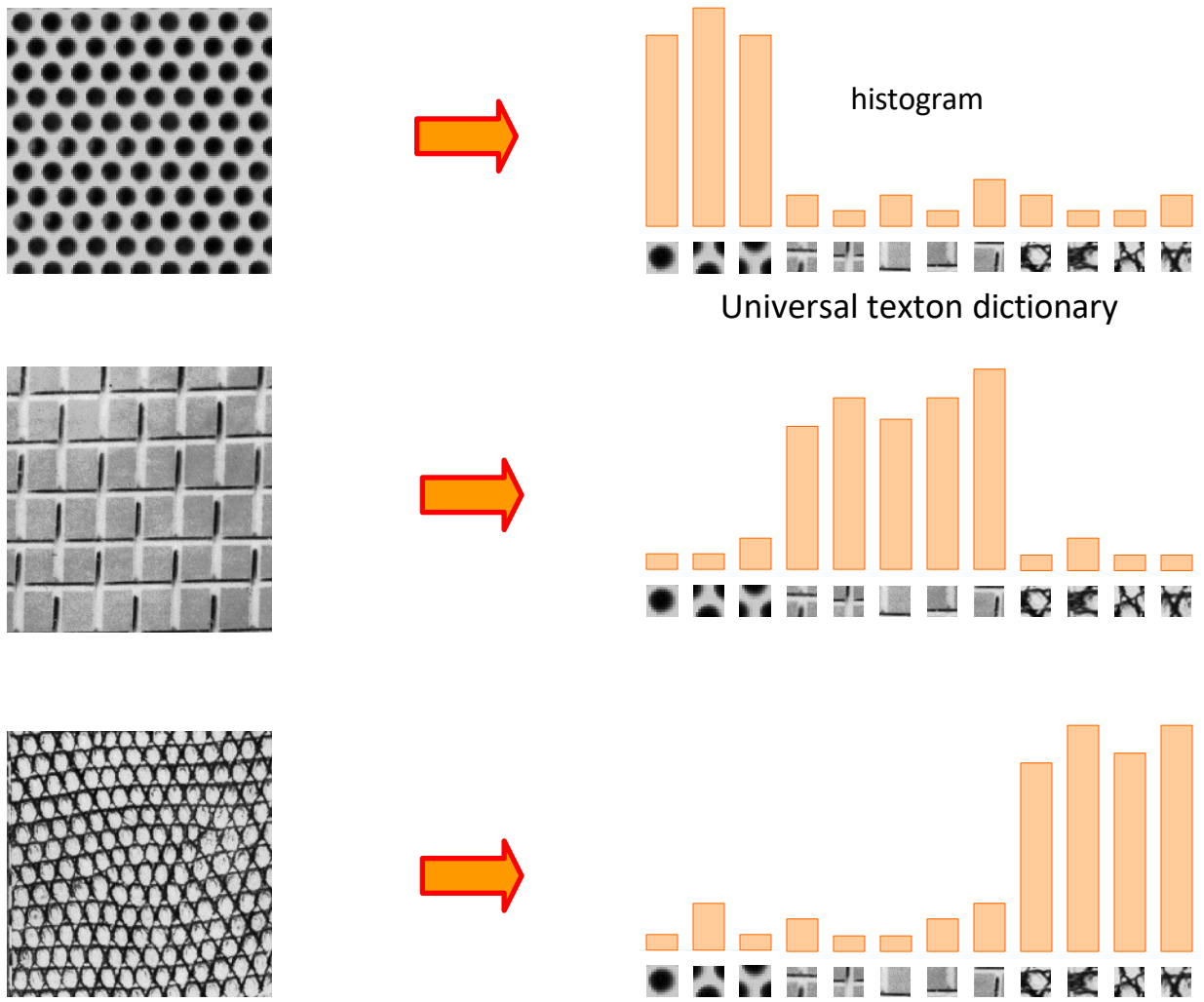
Learning Textons from data



Multiple training images of the same texture

Filter response over a bank of filters

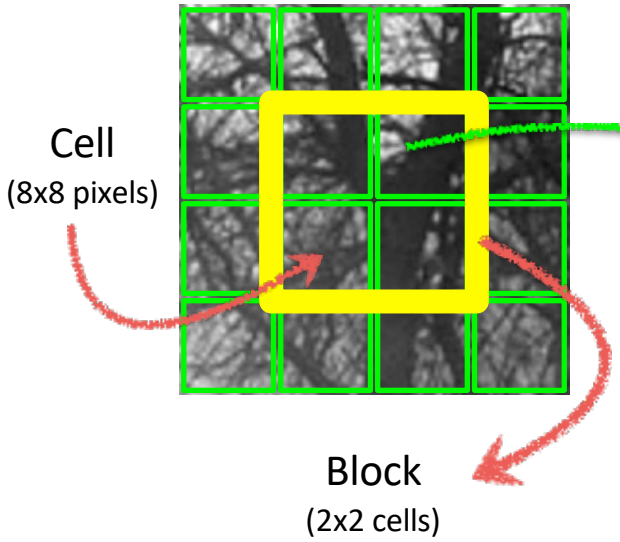
Bag of Words!



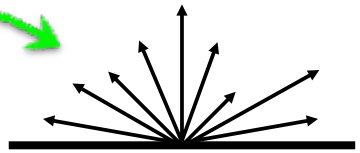
Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

HOG Descriptor

Dalal, Triggs. **Histograms of Oriented Gradients** for Human Detection. CVPR, 2005



histogram of 'unsigned' gradients



soft binning

gradient magnitude histogram
(one for each cell)

Concatenate and L-2 normalization

Single scale, no dominant orientation

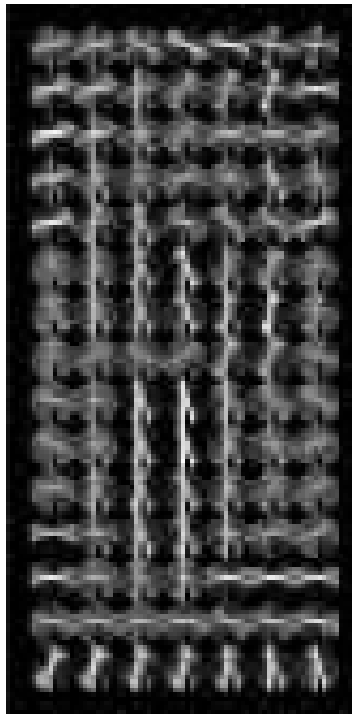
HOG Pedestrian Detection

1 cell step size

128 pixels
16 cells
15 blocks

64 pixels
8 cells
7 blocks

visualization



15 x 7 x 4 x 9 =
3780

Redundant representation due to overlapping blocks

How many times is each inner cell encoded?

http://chrisjmccormick.wordpress.com/2013/05/09/hog-person-detector-tutorial/