Image Classification

COS 429

Princeton University

High-level goal: scene understanding

Source: Fei-Fei Li, Rob Fergus, Antonio Torralba.

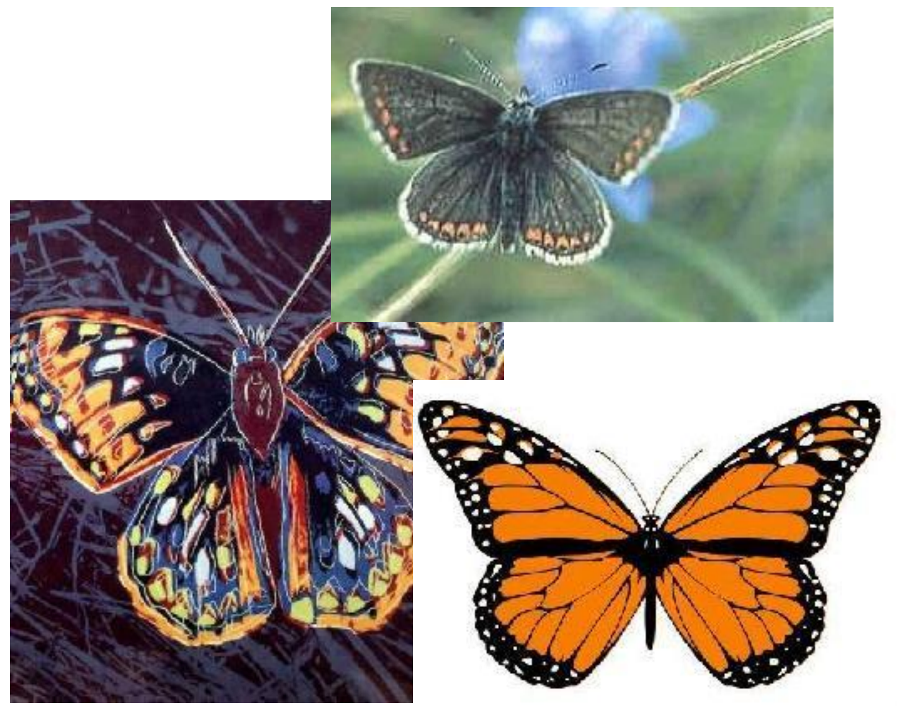
Today: image classification

- Given an image, add category-level annotations

Airplane

Today: image classification

- e.g., annotate basic-level object categories



Today: image classification

- Or, scene categories

Today: image classification

- Or, action categories

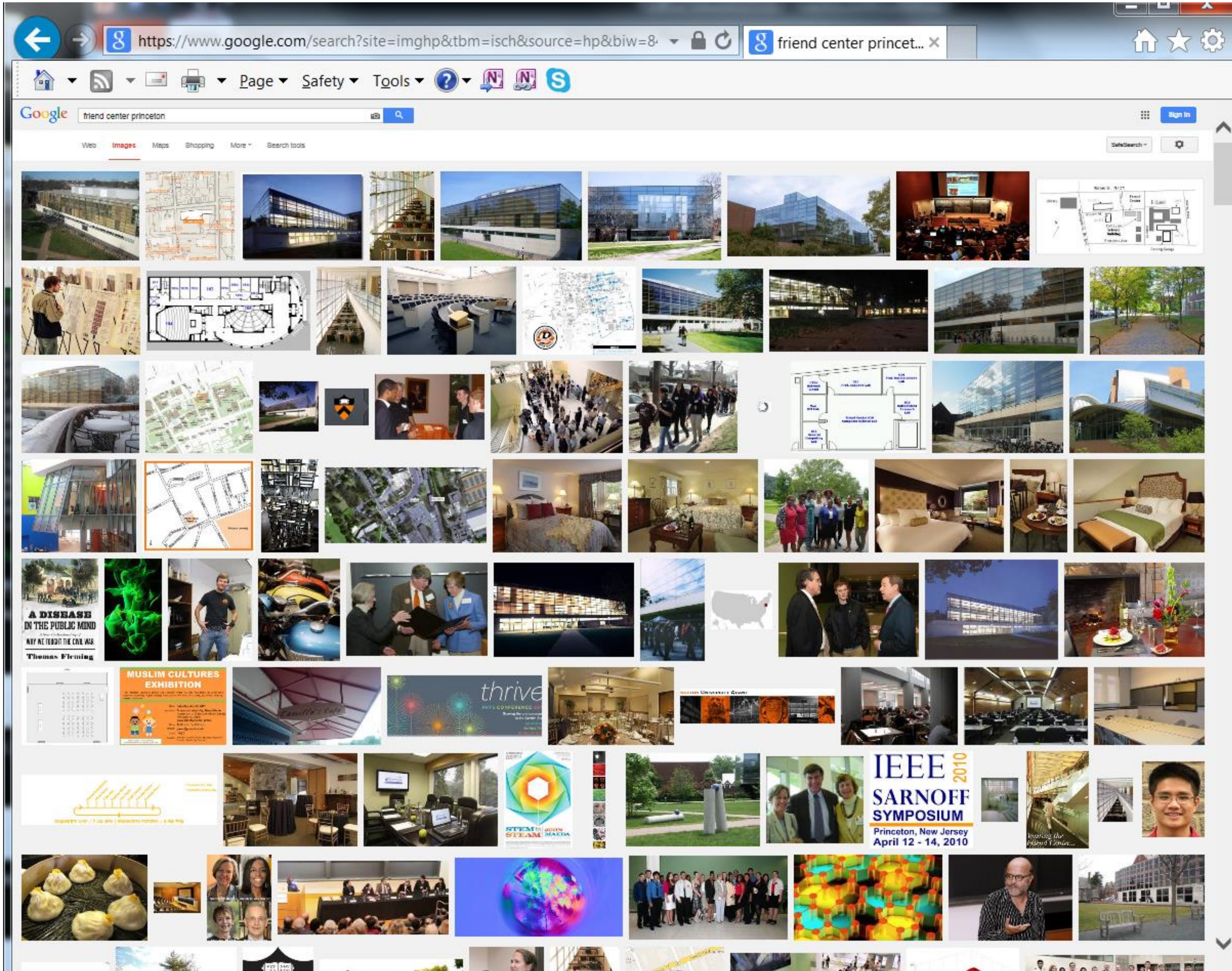
Today: image classification

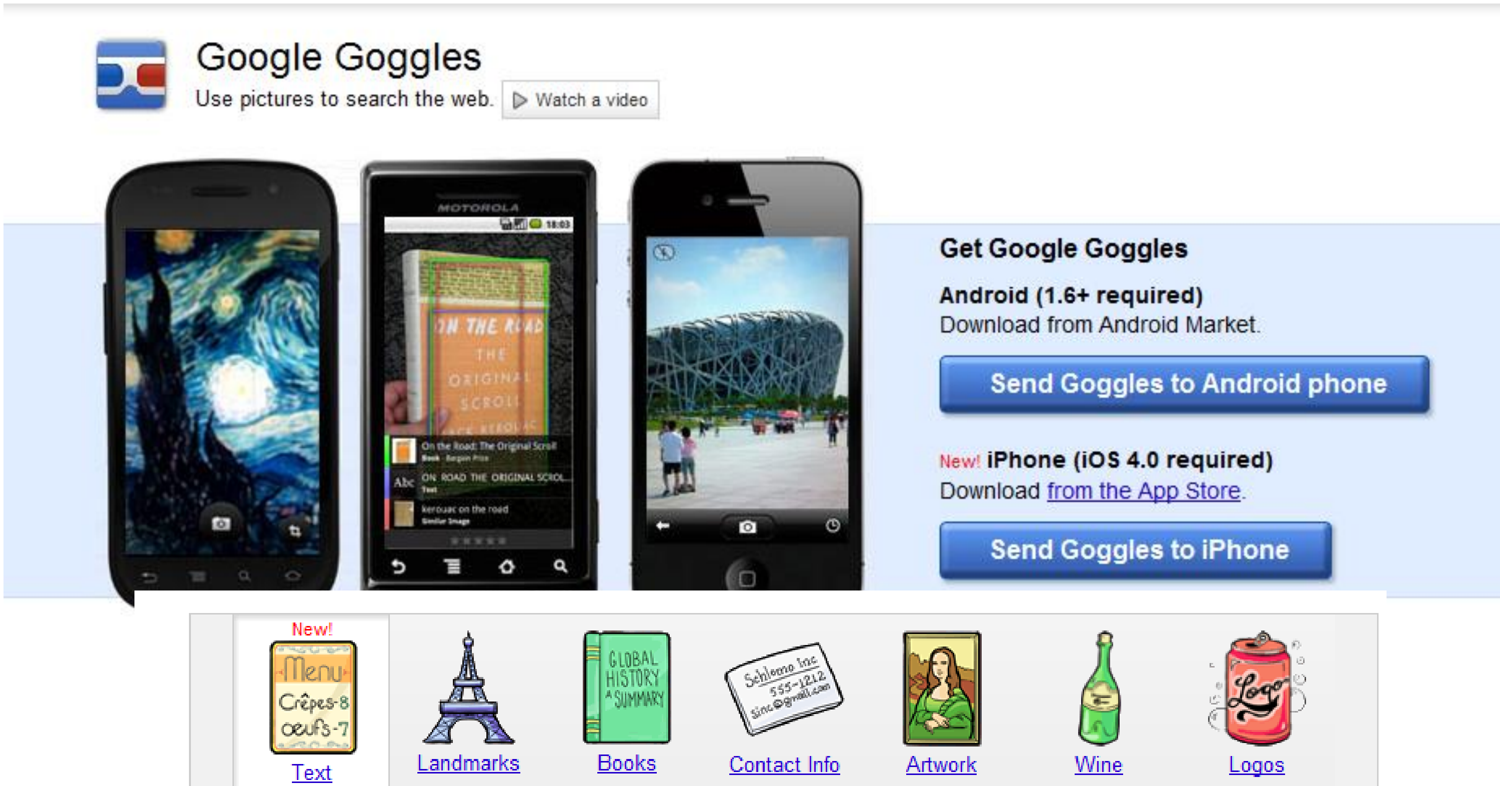
- Or, specific instances

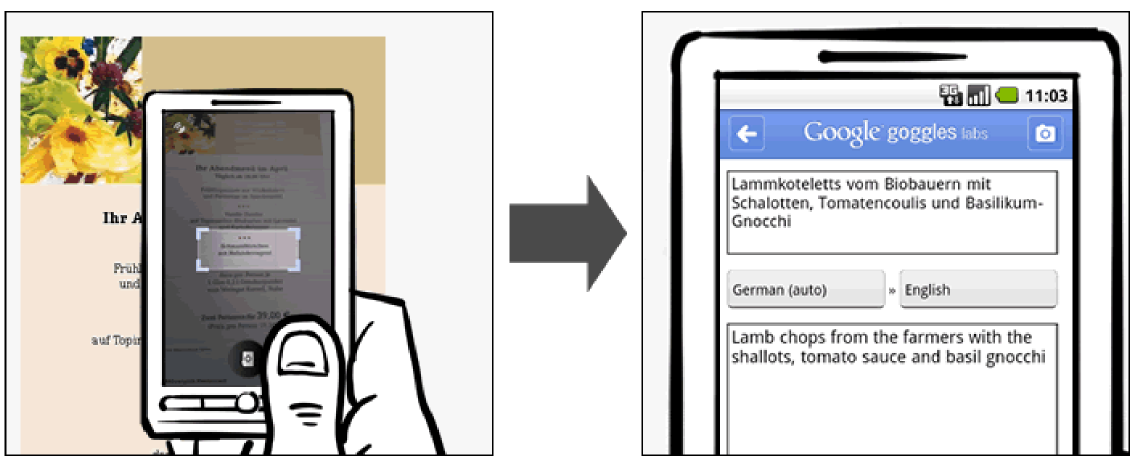
Today: image classification

- Or what else?

Applications?


https://www.google.com/search?site=imghp&tbm=isch&source=hp&biw=8
friend center princet...
Page
Safety
Tools
Google
friend center princeton
Sign in
Web
Images
Maps
Shopping
More
Search tools
SafeSearch
A DISEASE IN THE PUBLIC MIND
Thomas Fleming
MUSLIM CULTURES EXHIBITION
thrive
IEEE 2010
SARNOFF SYMPOSIUM
Princeton, New Jersey
April 12 - 14, 2010

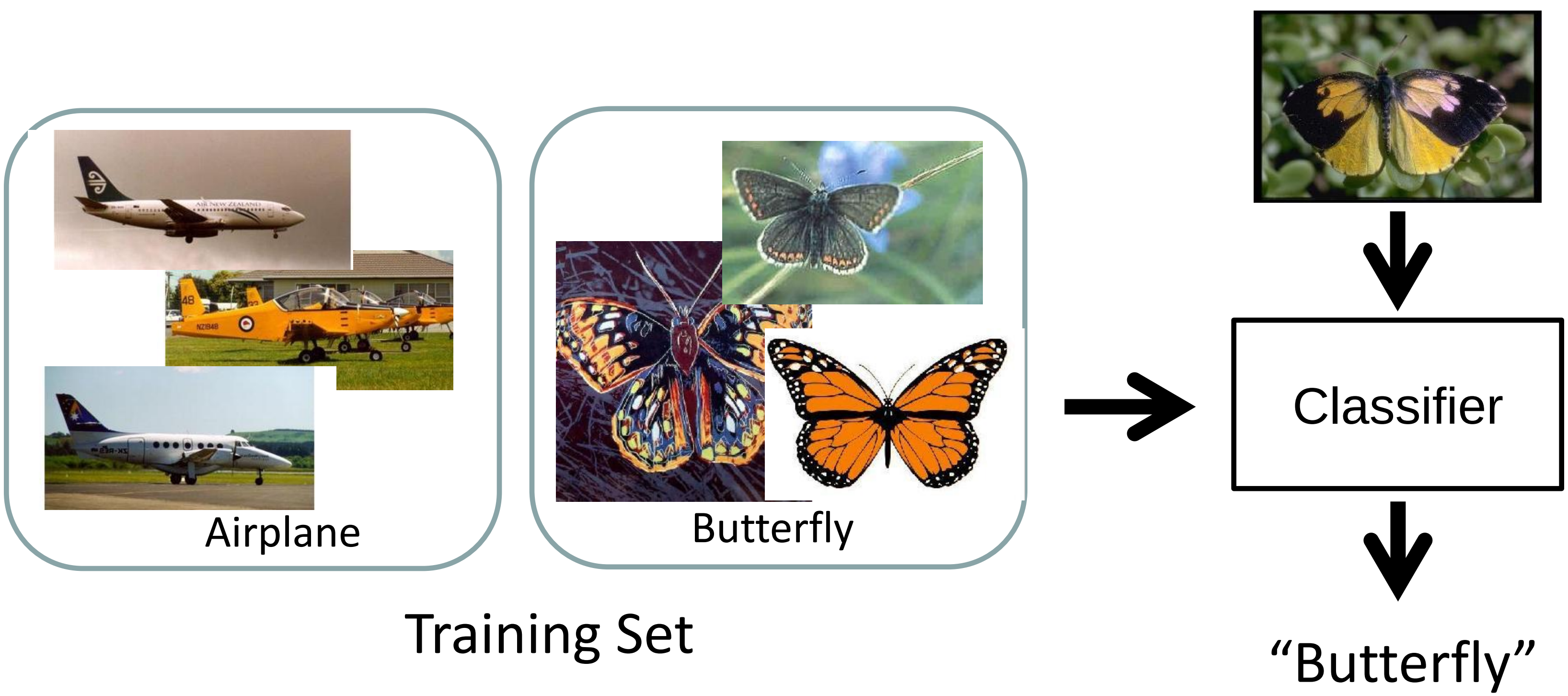

Google Goggles
Use pictures to search the web.
Watch a video
Get Google Goggles
Android (1.6+ required)
Download from Android Market.
Send Goggles to Android phone
New! iPhone (iOS 4.0 required)
Download from the App Store.
Send Goggles to iPhone
New!
Menu
Crêpes-8
oeufs-7
Text
Landmarks
GLOBAL HISTORY A SUMMARY
Books
Schlomo Inc
555-1212
Contact Info
Artwork
Wine
Logo
Logos


11:03
Google goggles labs
Lammkoteletts vom Biobauern mit Schalotten, Tomatencoulis und Basilikum-Gnocchi
German (auto)
English
Lamb chops from the farmers with the shallots, tomato sauce and basil gnocchi

Methods?

Train a Classifier

- Train a classifier on features extracted from categorized images, and then use it to predict the category of new images



Questions

- What training data?
- What features?
- What classifier?

Questions

- What training data?
- What features?
- What classifier?

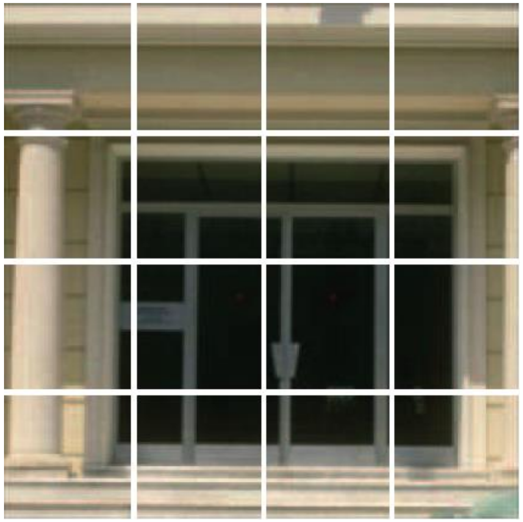
Image Classification Data

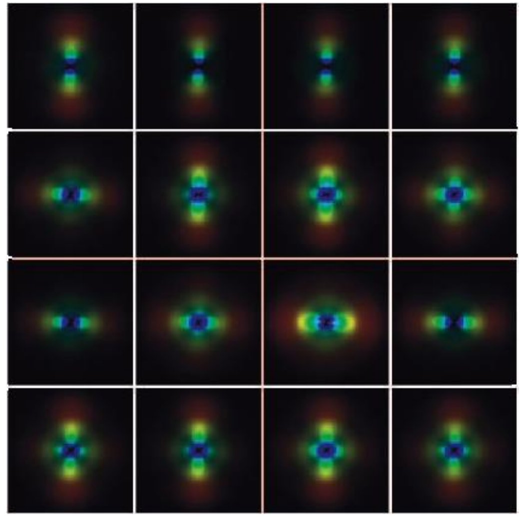
Questions

- What training data?
- What features?
- What classifier?

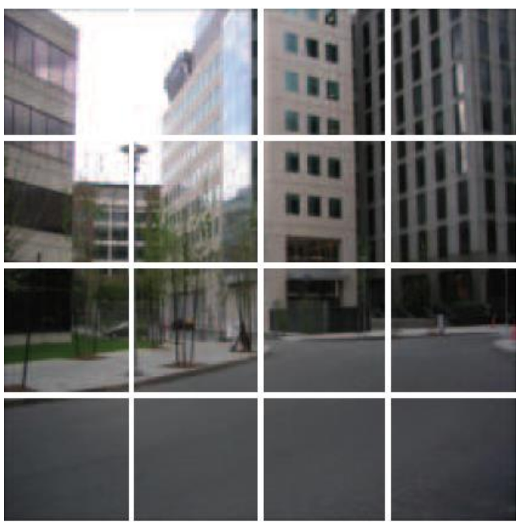
Example: Gist descriptor

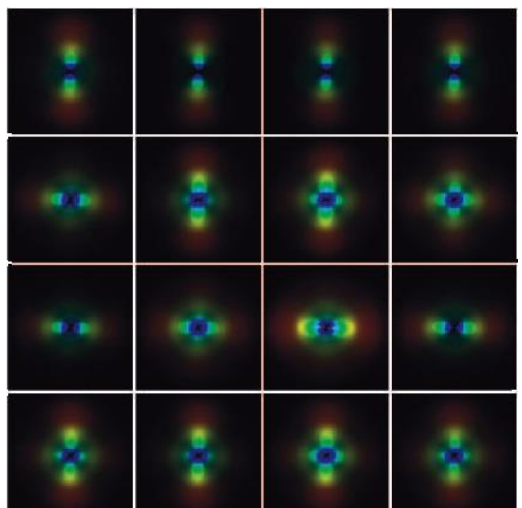
Oliva and Torralba, 2001





- Apply oriented Gabor filters over different scales
- Average filter energy in each bin

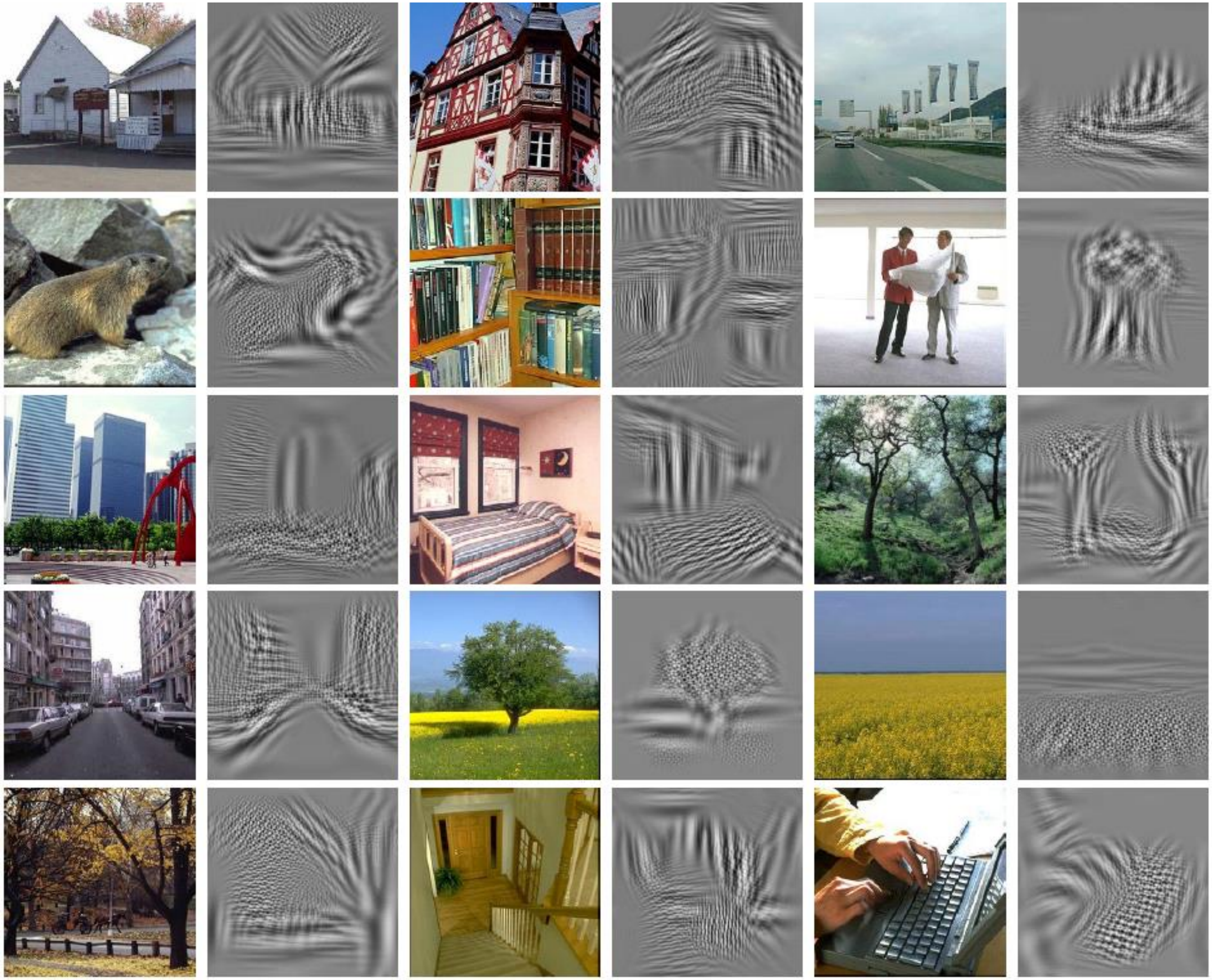




8 orientations
4 scales
x 16 bins
512 dimensions

M. Gorkani, R. Picard, ICPR 1994; Walker, Malik. Vision Research 2004; Vogel et al. 2004; Fei-Fei and Perona, CVPR 2005; S. Lazebnik, et al, CVPR 2006; …

Example: Gist descriptor



Global features (I) ~ global features (I')

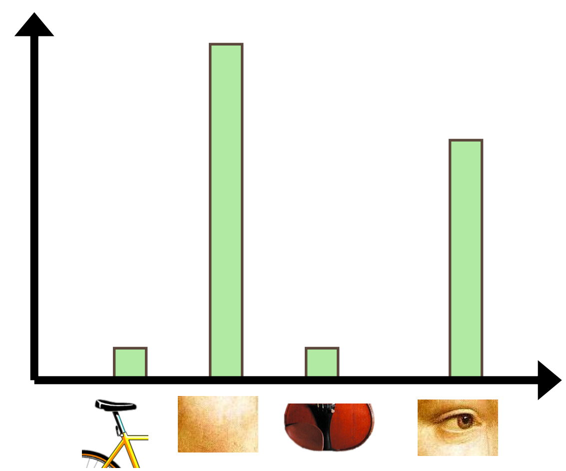
Oliva & Torralba (2001)

Example: Bag-of-words

Image → **Bag of ‘words’**

Example: Bag-of-words

Image → **Histogram of ‘word’ frequencies**



Bag-of-words models

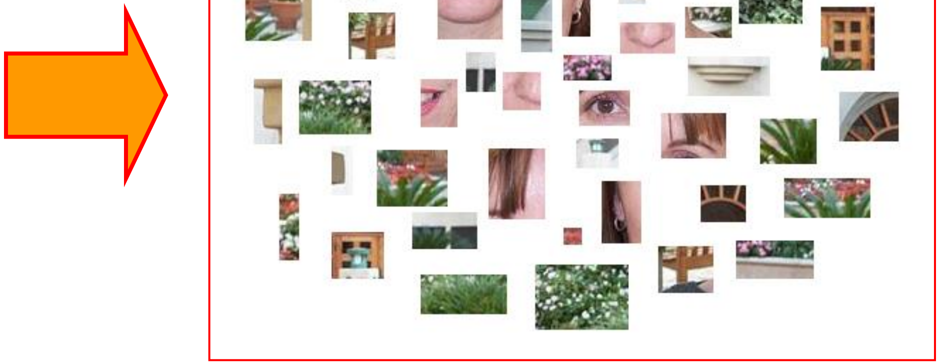
- Origin = common document representation
  Salton & McGill (1983)

US Presidential Speeches Tag Cloud
`http://chir.ag/phernalia/preztags/`

Bag of words models

- Hierarchical Bayesian models for documents (pLSA, LDA, etc.)
  - Hoffman 1999; Blei, Ng & Jordan, 2004; Teh, Jordan, Beal & Blei, 2004
- Texture recognition
  - Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003;
- Object categorization
  - Csurka, Bray, Dance & Fan, 2004; Sivic, Russell, Efros, Freeman & Zisserman, 2005; Sudderth, Torralba, Freeman & Willsky, 2005;
- Natural scene categorization
  - Vogel & Schiele, 2004; Fei-Fei & Perona, 2005; Bosch, Zisserman & Munoz, 2006

Bags of words for image classification



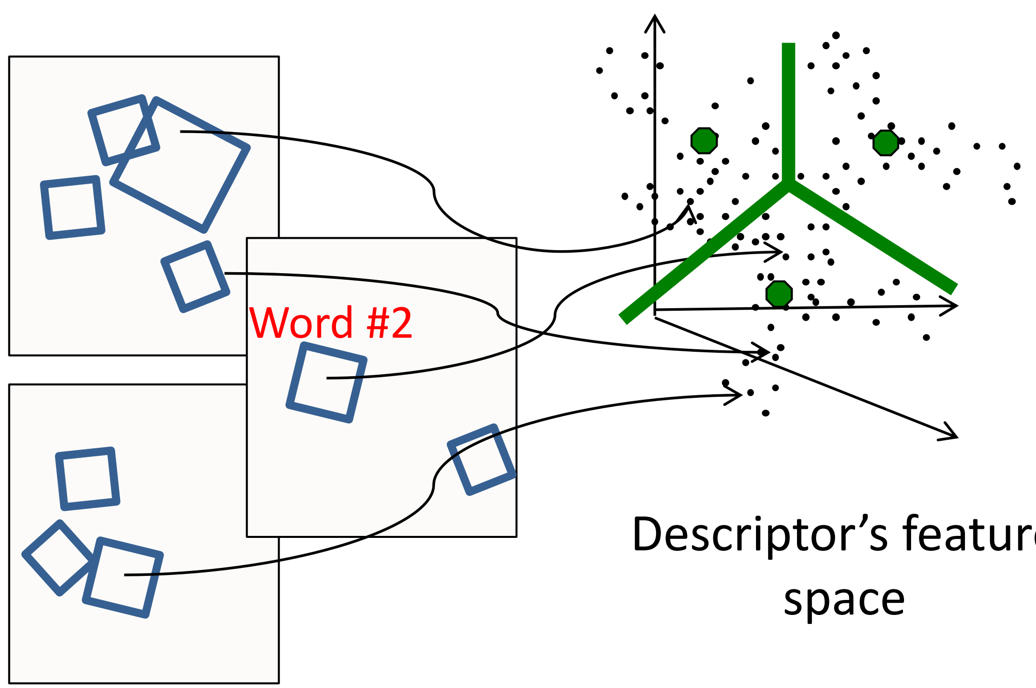
face, flowers, building

Bag of words image representation

- First, take a bunch of images, extract features, and build up a “dictionary” or “visual vocabulary” – a list of common features
- Given a new image, extract features and build a histogram – for each feature, find the closest visual word in the dictionary

Bag of words image representation

- Map high-dimensional descriptors to tokens/words by quantizing patch descriptors



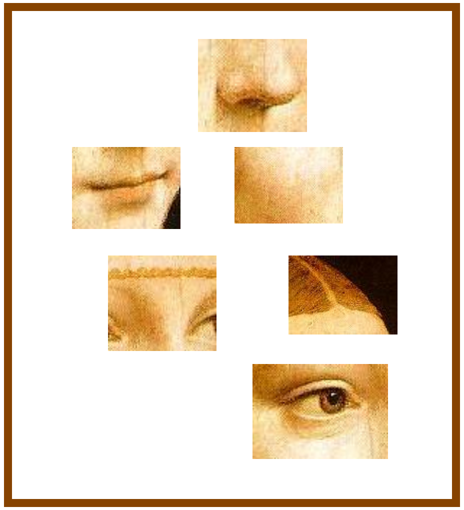
Descriptor's feature space

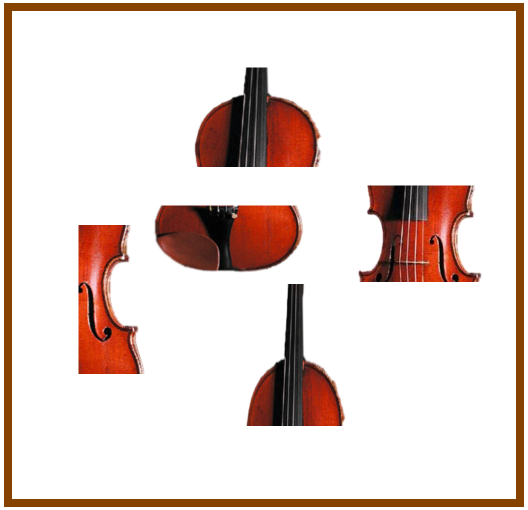
- Quantize via clustering, let cluster centers be the prototype "words"
- Determine which word to assign to each new image region by finding the closest cluster center.

Kristen Grauman

Bag of words: outline

1. Extract patches





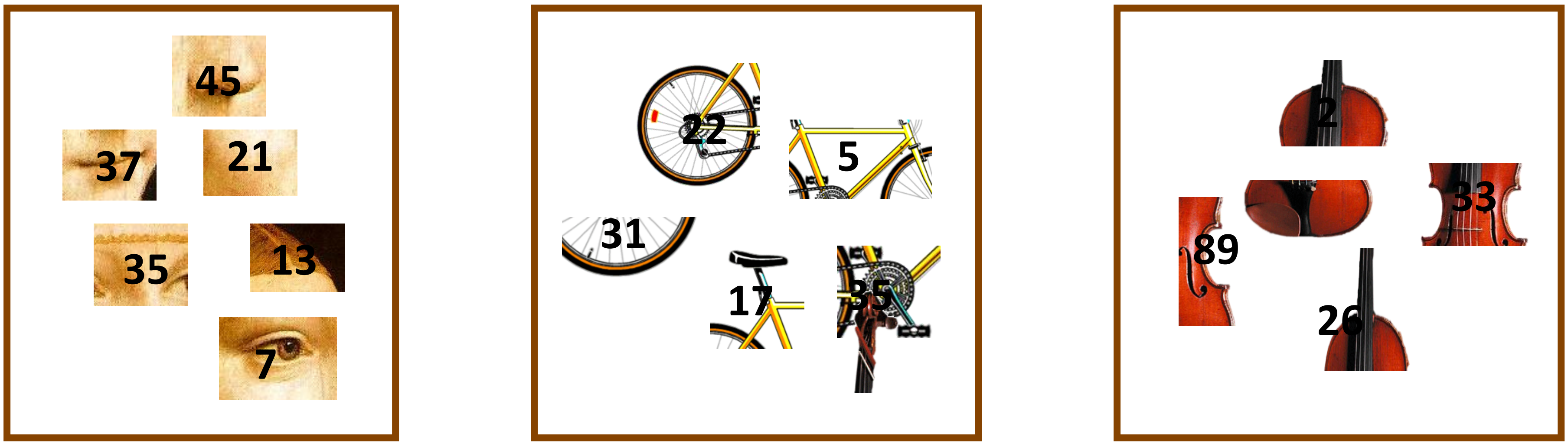
Bag of words: outline

1. Extract patches
2. Compute patch descriptors

37, 34, 555, 23, 1, 17, …

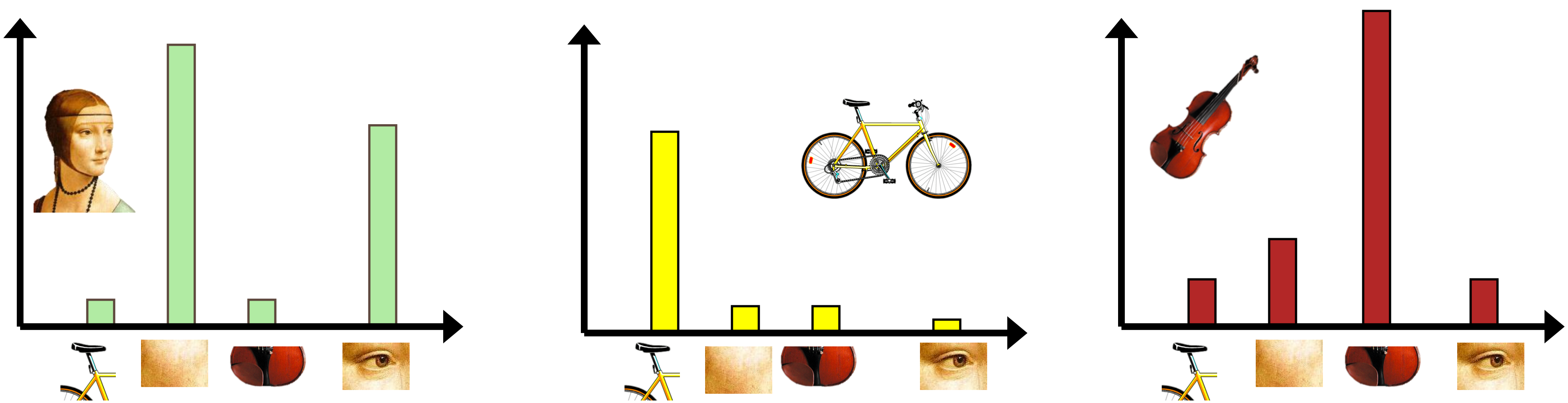
Bag of words: outline

1. Extract patches
2. Compute patch descriptors
3. Learn mapping from patch descriptors to visual “words” (cluster)



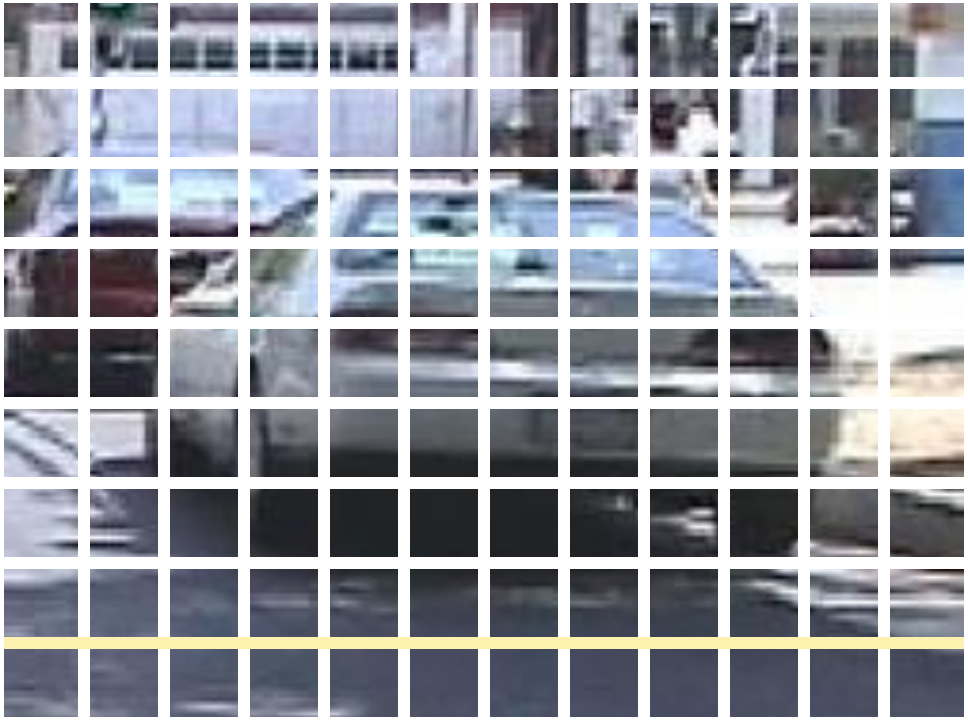
Bag of words: outline

1. Extract patches
2. Compute patch descriptors
3. Learn mapping from patch descriptors to visual "words" (cluster)
4. Represent images by "word" frequencies



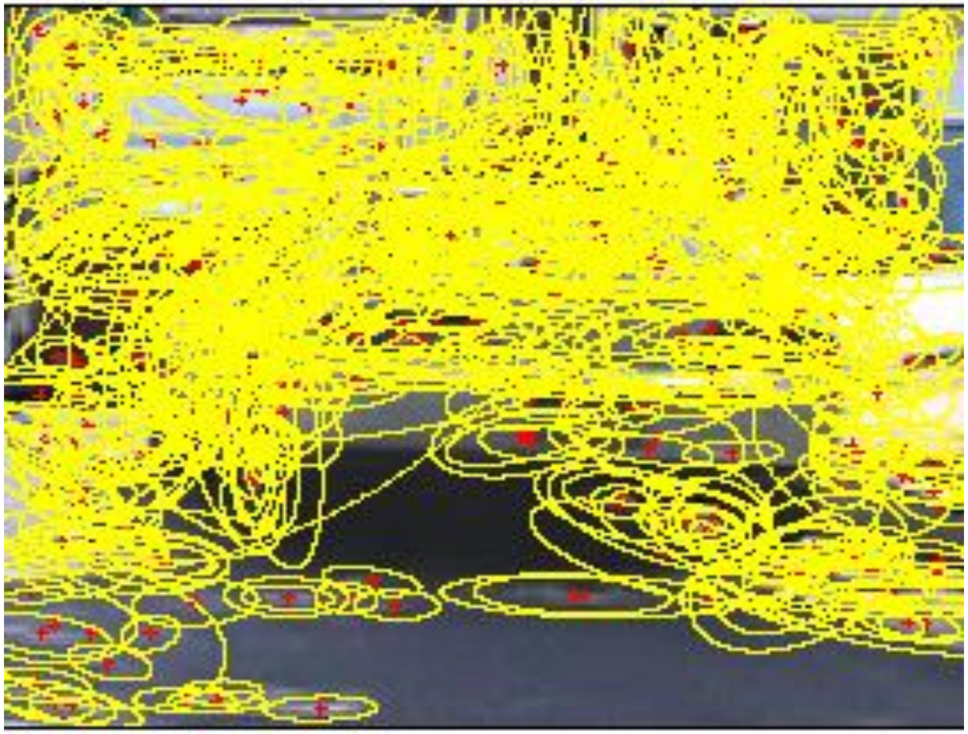
1. Extract patches

- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005



1. Extract patches

- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005
- Interest point detector
  - Csurka et al. 2004
  - Fei-Fei & Perona, 2005
  - Sivic et al. 2005

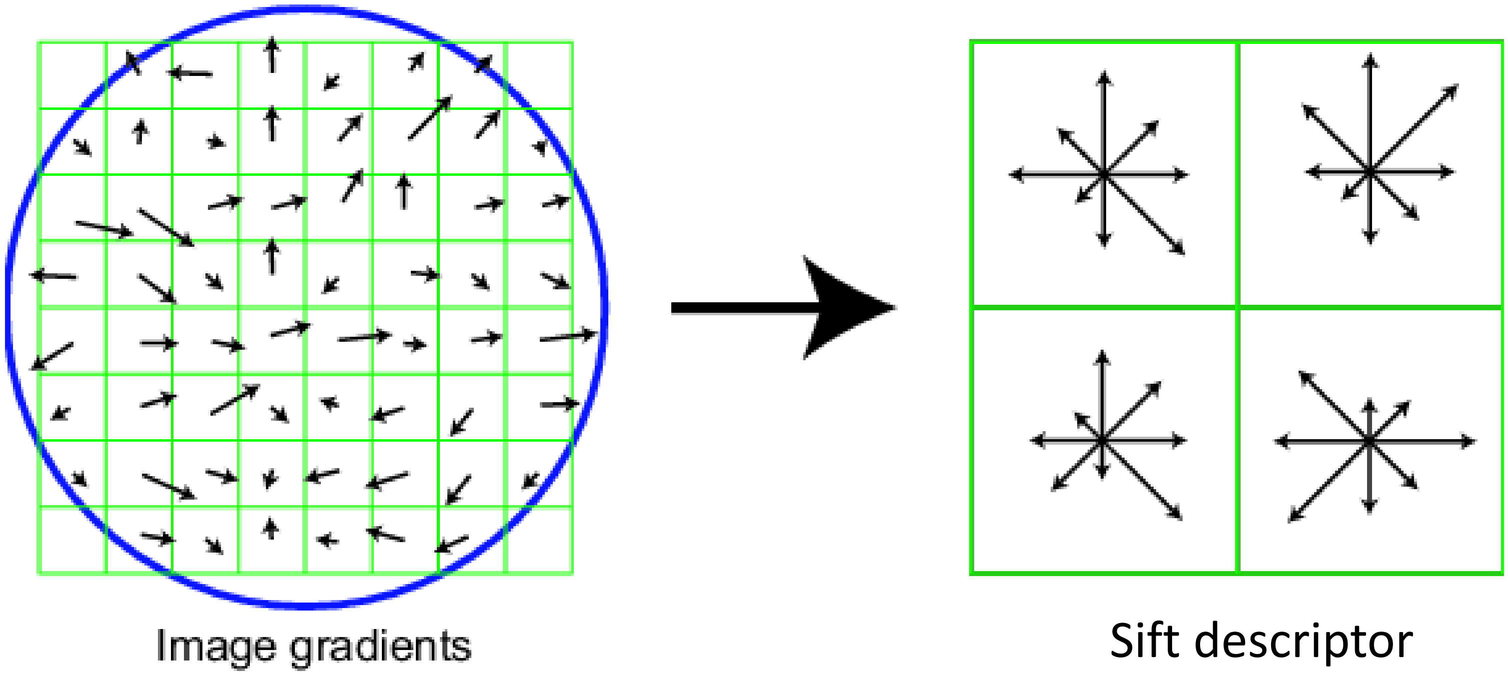


1. Extract patches

- 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)

2. Compute patch descriptors

- “Window”
- Sift
- etc.



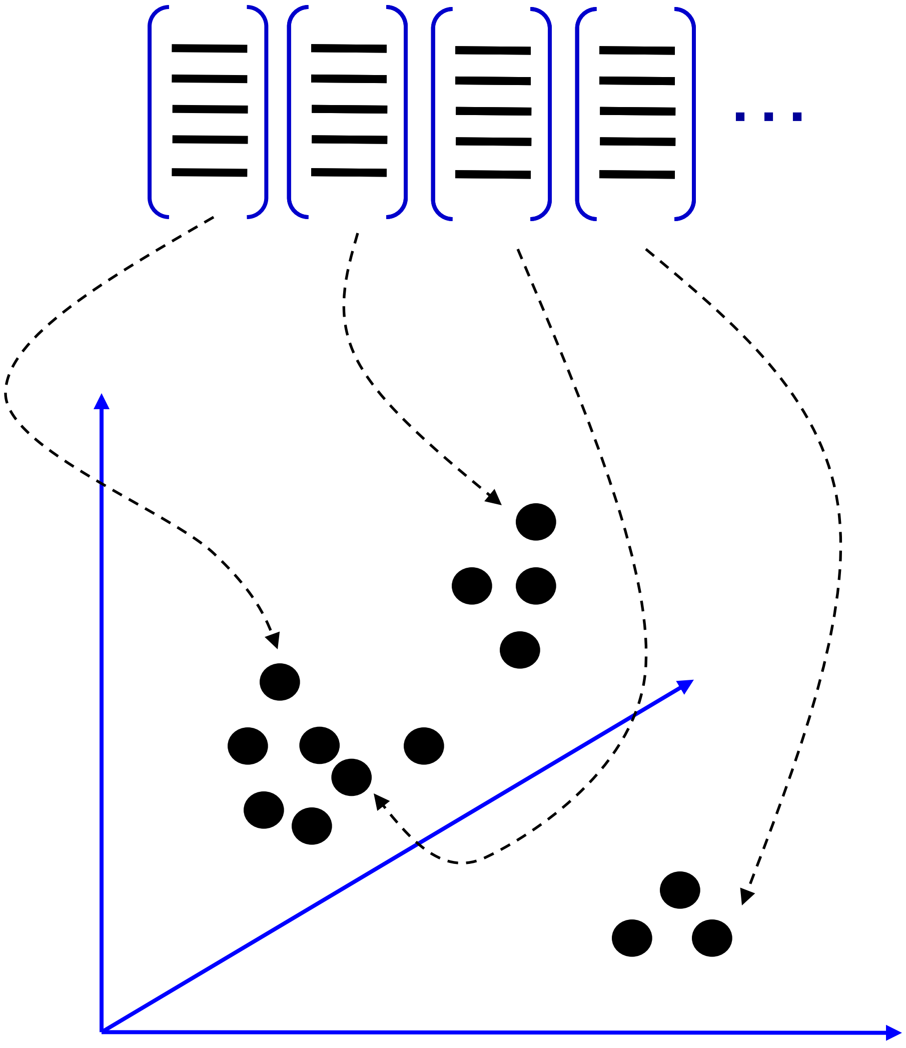
3. Learn the codebook

- **Simple option:** represent each codeword by a cluster center m_k in “descriptor space”
- **Building the codebook:** find k cluster centers than minimize the sum of distances from the patches to its closest cluster center (k-means clustering)

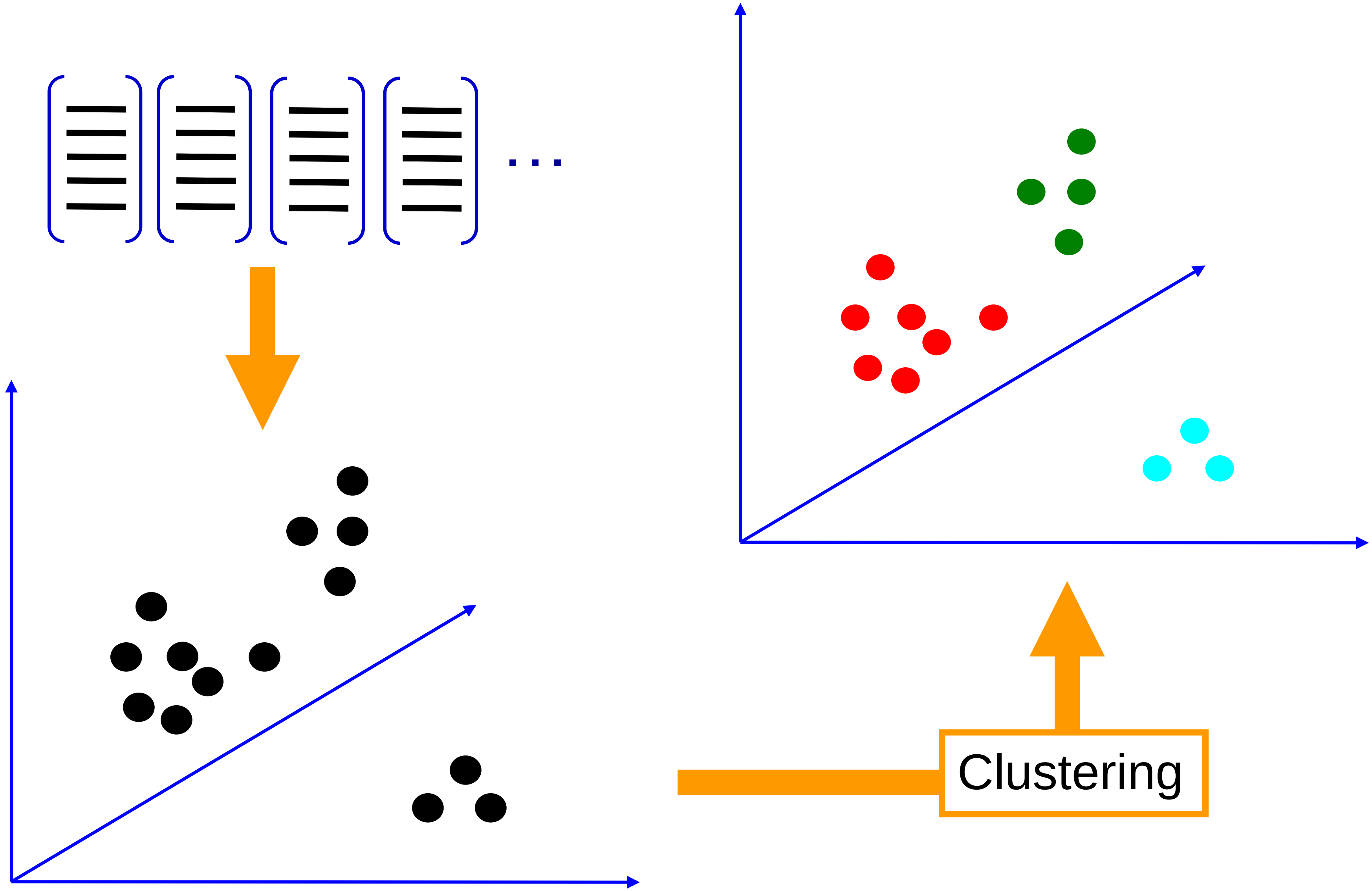
$$D(X,M) = \sum_{\text{cluster } k} \sum_{\substack{\text{point } i \text{ in} \\ \text{cluster } k}} (x_i - m_k)^2$$

- **Finding the codeword for a patch:** find closest cluster center in descriptor space

3. Learn the codebook

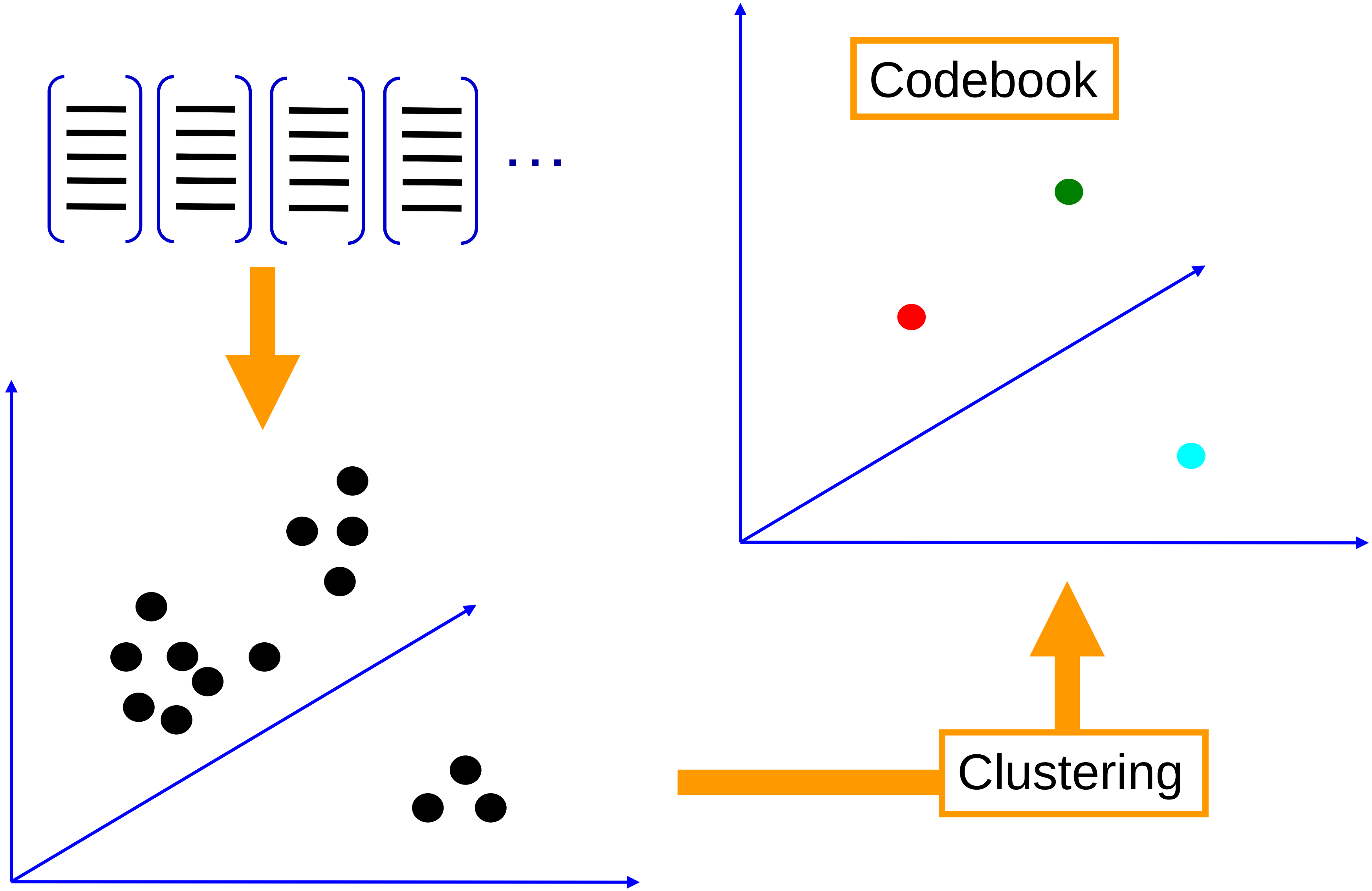


3. Learn the codebook



Slide credit: Josef Sivic

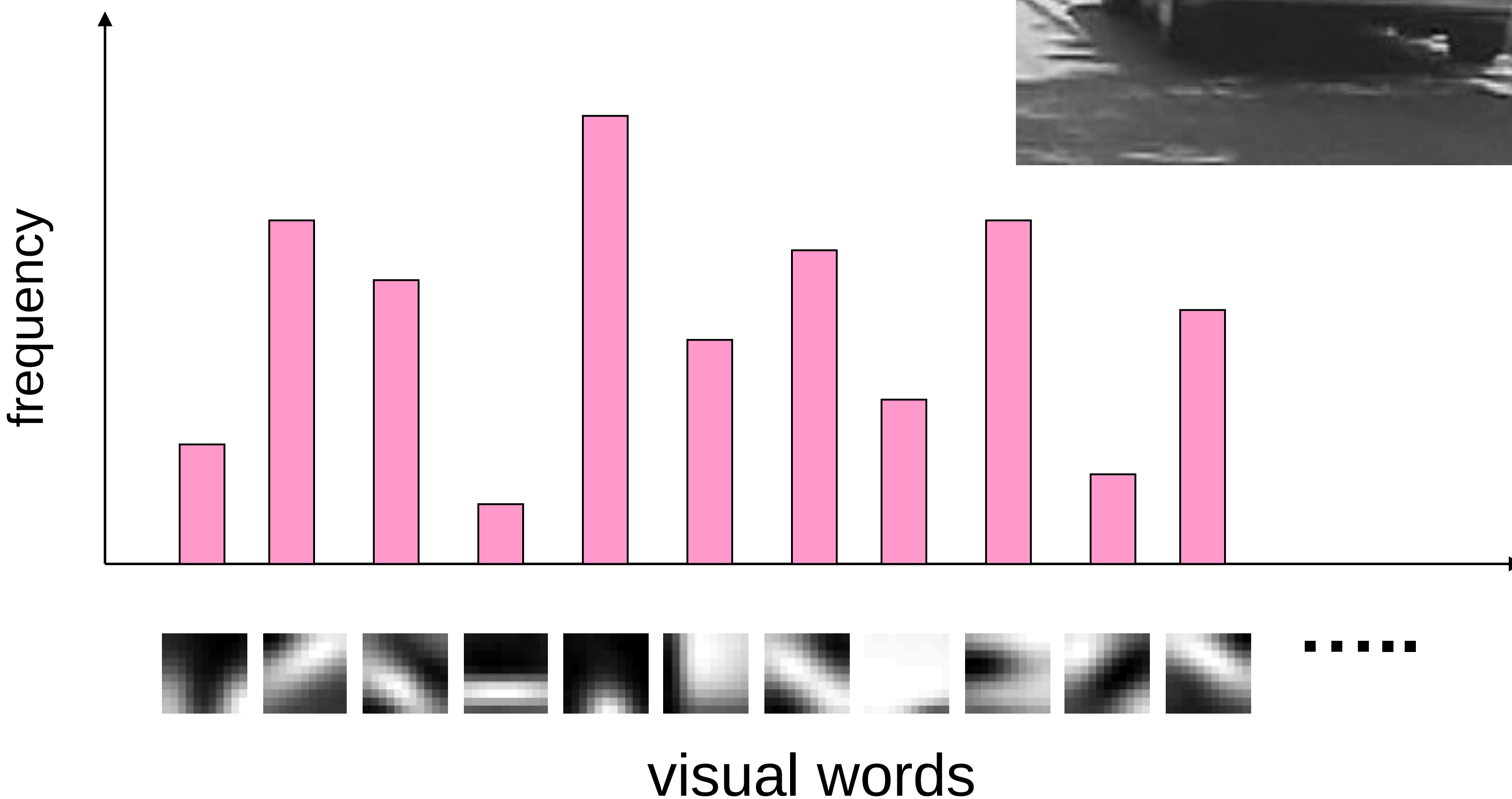
3. Learn the codebook



Slide credit: Josef Sivic

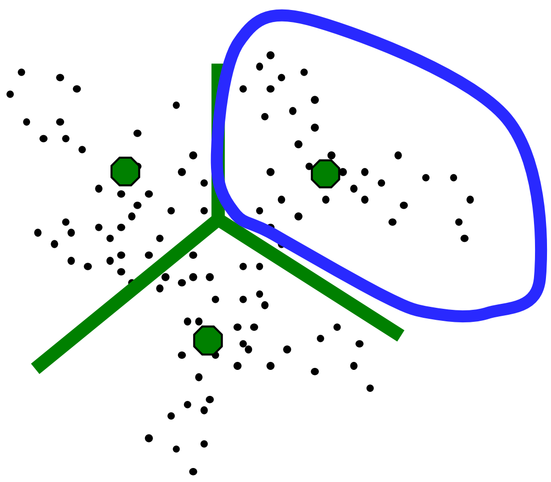
4. Represent images with histograms of word frequencies

Example Visual words



Example Visual words

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



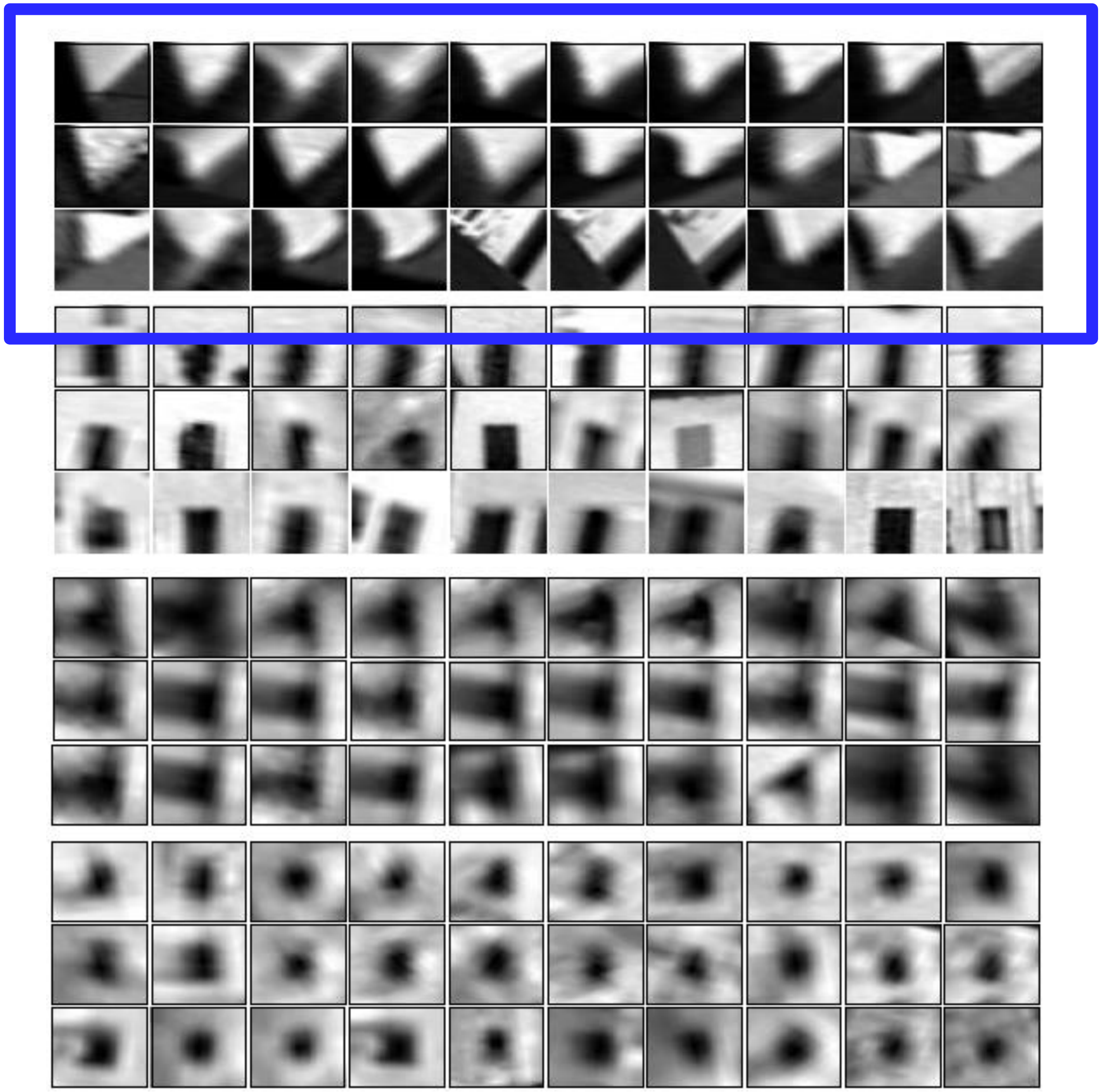
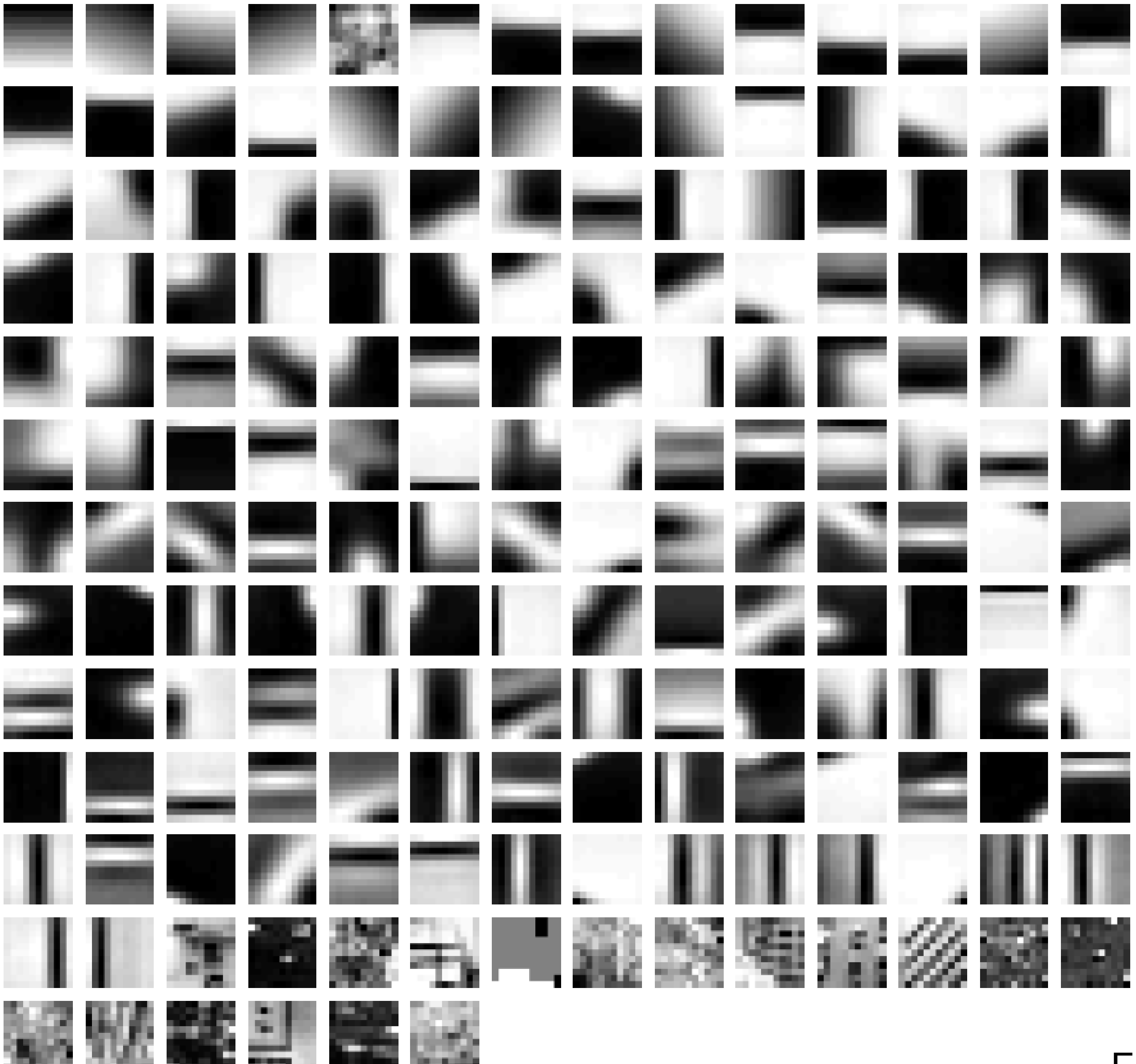


Figure from Sivic & Zisserman, ICCV 2003

Kristen Grauman

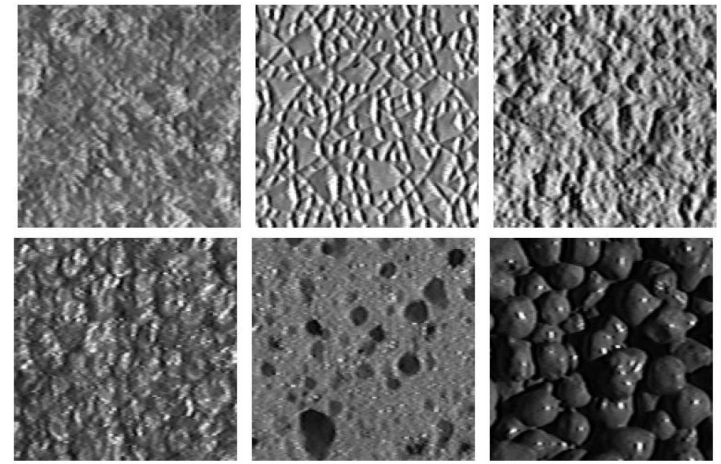
Example Codebook

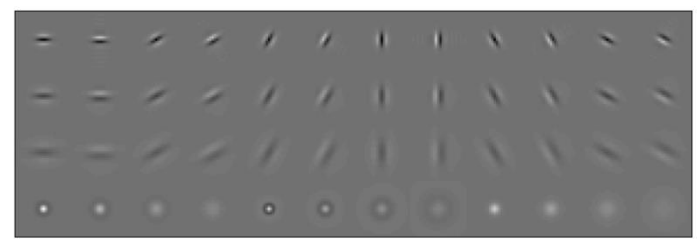


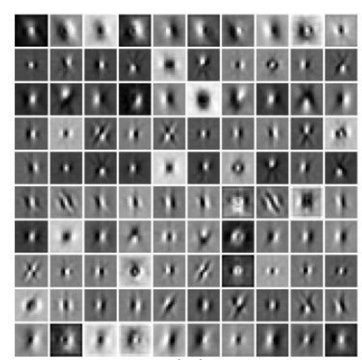
Fei-Fei et al. 2005

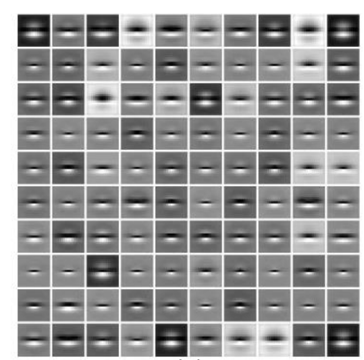
Visual words and textons

- First explored for texture and material representations
- *Texton* = cluster center of filter responses over collection of images
- Describe textures and materials based on distribution of prototypical texture elements.









Leung & Malik 1999; Varma & Zisserman, 2002

Kristen Grauman

Questions

- What training data?
- What features?
- What classifier?

Image classification

- Given a feature vector (bag-of-words) representation of images, how do we learn a model for distinguishing them from training data?

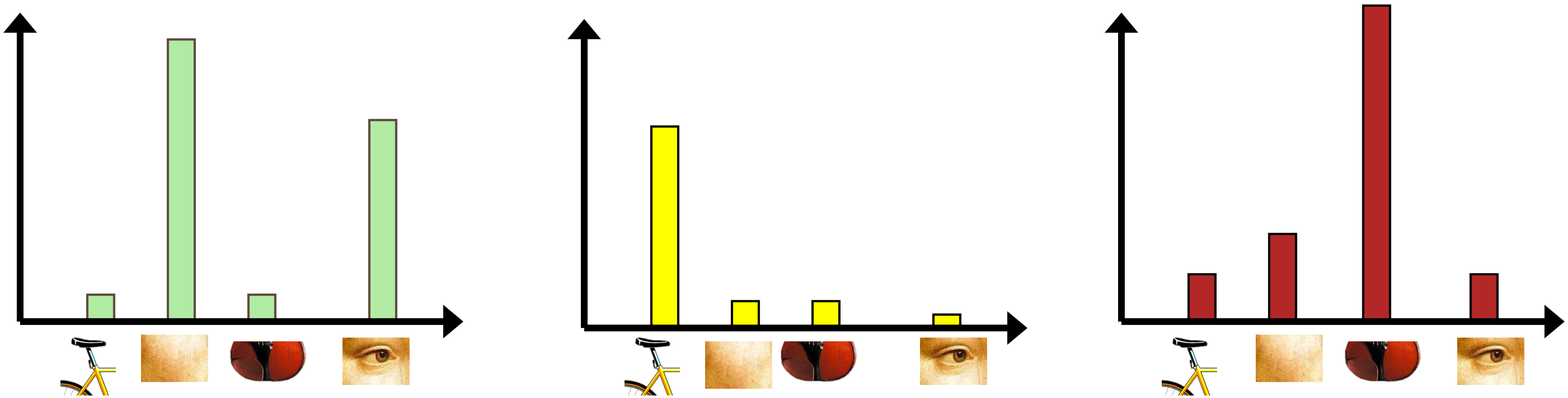


Image classification

Some classifiers:

- Nearest neighbor
- K-Nearest neighbors
- Linear classification
- Support vector machines
- Decision trees
- Naïve Bayes
- etc.

Image classification

Some classifiers:

- Nearest neighbor
- K-Nearest neighbors
- Linear classification
- Support vector machines
- Decision trees
- Naïve Bayes
- etc.

We have talked about these before

Image classification

Some classifiers:

- Nearest neighbor
- K-Nearest neighbors
- Linear classification
- Support vector machines
- Decision trees
- Naïve Bayes ← This time
- etc.

Bayes Classifier

Estimate the probability that an image belongs to each class C_i, and then choose the class $\hat{C}$ with maximum aposteri probability (MAP)

$$\hat{C} = \arg\max_i \; p(C_i \,/\, I)$$

Bayes Classifier

Assume that the probability that an image belongs to a class C_i is based on the joint probability of its patches p_j belonging to the class

$$p(C_i \mid I) = p(C_i \mid p_1, p_2, p_3, \ldots)$$

By substitution:

$$\hat{C} = \arg\max_i \; p(C_i \mid I)$$

$$\hat{C} = \arg\max_i \; p(C_i \mid p_1, p_2, p_3, \ldots)$$

Bayes Classifier

By Bayes rule:

$$p(C_i|p_1, p_2, \ldots) = p(p_1, p_2, \ldots|C_i) \frac{p(C_i)}{p(p_1, p_2, \ldots)}$$

By substitution:

$$\hat{C} = \arg\max_i \ p(C_i \,/\, p_1, p_2, p_3, \ldots)$$

$$\hat{C} = \arg\max_i \ p(p_1, p_2, \ldots|C_i) \frac{p(C_i)}{p(p_1, p_2, \ldots)}$$

Naïve Bayes Classifier

If we assume that patches are independent:

$$p(p_1, p_2, \ldots | C_i) = \prod_j p(p_j | C_i)$$

By substitution:

$$\hat{C} = \arg\max_i \; p(p_1, p_2, \ldots | C_i) \, \frac{p(C_i)}{p(p_1, p_2, \ldots)}$$

$$\hat{C} = \arg\max_i \; \prod_j p(p_j | C_i) \, \frac{p(C_i)}{p(p_1, p_2, \ldots)}$$

Naïve Bayes Classifier

If all classes are equally likely and we only care about finding the $\hat{C}$ with the maximum a posteri (MAP) probability, then the rightmost factor is irrelevant:

$$\hat{C} = \arg\max_i \prod_j p(p_j|C_i) \frac{p(C_i)}{p(p_1, p_2, \ldots)}$$

$$\hat{C} = \arg\max_i \prod_j p(p_j|C_i)$$

Naïve Bayes Classifier

If we detect the same number of patches in every image, then the patch probabilities are proportional to the counts

$$p(p_j|C_i) \propto \mathrm{count}\big(p_j \in training(C_i)\big)$$

By substitution:

$$\hat{C} = \arg\max_i \prod_j p(p_j|C_i)$$

$$\hat{C} = \arg\max_i \prod_j \mathrm{count}\big(p_j \in training(C_i)\big)$$

Naïve Bayes Classifier

To avoid effects of zero patch probabilities:

$$\hat{C} = \arg\max_i \prod_j (1 + \mathrm{count}\big(p_j \in training(C_i)\big)$$

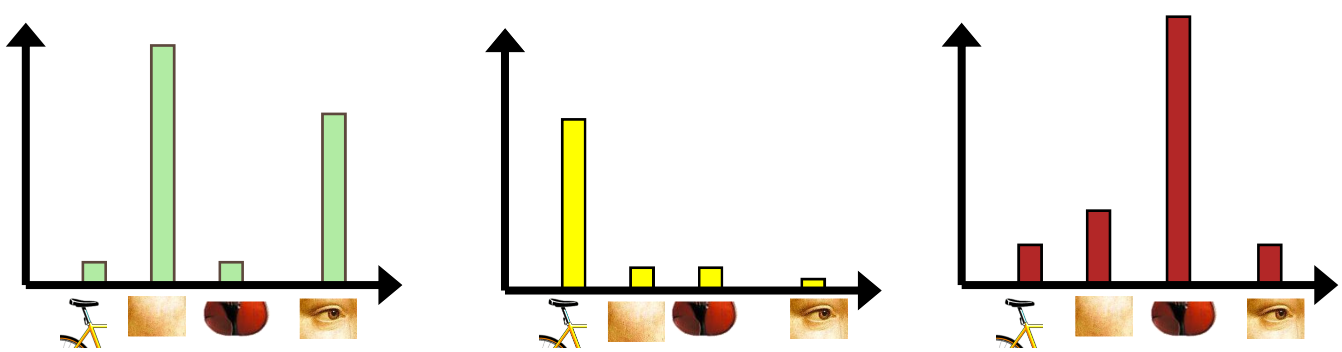
To avoid effects of finite precision math:

$$\hat{C} = \arg\max_i \sum_j \log(1 + \mathrm{count}\big(p_j \in training(C_i)\big))$$

Putting it all together

Training phase:

1. Select N patch locations from every training image
2. Compute descriptor for every patch
3. Cluster patch descriptors into codewords
   (store center of each cluster found with k-means
   so can map patches to codewords later)
4. Learn histograms of codewords for each class



Class 1 Class 2 Class 3

Putting it all together

Testing phase:

1. Select N patch locations from every training image
2. Compute descriptor for every patch
3. Build histogram of codewords
4. Classify image based on its histogram of codewords; if Naïve Bayes:

$$\hat{C} = \arg\max_i \sum_j \log(\, 1 + \mathrm{count}\big(p_j \in training(C_i)\big)\,)$$

Some Useful Extensions

- Inverted file index
- Weighting of words
- Spatial indexing

Some Useful Extensions

- Inverted file index ←
- Weighting of words
- Spatial indexing

Inverted file index

- Can quickly use the inverted file to compute similarity between a new image and all the images in the database
  – Only consider database images whose bins overlap the query image

Inverted file index

Database images

Image #1

Image #2

Image #3

Word #	Image #
1	3
2 …	
7	1, 2
8	3
9	
10 …	
91	2

…

⋮ ⋮ ⋮

- Database images are loaded into the index mapping words to image numbers

Kristen Grauman

Some Useful Extensions

- Inverted file index
- Weighting of words
- Spatial indexing

Weighting the words

- Just as with text, some visual words are more discriminative than others

the, and, or vs. ***cow, AT&T, Cher***

- the bigger fraction of the documents a word appears in, the less useful it is for matching
  - e.g., a word that appears in *all* documents is not helping us

tf-idf weighting

- **T**erm **f**requency – **i**nverse **d**ocument **f**requency
- Describe frame by frequency of each word within it, downweight words that appear often in the database
- (Standard weighting for text retrieval)

Number of occurrences of word i in document d

Total number of documents in database

$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$$

Number of words in document d

Number of documents word i occurs in, in whole database

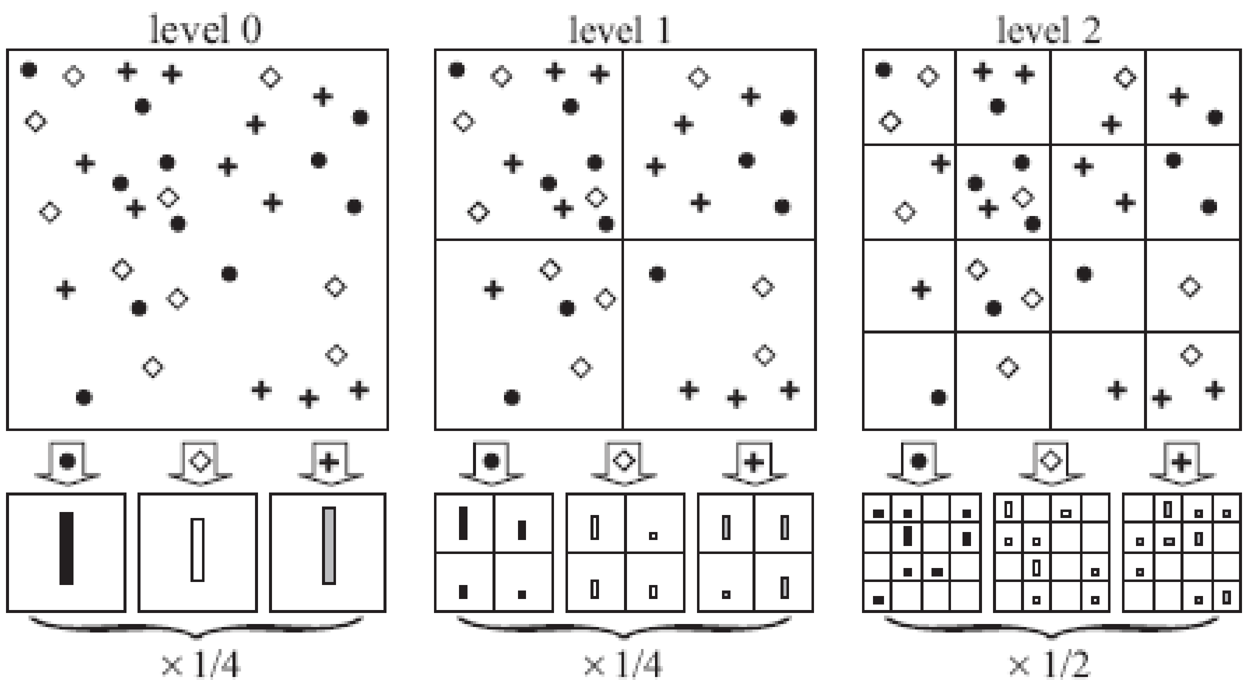
Kristen Grauman

Some Useful Extensions

- Inverted file index
- Weighting of words
- Spatial indexing

Spatial Indexing

- Build separate histograms of visual words for different regions of image
  - Pyramid match kernel for discriminative classification



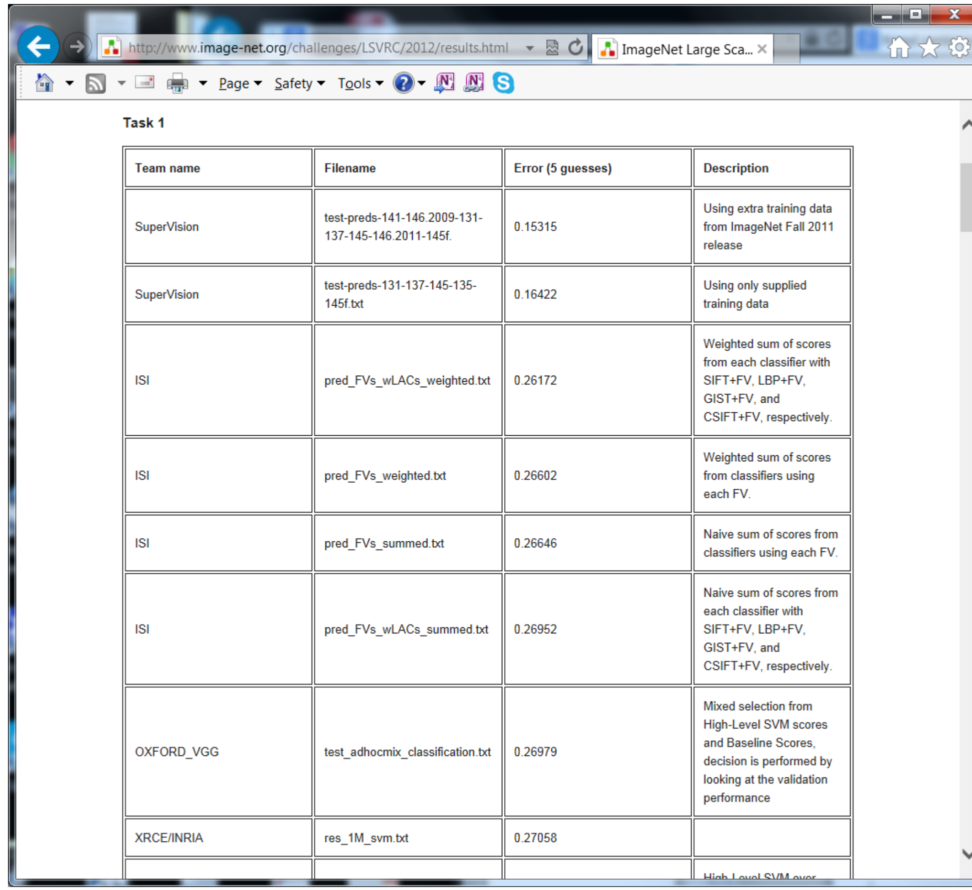
Labesnik

Other methods

Fisher kernels

Deep learning

etc.



Task 1

Team name	Filename	Error (5 guesses)	Description
SuperVision	test-preds-141-146.2009-131-137-145-146.2011-145f.	0.15315	Using extra training data from ImageNet Fall 2011 release
SuperVision	test-preds-131-137-145-135-145f.txt	0.16422	Using only supplied training data
ISI	pred_FVs_wLACs_weighted.txt	0.26172	Weighted sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and CSIFT+FV, respectively.
ISI	pred_FVs_weighted.txt	0.26602	Weighted sum of scores from classifiers using each FV.
ISI	pred_FVs_summed.txt	0.26646	Naive sum of scores from classifiers using each FV.
ISI	pred_FVs_wLACs_summed.txt	0.26952	Naive sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and CSIFT+FV, respectively.
OXFORD_VGG	test_adhocmix_classification.txt	0.26979	Mixed selection from High-Level SVM scores and Baseline Scores, decision is performed by looking at the validation performance
XRCE/INRIA	res_1M_svm.txt	0.27058	
			High-Level SVM over

ImageNet Challenge 2012

Evaluating the Results

How can we evaluate classification results?

Training / Test Sets

- Divide labeled data into two sets
- Use the first to train model (learn a classifier)
- Use the second to test model (classify and then check if right)

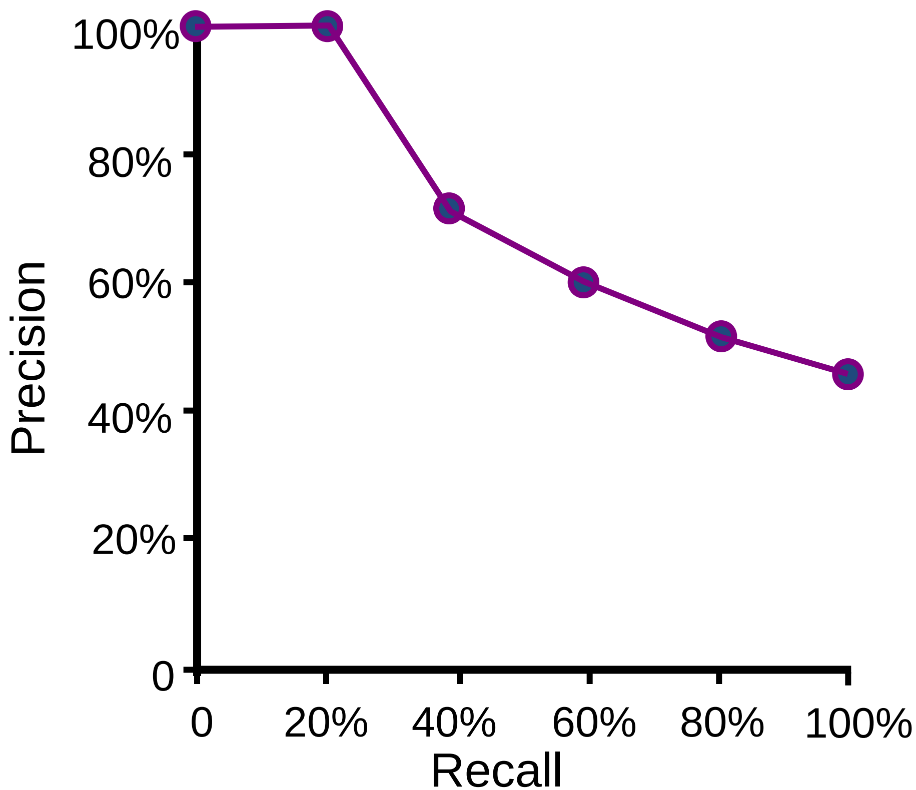
k-Way Cross-Validation

- Divide data into *k* (traditionally 10) partitions
- Train on *k*-1 of them, test on remaining one
- Repeat *k* times, report average test error

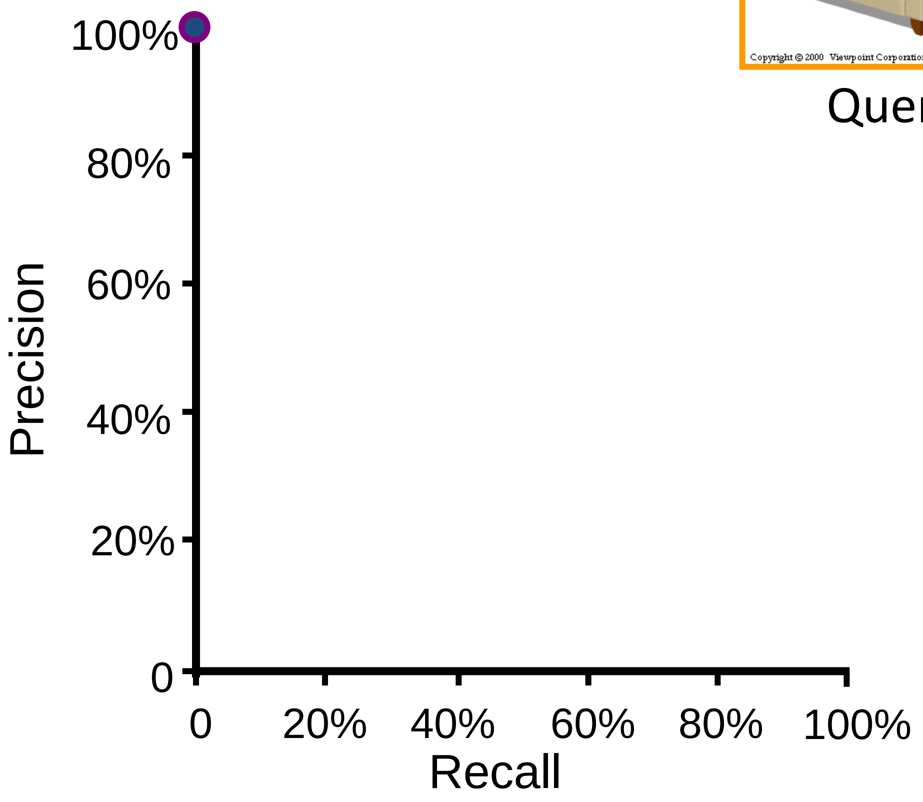
- Uses limited data more efficiently

Reporting Classification (Retrieval) Error: Precision-Recall Curves

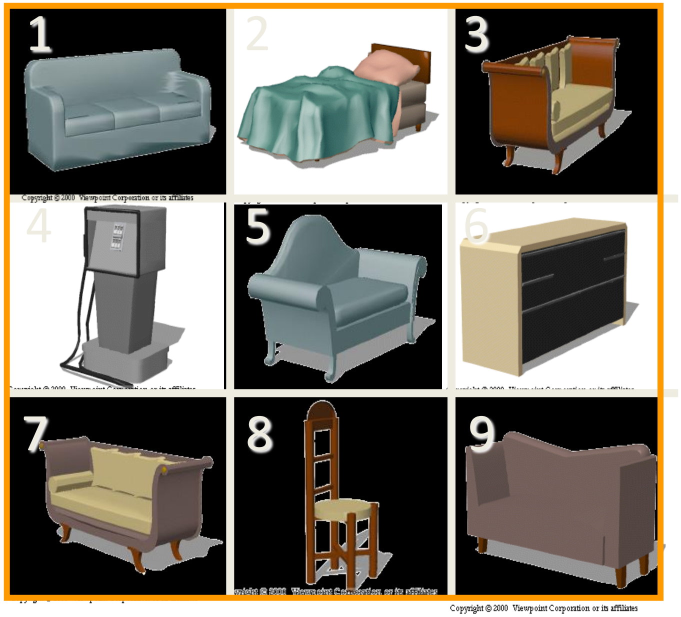
- Precision = retrieved_in_class / total_retrieved
- Recall = retrieved_in_class / total_in_class



Reporting Classification (Retrieval) Error: Precision-Recall Curves

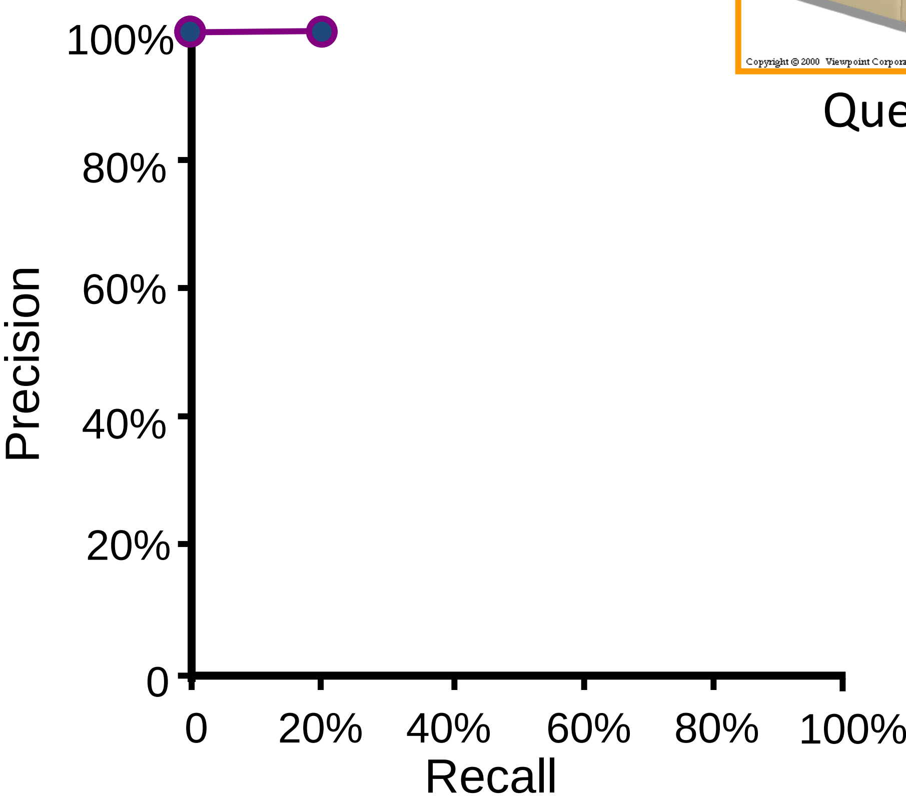


Query

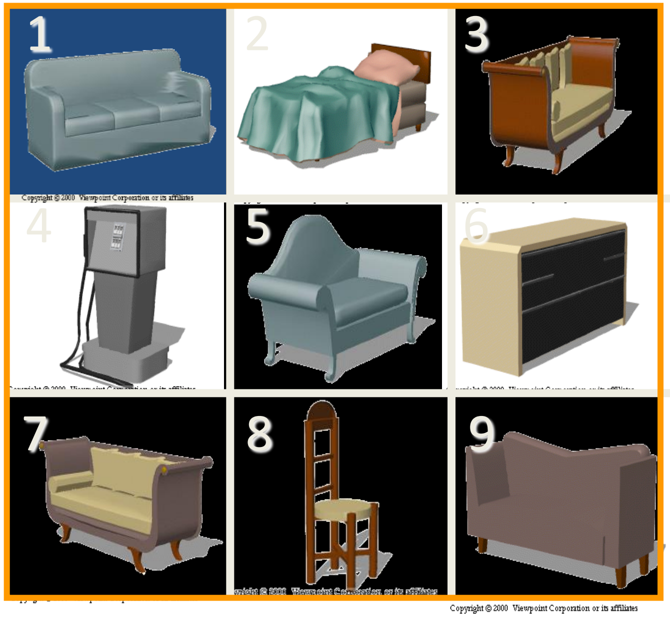


Ranked Matches

Reporting Classification (Retrieval) Error: Precision-Recall Curves

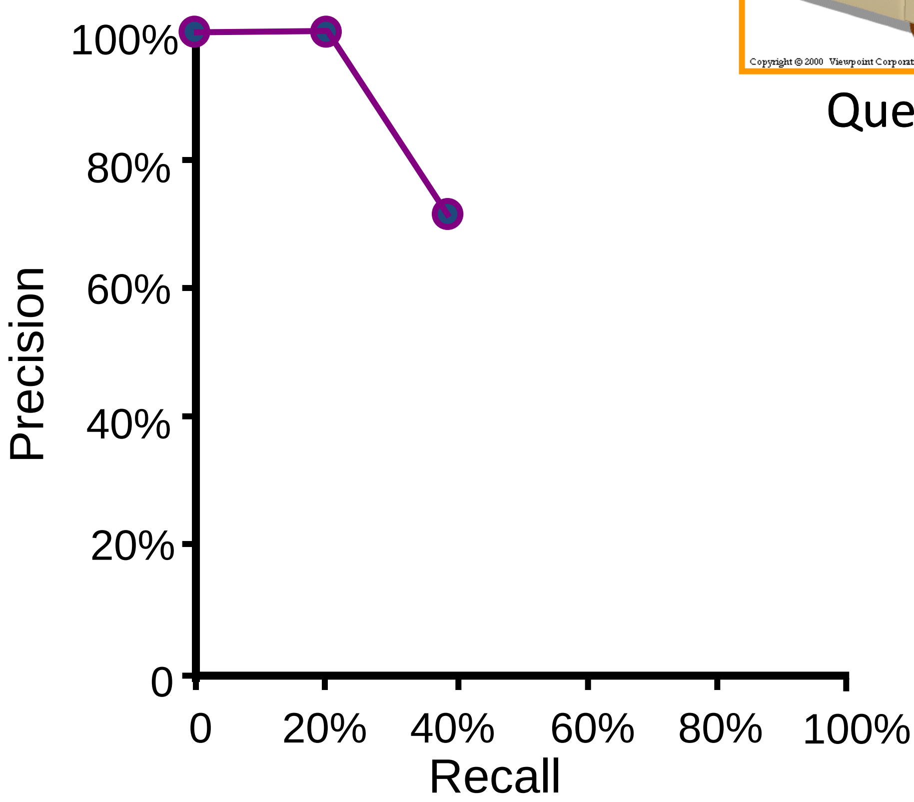


Query

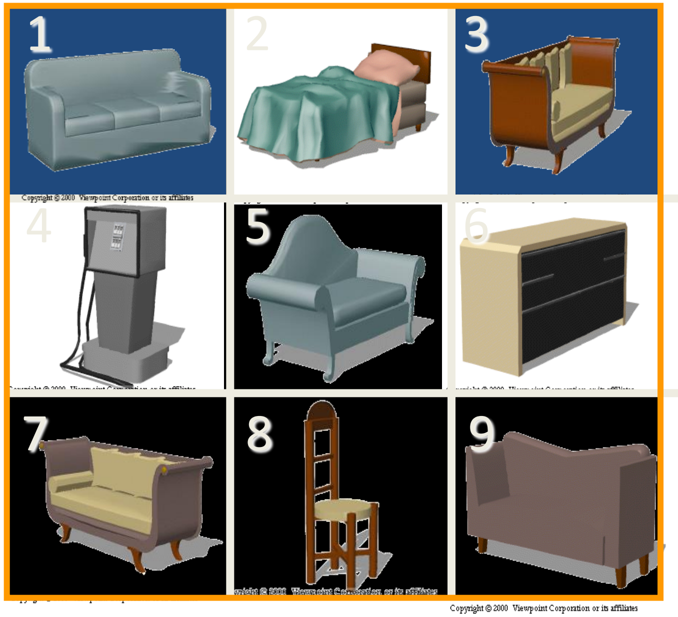


Ranked Matches

Reporting Classification (Retrieval) Error: Precision-Recall Curves

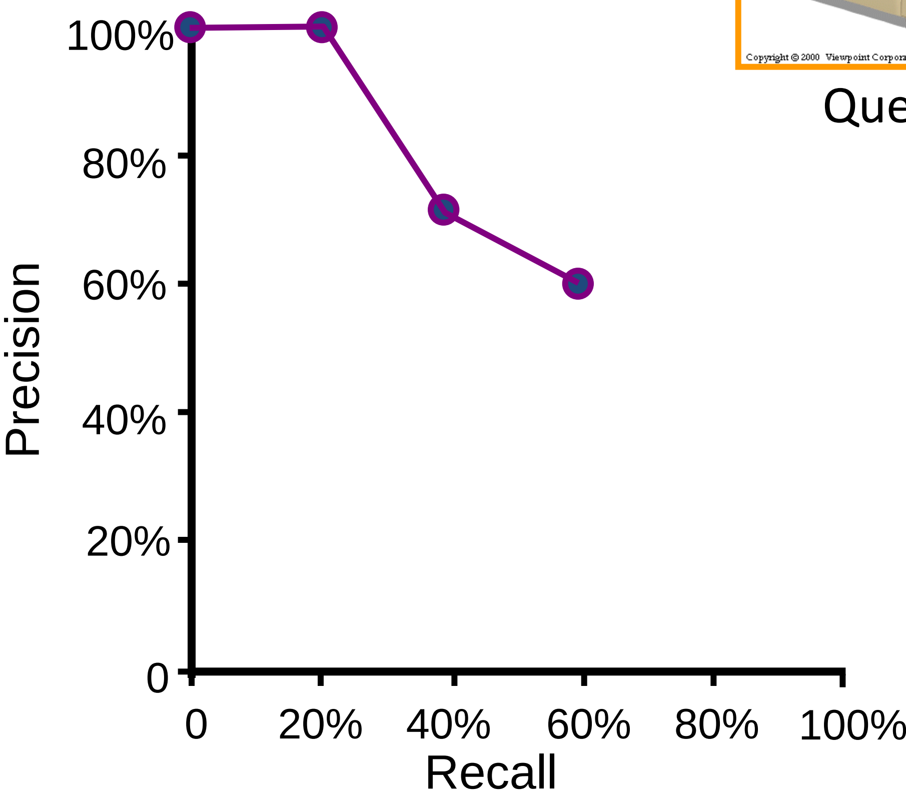


Query

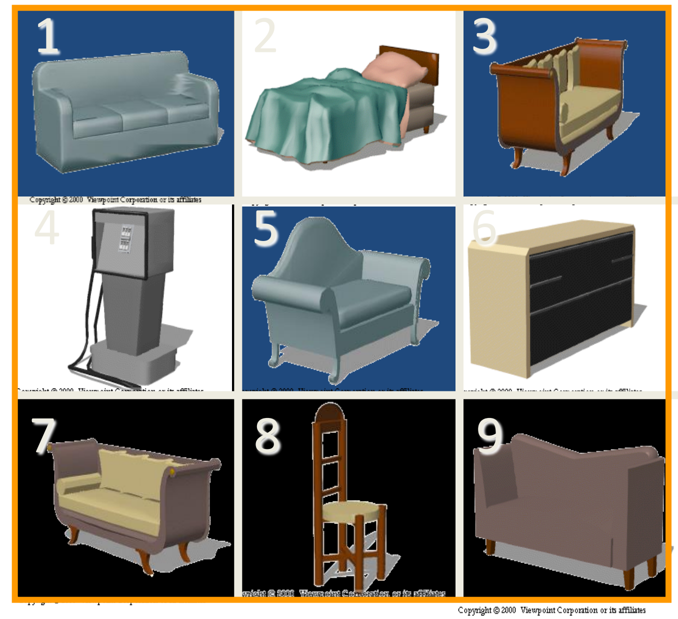


Ranked Matches

Reporting Classification (Retrieval) Error: Precision-Recall Curves

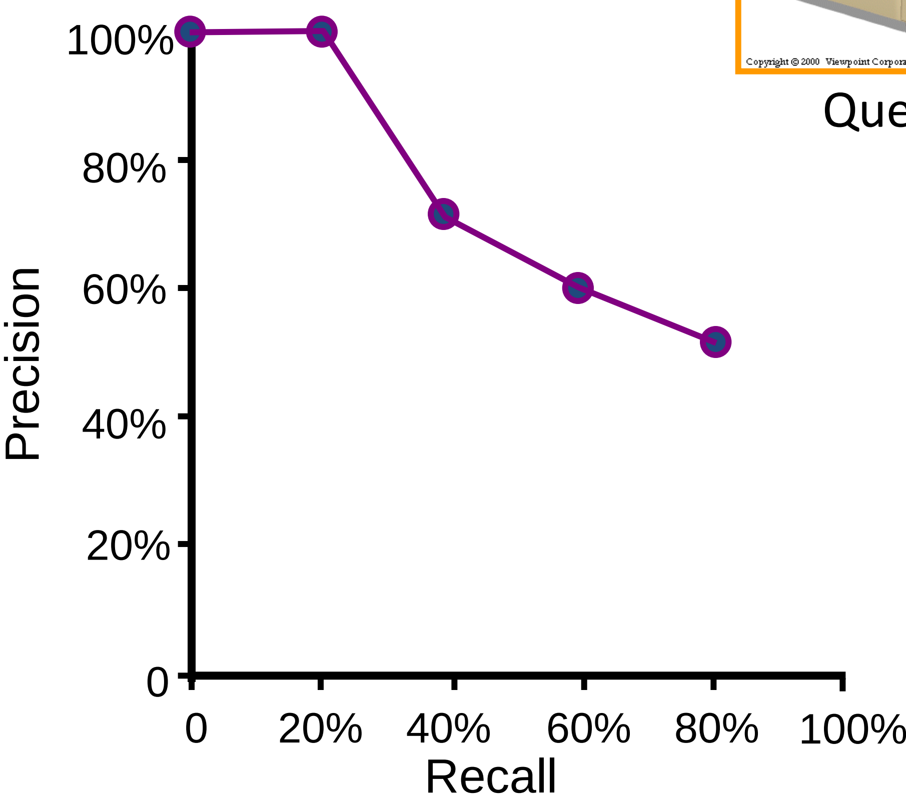


Query

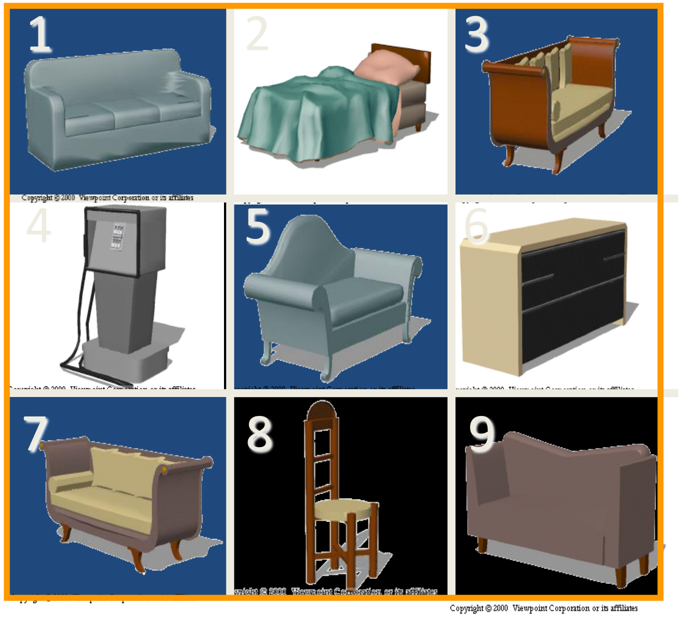


Ranked Matches

Reporting Classification (Retrieval) Error: Precision-Recall Curves

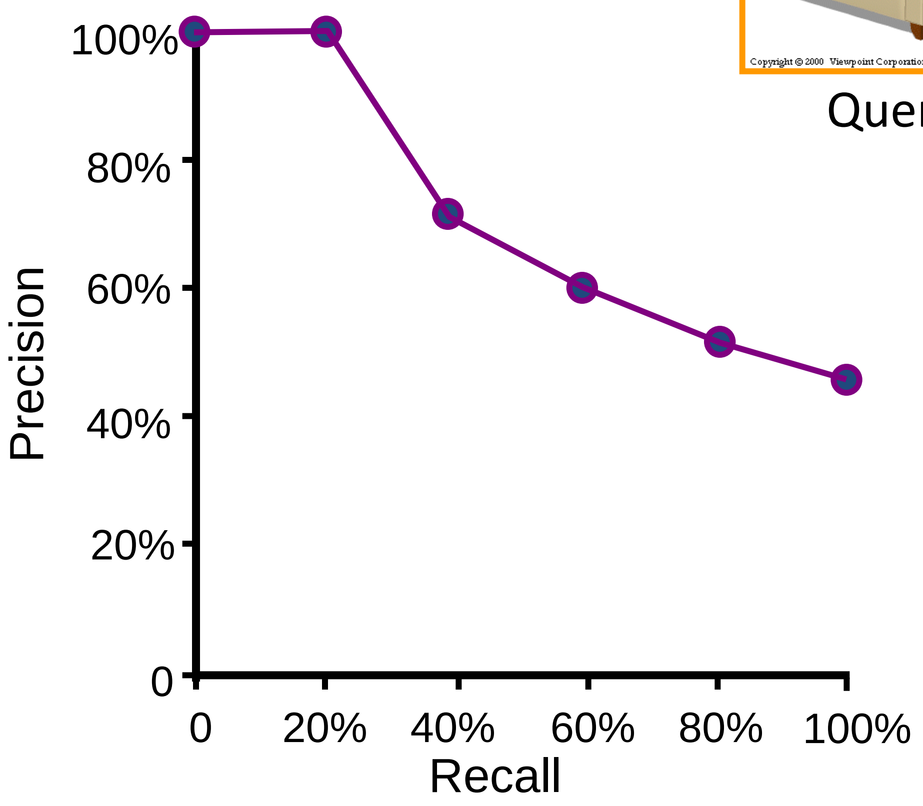


Query



Ranked Matches

Reporting Classification (Retrieval) Error: Precision-Recall Curves

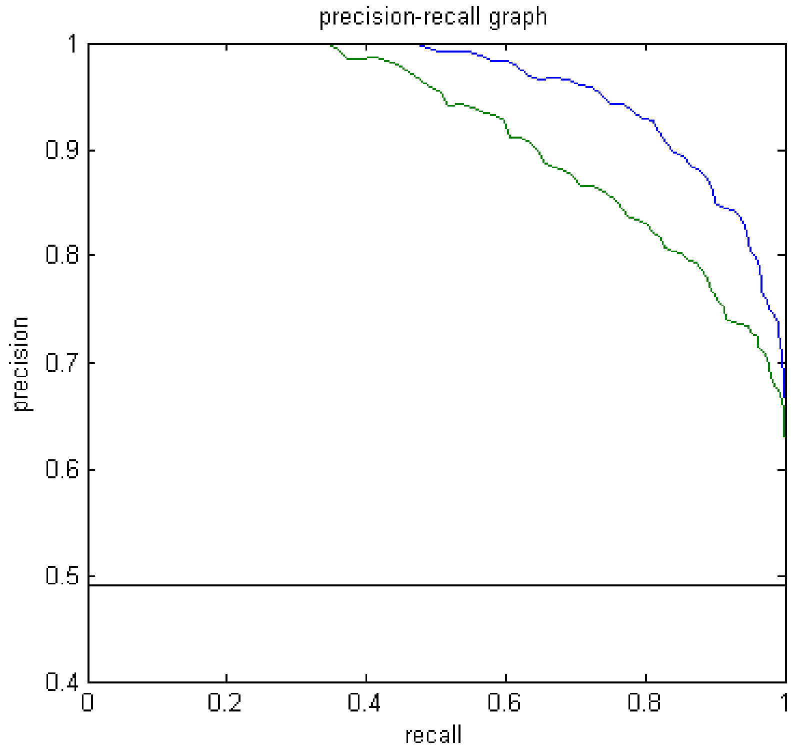


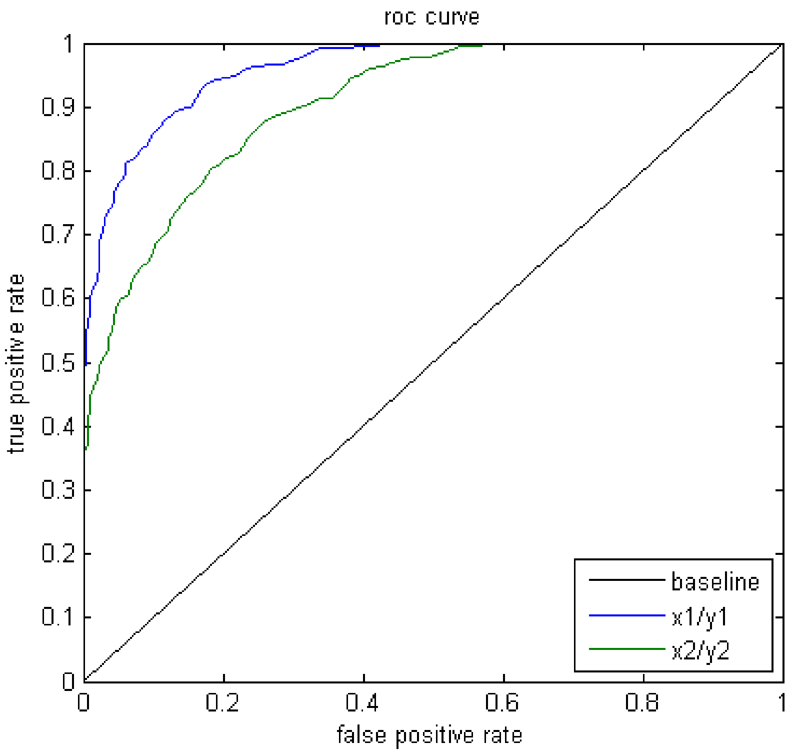
Query

Ranked Matches

ROC Curves

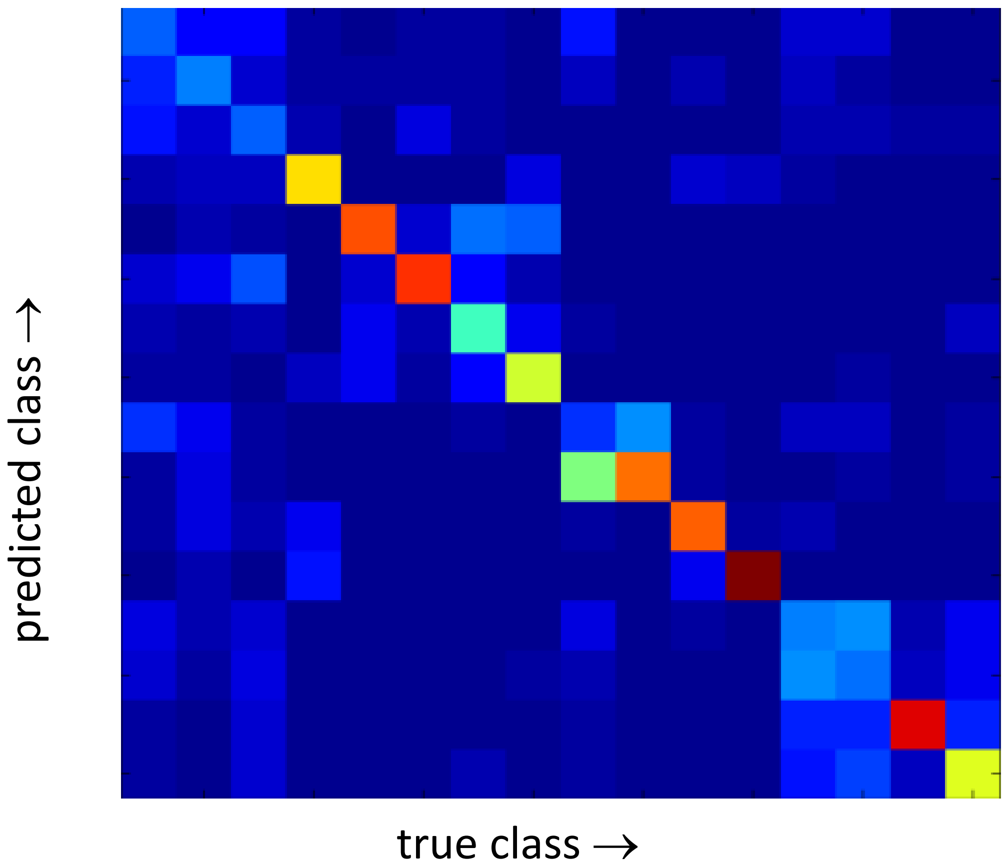
- True positive vs. false positive





Reporting Classification Error: Confusion Matrices

Entry (i,j) stores probability of image that is truly in class i being predicted as class j



Example: Video Google

Visually defined query

“Groundhog Day” [Rammis, 1993]

“Find this clock”

“Find this place”

Slide from Andrew Zisserman
Sivic & Zisserman, ICCV 2003

Video Google

1. Collect all words within query region
2. Inverted file index to find relevant frames
3. Compare word counts
4. Spatial verification

Query region

Retrieved frames

Slide from Andrew Zisserman
Sivic & Zisserman, ICCV 2003

Video Google

retrieved shots

Slide from Andrew Zisserman
Sivic & Zisserman, ICCV 2003

Summary

- Image classification
  - Predict annotations for image
- Bag of words image representation
  - Commonly used for image classification

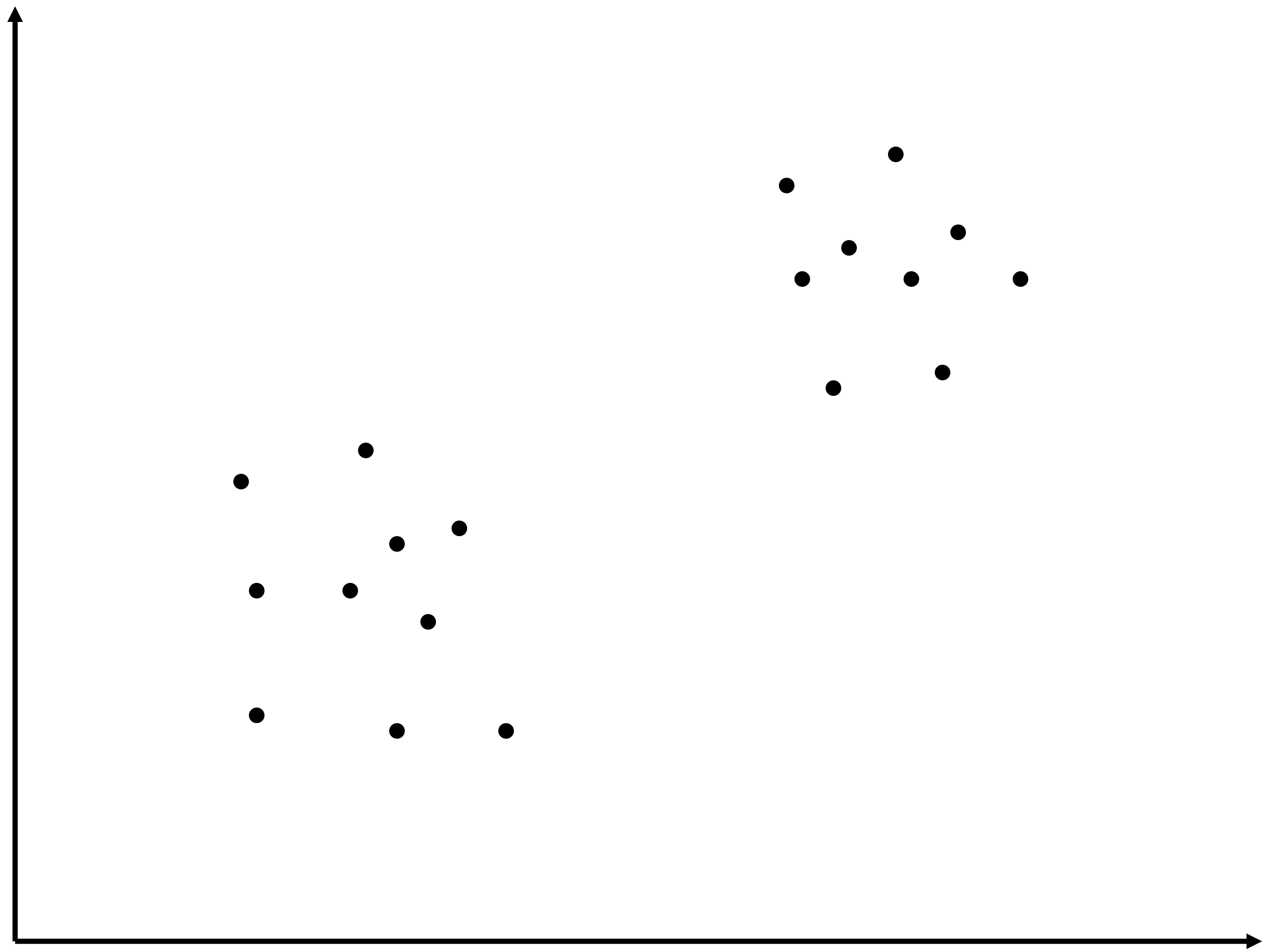
Bags of words: pros and cons

+ flexible to geometry / deformations / viewpoint
+ compact summary of image content
+ provides vector representation for sets
+ very good results in practice

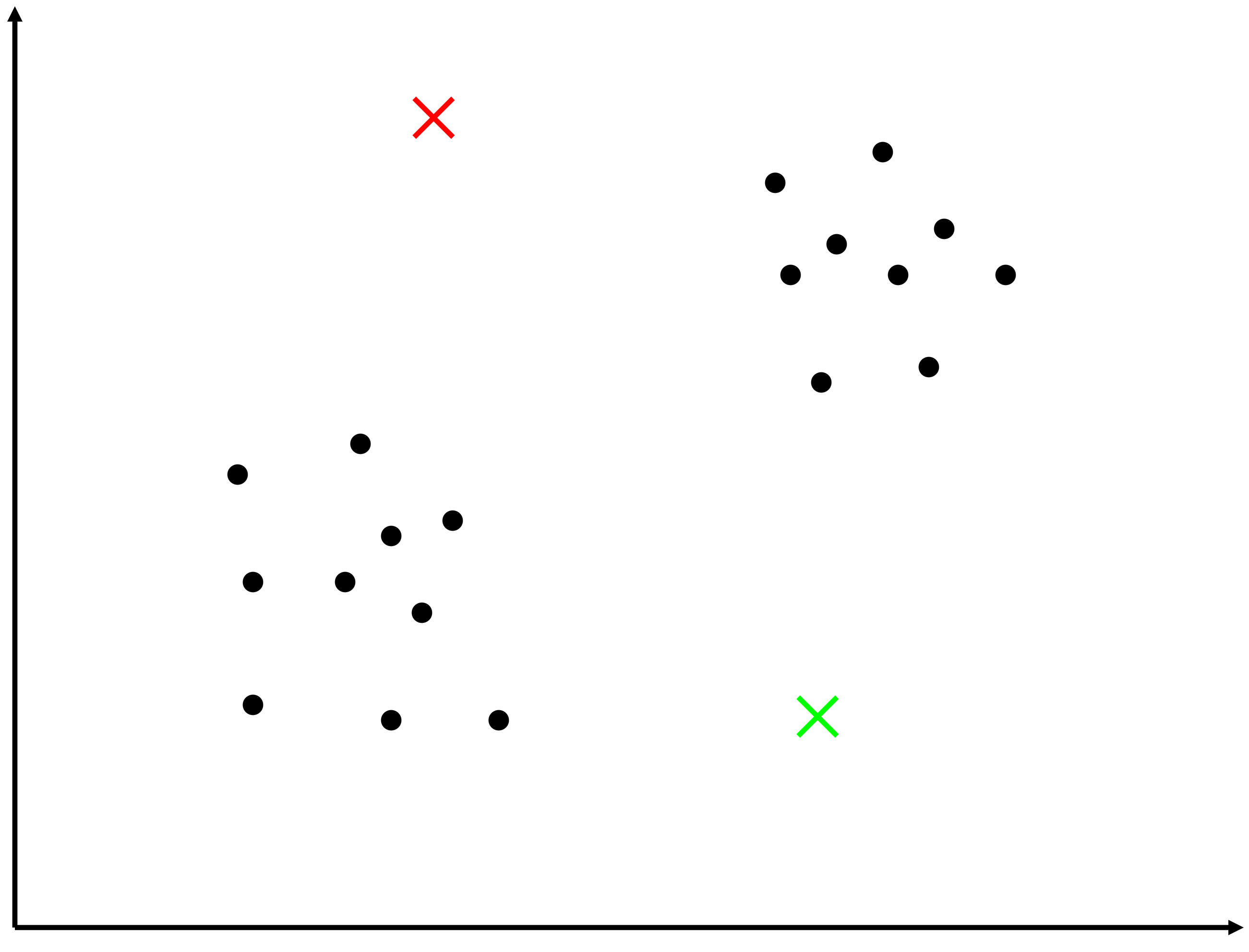
- basic model ignores geometry – must verify afterwards, or encode via features
- background and foreground mixed when bag covers whole image
- optimal vocabulary formation remains unclear

Kristen Grauman

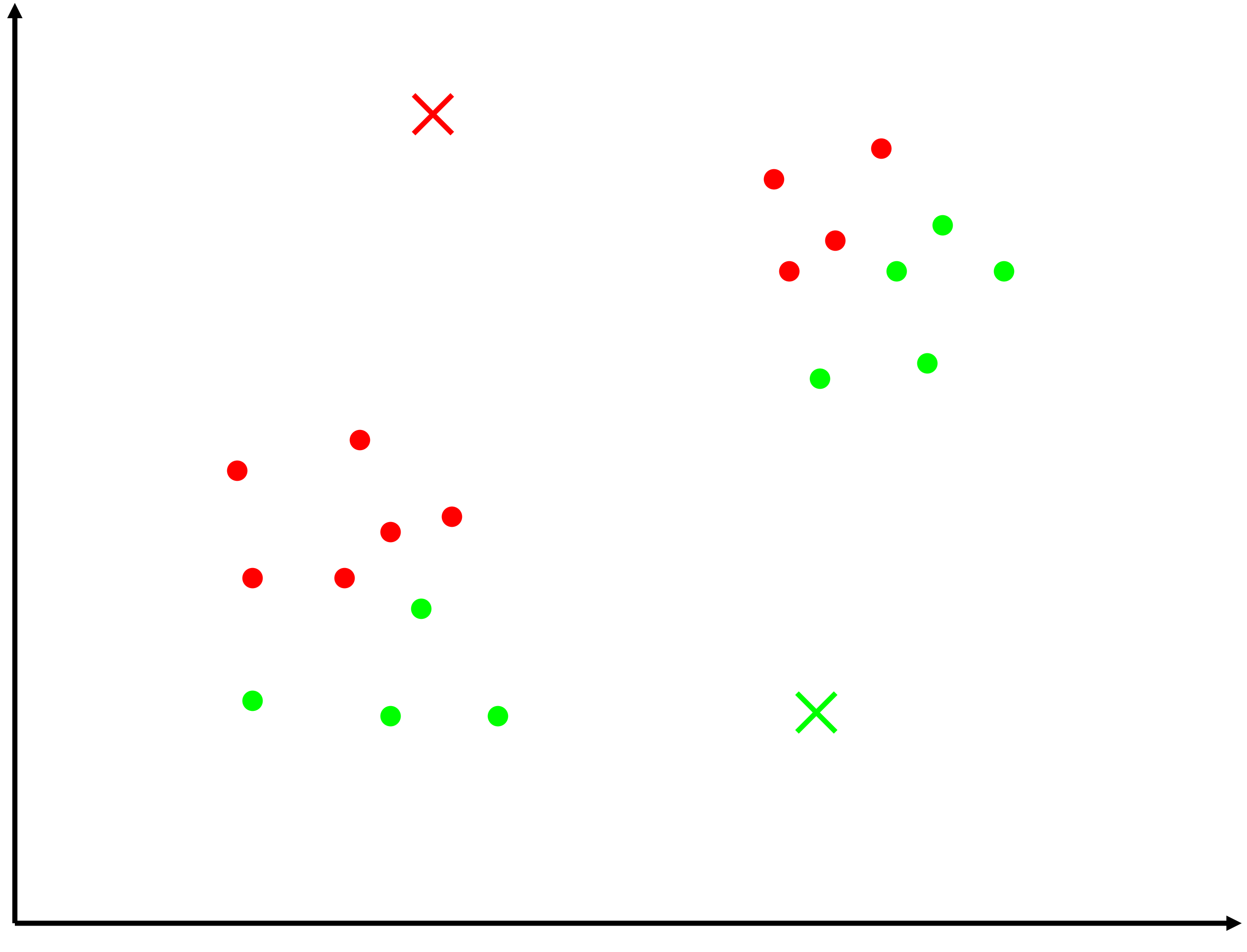
k-Means Clustering



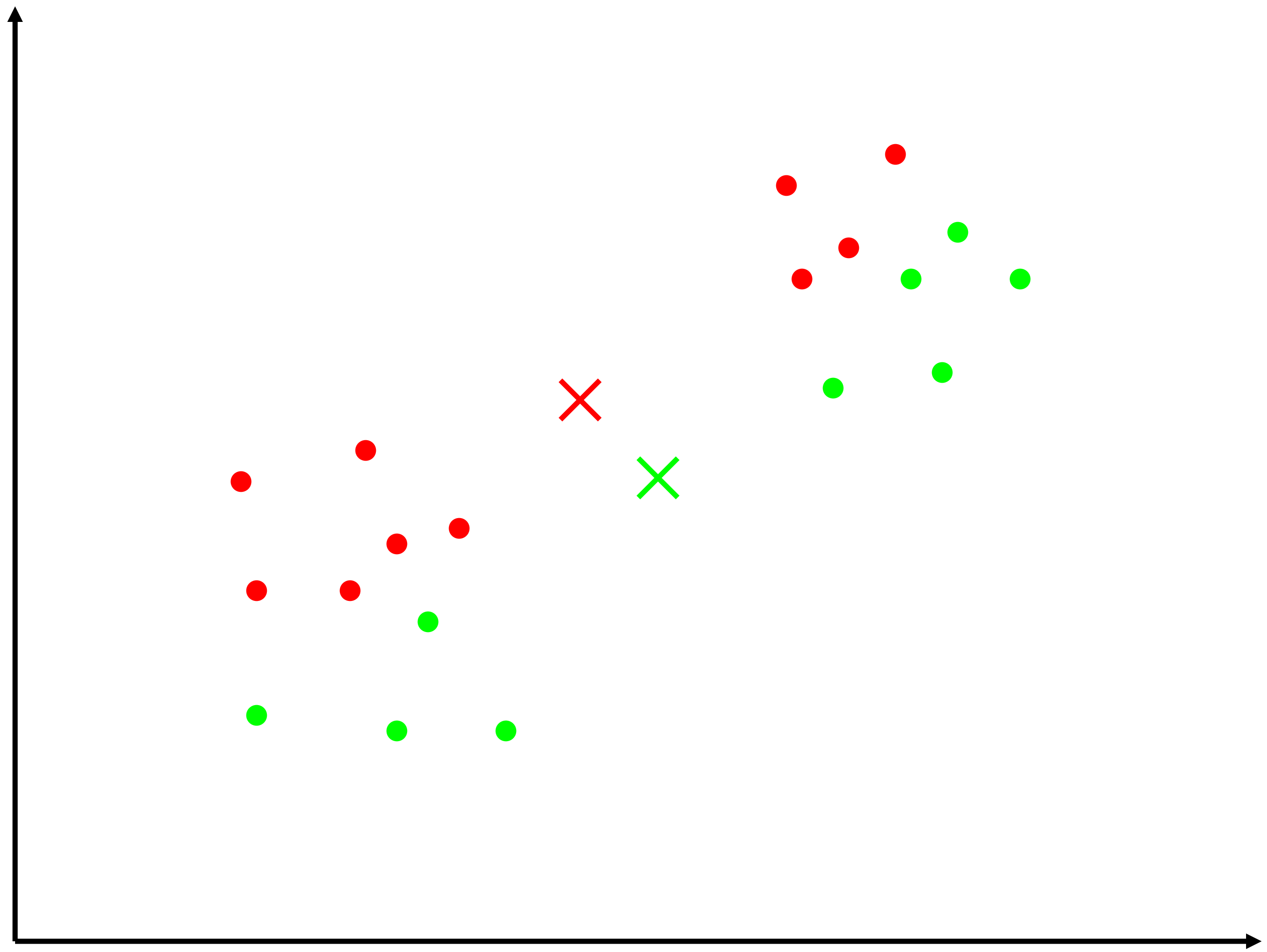
k-Means Clustering



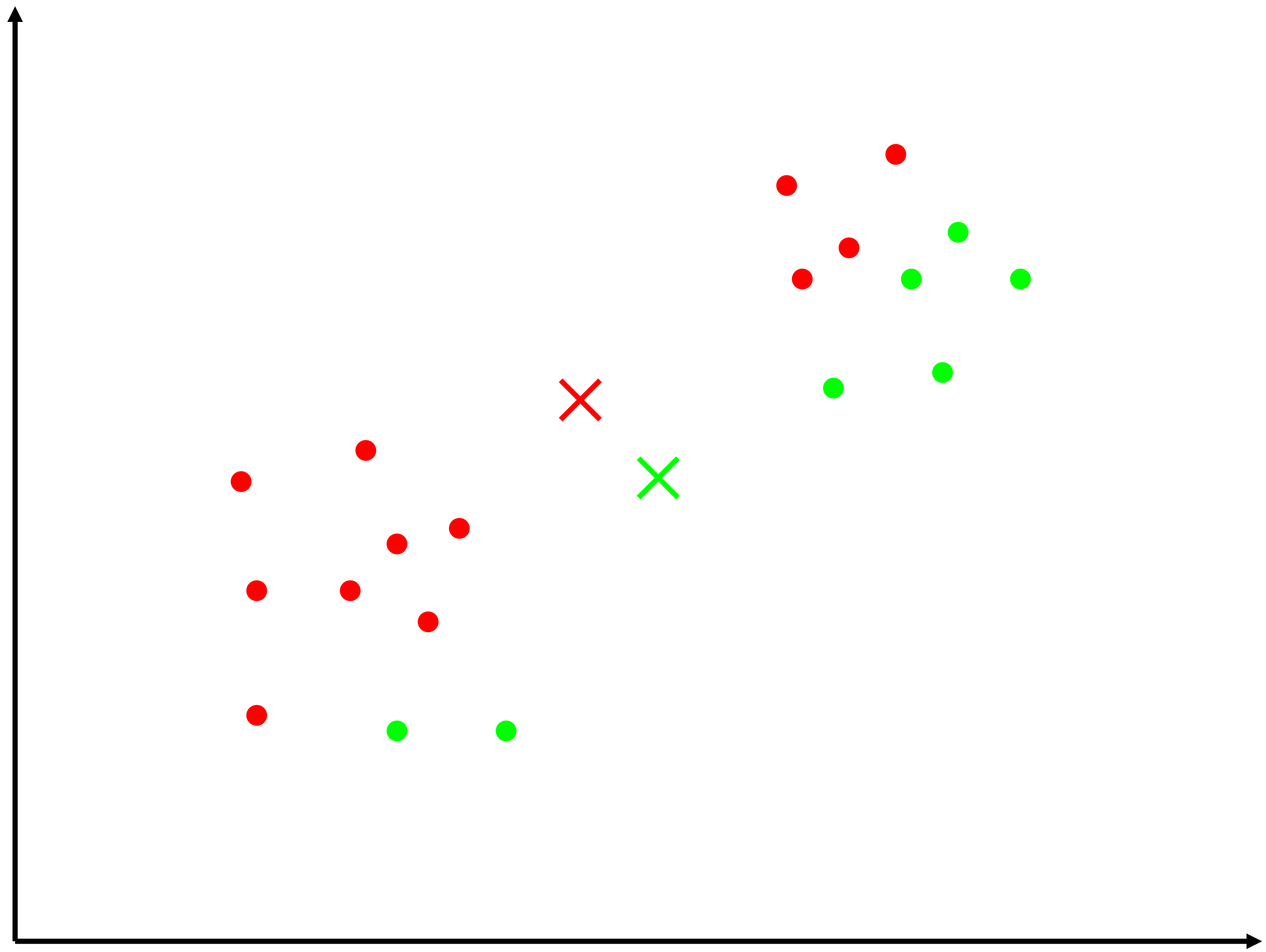
k-Means Clustering



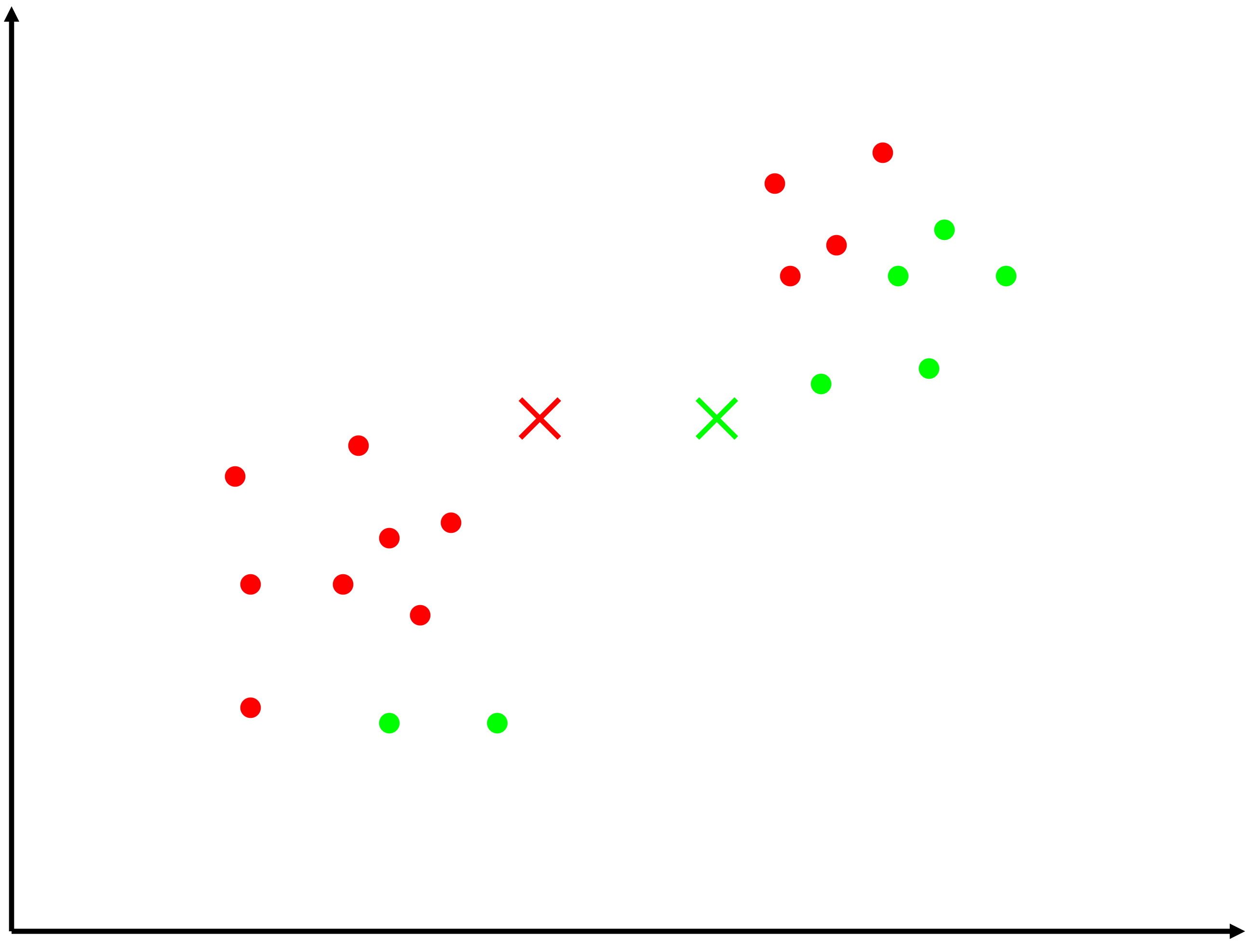
k-Means Clustering



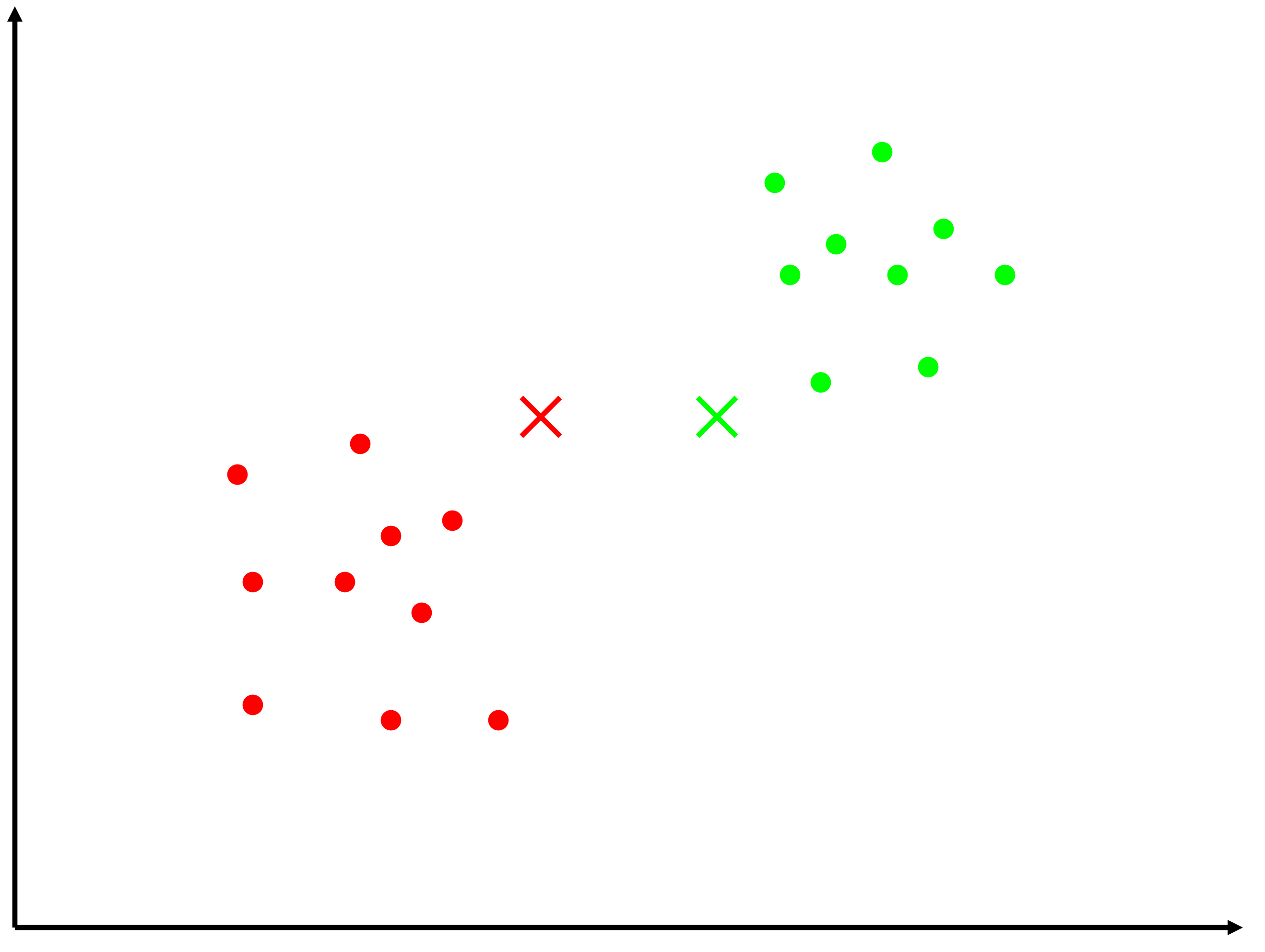
k-Means Clustering



k-Means Clustering



k-Means Clustering



k-Means Clustering