
Computer vision

Computer Vision

Lecture 5: Object Recognition I

Dr. Dina Khattab

dina.khattab@cis.asu.edu.eg

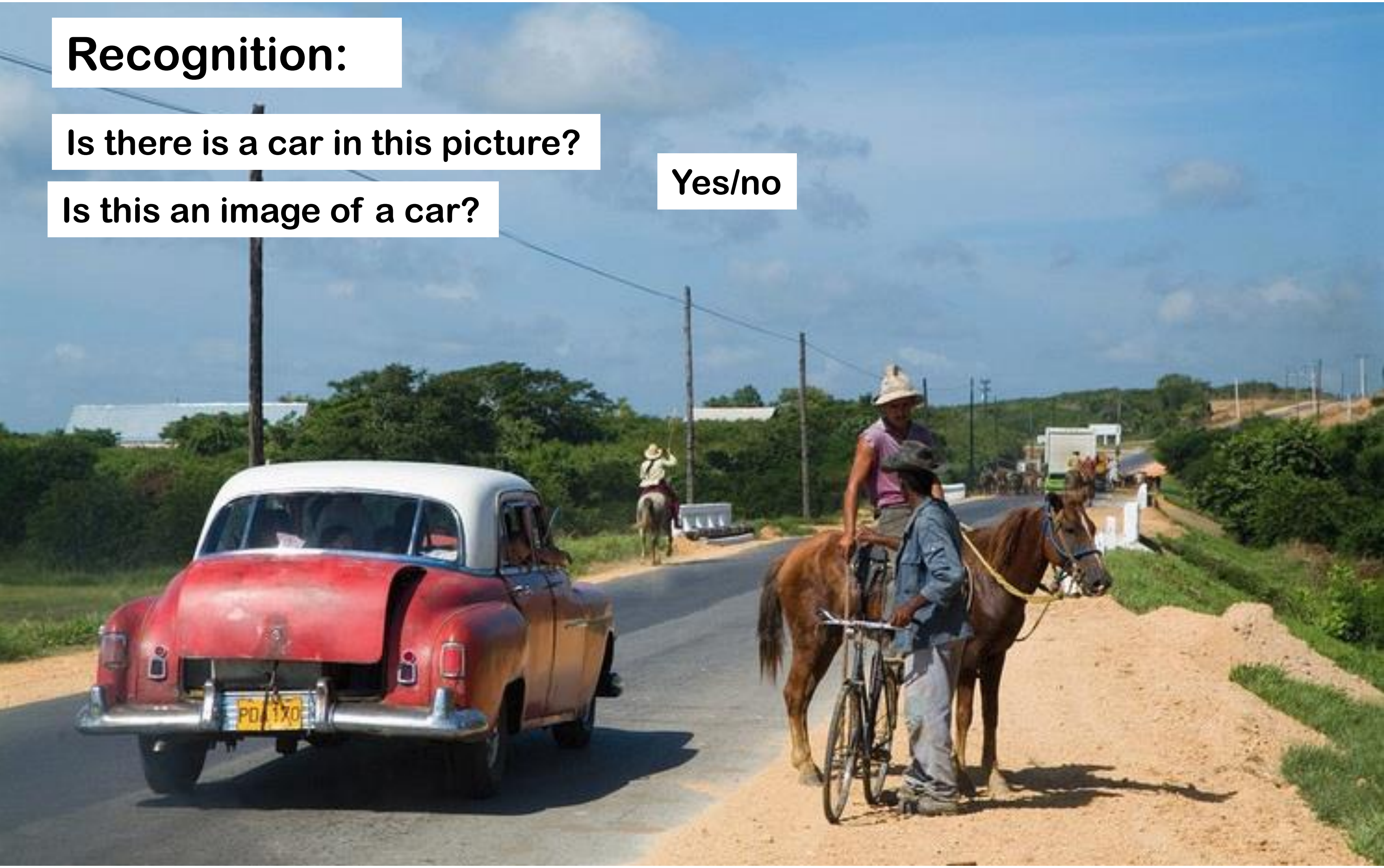
Scientific Computing Department

Instructor:	Dr. Dina Khattab
Email:	dina.khattab@cis.asu.edu.eg
Office:	Main Building – 4th floor – Room 302
Office Hours:	Monday 12:00 - 2:00 PM Thursday 11:00 AM to 12:00 PM

Agenda

- Object Recognition
  - Generative Models (Naïve Bayes)
    - App: Skin recognition
  - Discriminative Models (Linear classifiers)
    - Bag of Words (BOW)

Visual Recognition



Recognition:

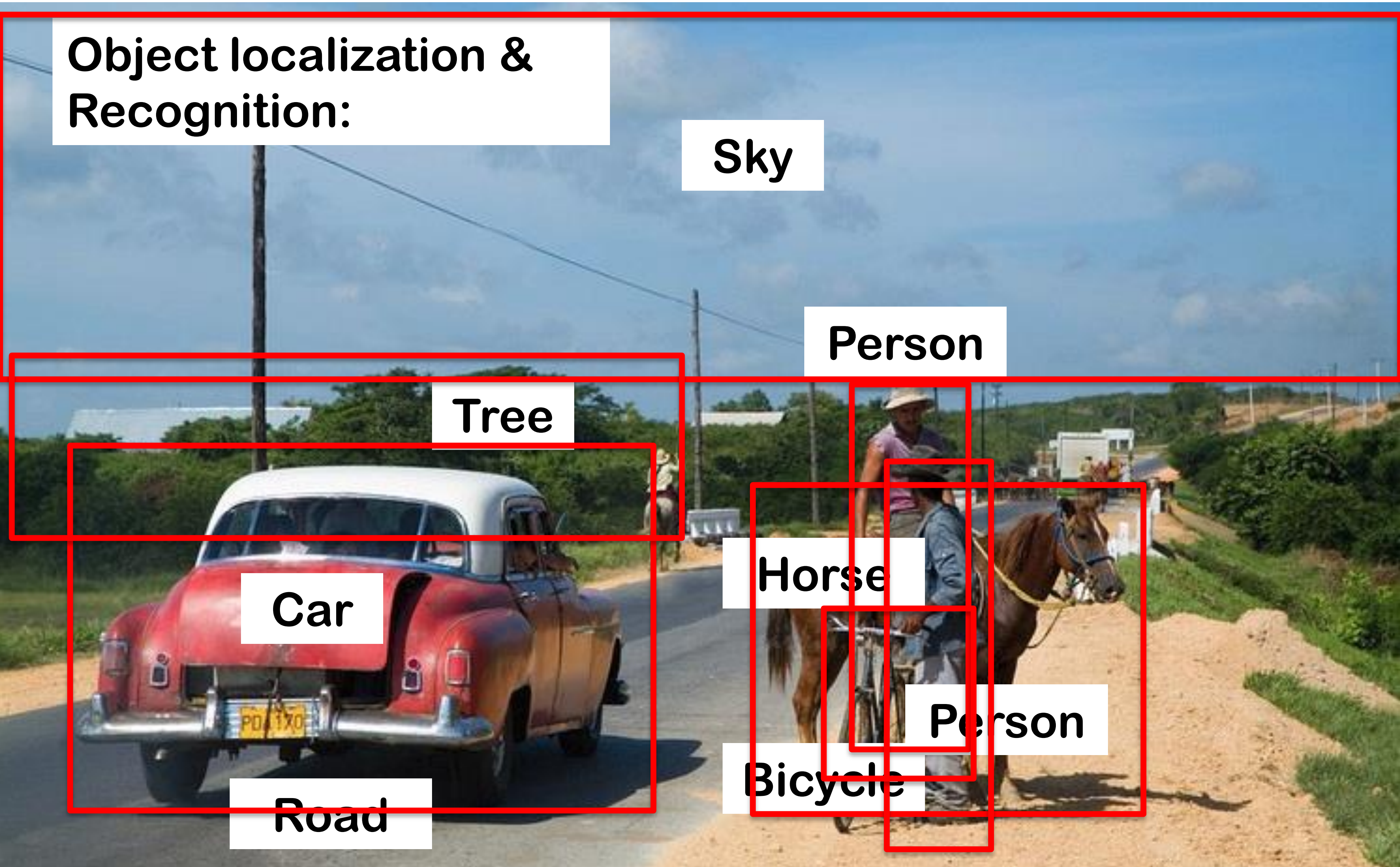
Is there is a car in this picture?

Is this an image of a car?

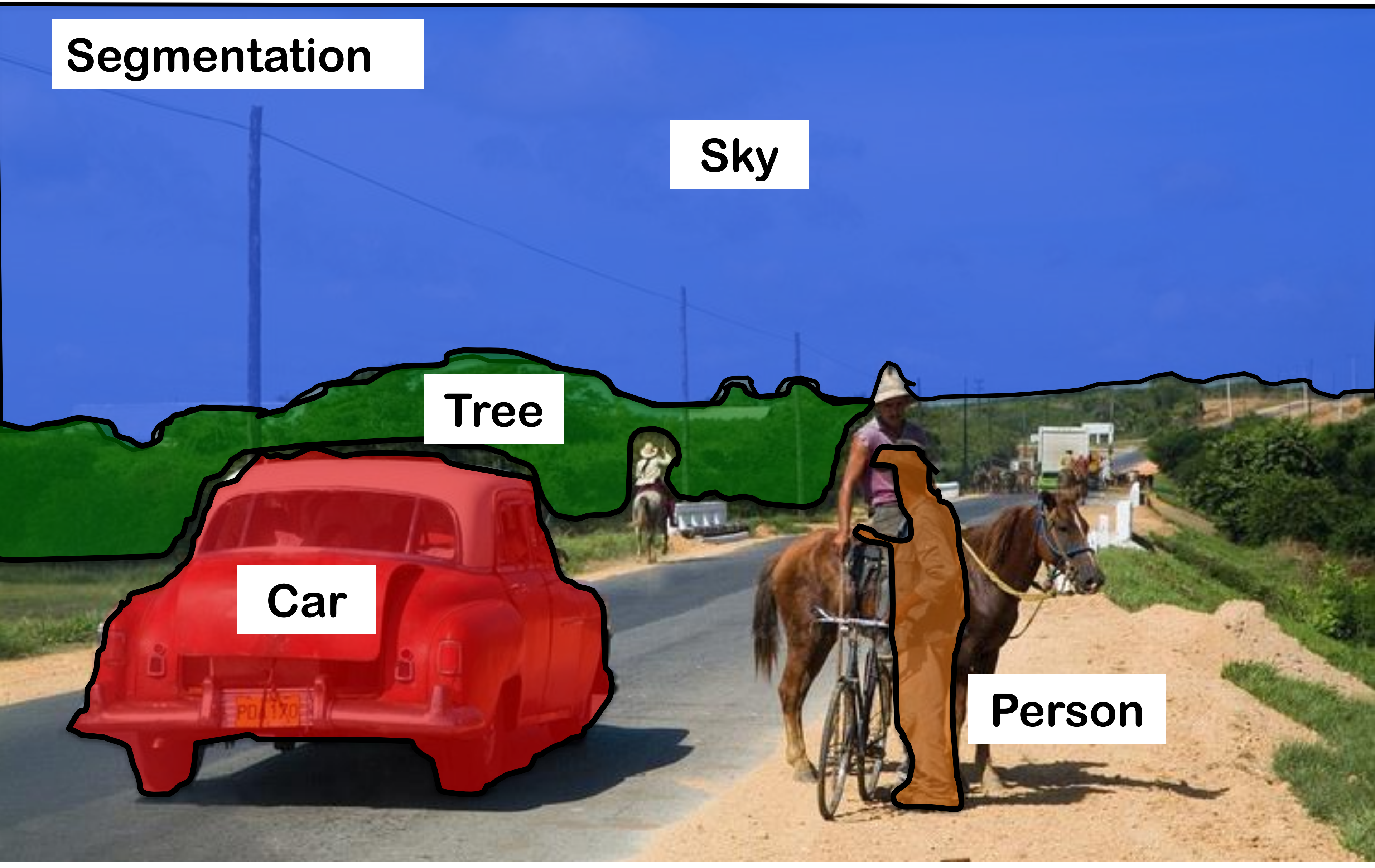
Yes/no

Visual Recognition

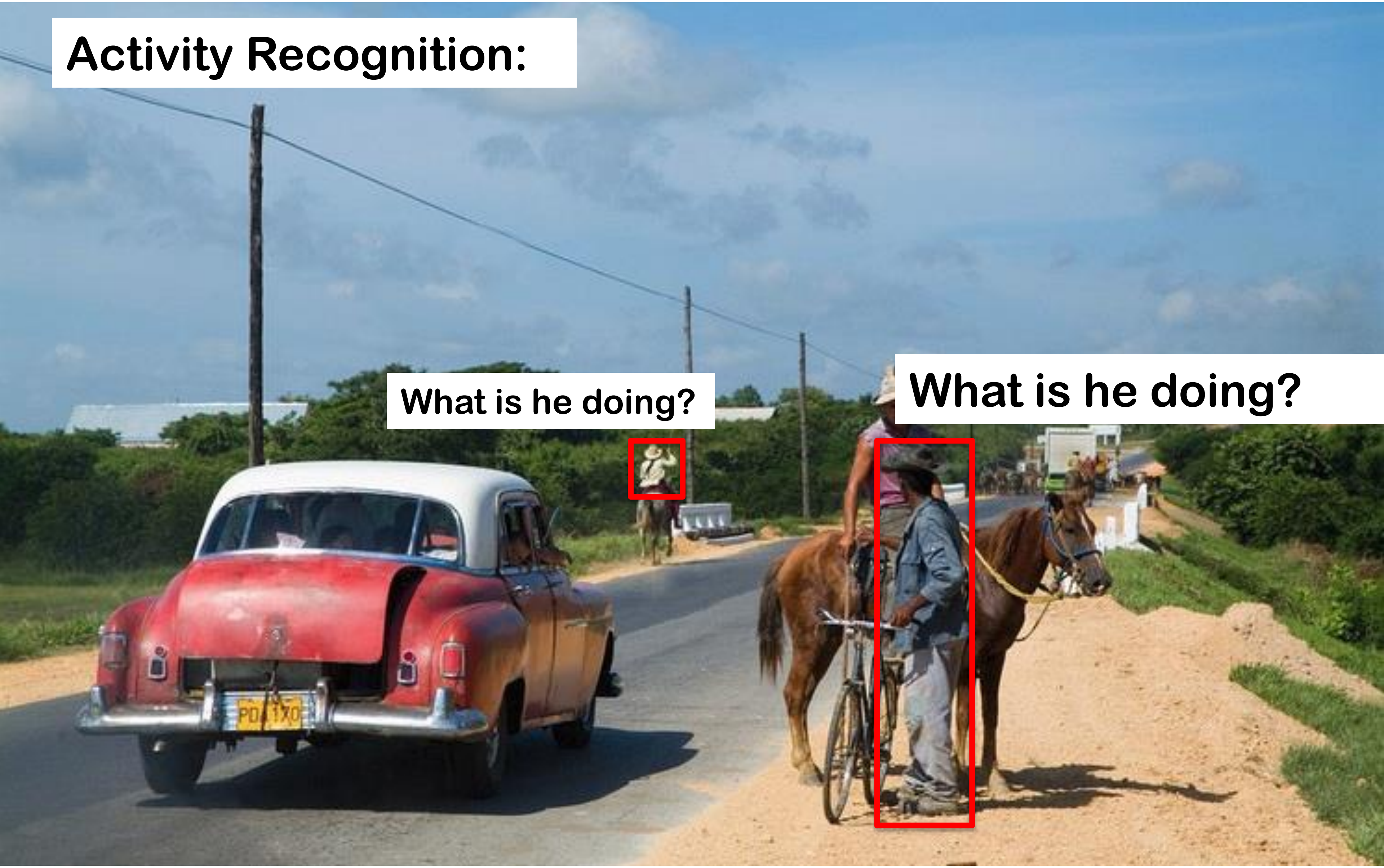
Visual Recognition



Visual Recognition



Visual Recognition



Challenges: Robustness

- **Viewpoint variation**: There are many potential ways to view an object, and the change in viewpoint can lead an object to look very different.

Challenges: Robustness

- **Illumination:** Different levels of light, particularly low light or a different light direction, will cause shadows to shift and the details of an object to become obscured.

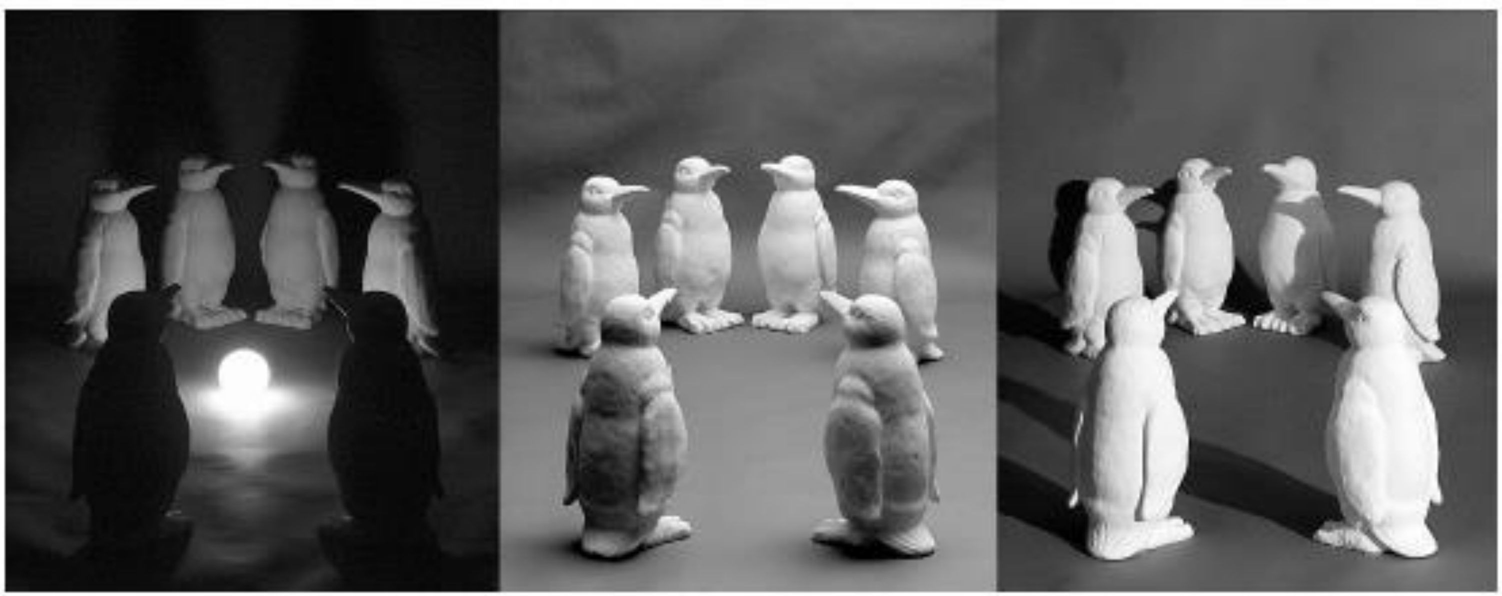


Image credit: J. Koenderink

Challenges: Robustness

- **Scale** Objects belonging to one category can come in a variety of sizes. If a classification is only trained on a particular size of object, then it will fail to recognize the same object in a different size.

Challenges: Robustness

- **Object pose:** Objects can change form and look very different, while still being considered the same object. For example, a person can be photographed in a number of poses, but is still considered a person if they’re bending over or if their arms are crossed.


Challenges: Robustness

- **Occlusion** Objects may be occluded, which could hide aspects of their characteristic geometry.

Challenges: Robustness

- **Background clutter** Similarities between the texture, color, and shapes in the background and the foreground can make it difficult to detect an object .

Challenges: Robustness

- **Intra-class variation** There can be significant shape variations event within one class of objects. For example, everything from a barstool to a lounge chair can be considered a chair.






Supervised Classification

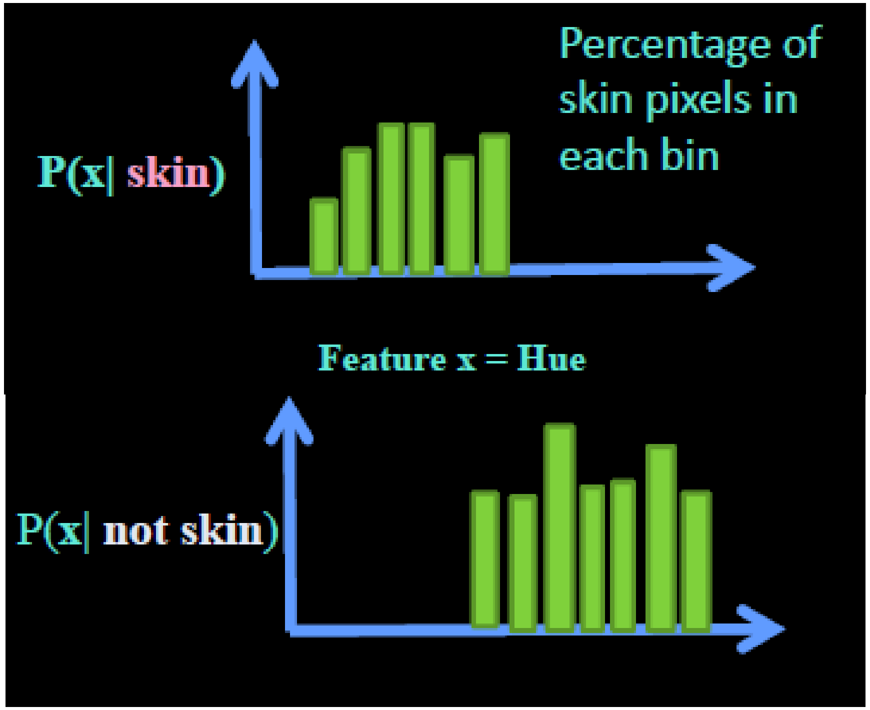
- Two General Strategies:

1. Generative:
   – Use the training data to build representative probability model for each class. (Model the distribution that generates the data).

2. Discriminative:
   – Construct a good decision boundary between different classes.

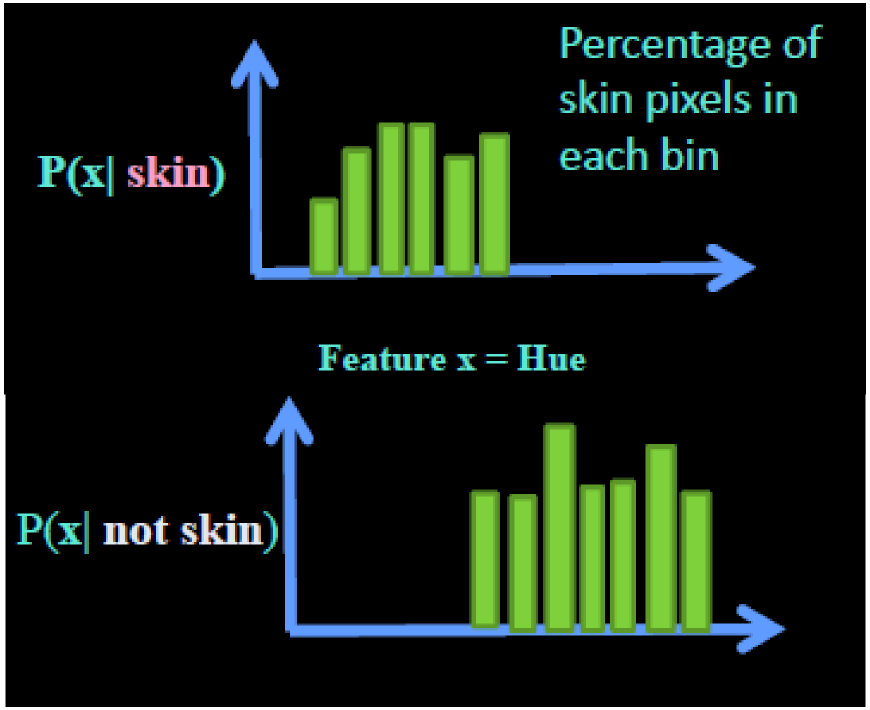
Supervised Classification: Generative (Naïve Bayes)

- Ex. Learning Skin colors



Ex. Learning Skin colors

- P(skin | x) ?!!



Bayes Rule

- $\overbrace{P(skin|x)}^{\text{Posterior}} = \dfrac{\overbrace{p(x|skin)}^{\text{Likelihood}}\overbrace{P(skin)}^{\text{Prior}}}{p(x)}$

- $P(skin|x)\ \alpha\ p(x|skin)P(skin)$

Bayes Rule

- If $P(skin|x) > P(\sim skin|x)$

OR $P(x|skin)p(skin) > P(x|\sim skin)p(\sim skin)$

then classify x as skin

We can get an evaluation If $P(skin|x) > \theta$

More general generative models

- For a given measurement x and set of classes c_i choose $c*$ by:

$$c^* = \arg\max_c p(c \mid \mathbf{x}) = \arg\max_c p(c)\, p(\mathbf{x} \mid c)$$

Generative Models

Advantages	**Disadvantages**
• Pure probabilistic models. • Parametric modeling of likelihood permits using small number of examples. • New classes don't disturb previous models.	• Where to get the priors? • Modeling non-c points? • It doesn't help if you have lots of data.

Some challenges for generative models

- But for the modern world there are some liabilities:
- Many signals are *high-dimensional* and *representing the complete density of class is data-hard*.
  - In some sense, we don't care about modeling the classes, *we only care about making the right decisions*.
  - *Model the hard cases-the ones near the boundaries.*

- We don't typically know which features of instances actually *models the different* classes.

So..

- We want to focus on ***discriminating*** between the class types.
- We want the machine to somehow ***learn*** the features that matter.
- This gets us to ***discriminative classification***

Discriminative Classification

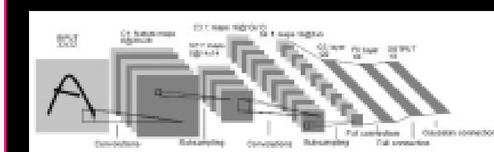
Find a division (surface) in feature space that separates the classes

Nearest neighbor

10^6 examples

Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005 …

Neural networks

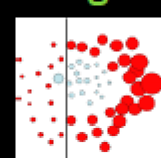


LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998 …

SVMs

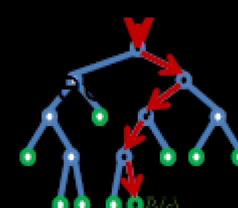
Guyon, Vapnik, Heisele, Serre, Poggio, 2001, …

Boosting



Viola, Jones 2001,
Torralba et al. 2004,
Opelt et al. 2006,…

Random Forests



Breiman, 1984
Shotton, et al CVPR 2008

Image Classification using SVM

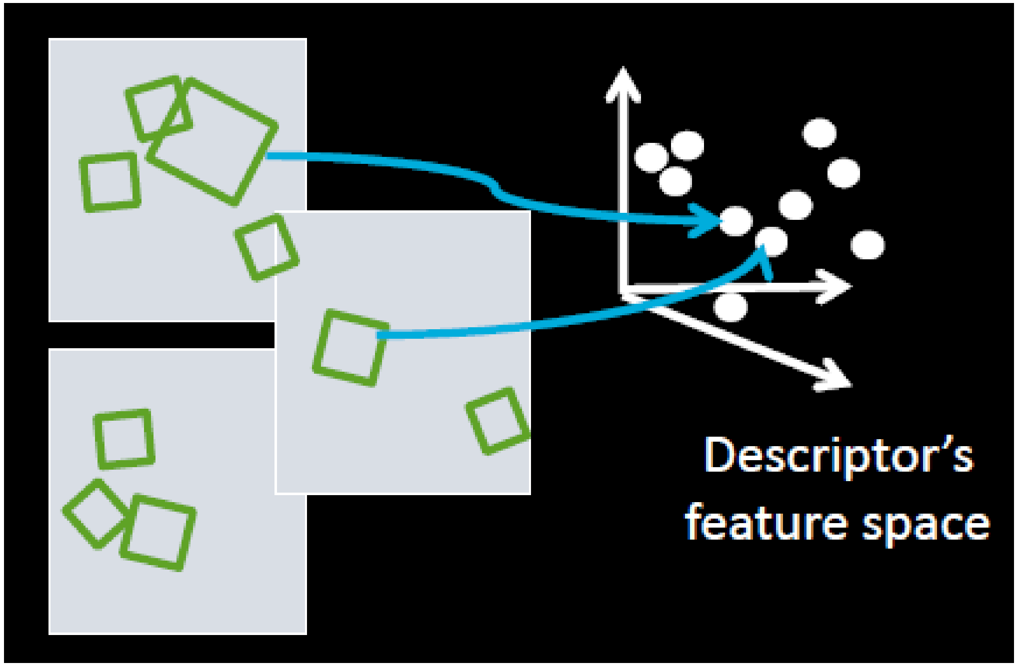
BAG OF VISUAL WORDS (BOW)

Image Classification using SVM

- Image → Feature extraction → classifier

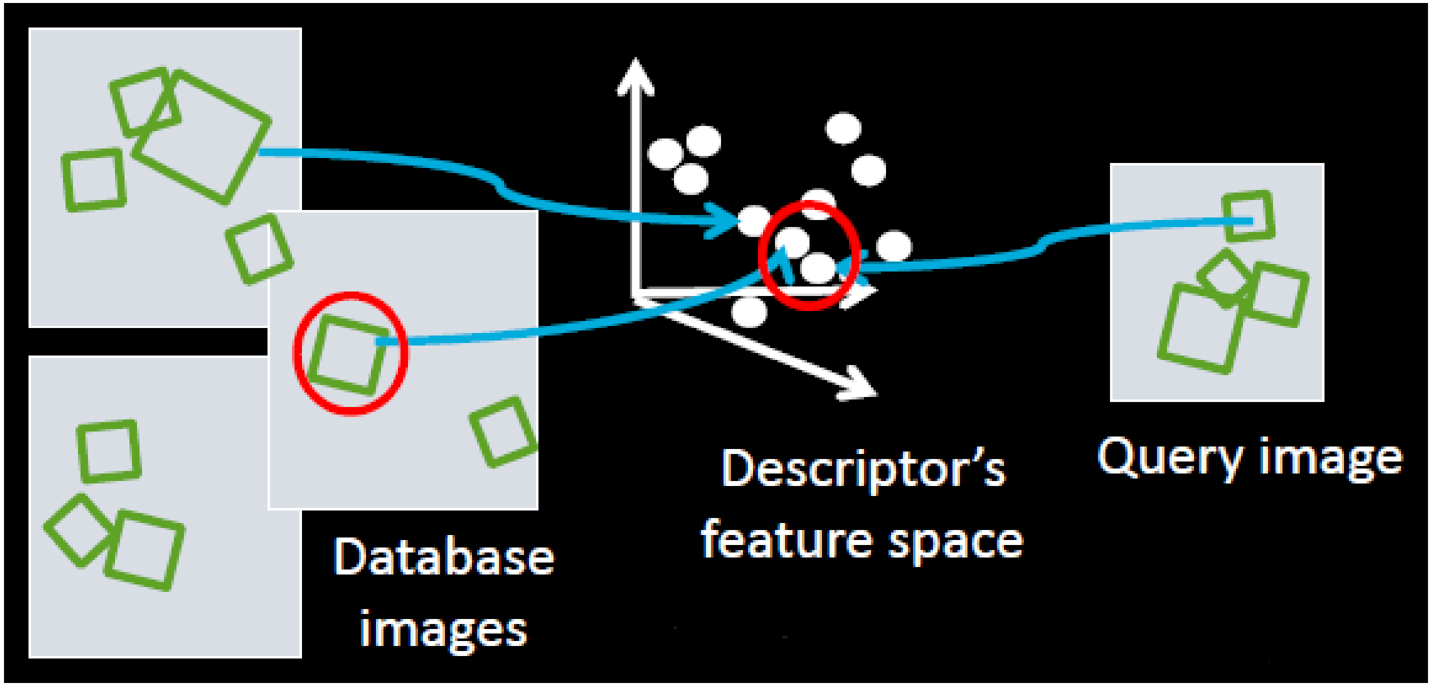
Indexing Local Features

Each patch / region has a descriptor, which is a point in some high-dimensional feature space (e.g., SIFT)



Indexing Local Features

When we see close points in feature space, we have similar descriptors, which indicates similar local content.



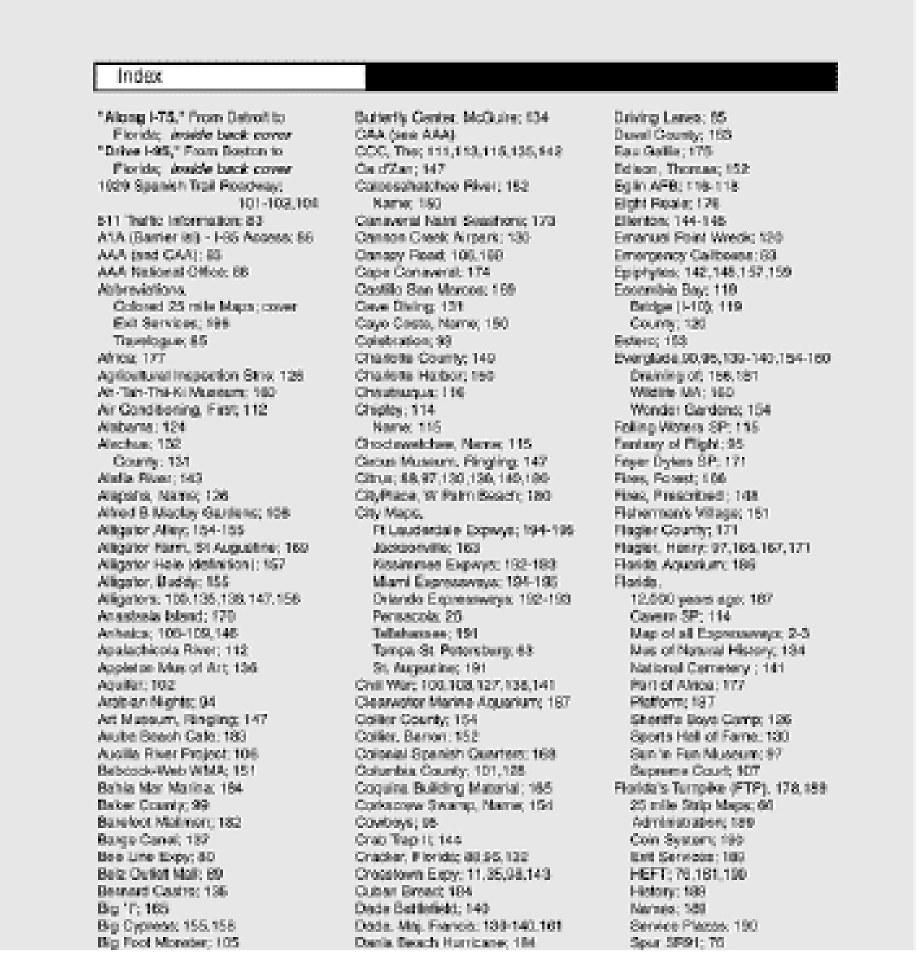
Easily can have millions of features to search!

Indexing Local Features

- With potentially thousands of features per image, and hundreds to millions of images to search, how to efficiently find those that are relevant to a new image?

Indexing local features: inverted file index

- For text documents, an efficient way to find all *pages* on which a *word* occurs is to use an index…
- We want to find all *images* in which a *feature* occurs.
- To use this idea, we'll need to map our features to "visual words"


Index

Bags of Features for Object Recognition

face, flowers, building

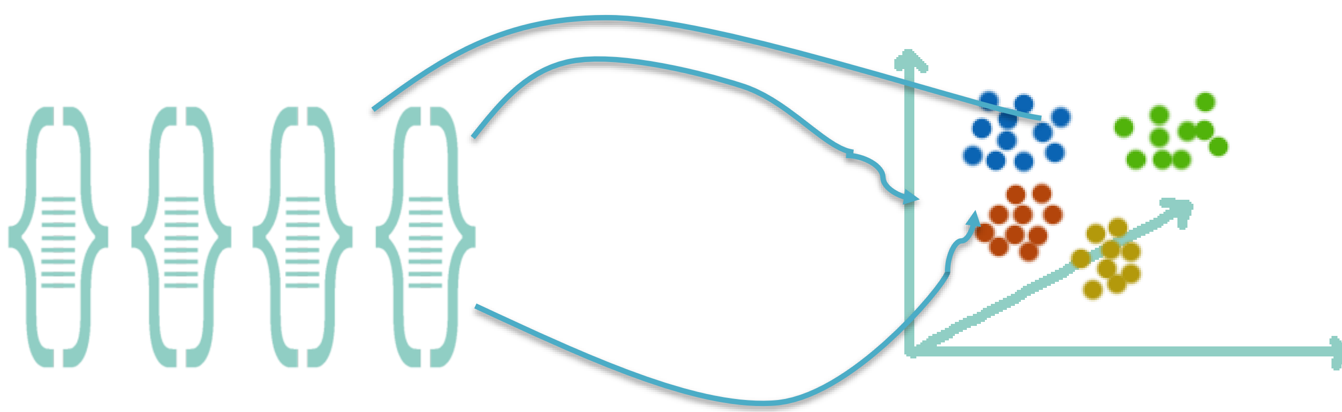
- "Bag of Words" is a way to simplify object representation as a collection of their subparts for purposes such as classification.

- Take a bunch of images, extract features, and build up a "dictionary" or "visual vocabulary" – a list of common features
- **Bag of features: outline**
  1. Extract features

  2. Learn "visual vocabulary"

  3. Quantize features using visual vocabulary

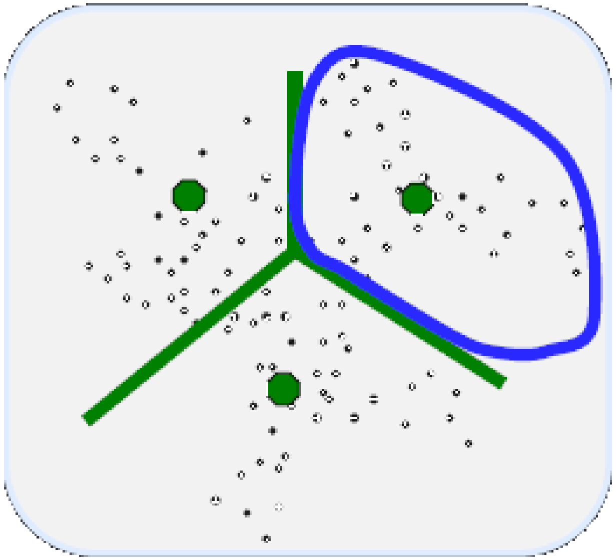
SIFT features

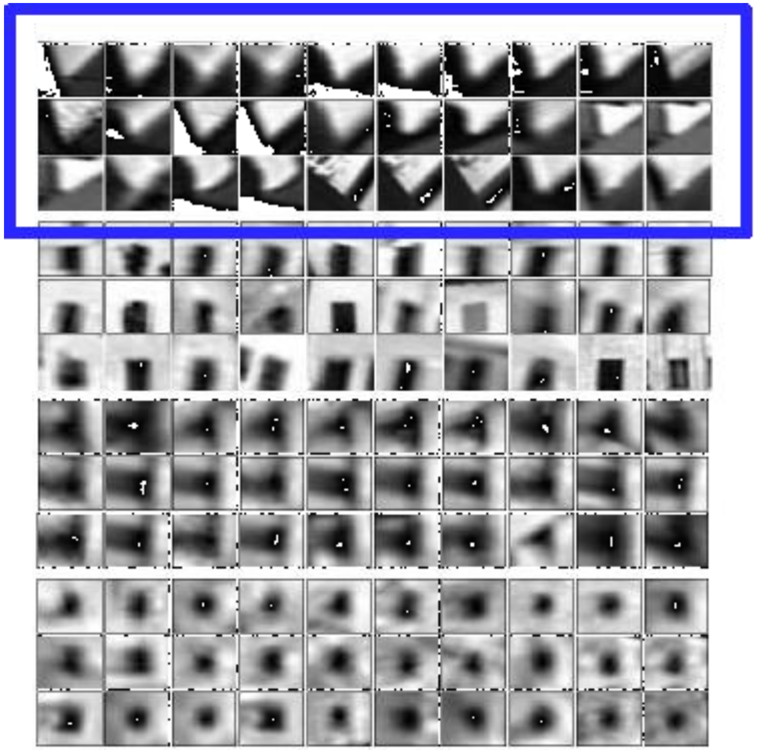


Clustering

Visual Words

Example: each group of patches belongs to the same visual word



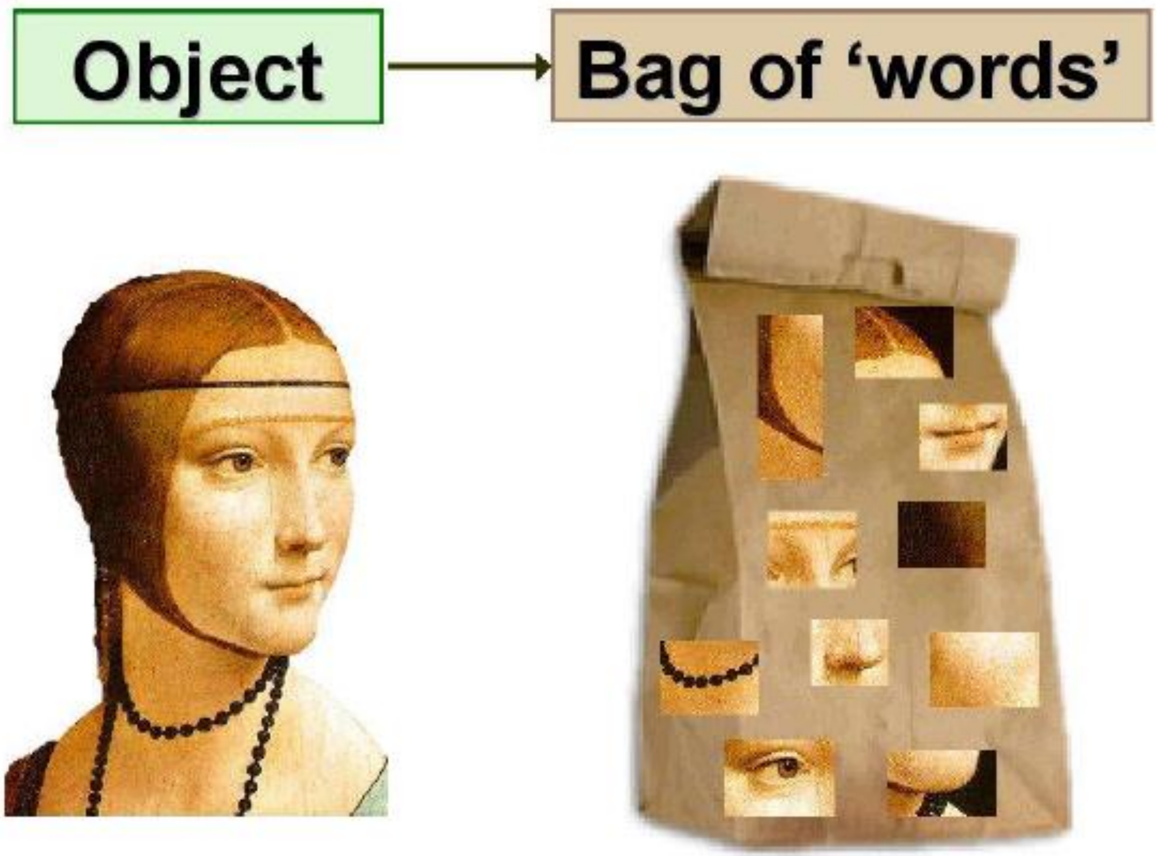


visual words are the centers of the clusters , which is the average of all patches in this cluster

From Clustering to Vector Quantization

- Clustering is a common method for learning a visual vocabulary or codebook
  - Unsupervised learning process
  - Each cluster center produced by k-means becomes a codevector
  - Provided the training set is sufficiently representative, the codebook will be "universal"

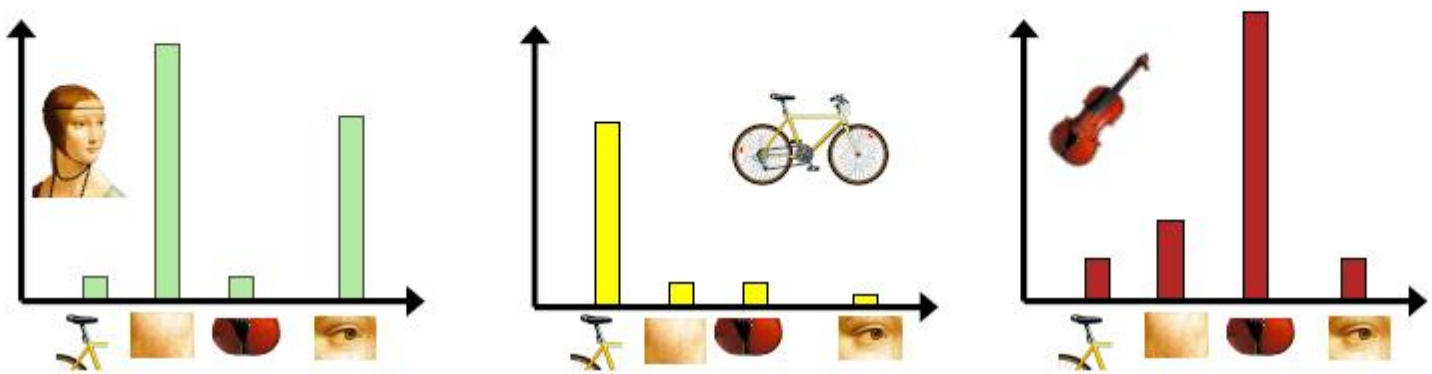
- The codebook is used for quantizing features
  - A *vector quantizer* takes a feature vector and maps it to the index of the nearest codevector in a codebook
  - Codebook = visual vocabulary
  - Codevector = visual word

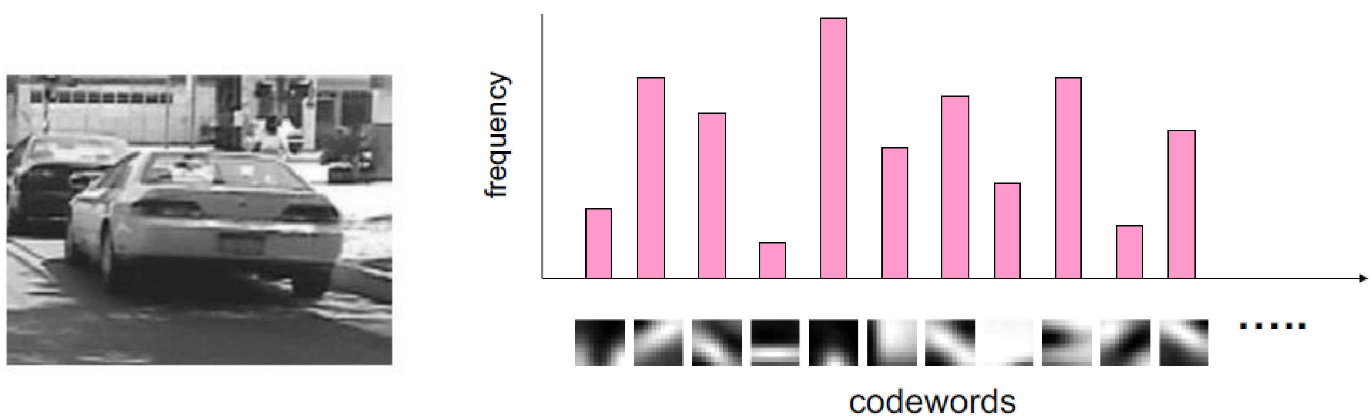


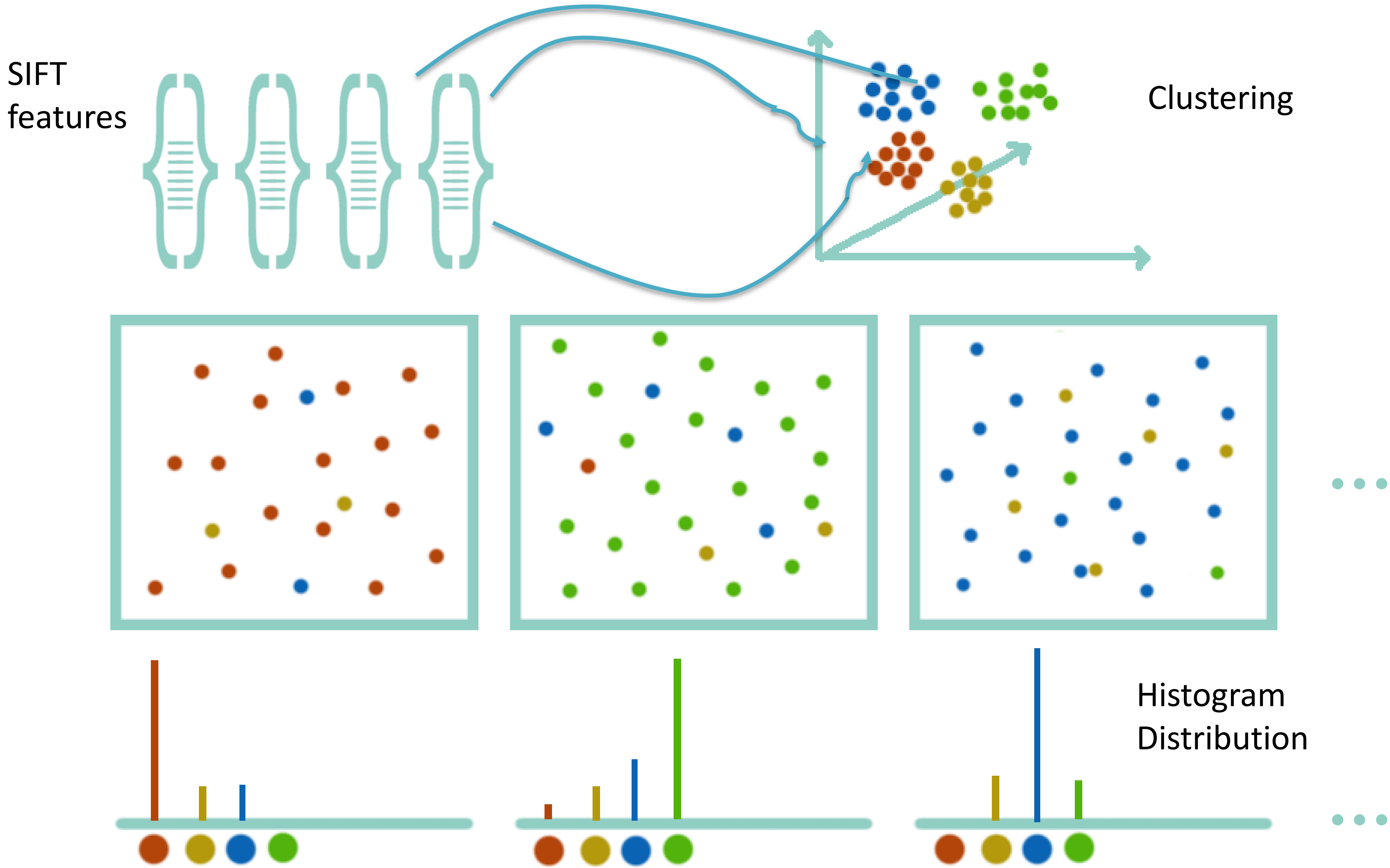
- How many of these code words show up in a particular picture?
- What is the distribution of these visual words in each image?
- We need to build a histogram for each image, so when we get a new image, we can find the images that has the same or similar histogram distribution of visual words.

Bag of features: outline

4. Represent images by frequencies of “visual words” (Summarize entire image based on its distribution/histogram of word occurrences)







5. convert each histogram into a normalized feature vector

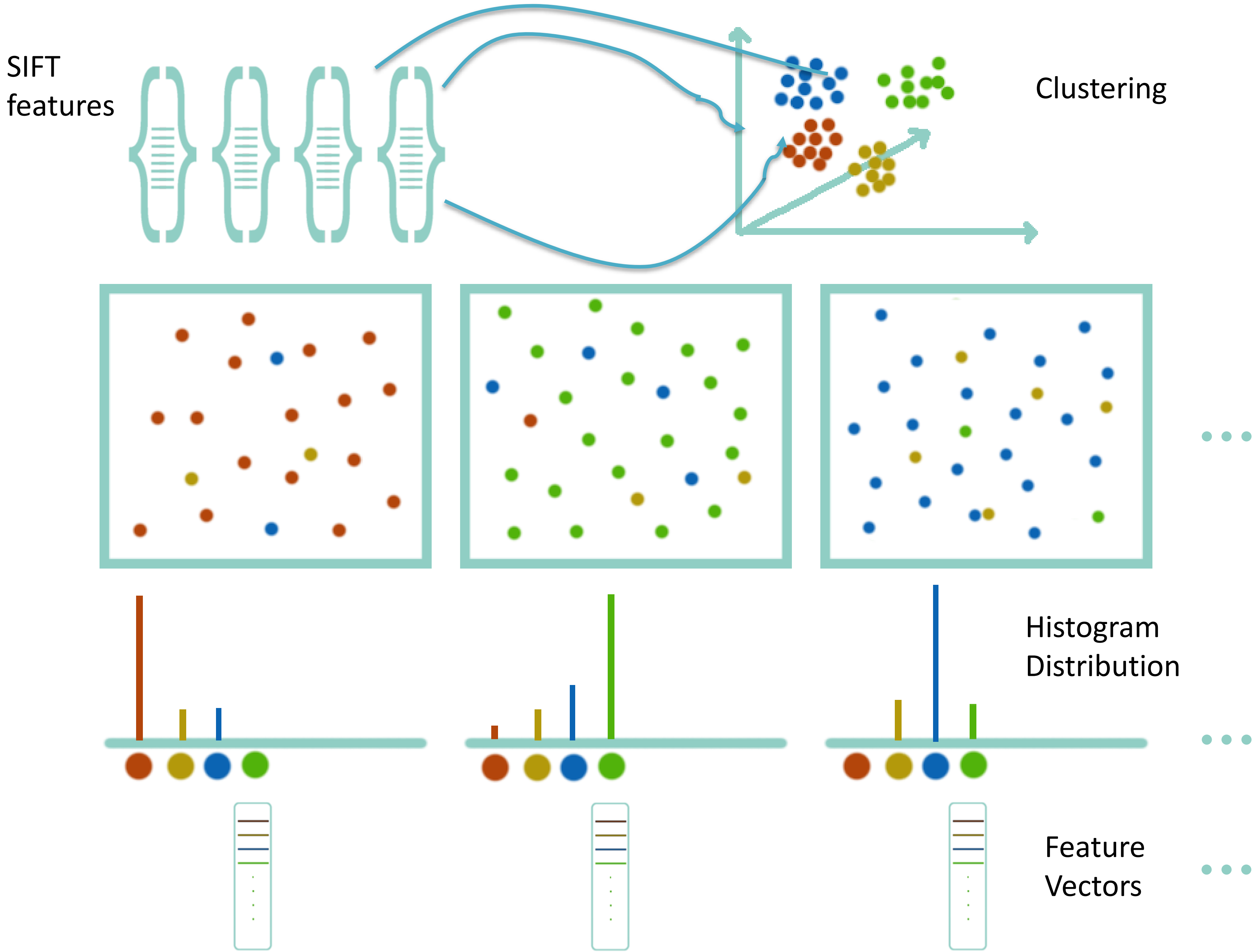
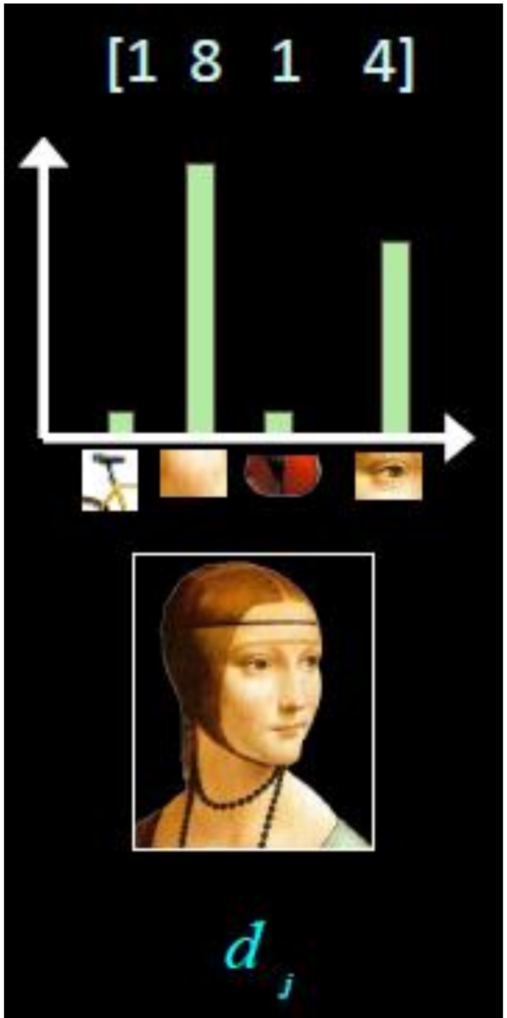
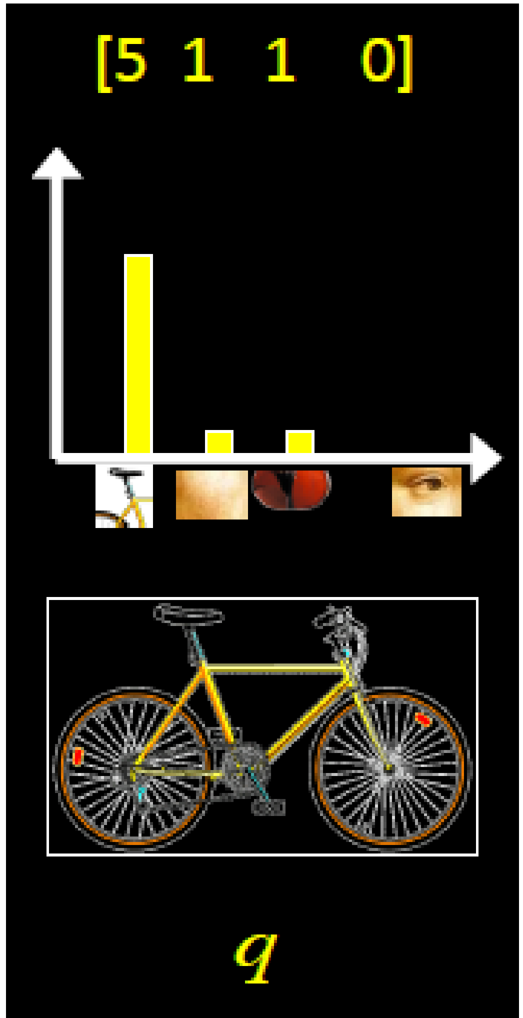

SIFT features
Clustering
Histogram Distribution
Feature Vectors

Image Classification using BOW

using k-NN or SVM





$$sim(d_j, q) = \frac{\langle d_j, q \rangle}{\|d_j\| \|q\|}$$

$$= \frac{\sum_{i=1}^{V} d_j(i) * q(i)}{\sqrt{\sum_{i=1}^{V} d_j(i)^2} * \sqrt{\sum_{i=1}^{V} q(i)^2}}$$

for vocabulary of *V* words

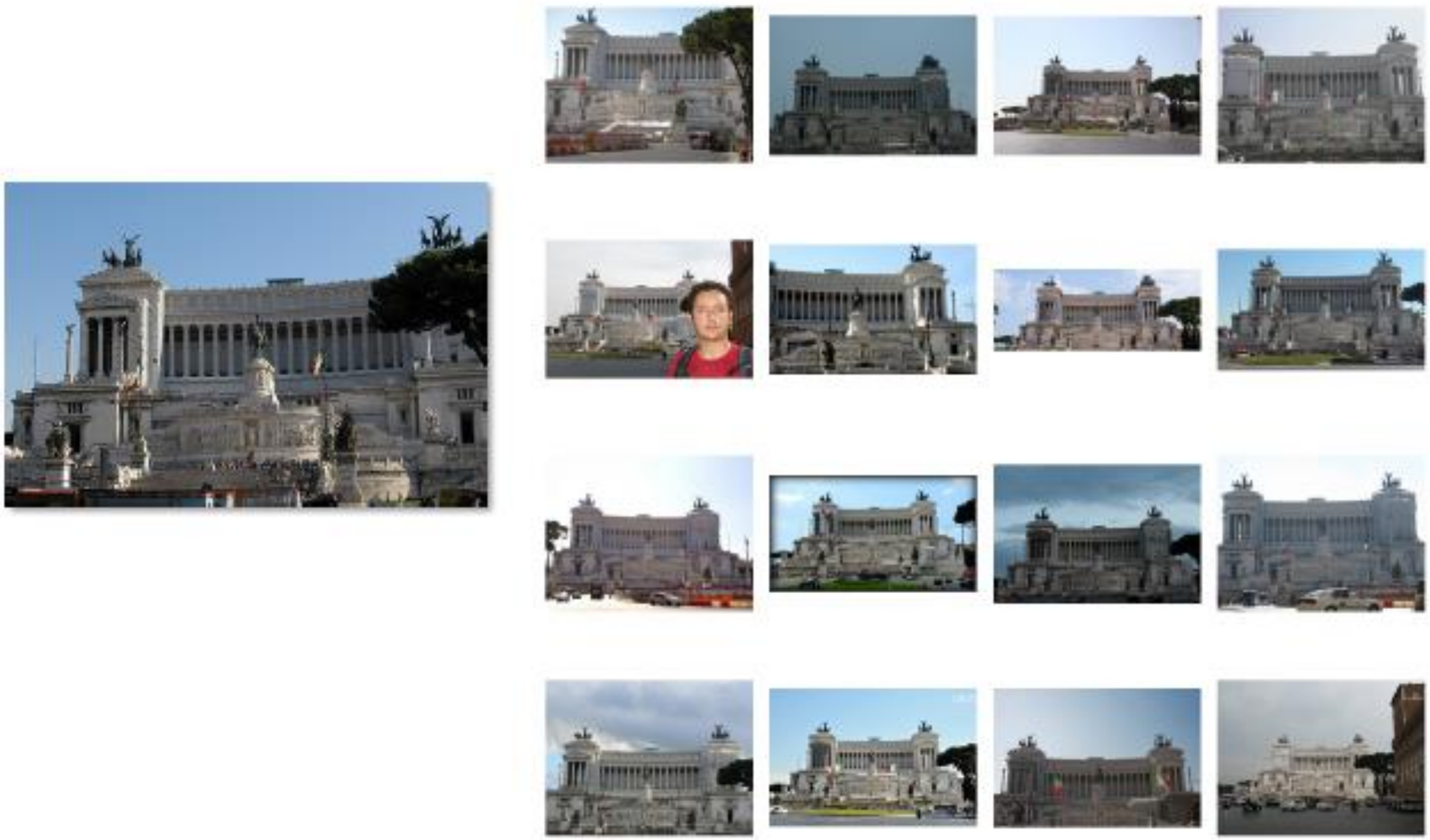
Object Classification with Bag of Words

- Performance on Caltech 101 dataset with linear SVM on bag-of-word vectors:

True classes →	*faces (frontal)*	*airplanes (side)*	*cars (rear)*	*cars (side)*	*motorbikes (side)*
faces(frontal)	**94**	0.4	0.7	0	1.4
airplanes (side)	1.5	**96.3**	0.2	0.1	2.7
cars (rear)	1.9	0.5	**97.7**	0	0.9
cars(side)	1.7	1.9	0.5	**99.6**	2.3
motorbikes (side)	0.9	0.9	0.9	0.3	**92.7**

[Csurka et al., ‘04]

Example bag-of-words matches



Example bag-of-words matches

Advantages

- We can create our visual vocabulary from a different dataset than the dataset that we are interested in classifying/clustering, so if first dataset is representative of the second, this algorithm will be successful (visual vocabulary can be universal).

Issues:

- How to choose vocabulary size (hyper-parameter)?
  - Too small: visual words not representative of all patches
  - Too large: quantization artifacts, overfitting

Credit for

CS 4495 Computer Vision (Spring 2015)

A. Bob - College of Computing, Georgia Tech.

CSE 455 Computer Vision (Winter 2017) by Linda Shapiro - University of Washington.

CS131 "Computer Vision: Foundations and Applications" by University of Stanford (Fall 2019)