

# Content-based image retrieval

Lecturer: Viktor Sakharchuk

UCU-2019

Slides courtesy:  
Kevin McGuinness  
Bastian Leibe  
Kristen Grauman

# Overview

- What is content based image retrieval?
- The classic SIFT retrieval pipeline
- Using off the shelf CNN features for retrieval
- Learning representations for retrieval

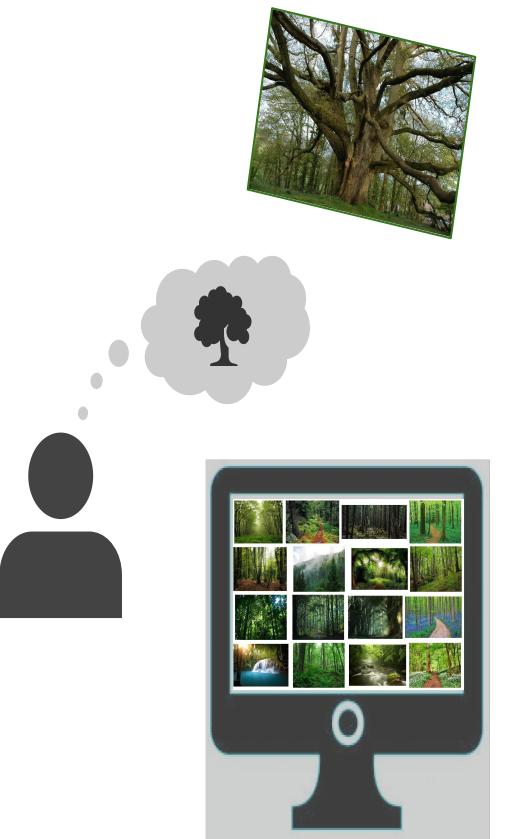
# The problem: query by example

Given:

- An example query image that illustrates the user's information need
- A very large dataset of images

Task:

Rank all images in the dataset according to how likely they are to fulfil the user's information need



# Challenges

**Similarity:** How to compare two images for similarity? What does it mean for two images to be similar?

**Speed:** How to guarantee sub-second query response times?

**Scalability:** How to handle millions of images and multiple simultaneous users?

How to ensure image representations are sufficiently compact?

# Challenges: comparing images



	120	125	131	135	136	134	130	128	125
	123	129	128	137	140	137	131	129	130
119	123	131	134	137	135	130	128	125	141
122	119	124	134	139	137	132	131	131	142
135	133	132	130	133	136	138	135	138	147
120	128	137	140	137	131	129	130	131	147
123	123	129	132	141	141	142	142	142	143
125	128	134	135	141	148	155	153	159	139
127	128	136	145	145	147	145	147	147	138
127	129	135	140	142	142	143	142	148	135
132	135	138	138	139	138	134	137	142	136
133	135	136	135	134	136	131	130	131	136
135	135	135	134	135	132	132	132	130	135
137	134	137	135	137	129	130	128	133	129
138	134	135	134	132	127	130	131	132	129

**Image representation:** 150528 dimensional vector consisting in the concatenation of the flatten color channels. Final vector is l2 normalized.

```
120 128 137 ... 130 131 132 | 119 123 131 ... 136 138 135 | 120 125 131 ... 129 130 128
```

Distance: 0.0



Distance: 0.654



Distance: 0.661



# Challenges: comparing images

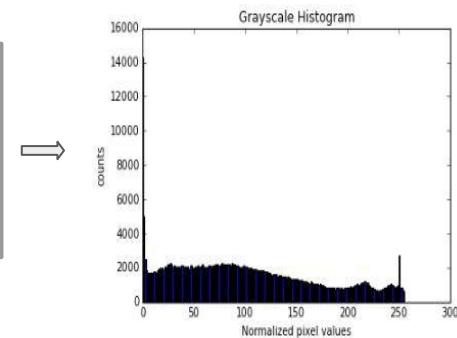


120 125 131 135 136 134 130 128 125
123 126 128 137 140 137 131 129 130
120 125 131 134 137 130 128 125
122 119 124 134 139 137 130 131 131
120 128 137 140 137 131 129 130 131
125 132 133 138 133 137 130 135 138
120 128 137 140 137 131 129 130 131
122 123 129 132 141 141 142 142 142
123 123 129 132 141 141 142 142 142
125 128 134 135 141 148 155 153 153
127 128 136 145 145 145 145 145 145
127 129 135 140 142 142 143 142 148
132 135 138 138 138 139 138 134 137
133 135 136 135 134 136 131 130 131
135 135 135 134 135 135 132 132 130
137 134 137 135 137 129 130 128 133
138 134 135 134 132 127 130 131 132



119 123 131 134 137 130 128 125
120 125 131 134 136 134 130 128
123 126 128 137 140 137 131 129
120 128 137 140 137 131 129 130
125 132 133 138 133 137 130 135
120 128 137 140 137 131 129 130
122 123 129 132 141 141 142 142
123 123 129 132 141 141 142 142
125 128 134 135 141 148 155 153
127 128 136 145 145 145 145 145
127 129 135 140 142 142 143 142
132 135 138 138 138 139 138 134 137
133 135 136 135 134 136 131 130 131
135 135 135 134 135 135 132 132 130
137 134 137 135 137 129 130 128 133
138 134 135 134 132 127 130 131 132

**Image representation:** 256 dimensional vector based on the histogram of the grayscale pixels. Vector is l2 normalized.



Distance: 0.0



Distance: 0.718



Distance: 0.080



# Classification

Query: This chair



Results from dataset classified as “chair”

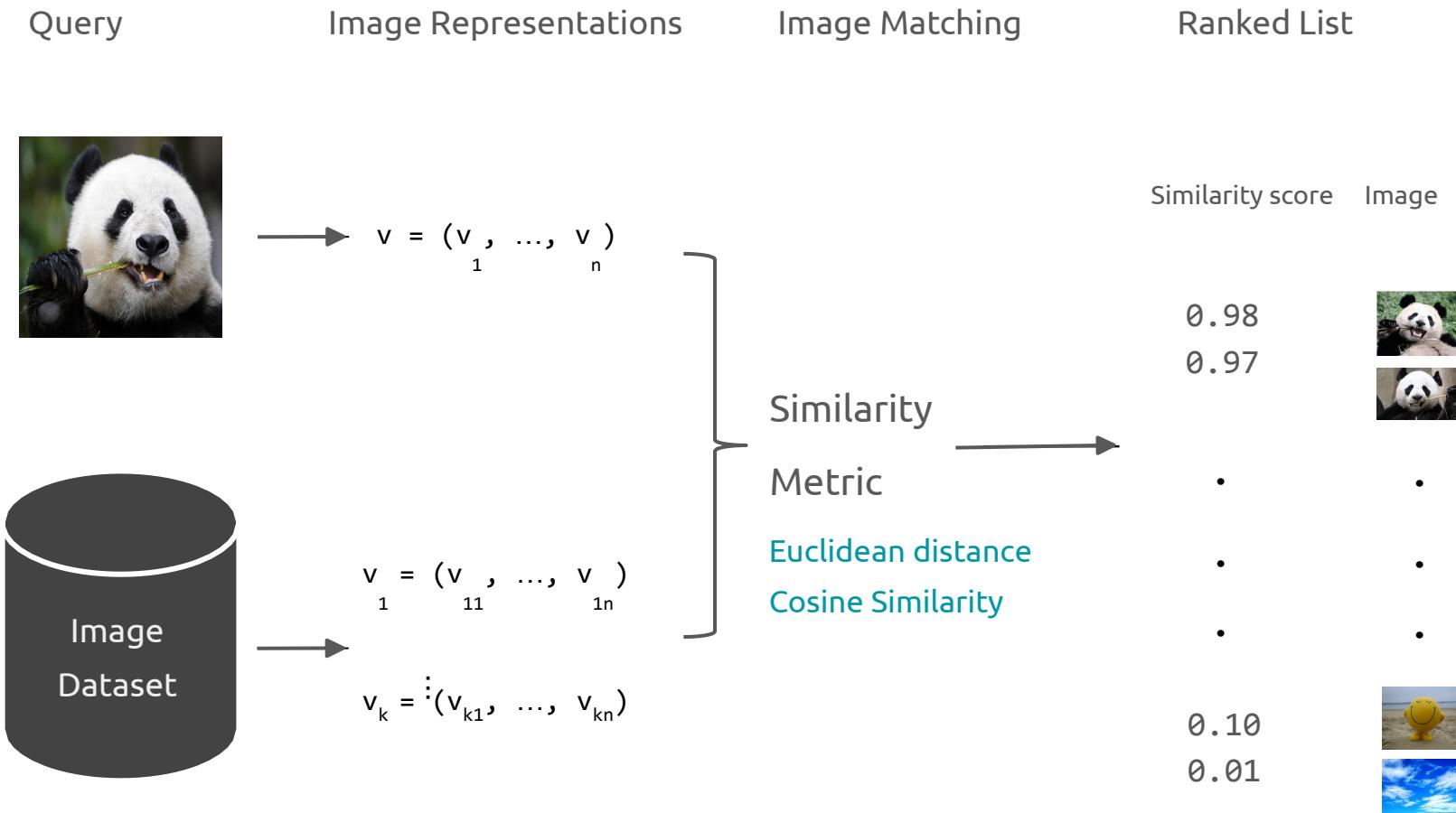
# Retrieval

Query: This chair



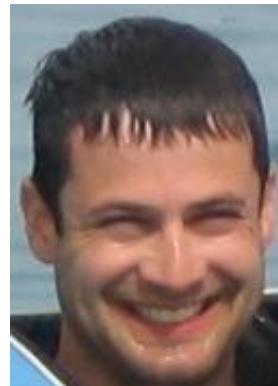
Results from dataset ranked by similarity to the query

# The retrieval pipeline



# Global representations: limitations

- Success may rely on alignment  
-> sensitive to viewpoint
- All parts of the image or window impact the description  
-> sensitive to occlusion, clutter

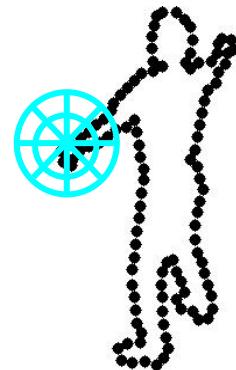


# Local representations

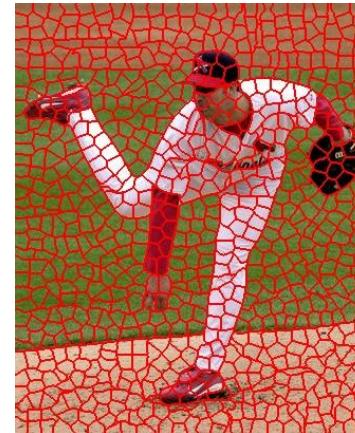
- Describe component regions or patches separately.
- Many options for detection & description...



SIFT [Lowe 99]



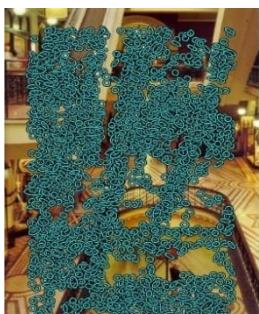
Shape context  
[Belongie 02]



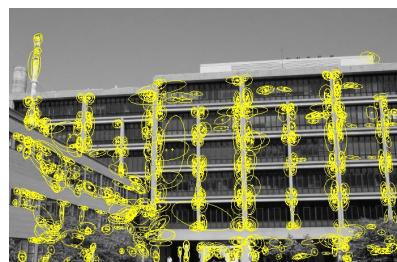
Superpixels  
[Ren et al.]



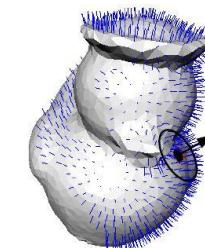
Maximally Stable  
Extremal Regions  
[Matas 02]



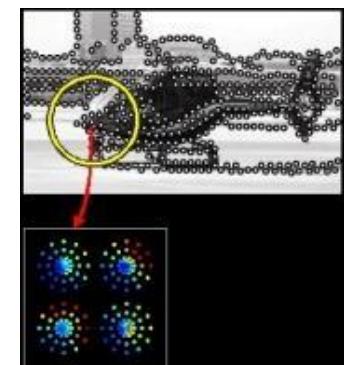
Salient regions  
[Kadir 01]



Harris-Affine  
[Mikolajczyk 04]



Spin images  
[Johnson 99]



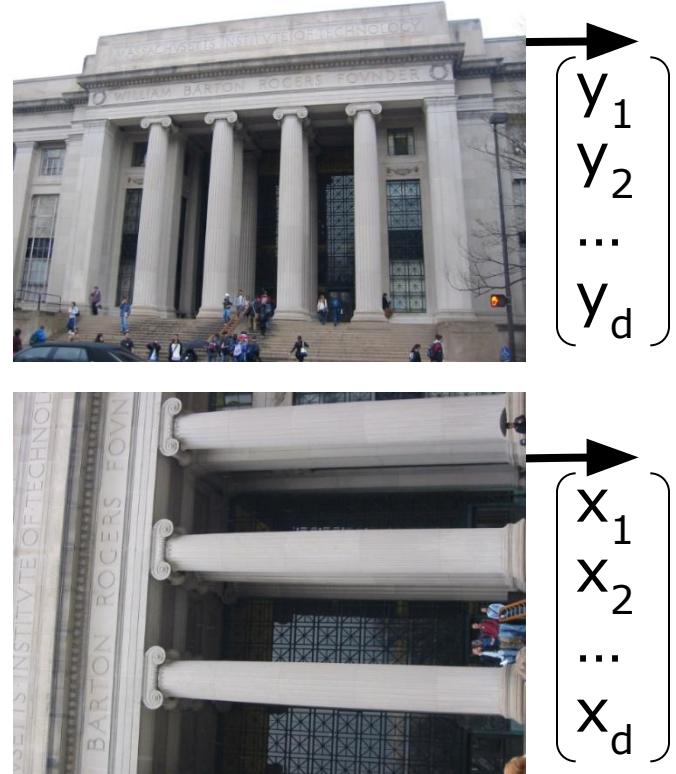
Geometric Blur  
[Berg 05]

# Recall: Invariant local features

Subset of local feature types designed to be invariant to

- Scale
- Translation
- Rotation
- Affine transformations
- Illumination

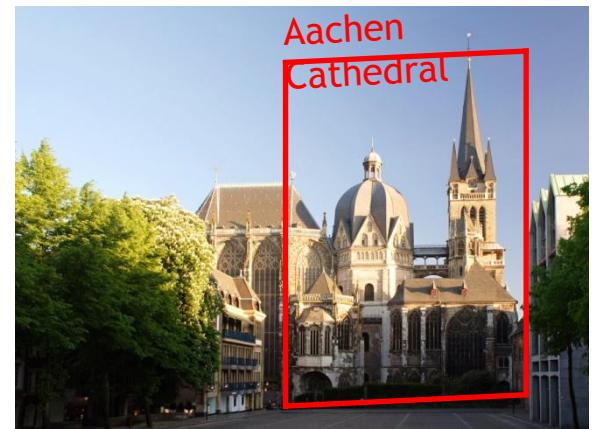
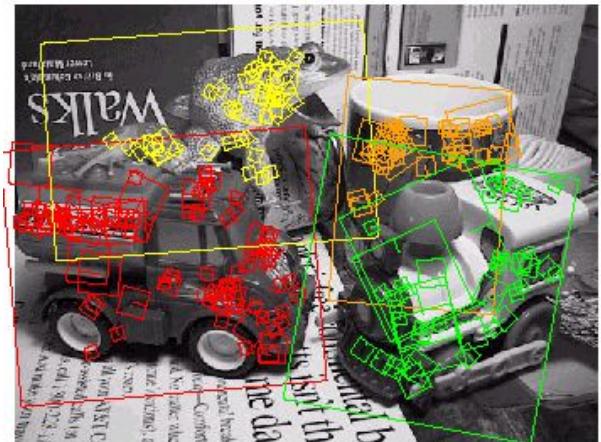
Detect interest points  
Extract descriptors



[Mikolajczyk01, Matas02, Tuytelaars04, Lowe99, Kadir01, ... ]

# Recognition with local feature sets

- Previously, we saw how to use local invariant features + a global spatial model to recognize specific objects, using a planar object assumption.
- Now, we'll use local features for
  - Indexing-based recognition
  - Bags of words representations
  - Correspondence / matching kernels

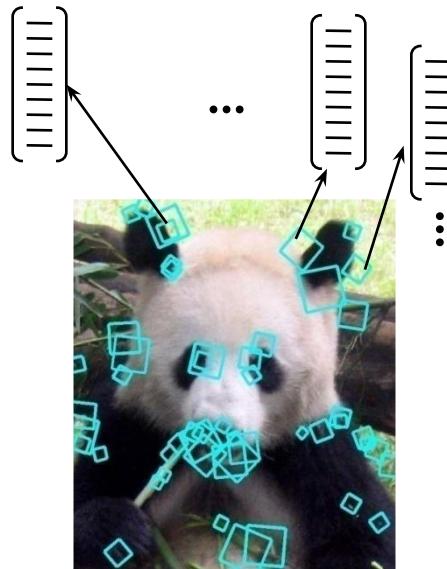


# Basic flow



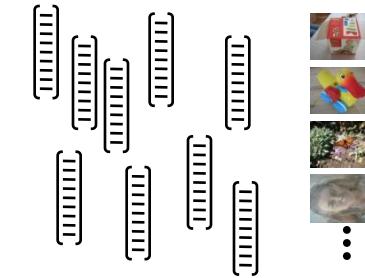
**Detect or sample  
features**

List of positions,  
scales,  
orientations



**Describe  
features**

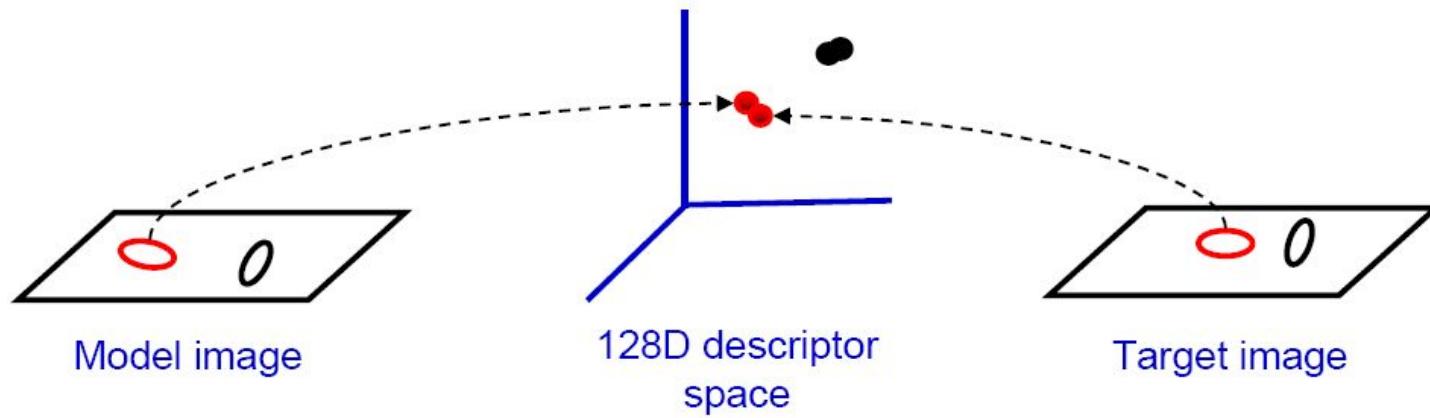
Associated list of  
d-dimensional  
descriptors



**Index each one into pool  
of descriptors from  
previously seen images**

# Indexing local features

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



# Indexing local features

- We saw in the previous section how to use voting and pose clustering to identify objects using local features

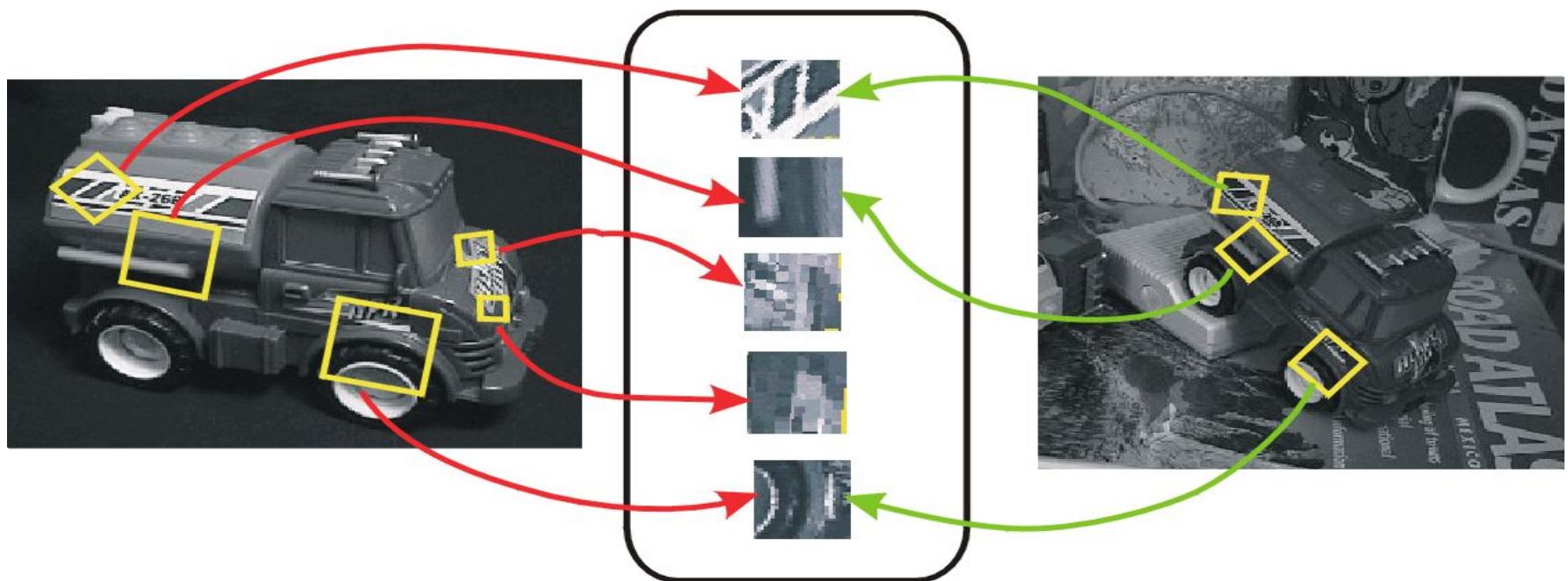
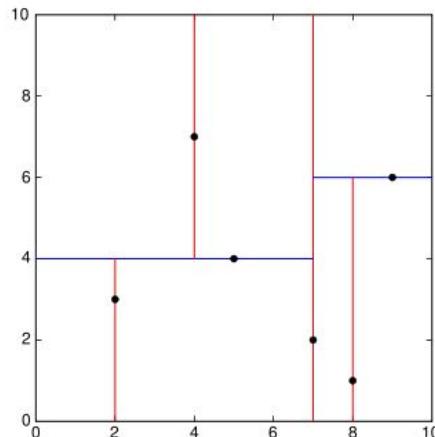


Figure credit: David Lowe

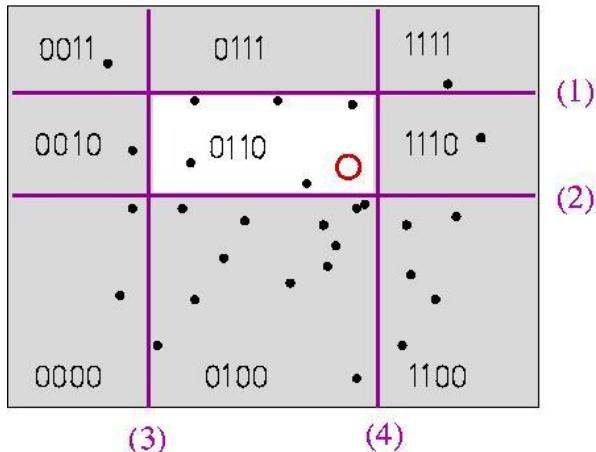
# **Indexing local features**

- 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?
  - Low-dimensional descriptors : can use standard efficient data structures for nearest neighbor search
  - High-dimensional descriptors: approximate nearest neighbor search methods more practical
  - Inverted file indexing schemes

# Indexing local features: approximate nearest neighbor search



**Best-Bin First (BBF)**, a variant of k-d trees that uses priority queue to examine most promising branches first [Beis & Lowe, CVPR 1997]



**Locality-Sensitive Hashing (LSH)**, a randomized hashing technique using hash functions that map similar points to the same bin, with high probability [Indyk & Motwani, 1998]

# Indexing local features: inverted file index

Index	
"Along I-75," From Detroit to Florida; <i>Inside back cover</i>	Butterfly Center, McGuire; 134
"Drive I-95," From Boston to Florida; <i>Inside back cover</i>	CAA (see AAA)
1929 Spanish Trail Roadway; 101-102,104	CCC, The; 111,113,115,135,142
511 Traffic Information; 83	Ca'Zan; 147
A1A (Barrier Isl) - I-95 Access; 86	Caloosahatchee River; 152
AAA (and CAA); 83	Name; 150
AAA National Office; 88	Canaveral Natnl Seashore; 173
Abbreviations,	Cannon Creek Airpark; 130
Colored 25 mile Maps; cover	Canopy Road; 106,160
Exit Services; 196	Cape Canaveral; 174
Travelogue; 85	Castillo San Marcos; 169
Africa; 177	Cave Diving; 131
Agricultural Inspection Stns; 126	Cayo Costa, Name; 150
Ah-Tah-Thi-Ki Museum; 180	Celebration; 93
Air Conditioning, First; 112	Charlotte County; 149
Alabama; 124	Charlotte Harbor; 150
Alachua; 132	Chautauqua; 116
County; 131	Clipley; 114
Alafia River; 143	Name; 115
Alapaha, Name; 126	Choctawatchee, Name; 115
Alfred B Macay Gardens; 106	Circus Museum, Ringling; 147
Alligator Alley; 154-155	Citrus; 88,97,130,136,140,180
Alligator Farm, St Augustine; 169	CityPlace, W Palm Beach; 180
Alligator Hole (definition); 157	City Maps,
Alligator, Buddy; 155	Ft Lauderdale Expwys; 194-195
Alligators; 100,135,138,147,156	Jacksonville; 163
Anastasia Island; 170	Kissimmee Expwys; 192-193
Anhaica; 108-109,146	Miami Expressways; 194-195
Apalachicola River; 112	Orlando Expressways; 192-193
Appleton Mus of Art; 136	Pensacola; 26
Aquifer; 102	Tallahassee; 191
Arabian Nights; 94	Tampa-St. Petersburg; 63
Art Museum, Ringling; 147	St. Augustine; 191
Aruba Beach Cafe; 183	Civil War; 100,108,127,138,141
Aucilla River Project; 106	Clearwater Marine Aquarium; 187
Babcock-Web WMA; 151	Collier County; 154
Bahia Mar Marina; 184	Colonial Spanish Quarters; 168
Baker County; 99	Columbia County; 101,128
Barefoot Mallmen; 182	Coquina Building Material; 165
Barge Canal; 137	Corkscrew Swamp, Name; 154
Bee Line Expy; 80	Cowboys; 95
Belz Outlet Mall; 89	Crab Trap II; 144
Bernard Castro; 136	Cracker, Florida; 88,95,132
Big "I"; 165	Crosstown Expy; 11,35,98,143
Big Cypress; 155,158	Cuban Bread; 184
Big Foot Monster; 105	Dade Battlefield; 140
Billie Swamp Safari; 160	Dade, Maj. Francis; 139-140,161
Blackwater River SP; 117	Danie Beach Hurricane; 184
Blue Angels	Daniel Boone, Florida Walk; 117
	Daytona Beach; 172-173
	De Land; 87
	Driving Lanes; 85
	Duval County; 163
	Eau Gallie; 175
	Edison, Thomas; 152
	Eglin AFB; 116-118
	Eight Reale; 176
	Ellenton; 144-145
	Emanuel Point Wreck; 120
	Emergency Callboxes; 83
	Epiphytes; 142,148,157,159
	Escambia Bay; 119
	Bridge (I-10); 119
	County; 120
	Estero; 153
	Everglade; 90,95,139-140,154-160
	Draining of; 156,181
	Wildlife MA; 160
	Wonder Gardens; 154
	Falling Waters SP; 115
	Fantasy of Flight; 95
	Fayer Dykes SP; 171
	Fires, Forest; 166
	Fires, Prescribed; 148
	Fisherman's Village; 151
	Flagler County; 171
	Flagler, Henry; 97,165,167,171
	Florida Aquarium; 186
	Florida,
	12,000 years ago; 187
	Cavern SP; 114
	Map of all Expressways; 2-3
	Mus of Natural History; 134
	National Cemetery ; 141
	Part of Africa; 177
	Platform; 187
	Sheriff's Boys Camp; 126
	Sports Hall of Fame; 130
	Sun 'n Fun Museum; 97
	Supreme Court; 107
	Florida's Turnpike (FTP); 178,189
	25 mile Strip Maps; 66
	Administration; 189
	Coin System; 190
	Exit Services; 189
	HEFT; 76,161,190
	History; 189
	Names; 189
	Service Plazas; 190
	Spur SR91; 76
	Ticket System; 190
	Toll Plazas; 190
	Ford, Henry; 152

- 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”.

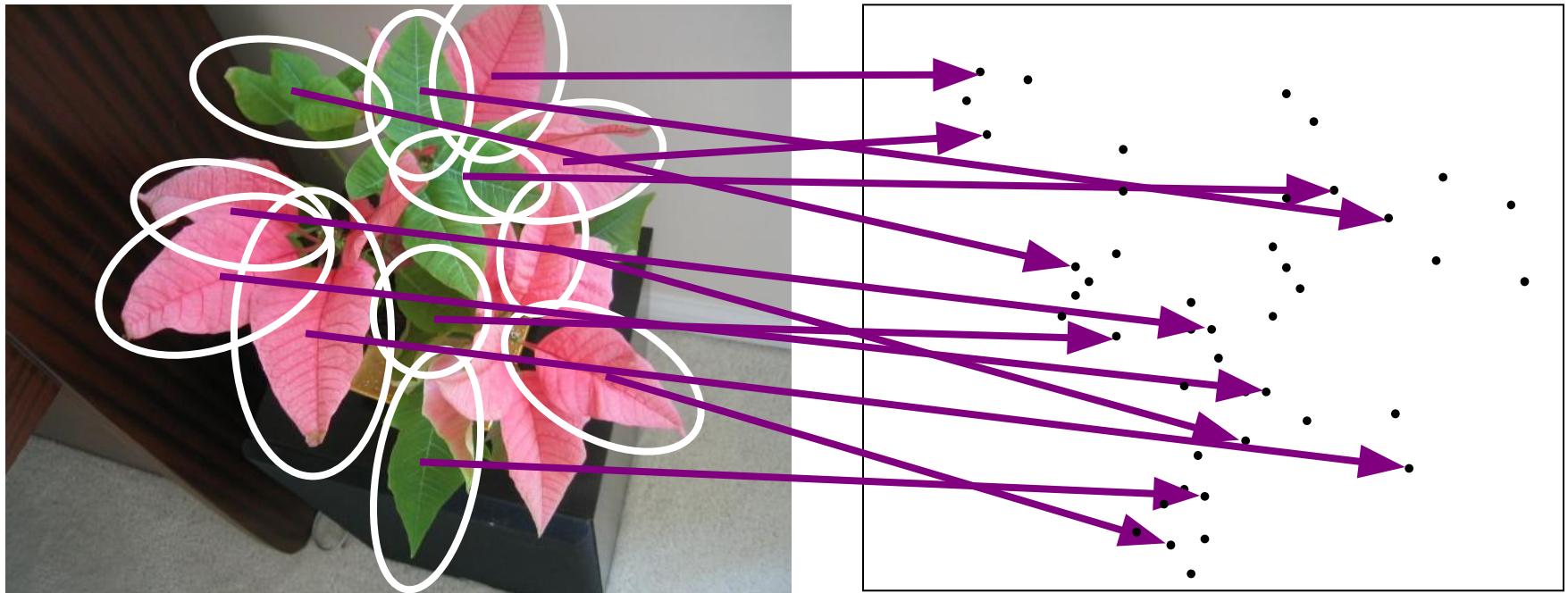
# Visual words: main idea

- Extract some local features from a number of images ...

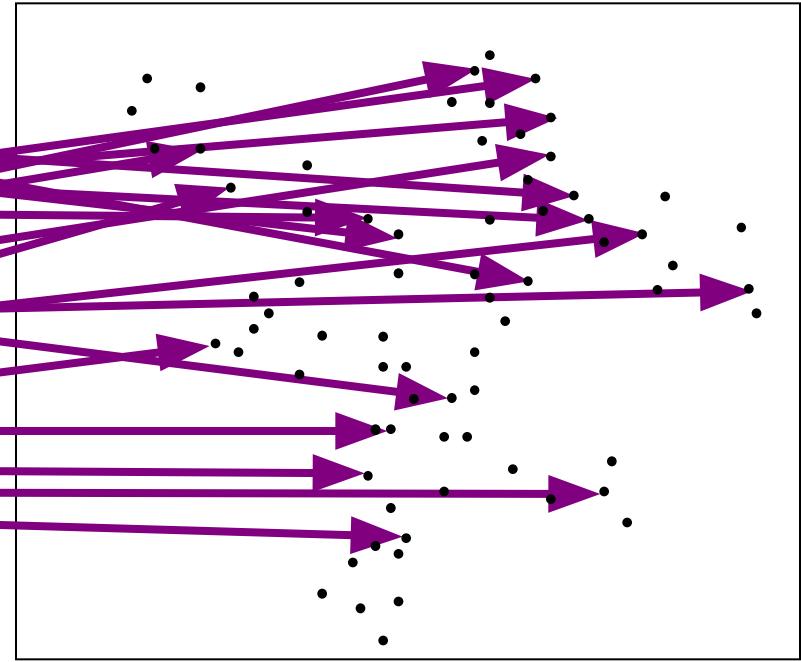
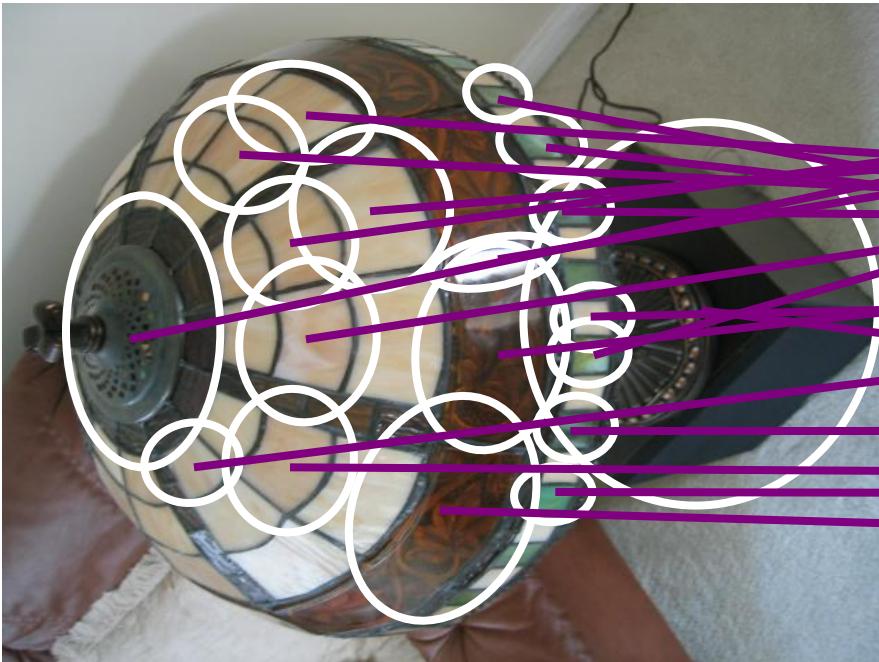


e.g., SIFT descriptor space: each point is 128-dimensional

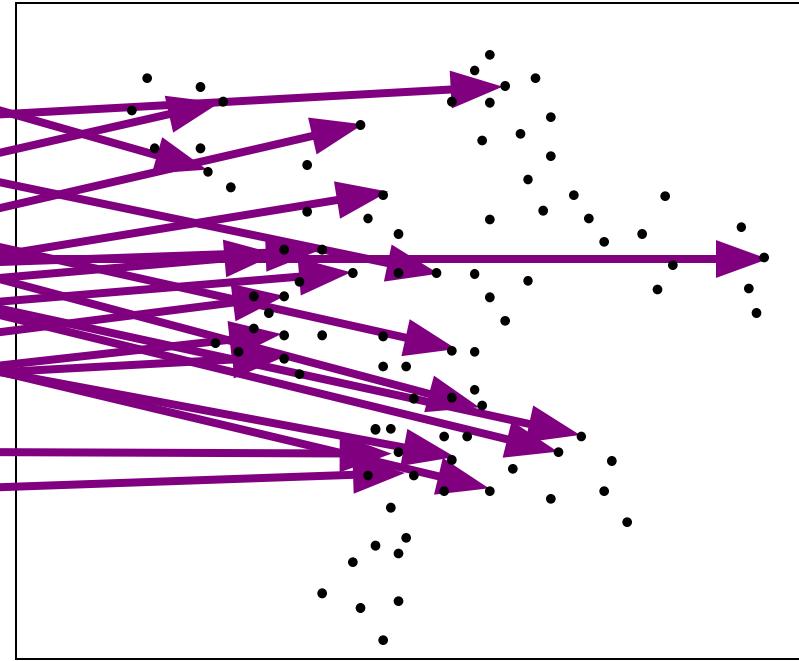
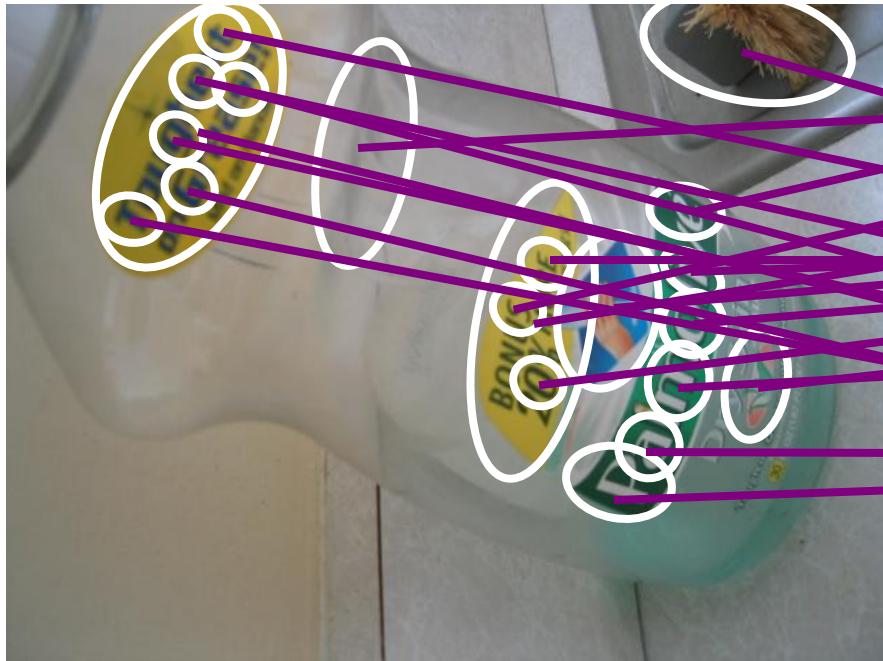
# Visual words: main idea

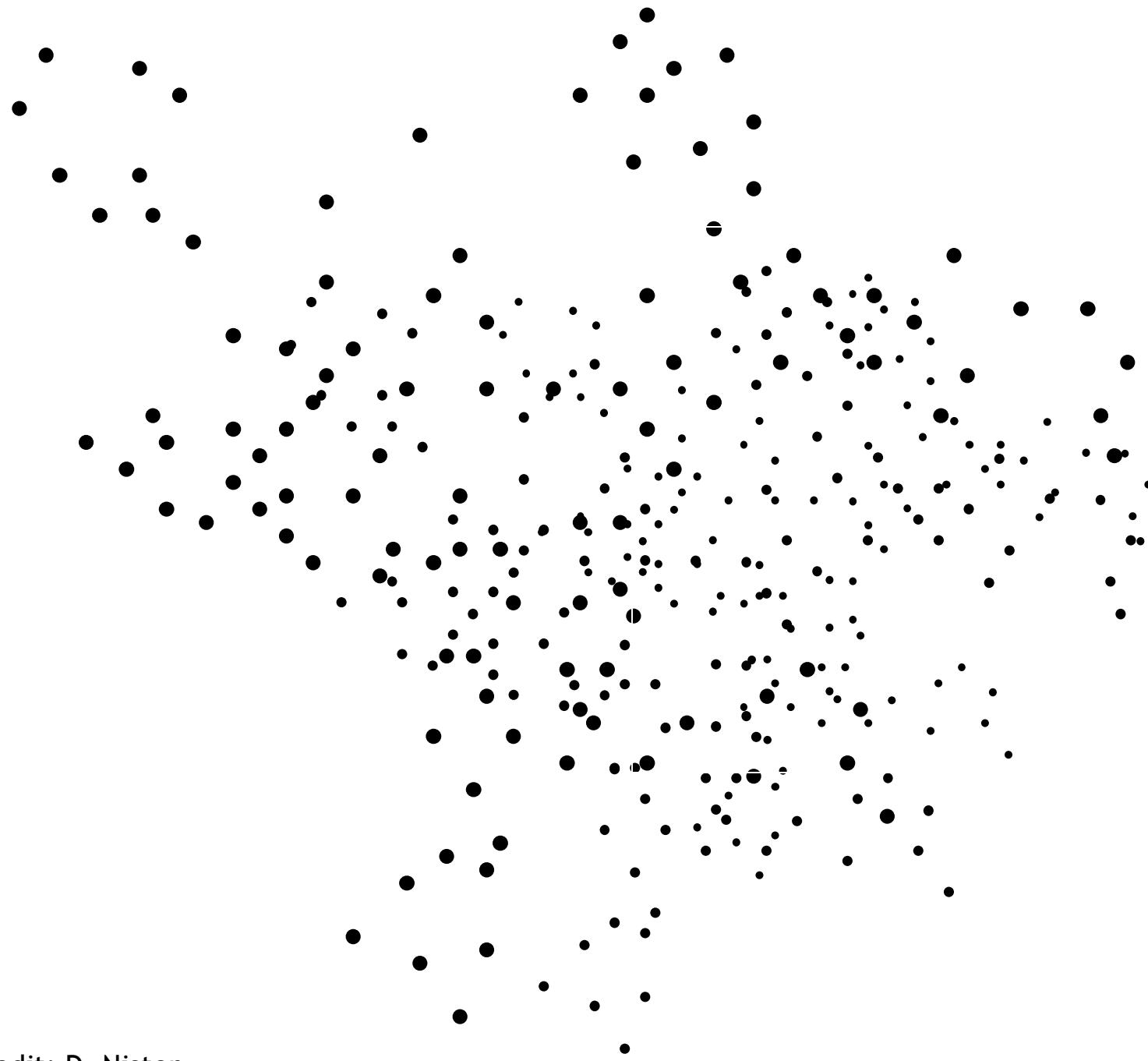


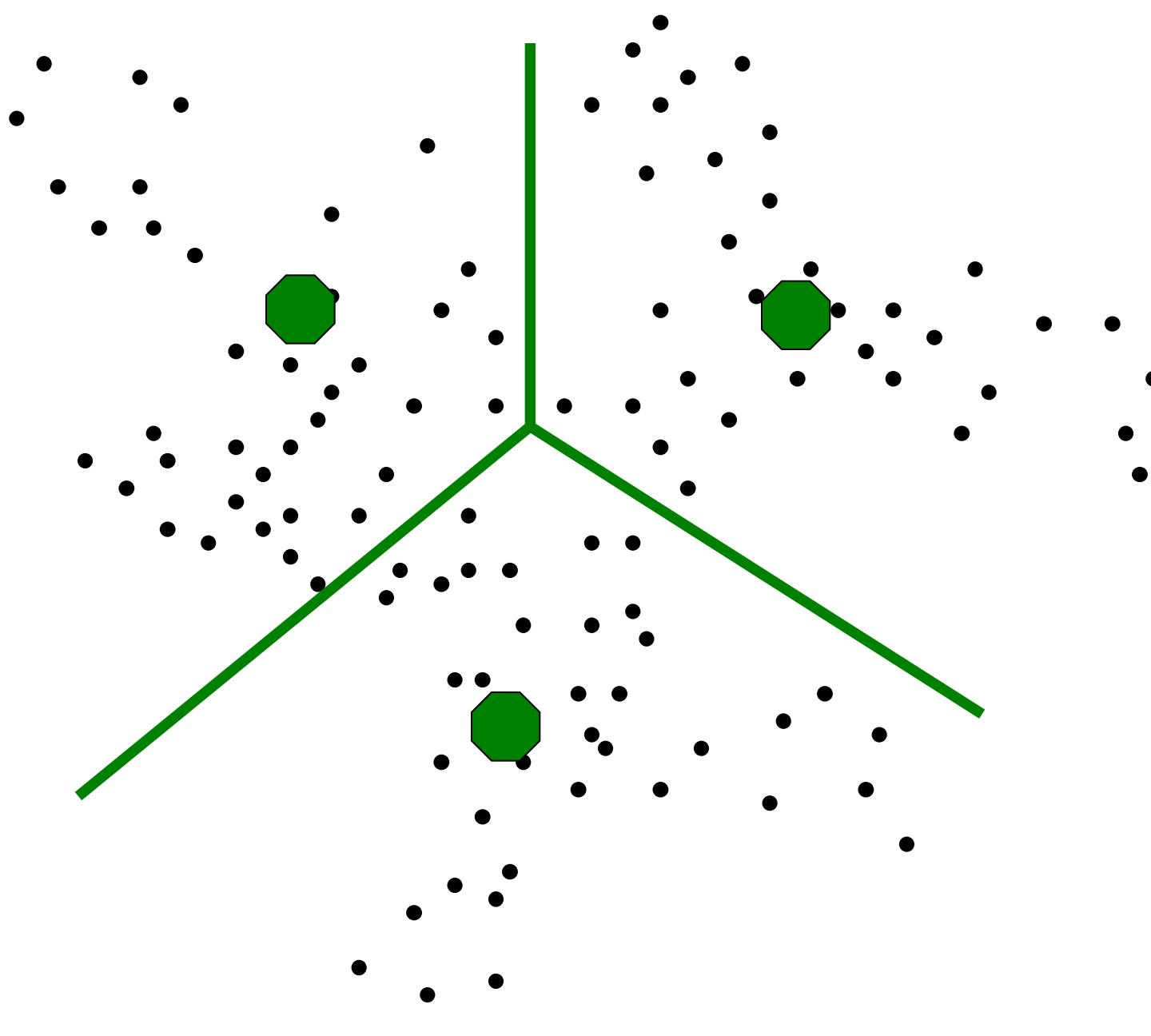
# Visual words: main idea



# Visual words: main idea



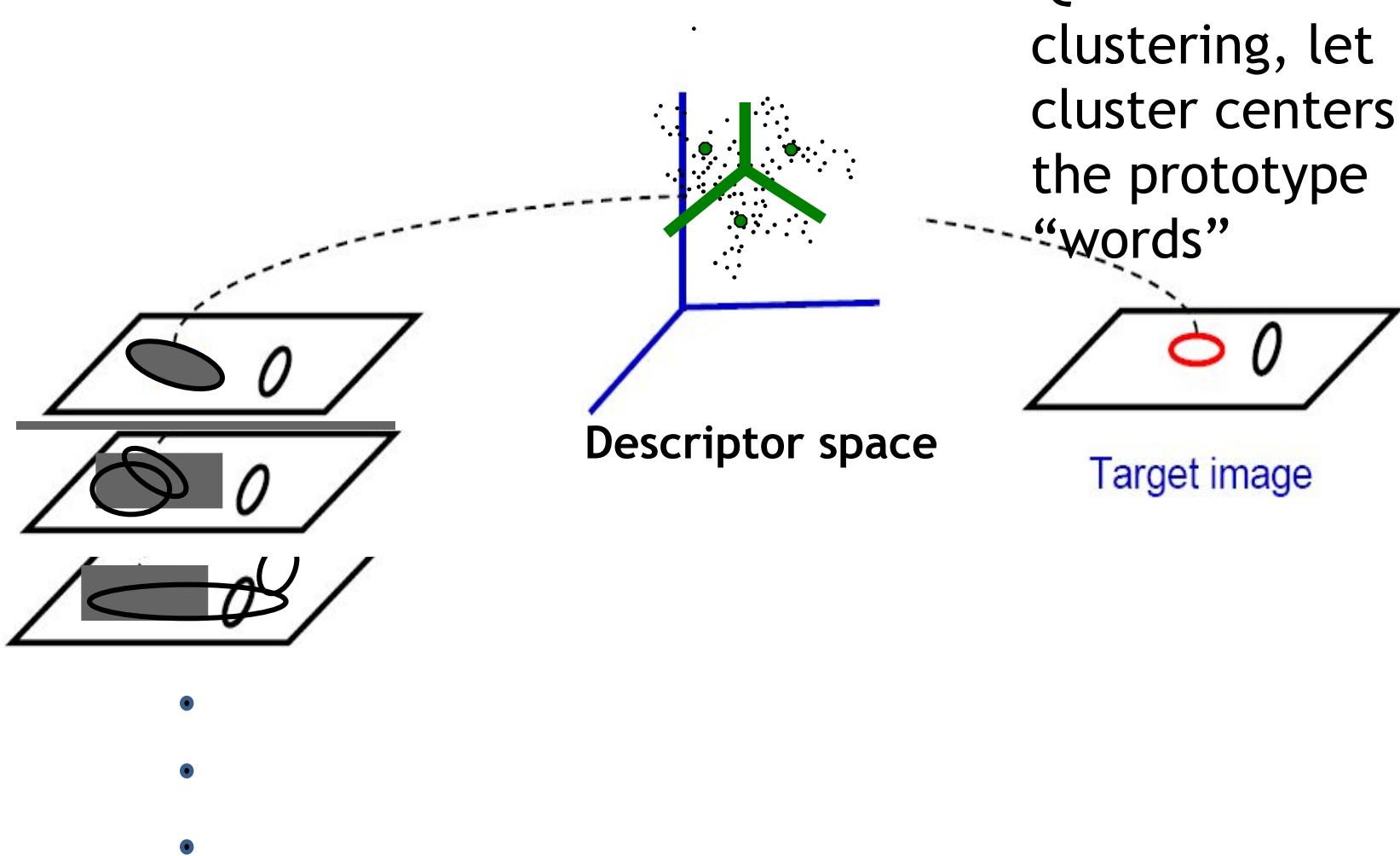




# Visual words: main idea

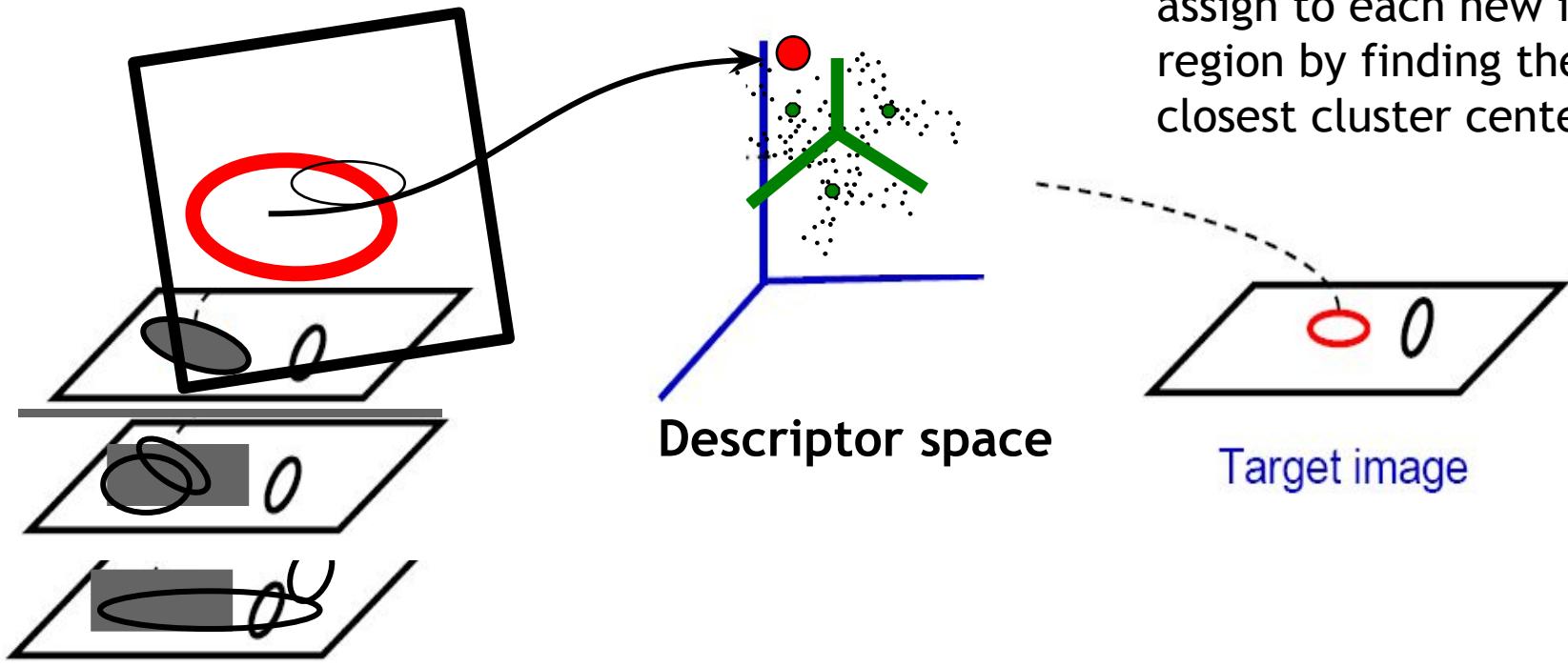
Map high-dimensional descriptors to tokens/words by quantizing the feature space

- Quantize via clustering, let cluster centers be the prototype “words”



# Visual words: main idea

Map high-dimensional descriptors to tokens/words by quantizing the feature space



- Determine which word to assign to each new image region by finding the closest cluster center.

# Visual words

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

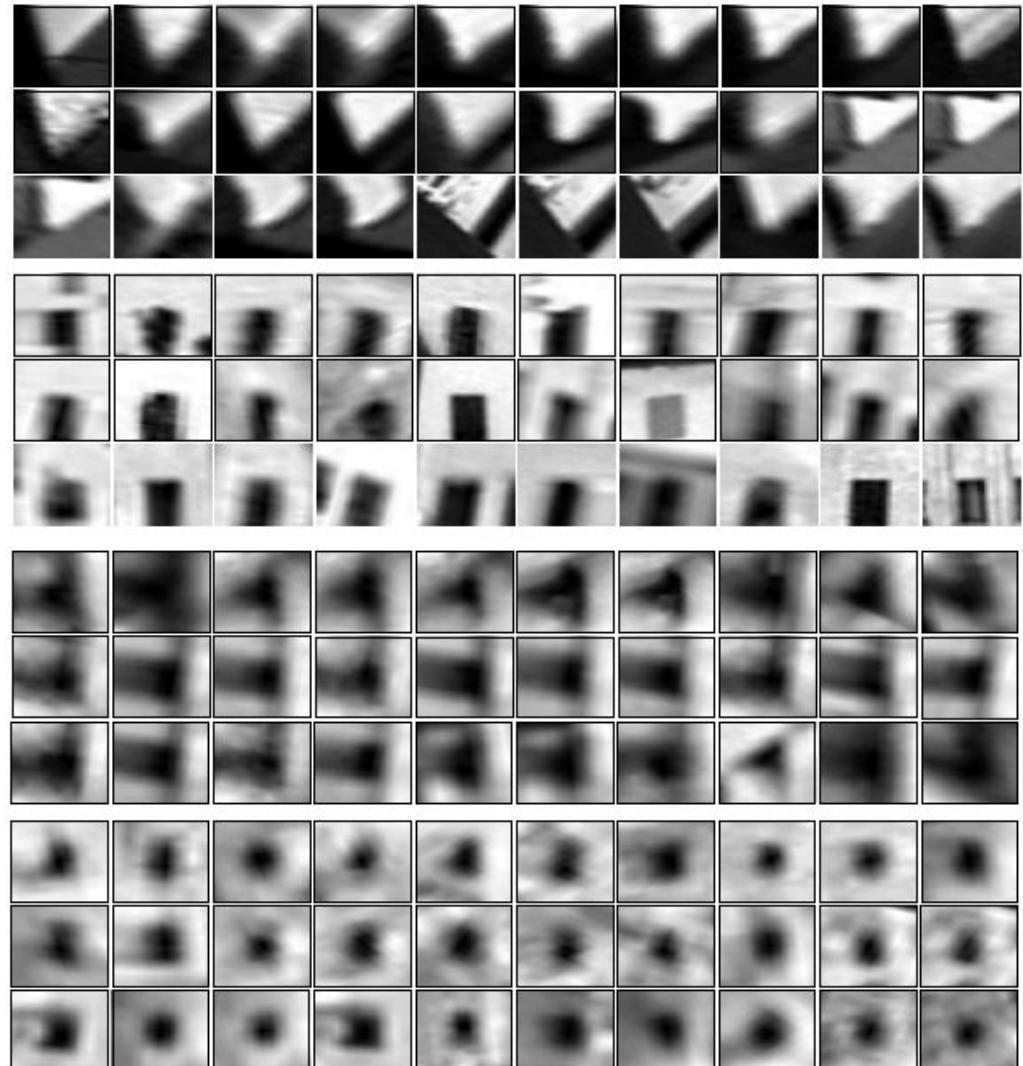
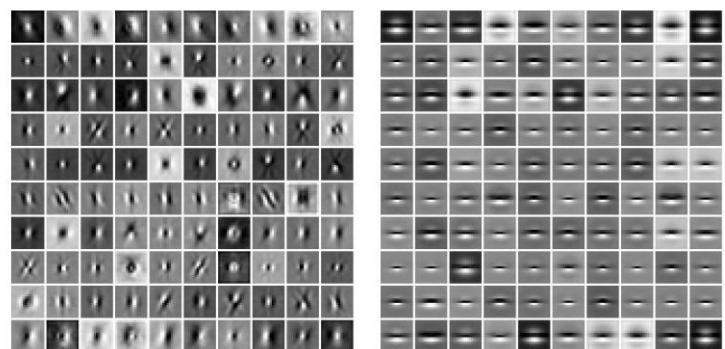
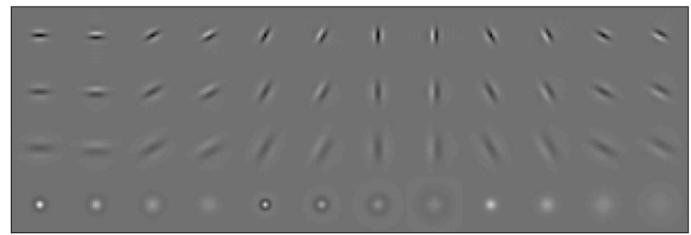
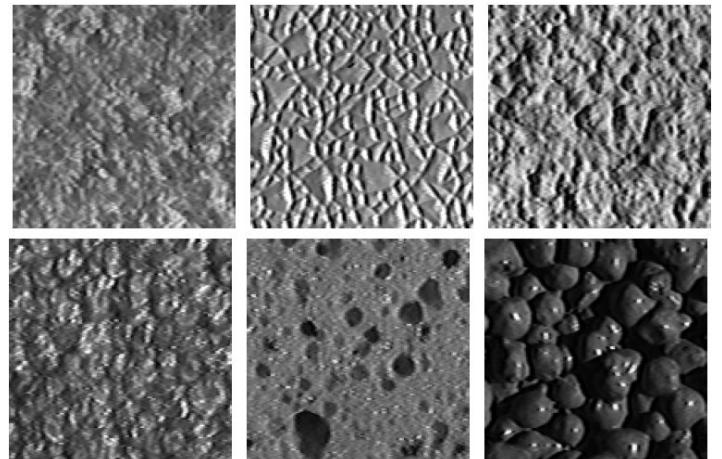


Figure from Sivic & Zisserman, ICCV 2003

# Visual words

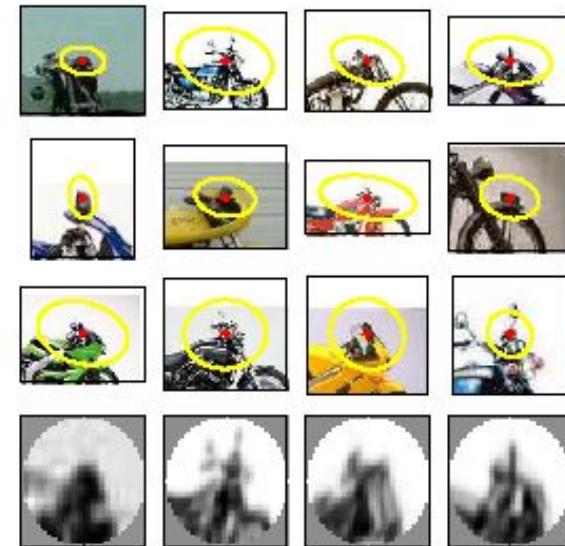
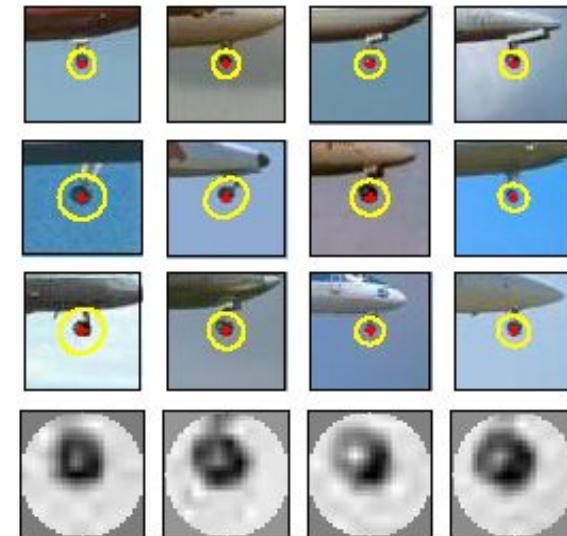
- First explored for texture and material representations
- *Texton* = cluster center of filter responses over collection of images
- Describe textures and materials based on distribution of prototypical texture elements.

Leung & Malik 1999; Varma & Zisserman, 2002; Lazebnik, Schmid & Ponce, 2003;



# Visual words

- More recently used for describing scenes and objects for the sake of indexing or classification.



Sivic & Zisserman 2003;  
Csurka, Bray, Dance, & Fan  
2004; many others.

# Inverted file index for images comprised of visual words



frame #5



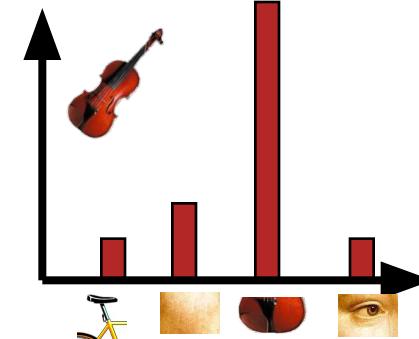
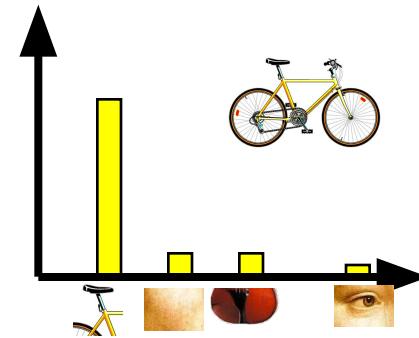
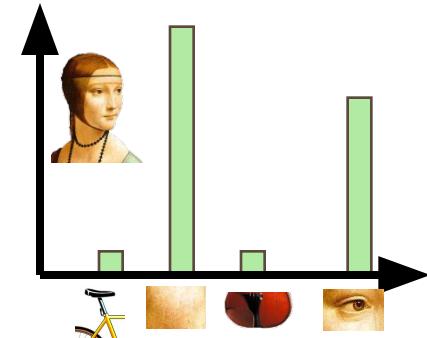
frame #10

Word number	List of image numbers
1	→ 5, 10, ...
2	→ 10, ...
...	...



# Bags of visual words

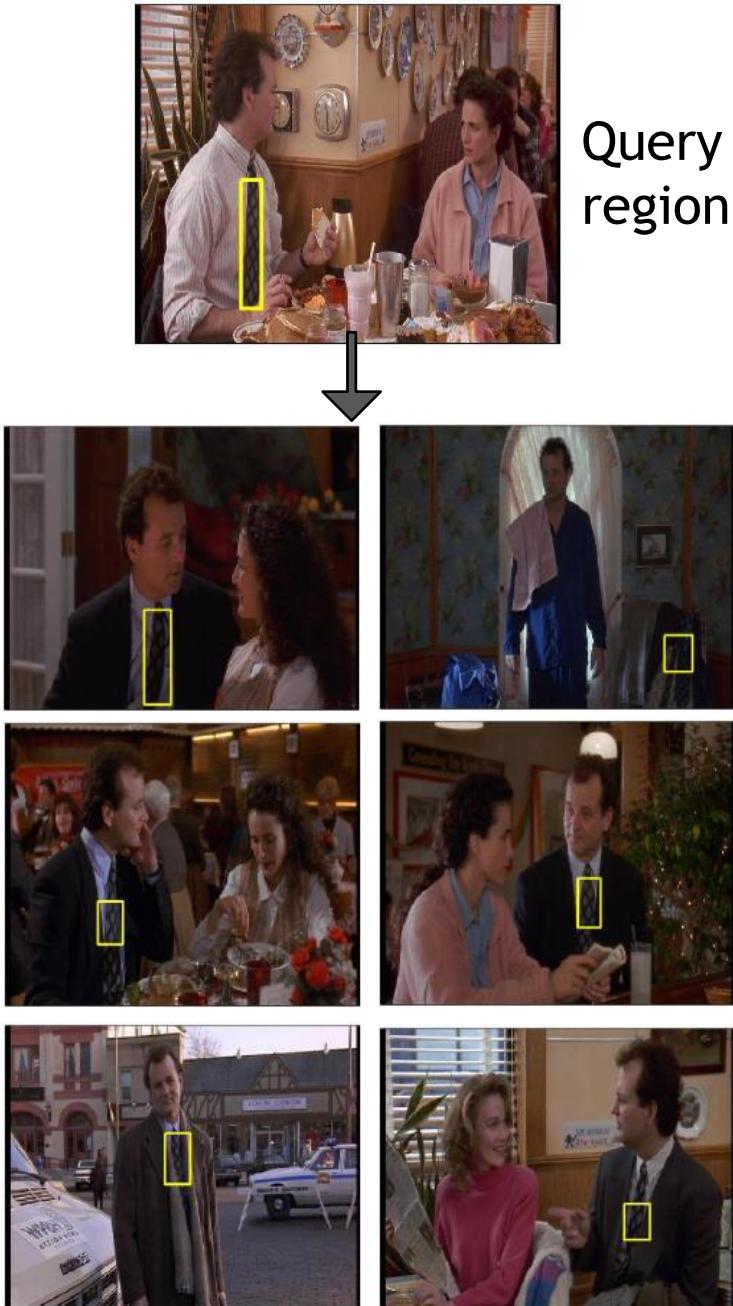
- Summarize entire image based on its distribution (histogram) of word occurrences.
- Analogous to bag of words representation commonly used for documents.



# Image Retrieval Altogether

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

Sivic & Zisserman, ICCV 2003

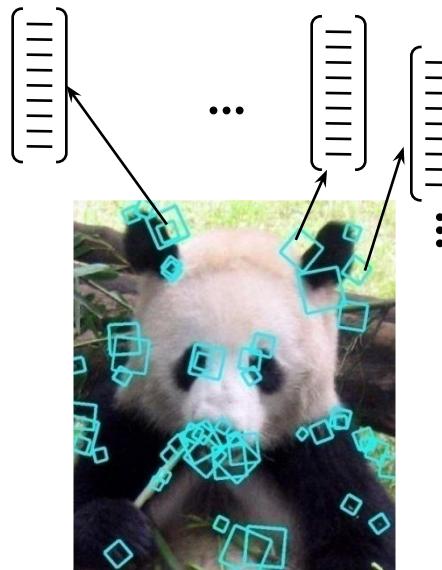


# Basic flow



**Detect or sample  
features**

List of positions,  
scales,  
orientations



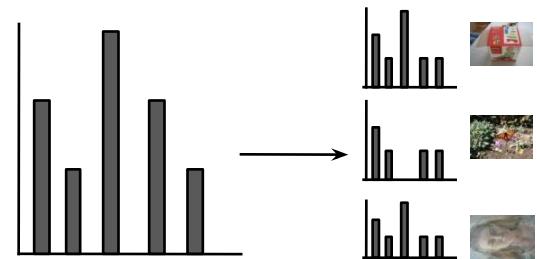
**Describe  
features**

Associated list of  
d-dimensional  
descriptors

or



**Index each one into pool  
of descriptors from  
previously seen images**



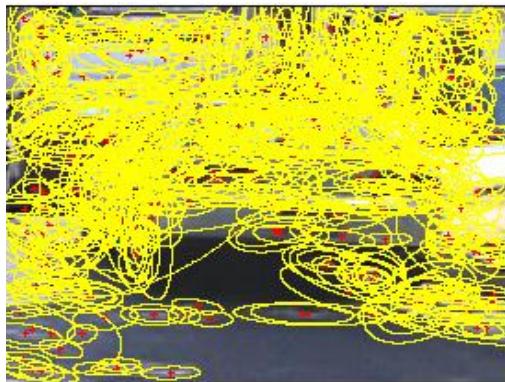
**Quantize to form  
bag of words vector  
for the image**

# Visual vocabulary formation

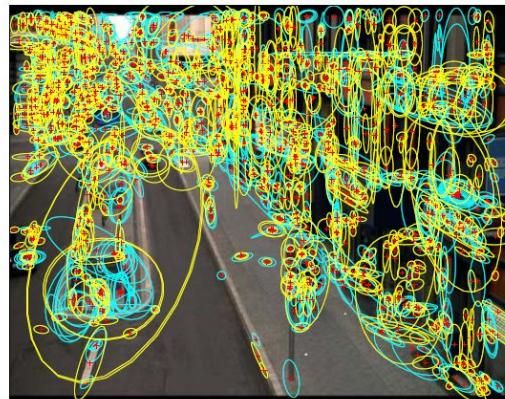
## Issues:

- Sampling strategy
- Clustering / quantization algorithm
- Unsupervised vs. supervised
- What corpus provides features (universal vocabulary?)
- Vocabulary size, number of words

# Sampling strategies



Sparse, at  
interest points



Multiple interest  
operators



Dense, uniformly



Randomly

- To find specific, textured objects, sparse sampling from interest points often more reliable.
- Multiple complementary interest operators offer more image coverage.
- For object categorization, dense sampling offers better coverage.

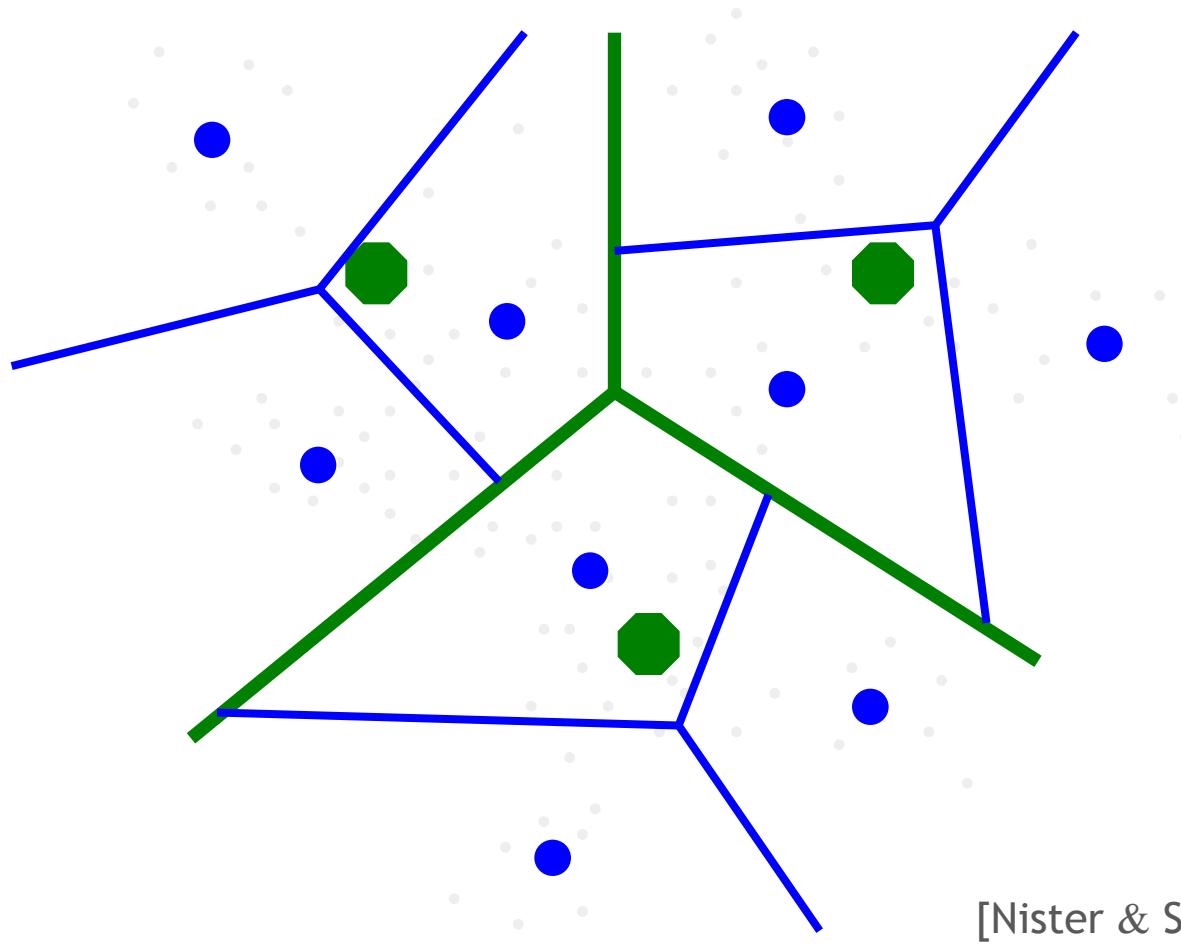
[See Nowak, Jurie & Triggs, ECCV 2006]

## Clustering / quantization methods

- k-means (typical choice), agglomerative clustering, mean-shift,...
- Hierarchical clustering: allows faster insertion / word assignment while still allowing large vocabularies
  - Vocabulary tree [Nister & Stewenius, CVPR 2006]

# Example: Recognition with Vocabulary Tree

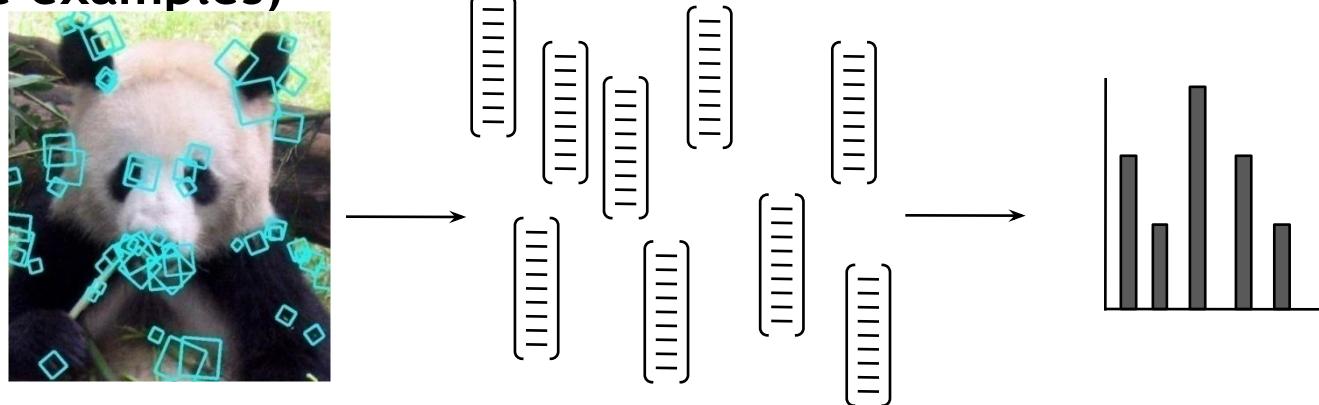
- Tree construction:



[Nister & Stewenius, CVPR'06]

# Learning and recognition with bag of words histograms

- Bag of words representation makes it possible to describe the unordered point set with a single vector (of fixed dimension across image examples)



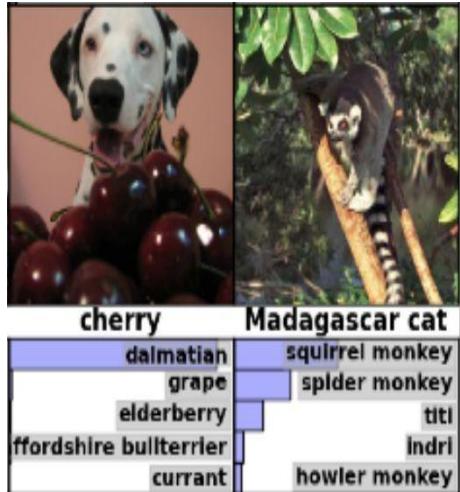
- Provides easy way to use distribution of feature types with various learning algorithms requiring vector input.

## Bags of words: pros and cons

- + flexible to geometry / deformations / viewpoint
- + compact summary of image content
- + provides vector representation for sets
- + has yielded good recognition results in practice
  
- basic model ignores geometry - must verify afterwards, or encode via features
- background and foreground mixed when bag covers whole image
- interest points or sampling: no guarantee to capture object-level parts
- optimal vocabulary formation remains unclear

# Convolutional neural networks and retrieval

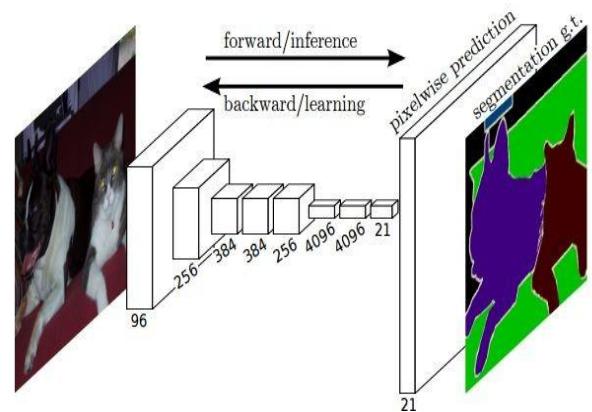
Classification



Object Detection

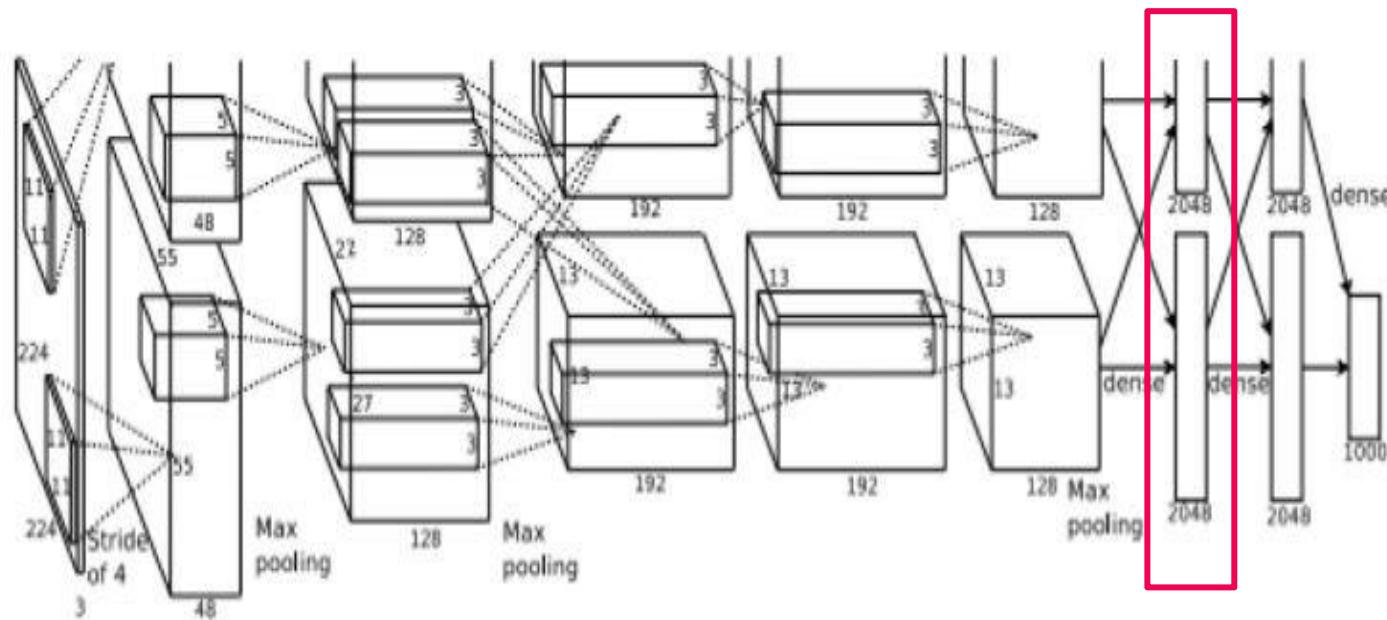


Segmentation



# Off-the-shelf CNN representations

FC layers as global feature representation



# Off-the-shelf CNN representations

## Neural codes for retrieval [1]

- FC7 layer (4096D)
- $L^2$  norm + PCA whitening +  $L^2$  norm
- Euclidean distance
- Only better than traditional SIFT approach after fine tuning on similar domain image dataset.

## CNN features off-the-shelf: an astounding baseline for recognition [2]

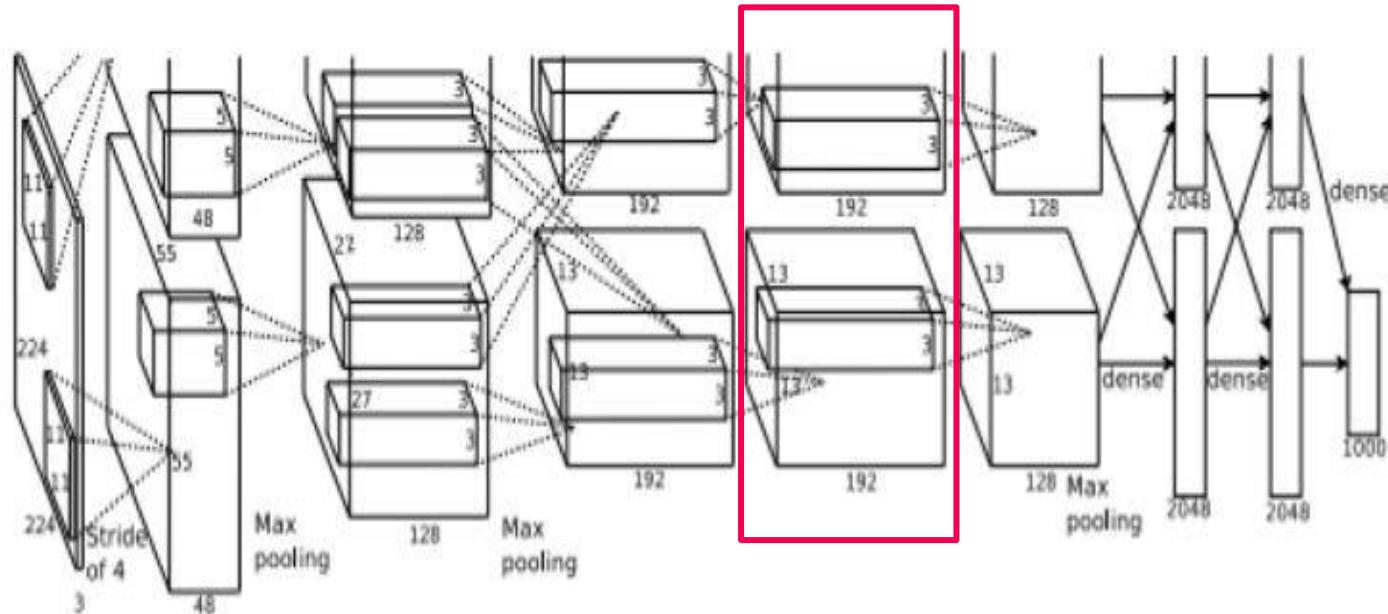
- Extending Babenko's approach with spatial search
- Several features extracted by image (sliding window approach)
- Really good results but too computationally expensive for practical situations

[1] Babenko et al, [Neural codes for image retrieval](#), CVPR

[2] Razavian et al, [CNN features off-the-shelf: an astounding baseline for recognition](#), CVPR  
2014

# Off-the-shelf CNN representations

sum/max pool conv features across filters



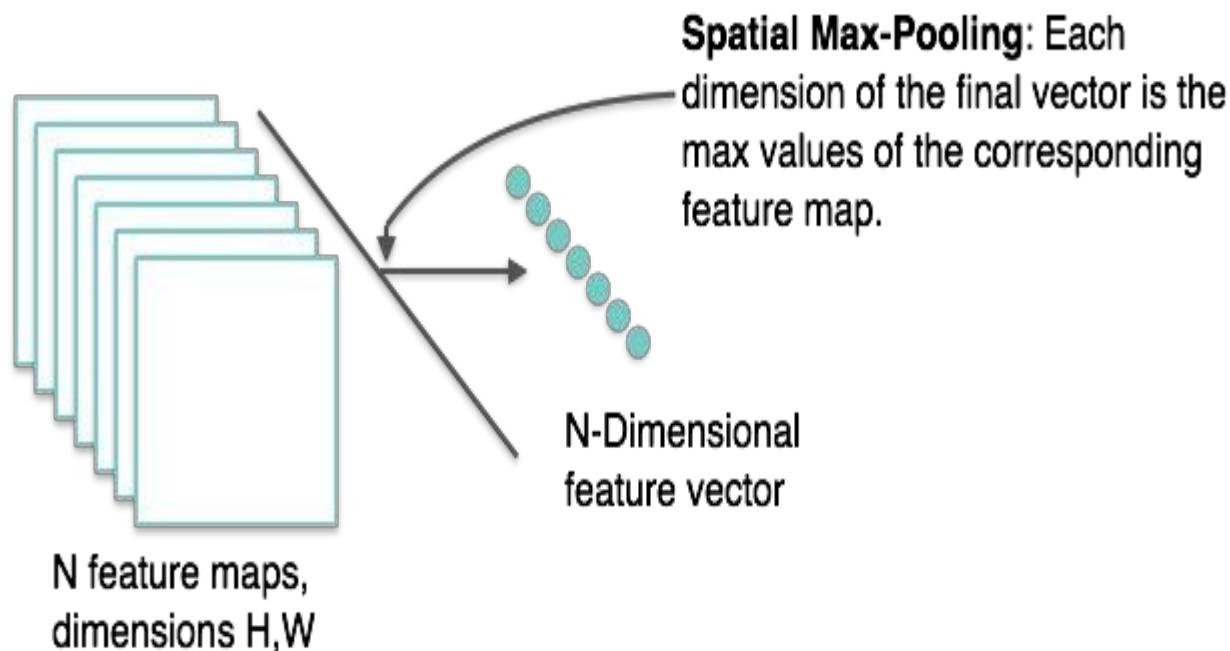
Babenko and Lempitsky, [Aggregating local deep features for image retrieval](#). ICCV 2015

Tolias et al. [Particular object retrieval with integral max-pooling of CNN activations](#). arXiv:1511.05879.

Kalantidis et al. [Cross-dimensional Weighting for Aggregated Deep Convolutional Features](#). arXiv:1512.04065.

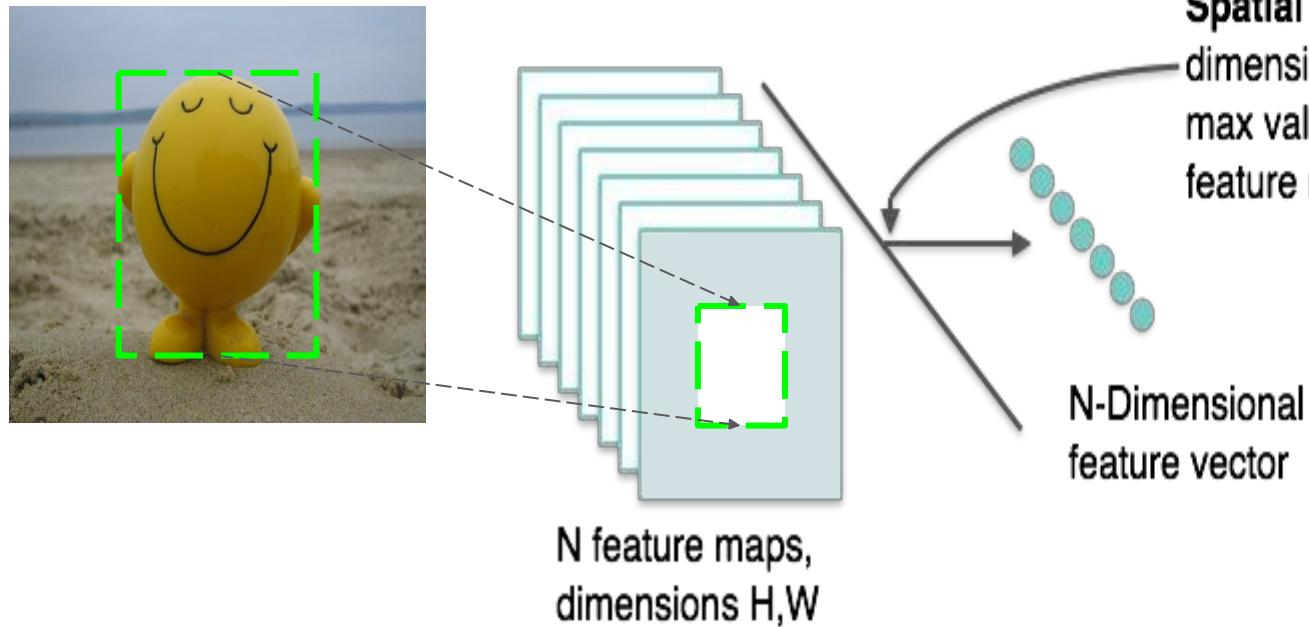
# Off-the-shelf CNN representations

Descriptors from convolutional layers



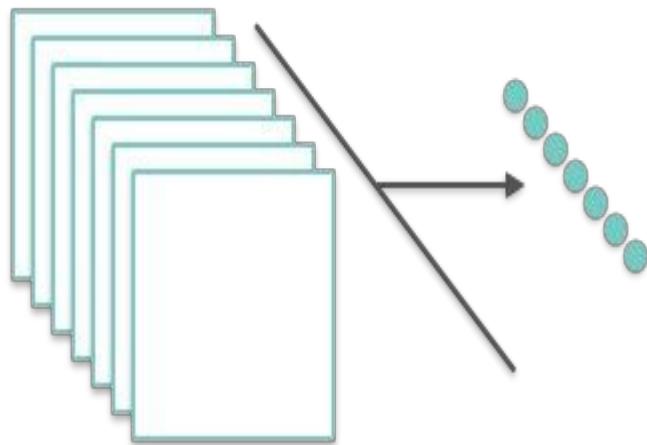
# Off-the-shelf CNN representations

Pooling features on conv layers allow to describe specific parts of an image

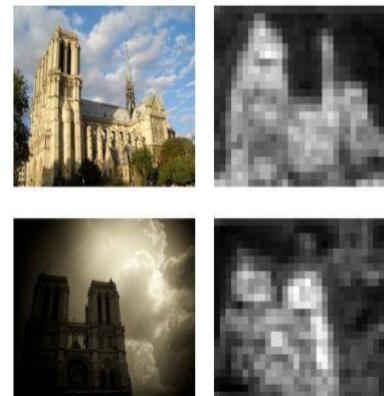


# Off-the-shelf CNN representations

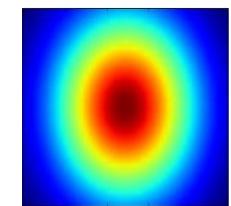
Sum/max pooling operation of a conv layer



Apply spatial weighting on the features before pooling them



[1] weighting based on 'strength' of the local features



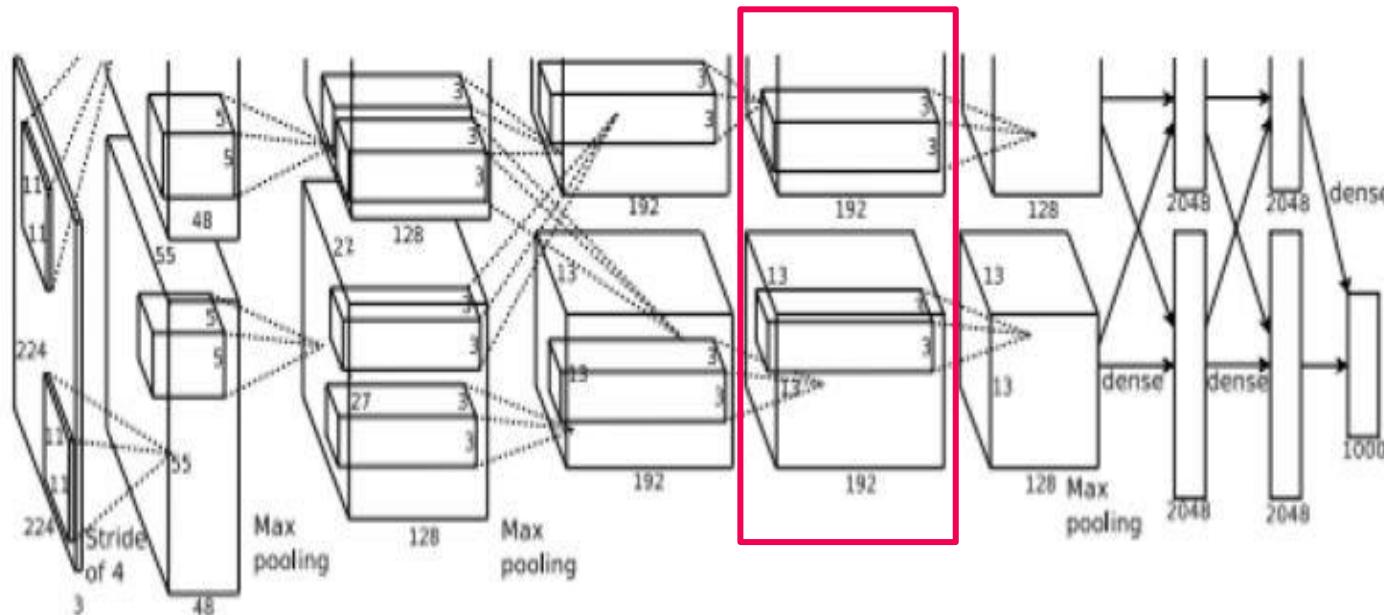
[2] weighting based on the distance to the center of the image

[1] Babenko and Lempitsky, [Aggregating local deep features for image retrieval](#). ICCV 2015

[2] Kalantidis et al. [Cross-dimensional Weighting for Aggregated Deep Convolutional Features](#). ECCV 2016

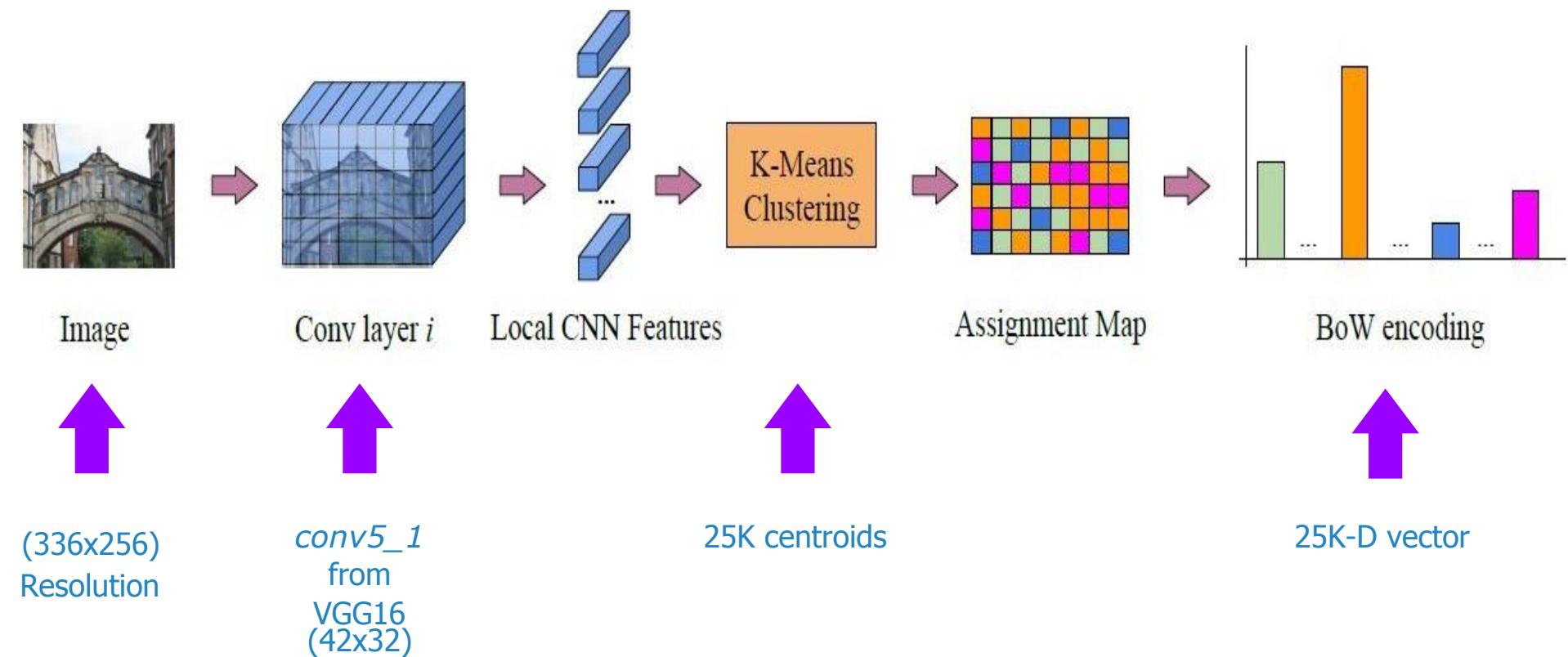
# Off-the-shelf CNN representations

BoW, VLAD encoding of conv features

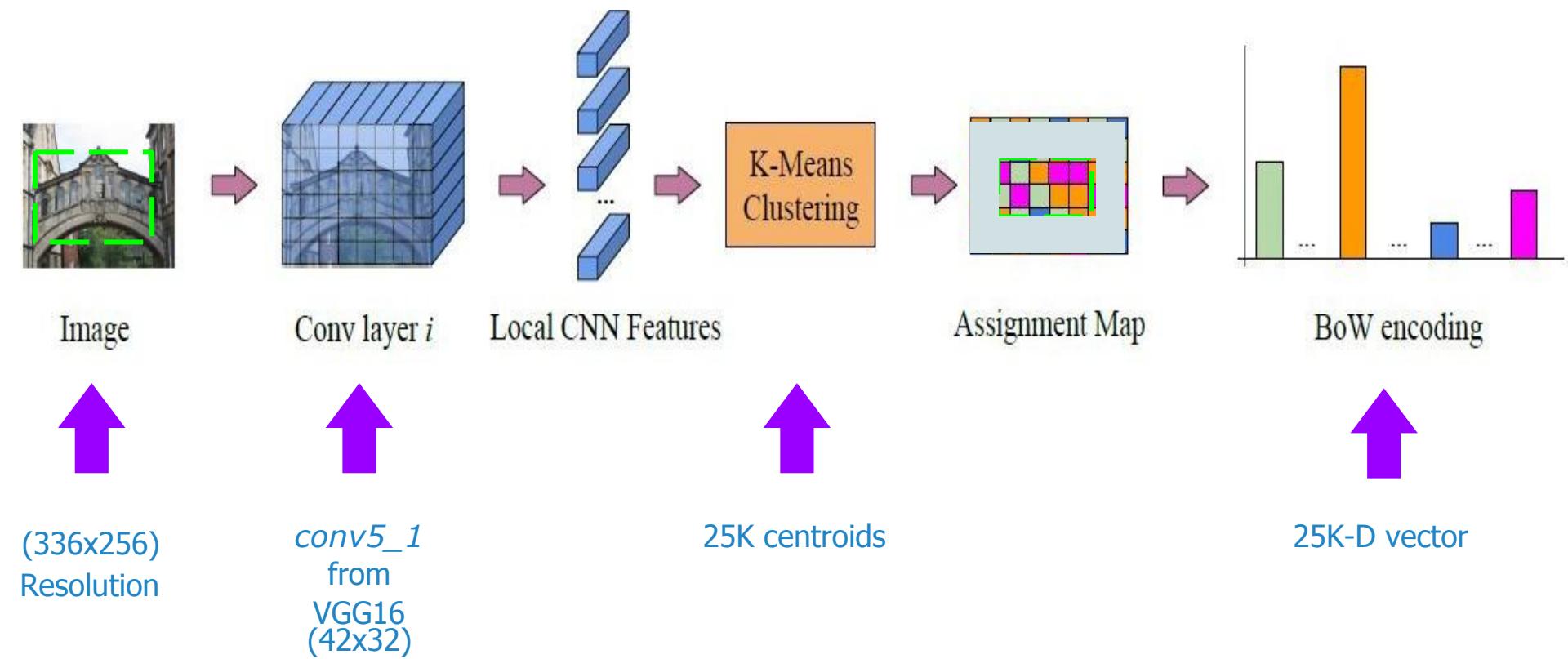


# Off-the-shelf CNN representations

Descriptors from convolutional layers



# Off-the-shelf CNN representations



# CNN as a feature extractor

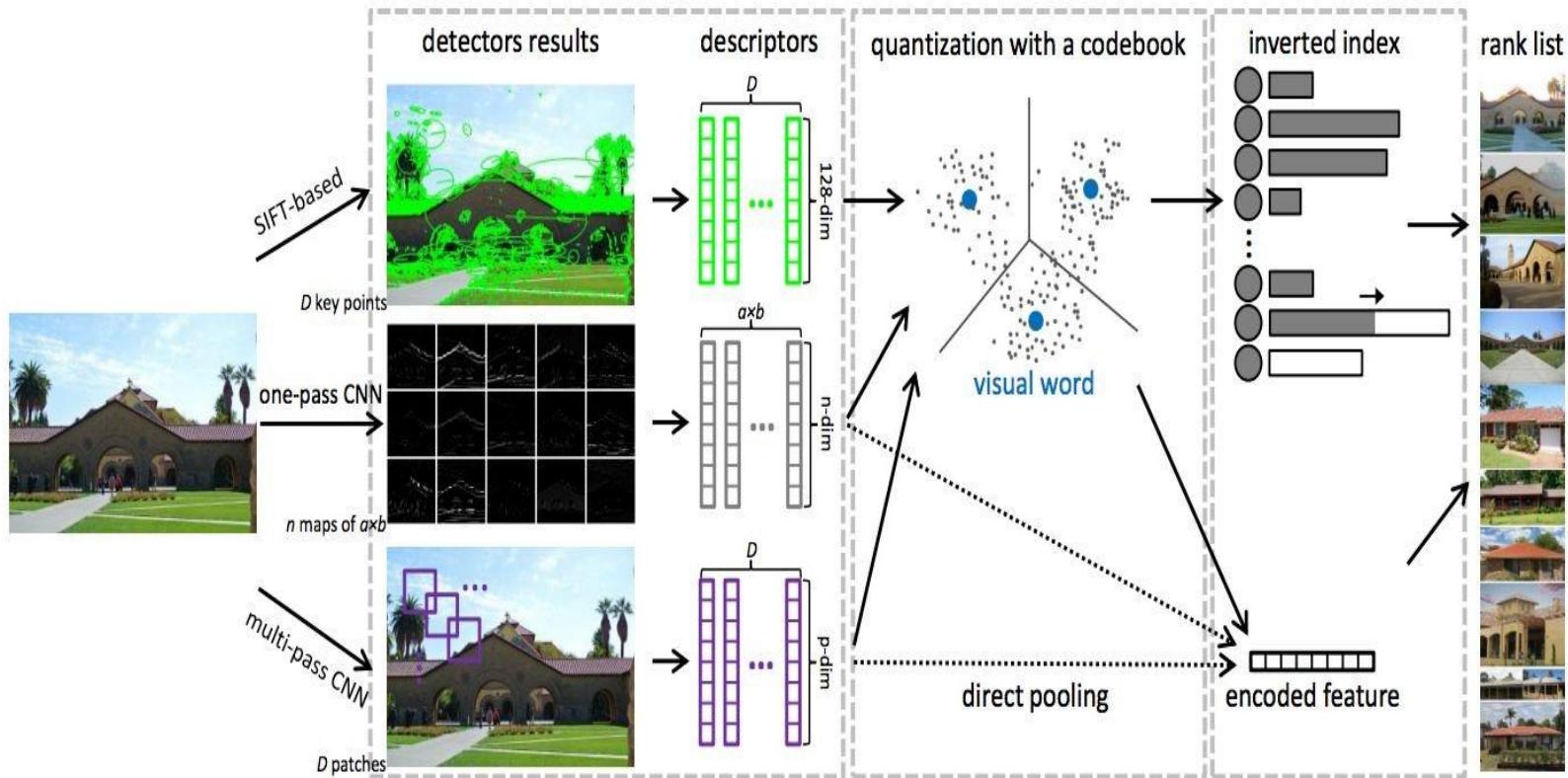


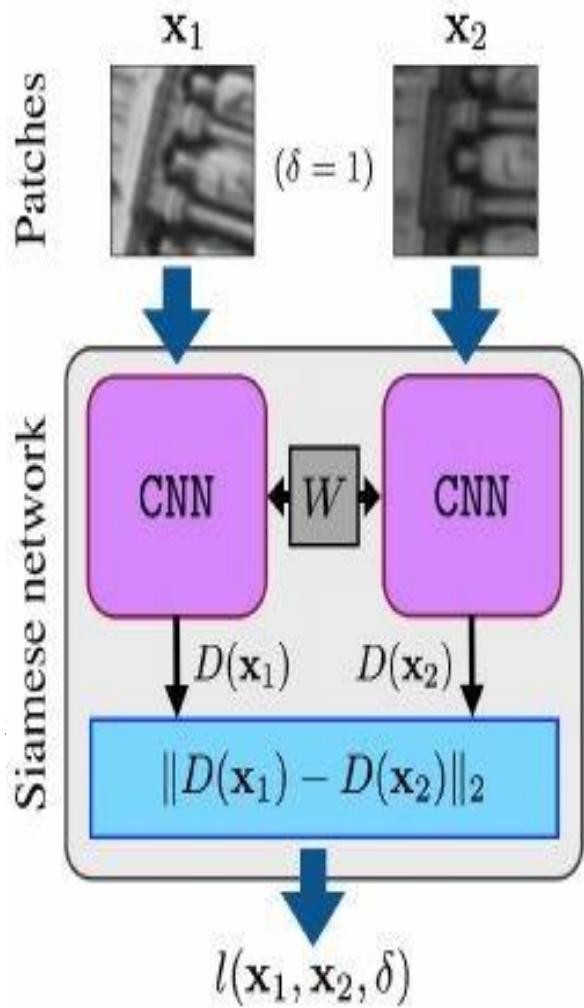
Diagram from [SIFT Meets CNN: A Decade Survey of Instance Retrieval](#)

# Learning representations for retrieval

**Siamese network:** network to learn a function that maps input patterns into a target space such that  $L^2$  norm in the target space approximates the semantic distance in the input space.

Applied in:

- Dimensionality reduction [1]
- Face verification [2]
- Learning local image representations [3]



[1] Song et al.: [Deep metric learning via lifted structured feature embedding](#). CVPR 2015

[2] Chopra et al. [Learning a similarity metric discriminatively, with application to face verification](#)

CVPR' [3] Simo-Serra et al. [Fracking deep convolutional image descriptors](#). CoRR, abs/1412.6537, 2014

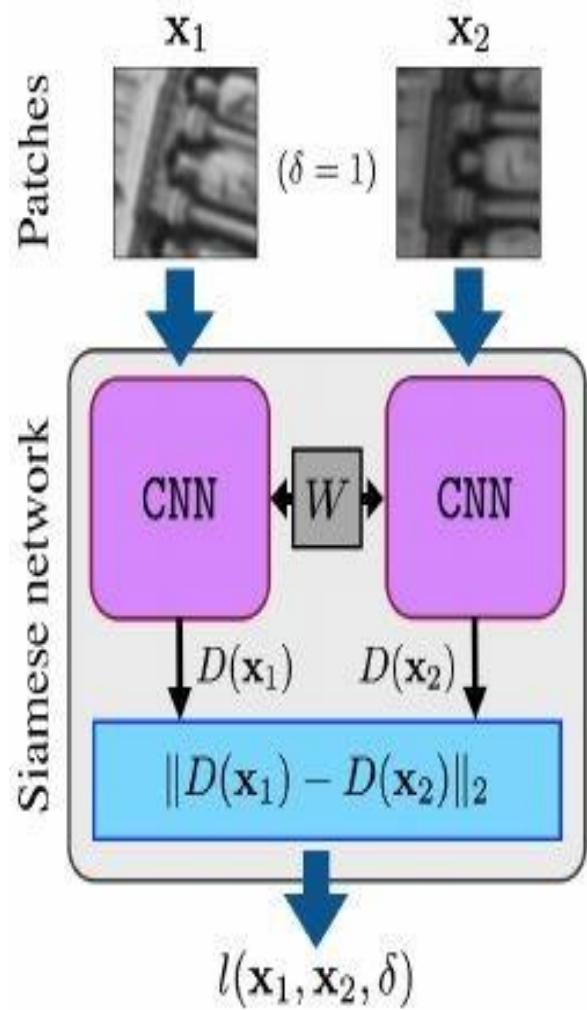
# Learning representations for retrieval

**Siamese network:** network to learn a function that maps input patterns into a target space such that  $L^2$  norm in the target space approximates the semantic distance in the input space.

$$l(\mathbf{x}_1, \mathbf{x}_2, \delta) = \delta \cdot l_P(d_D(\mathbf{x}_1, \mathbf{x}_2)) + (1 - \delta) \cdot l_N(d_D(\mathbf{x}_1, \mathbf{x}_2))$$

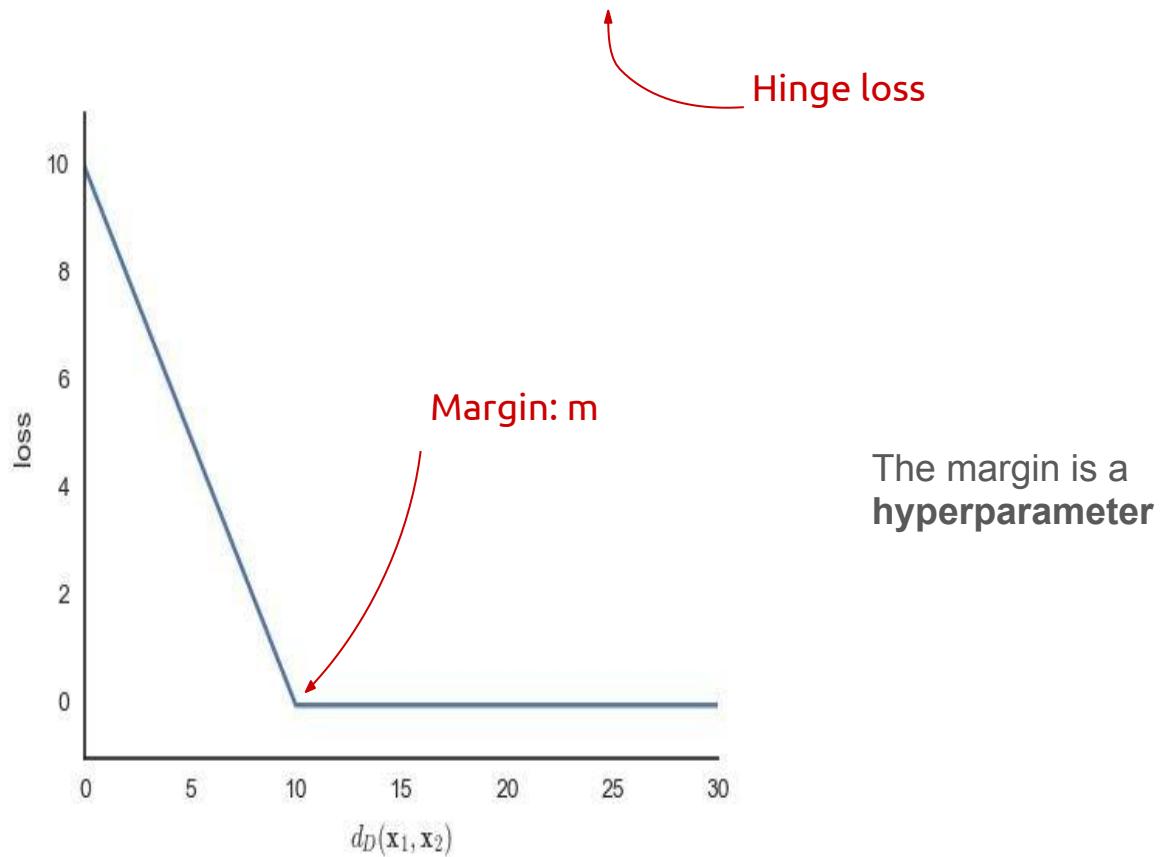
$$l_P(d_D(\mathbf{x}_1, \mathbf{x}_2)) = d_D(\mathbf{x}_1, \mathbf{x}_2)$$

$$l_N(d_D(\mathbf{x}_1, \mathbf{x}_2)) = \max(0, m - d_D(\mathbf{x}_1, \mathbf{x}_2))$$



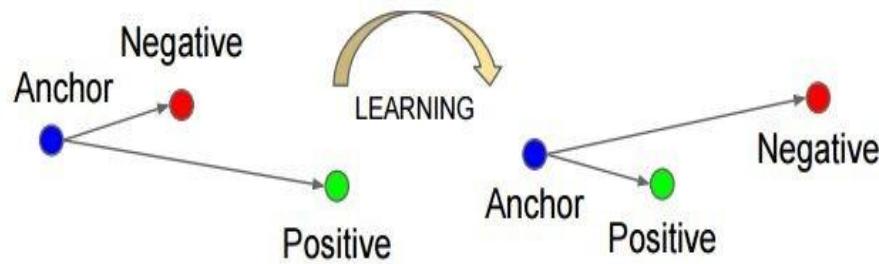
$$l_N(d_D(\mathbf{x}_1, \mathbf{x}_2)) = \max(0, m - d_D(\mathbf{x}_1, \mathbf{x}_2))$$

**Negative pairs:** if nearer than the margin, pay a linear penalty

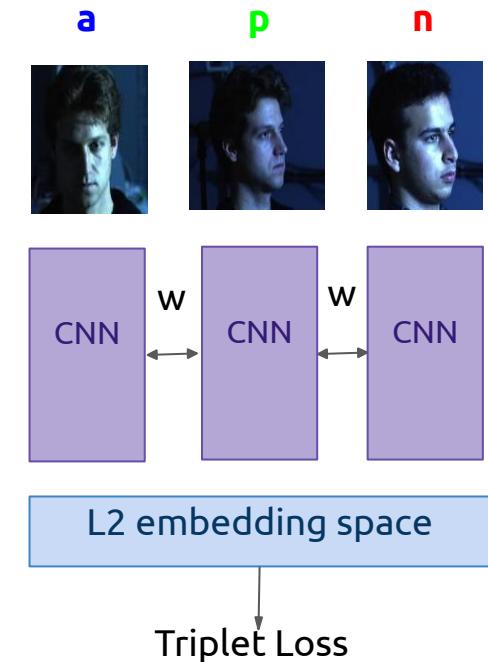


# Learning representations for retrieval

**Siamese network with triplet loss:** loss function minimizes distance between query and positive maximizes distance between query and negative



$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2$$

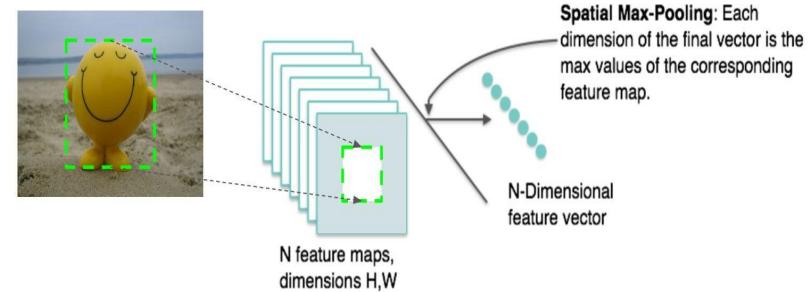


# Learning representations for retrieval

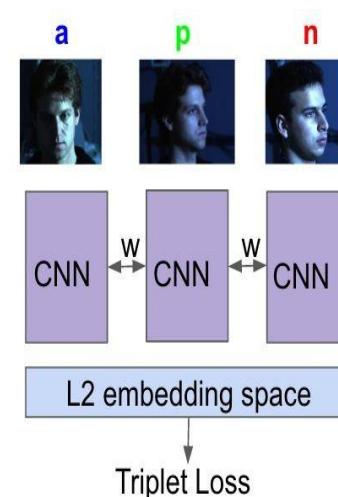
Deep Image Retrieval: Learning global representations for image search, Gordo A. et al. Xerox Research Centre, 2016

- R-MAC representation  
Learning descriptors for retrieval using three channels  
siamese loss: Ranking objective:

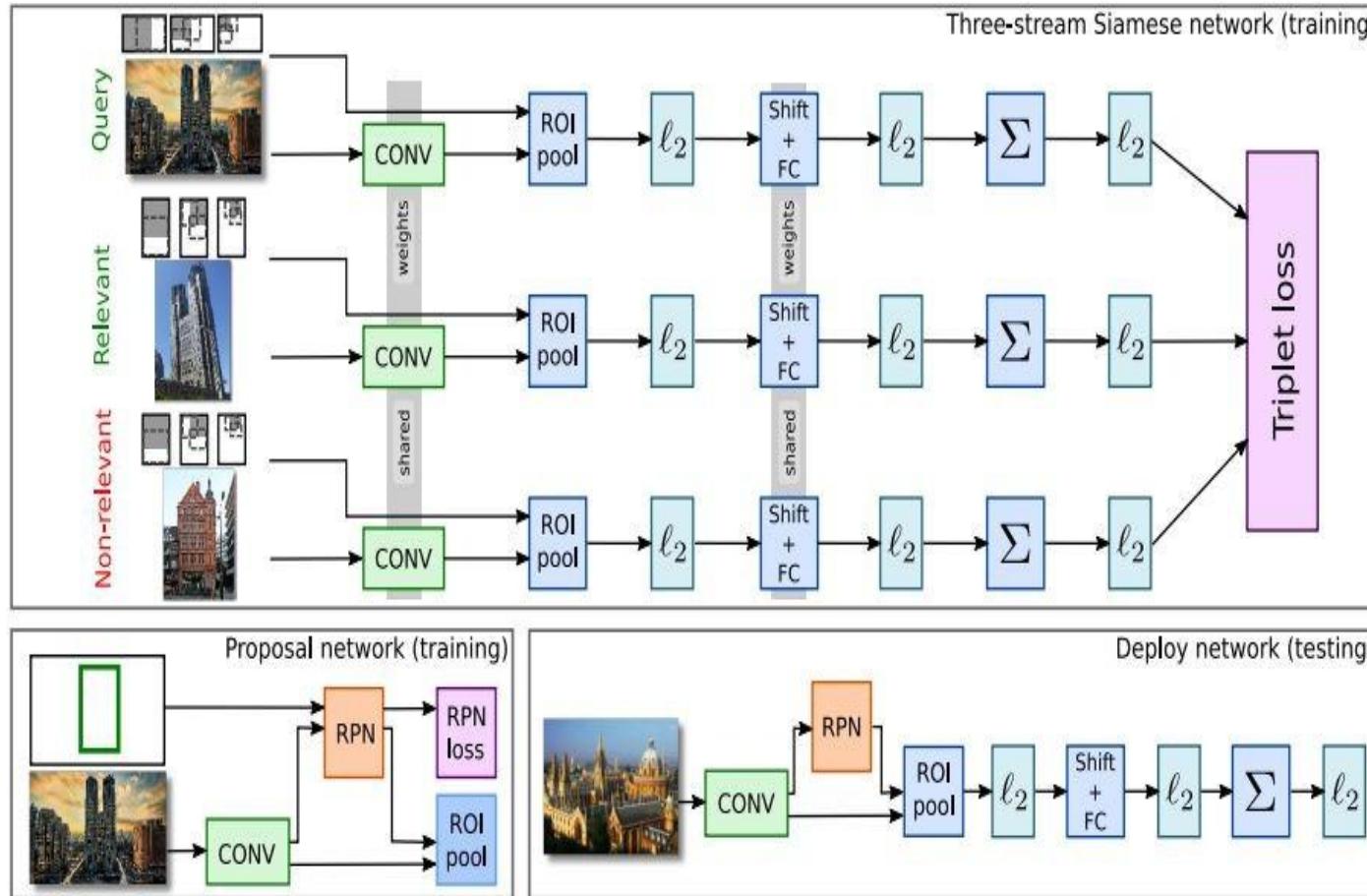
$$L(I_q, I^+, I^-) = \max(0, m + q^T d^- - q^T d^+)$$



- Learning where to pool within an image: predicting object locations
- Local features (from predicted ROI) pooled into a more discriminative space (learned fc)
- Building and cleaning a dataset to generate triplets
- 



# Learning representations for retrieval



# Questions?