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

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

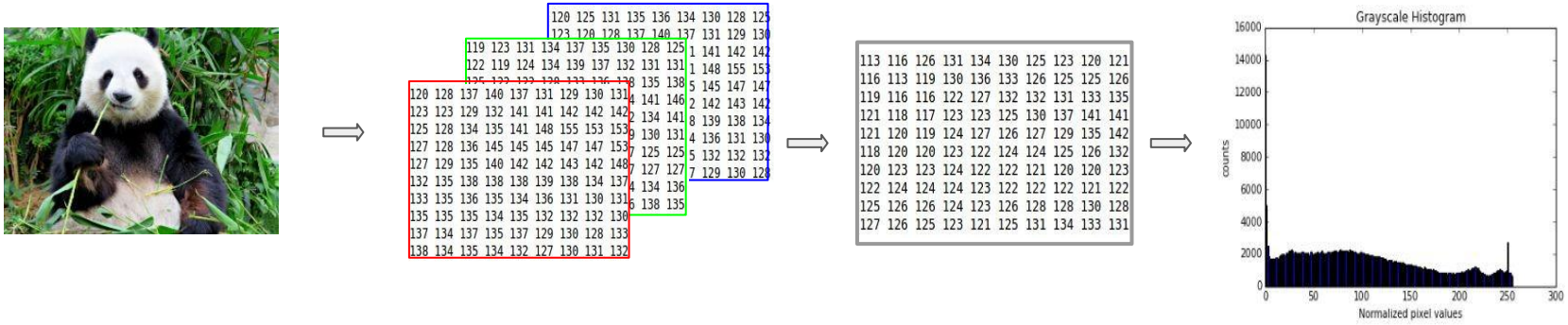
Distance: 0.0

Distance: 0.654

Distance: 0.661

Challenges: comparing images

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

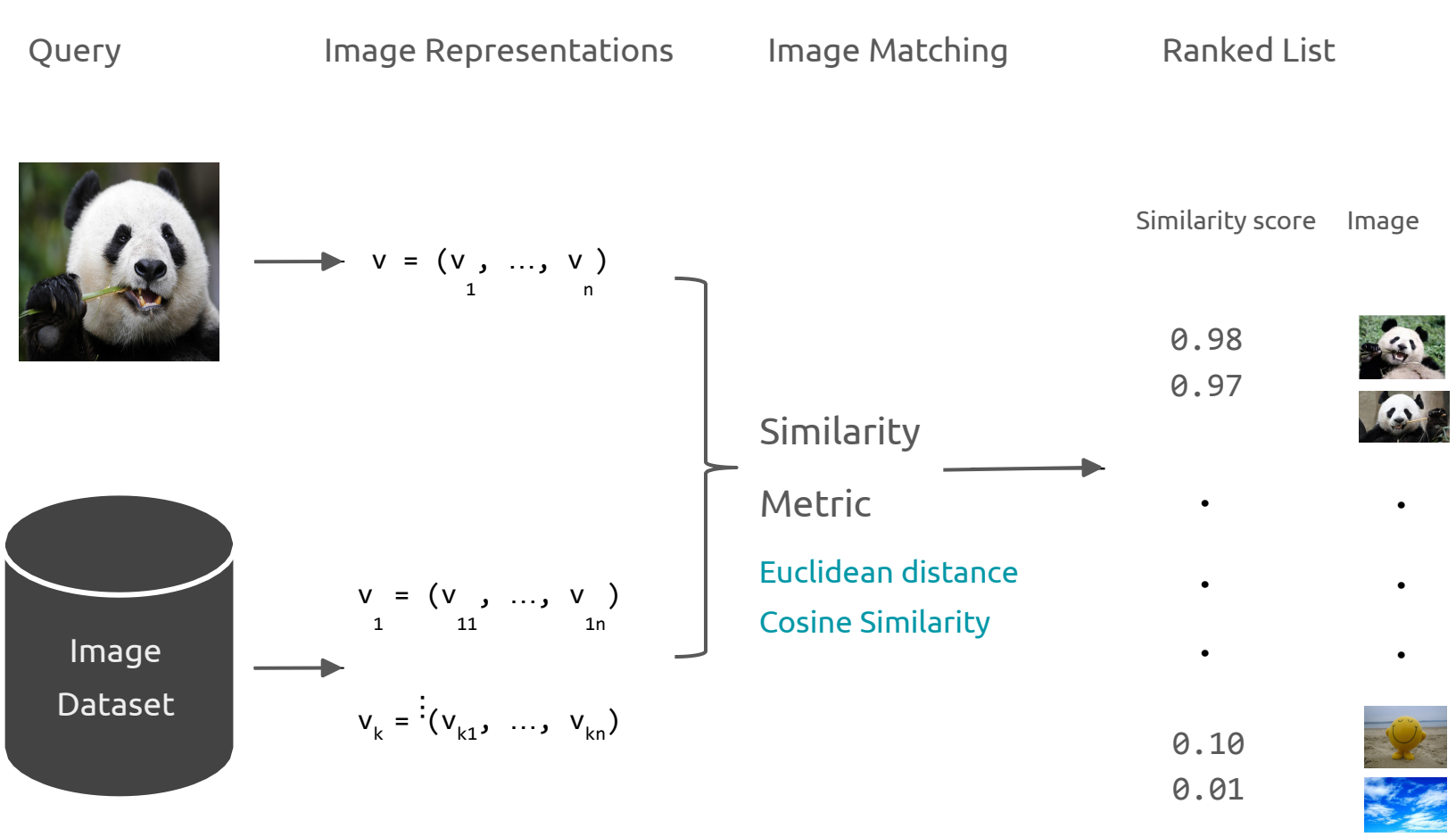
Results from dataset classified as “chair”

Retrieval

Query: This chair

Results from dataset ranked by similarity to the query

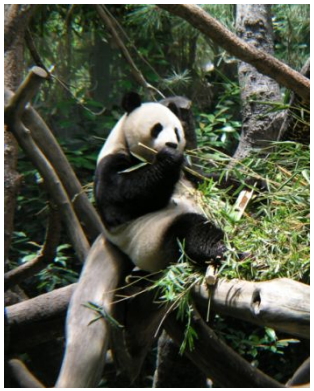
The retrieval pipeline



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

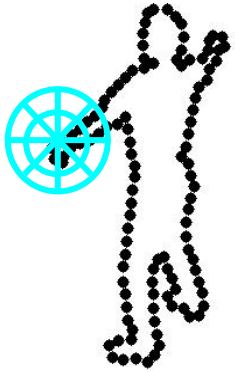




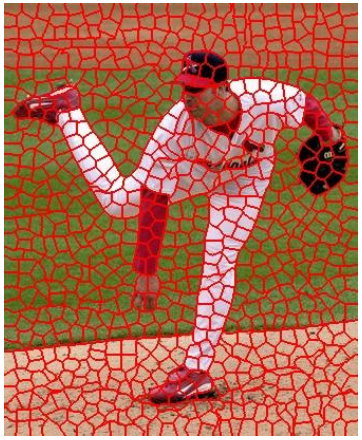

Local representations

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

SIFT [Lowe 99]

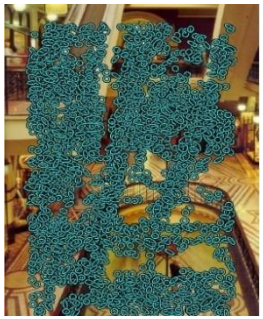


Shape context
[Belongie 02]

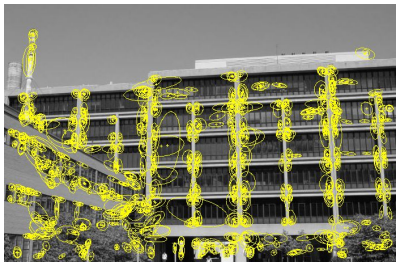


Superpixels
[Ren et al.]

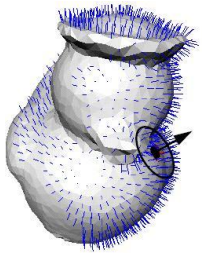
Maximally Stable
Extremal Regions
[Matas 02]



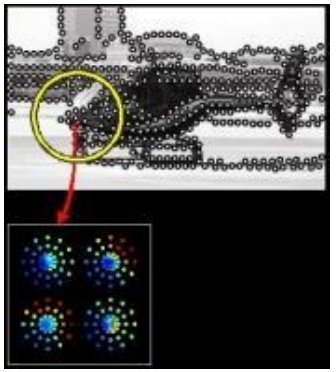
Salient regions
[Kadir 01]



Harris-Affine
[Mikolajczyk 04]



Spin images
[Johnson 99]



Geometric Blur
[Berg 05]

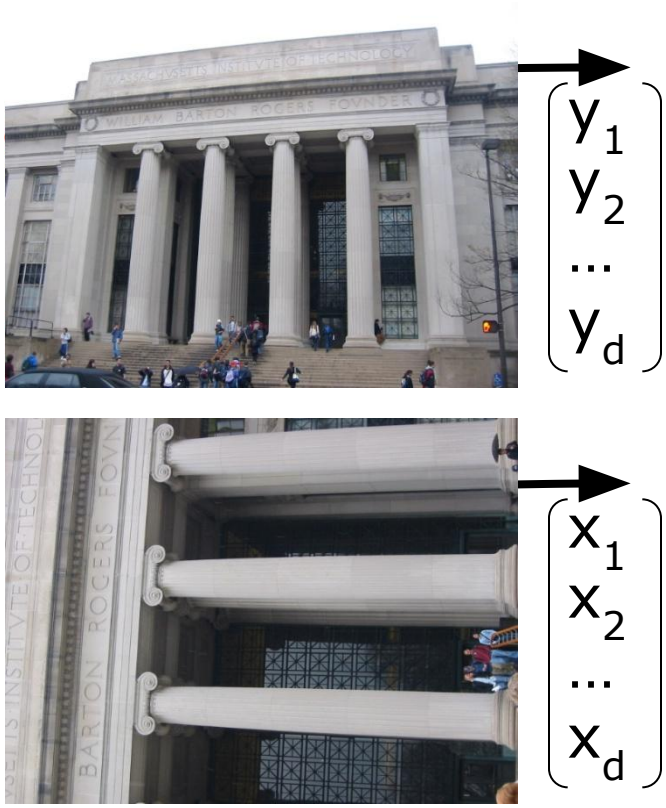
Recall: Invariant local features

Subset of local feature types designed to be invariant to

- Scale
- Translation
- Rotation
- Affine transformations
- Illumination

Detect interest points

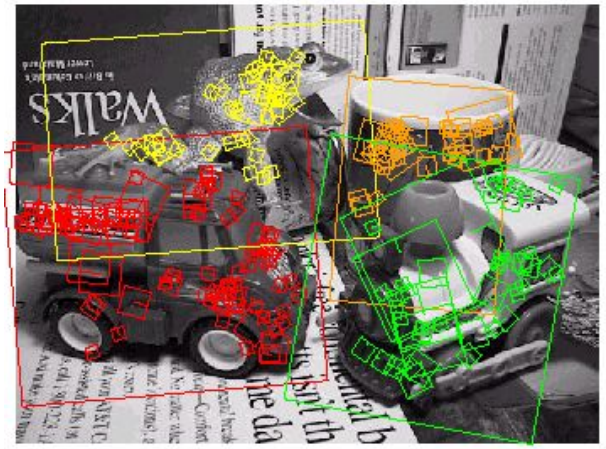
Extract descriptors

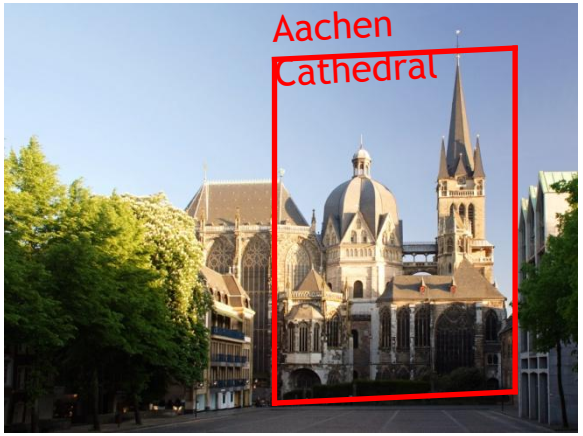


[Mikolajczyk01, Matas02, Tuytelaars04, Lowe99, Kadir01,…]

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

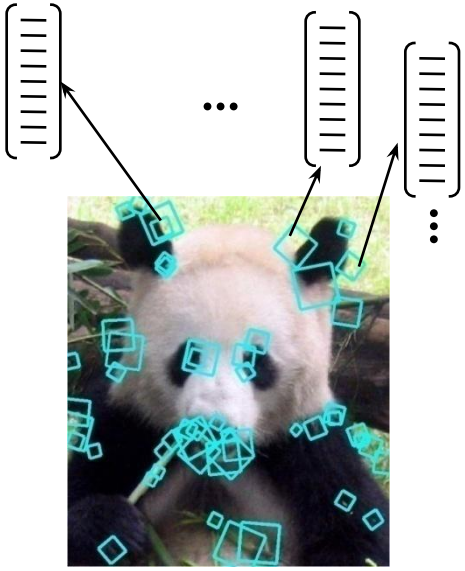




Basic flow

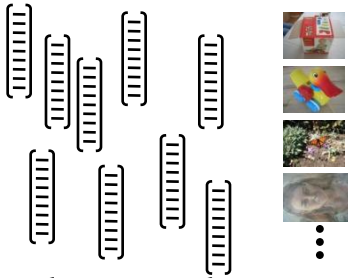
Detect or sample features

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.

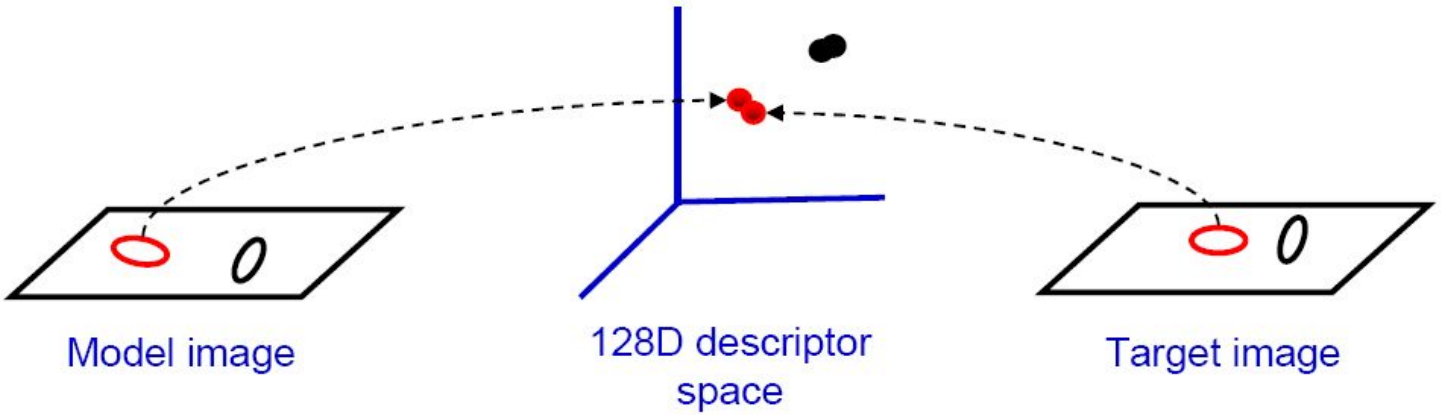


Figure credit: A. Zisserman

Indexing local features

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

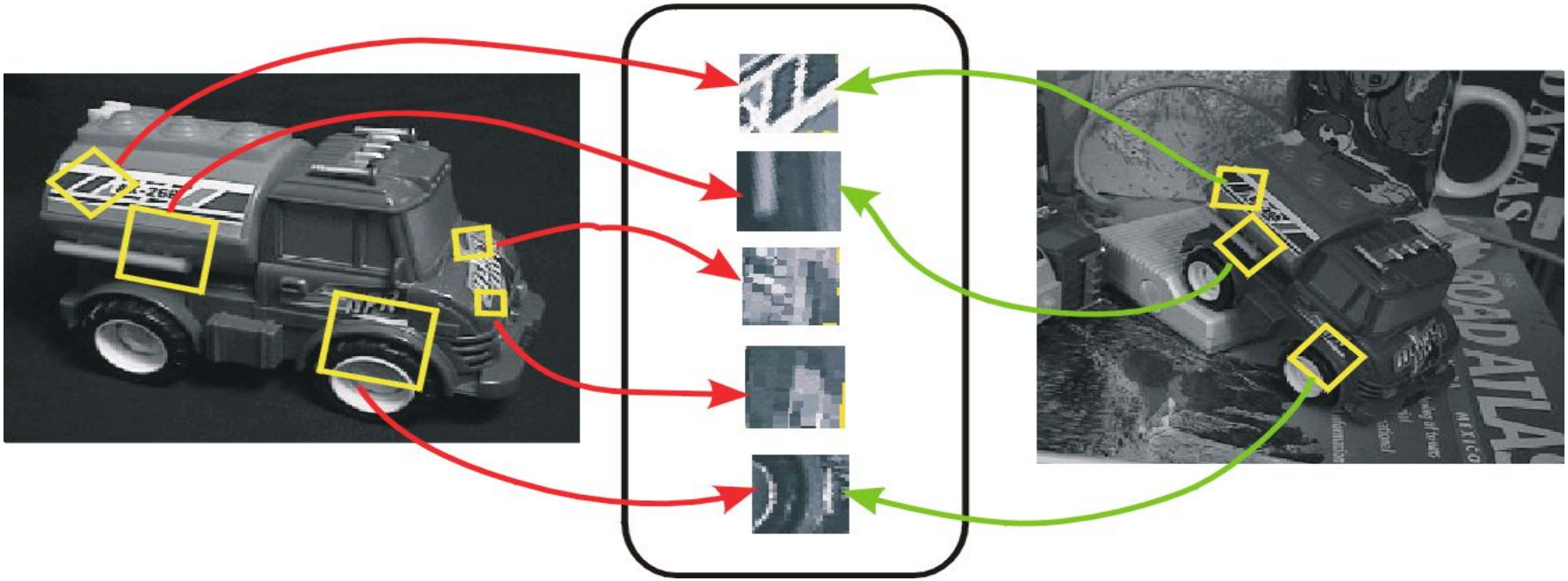
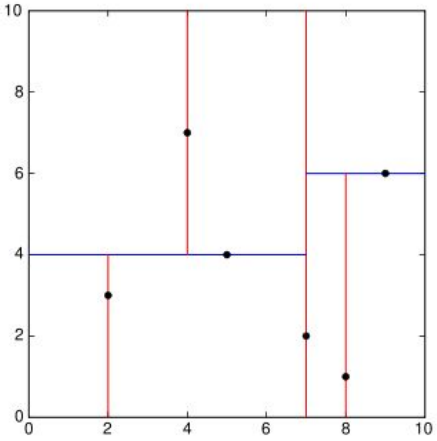


Figure credit: David Lowe

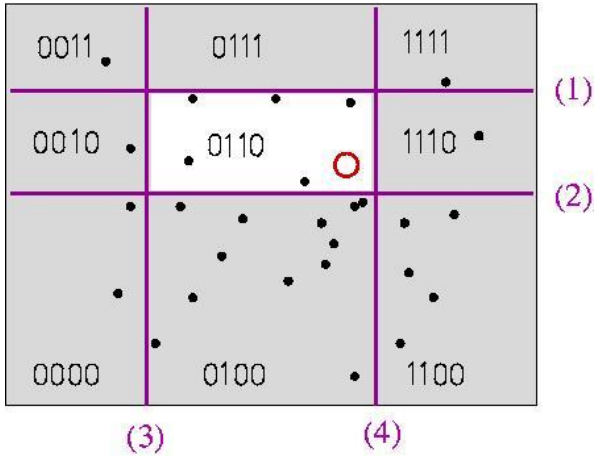
Indexing local features

- 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

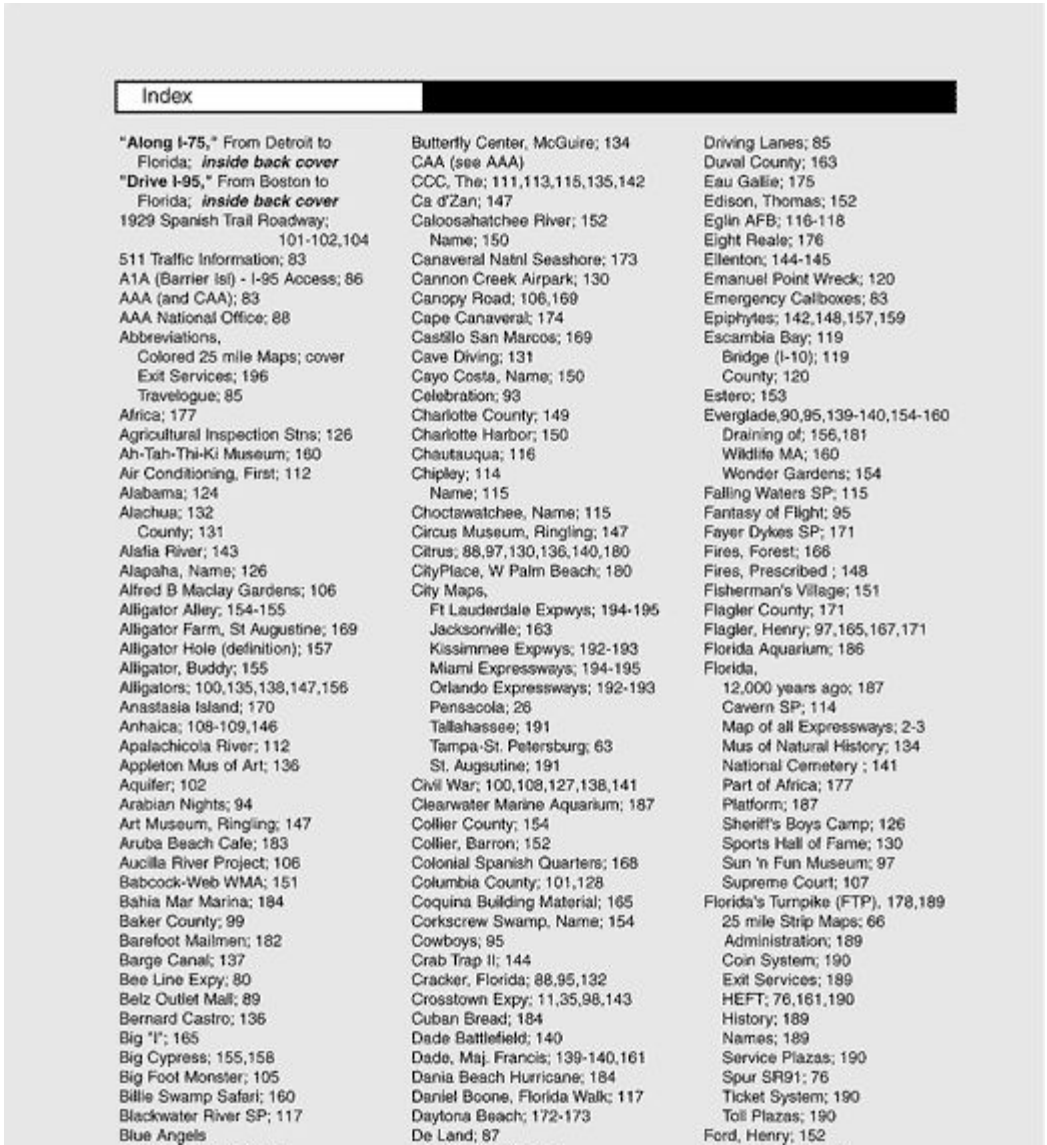


Best-Bin First (BBF), a variant of k-d trees that uses priority queue to examine most promising branches first [Beis & Lowe, CVPR 1997]



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

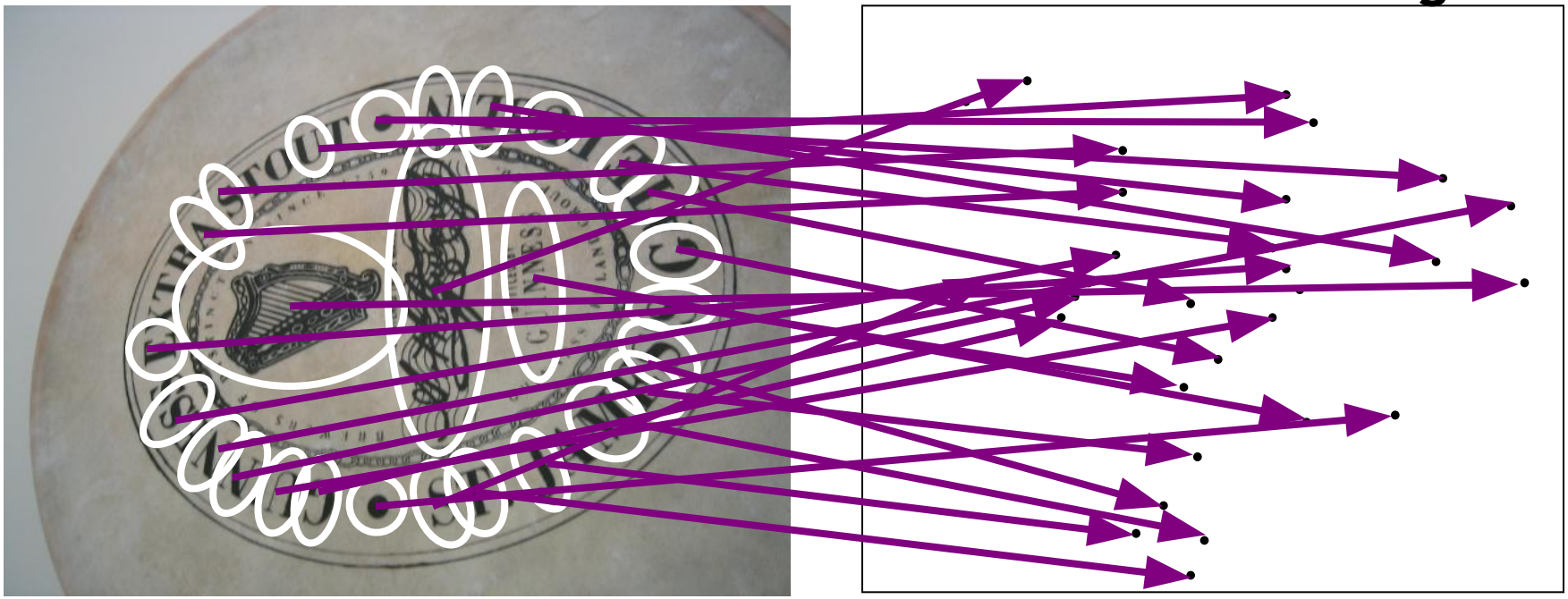


Index

- 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

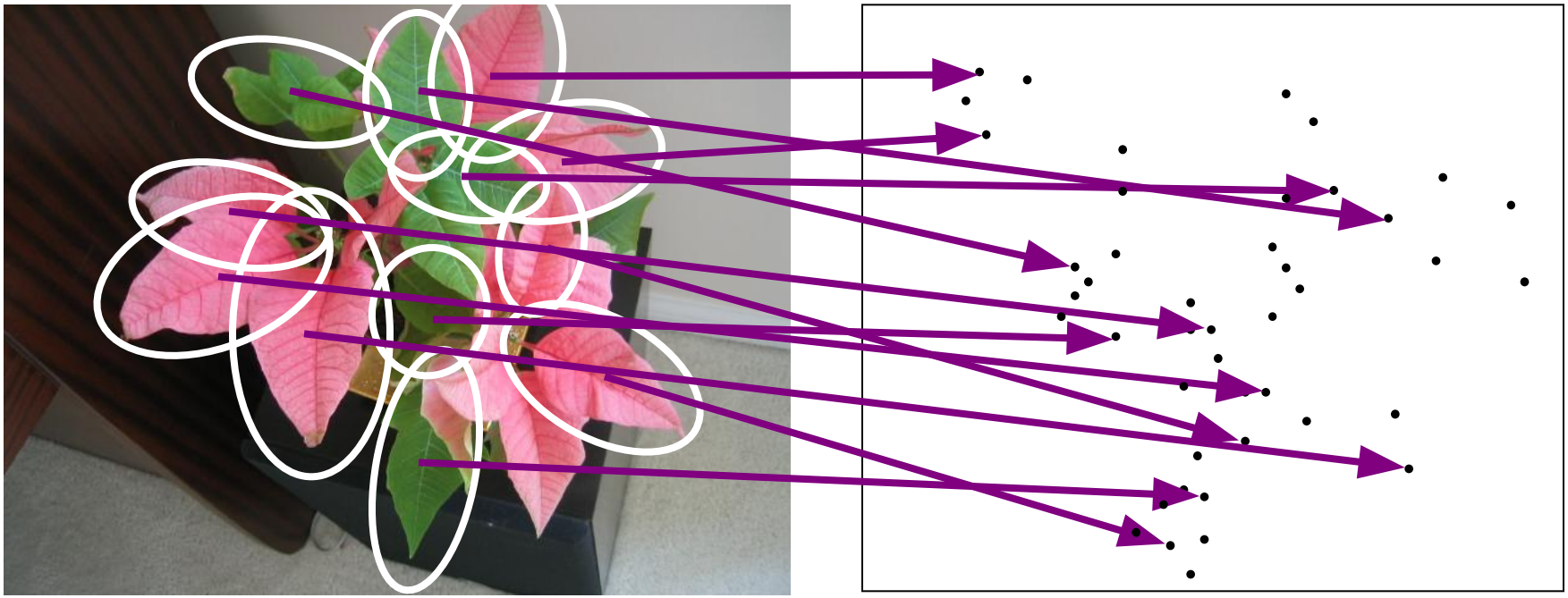
- Extract some local features from a number of images …



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

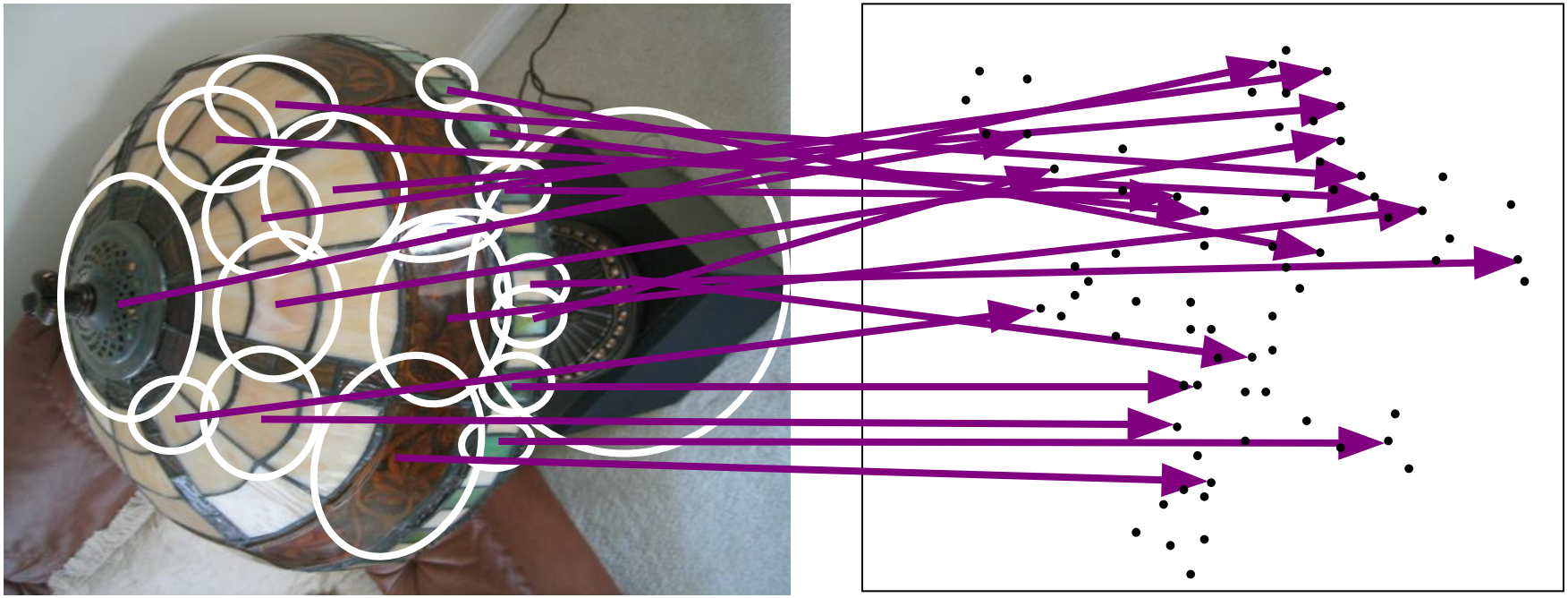
Slide credit: D. Nister

Visual words: main idea



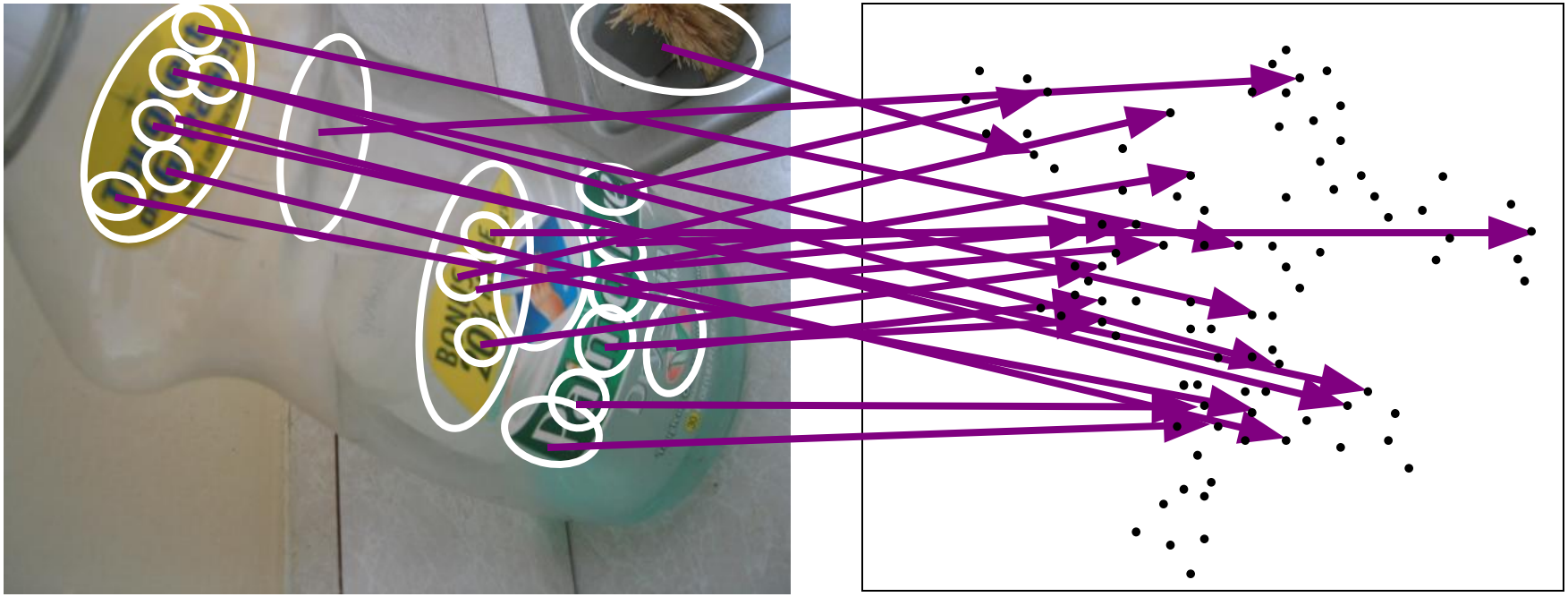
Slide credit: D. Nister

Visual words: main idea

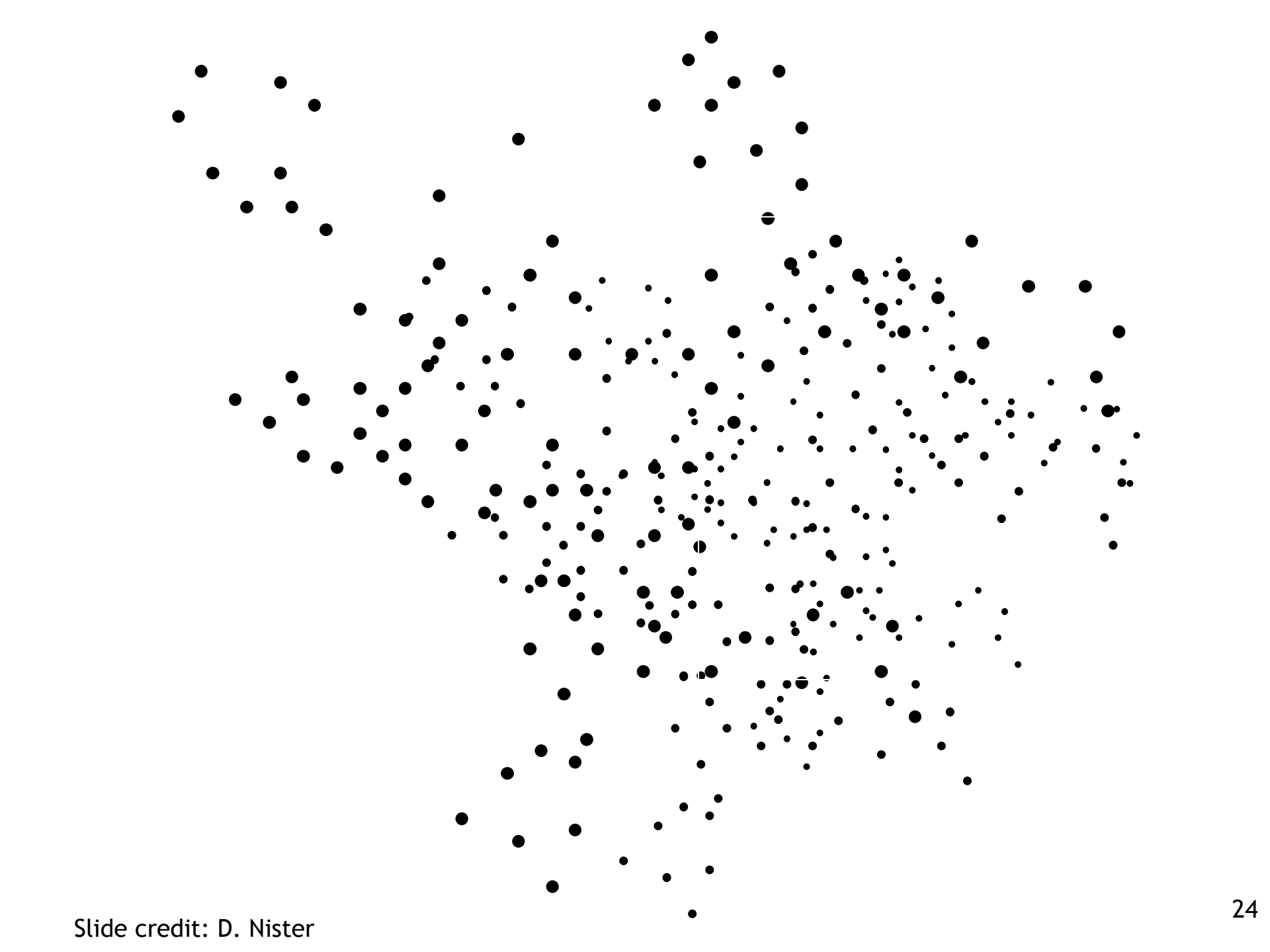


Slide credit: D. Nister

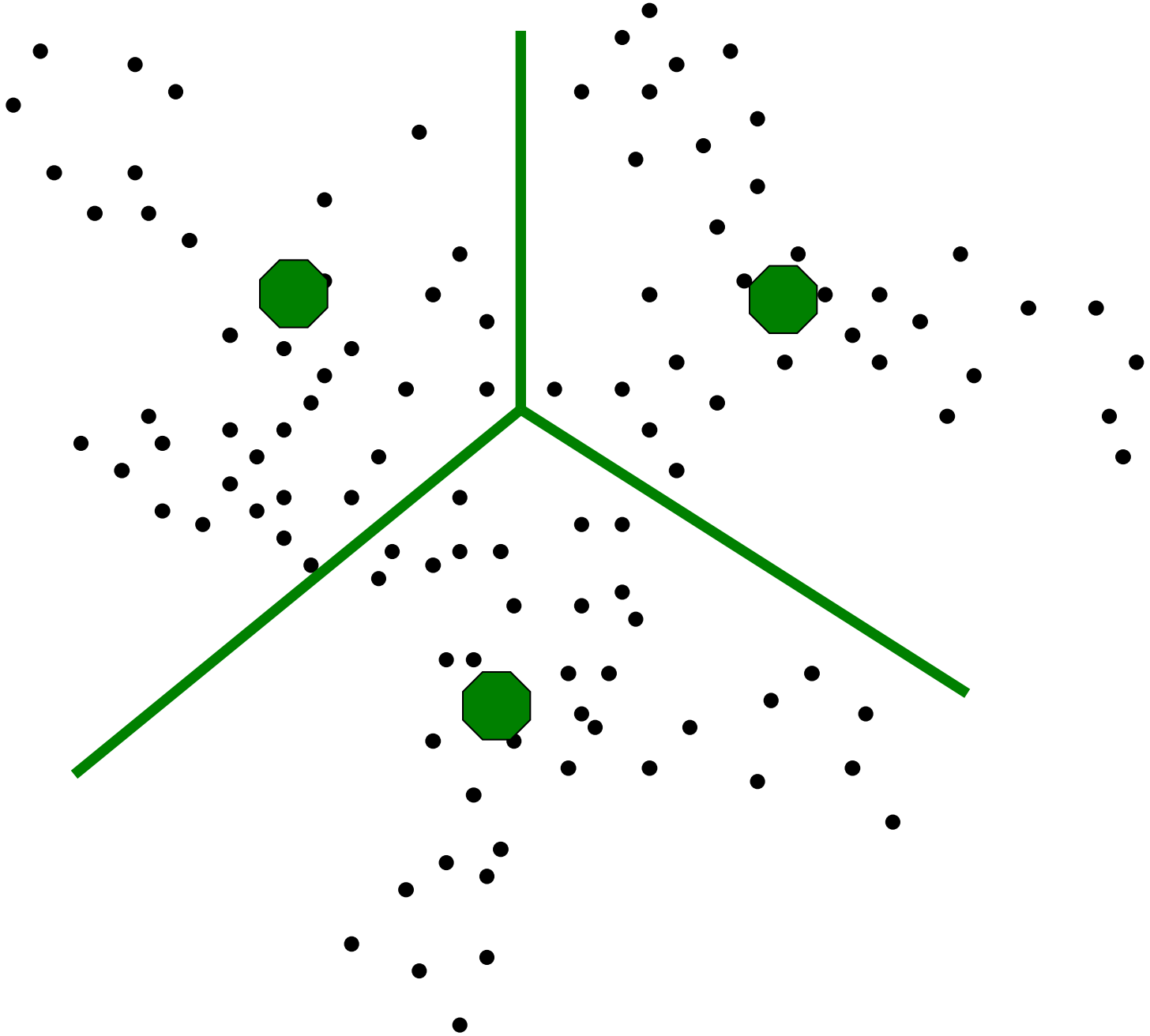
Visual words: main idea



Slide credit: D. Nister



Slide credit: D. Nister



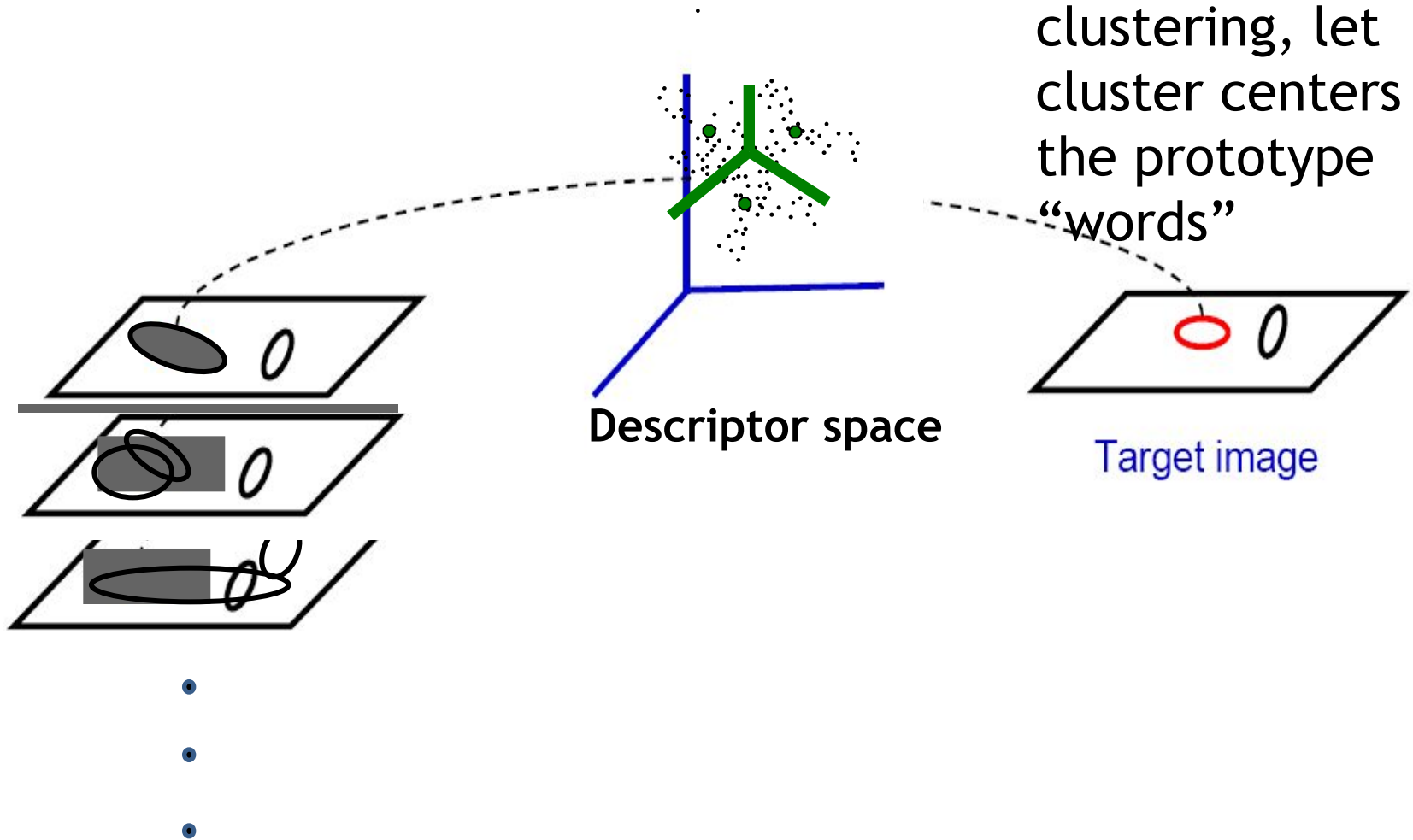
Slide credit: D. Nister

K. Grauman, B. Leibe

Visual words: main idea

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

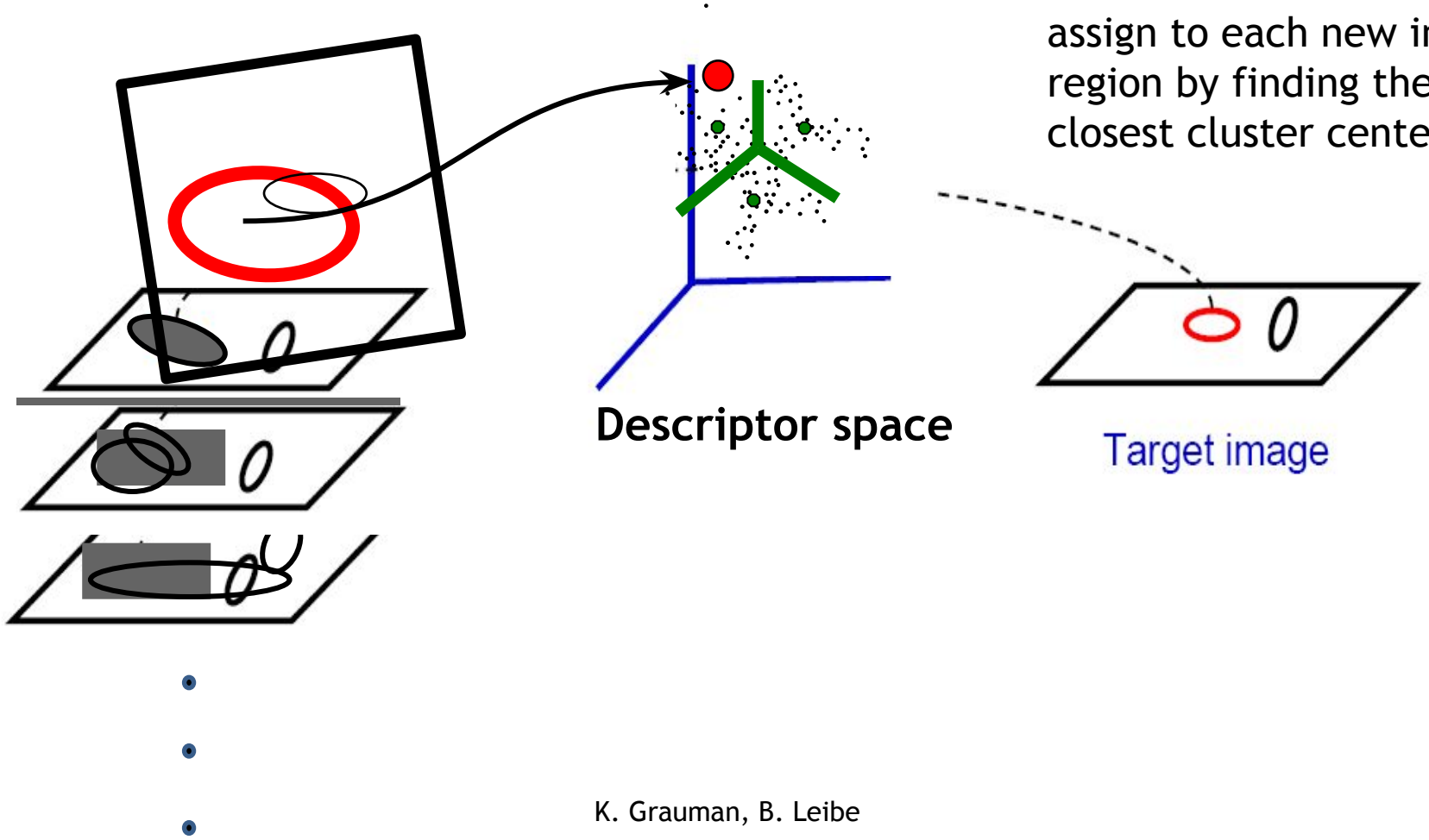
- Quantize via clustering, let cluster centers be the prototype "words"



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

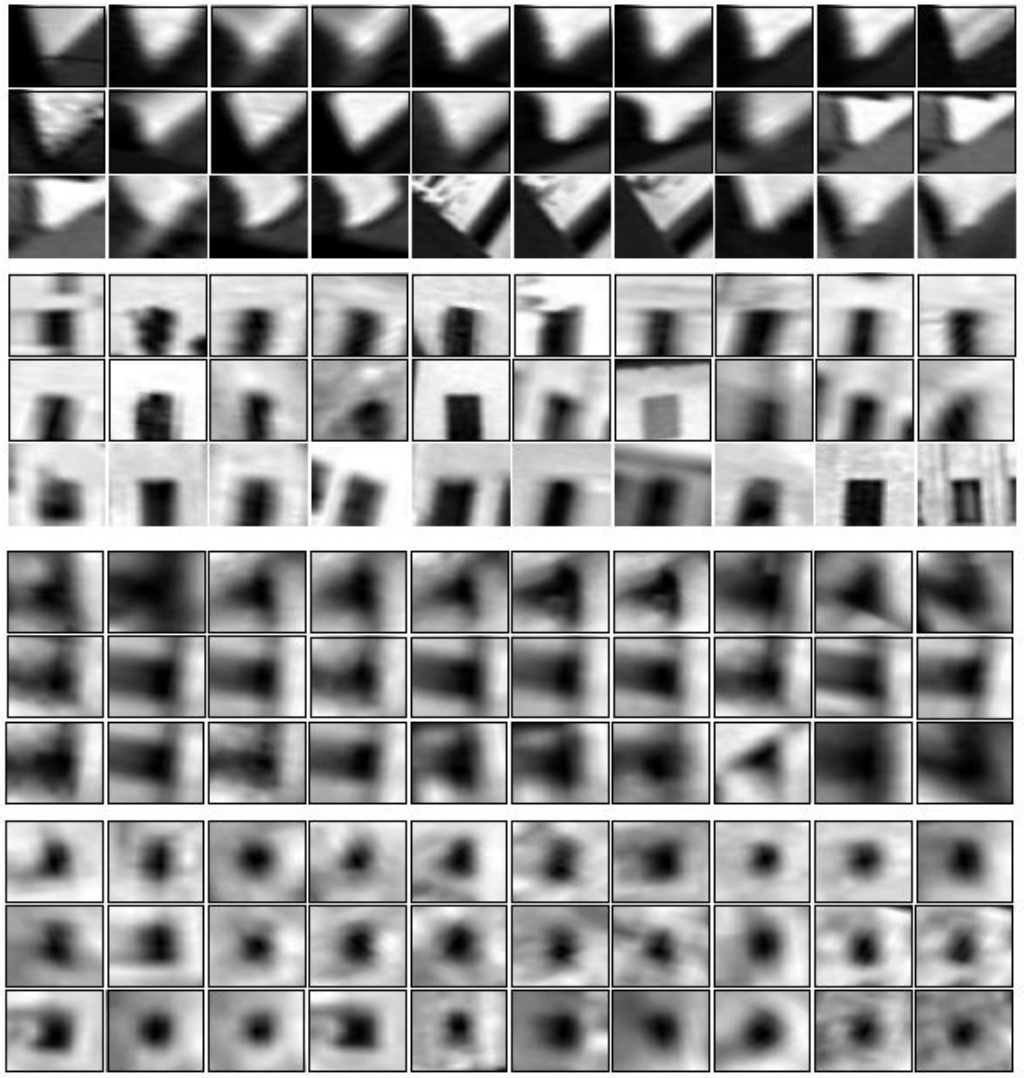
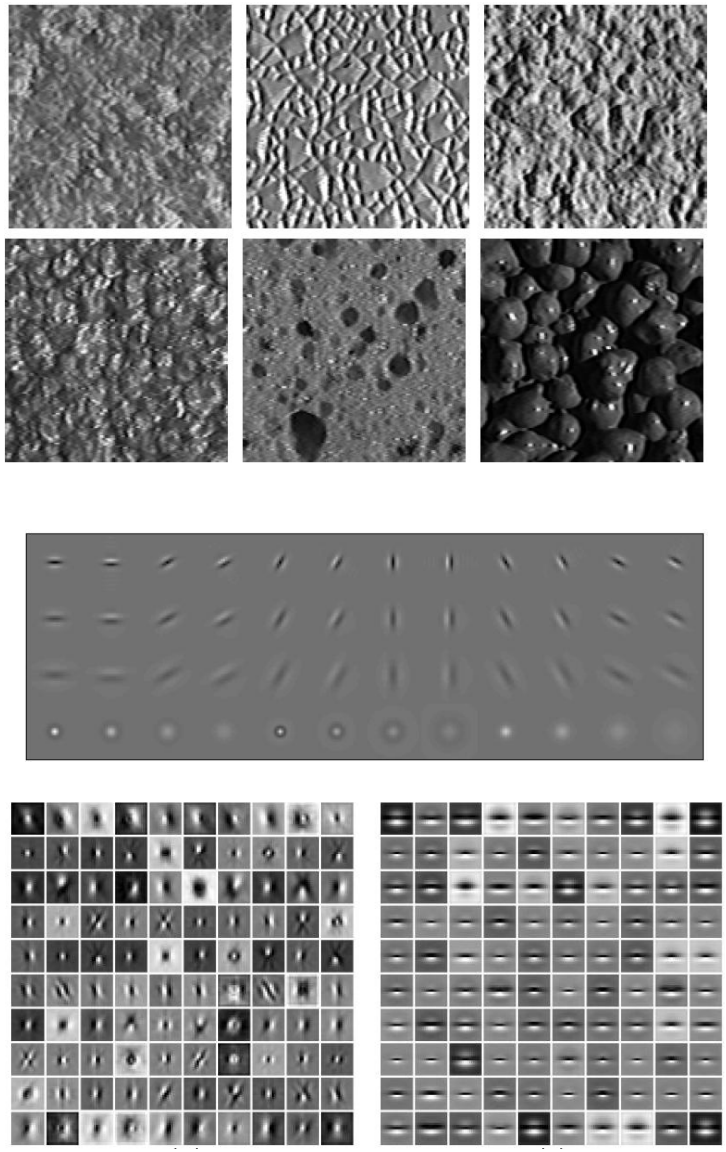


Figure from Sivic & Zisserman, ICCV 2003

Visual words

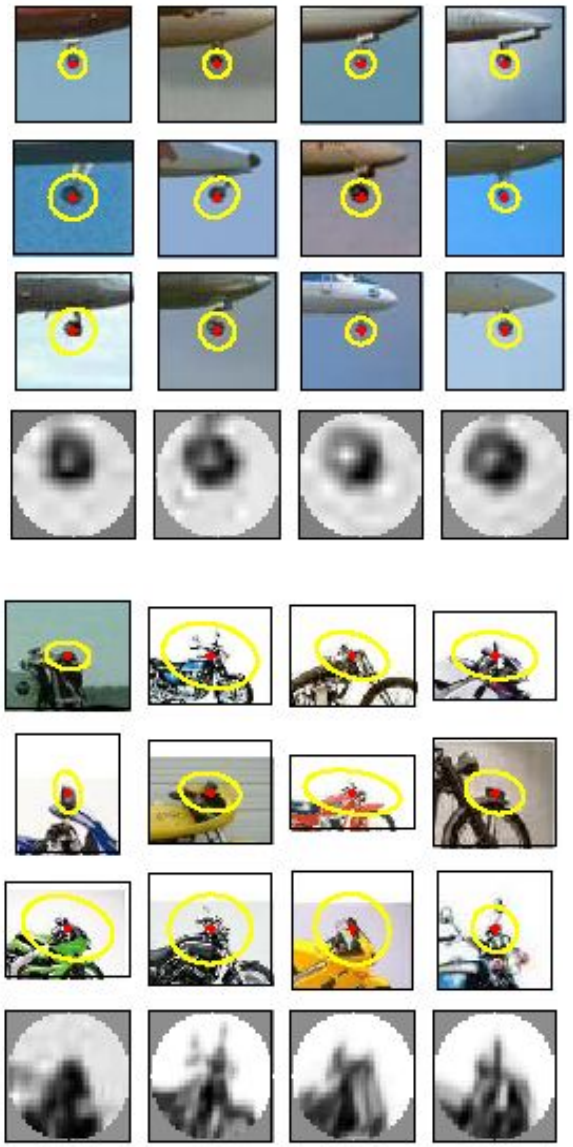
- First explored for texture and material representations
- *Texton* = cluster center of filter responses over collection of images
- Describe textures and materials based on distribution of prototypical texture elements.

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,...
...	...

Image credit: A. Zisserman

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.

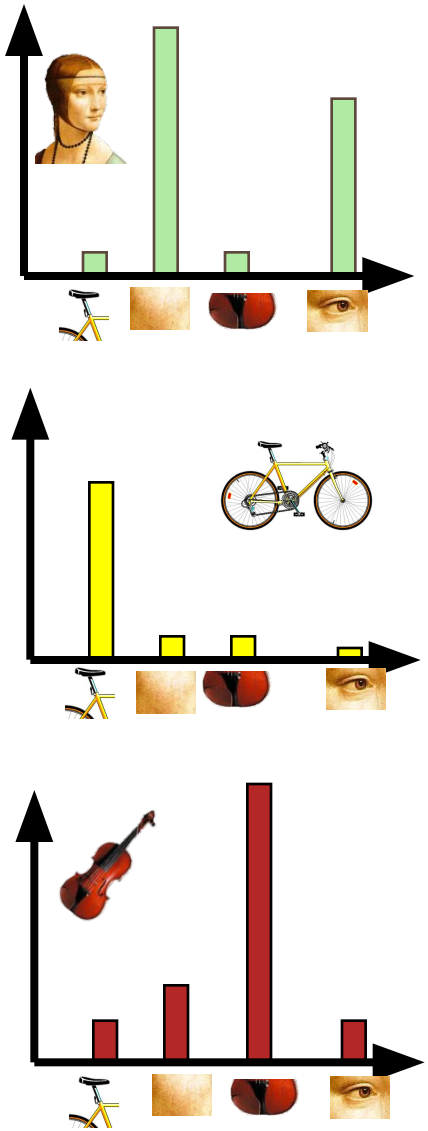
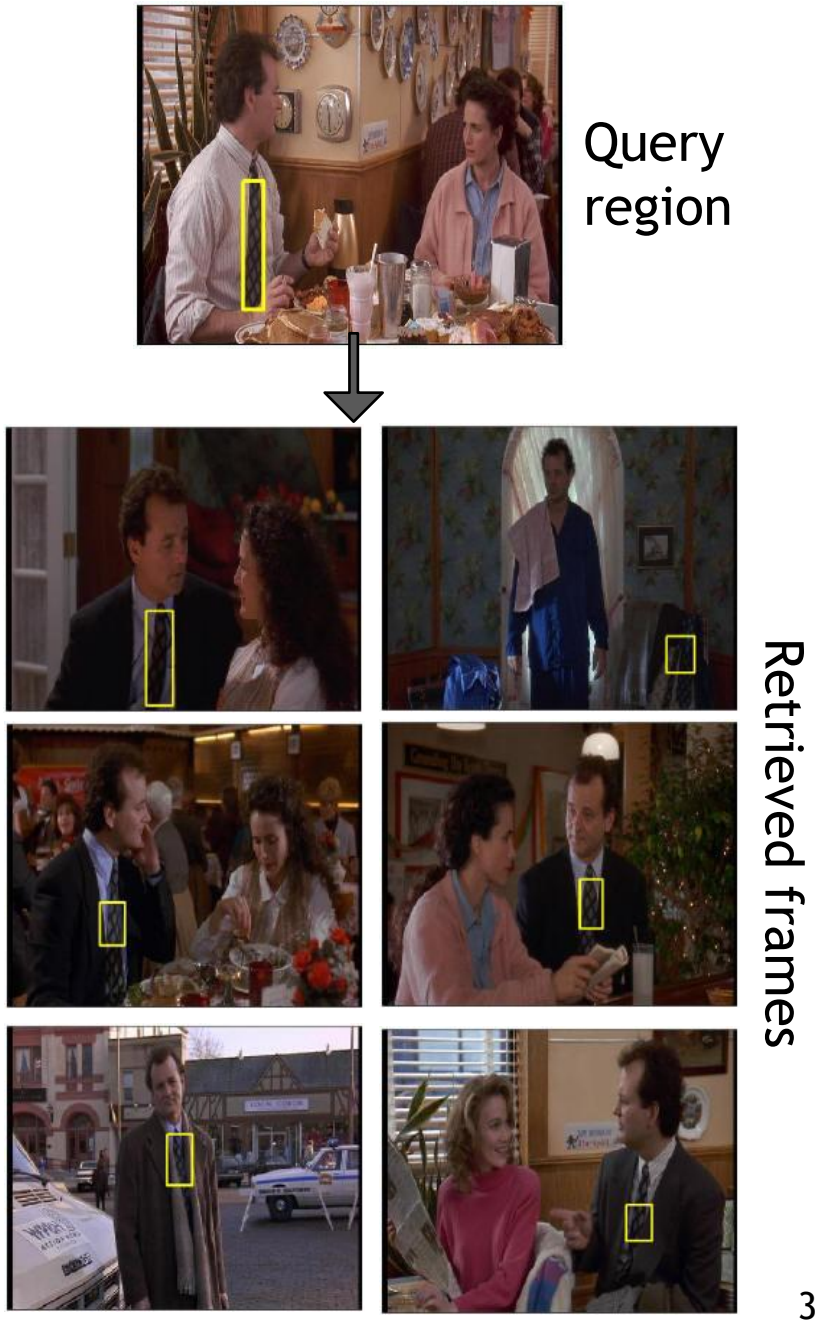


Image credit: Fei-Fei Li

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

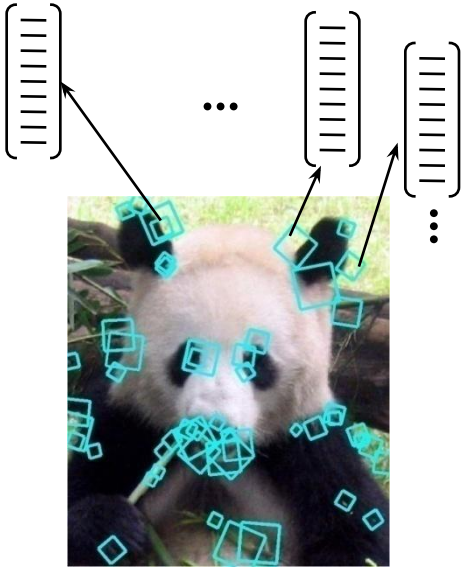
Sivic & Zisserman, ICCV 2003



Basic flow

Detect or sample features

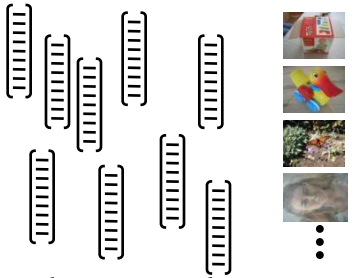
List of positions, scales, orientations



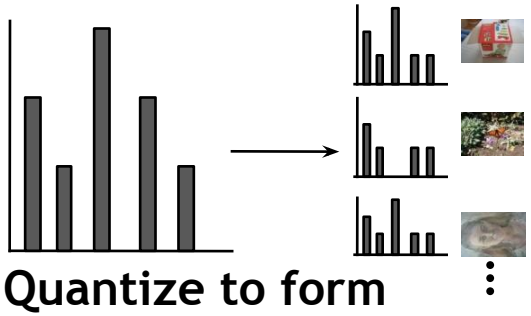
Describe features

Associated list of d-dimensional descriptors

or



Index each one into pool of descriptors from previously seen images



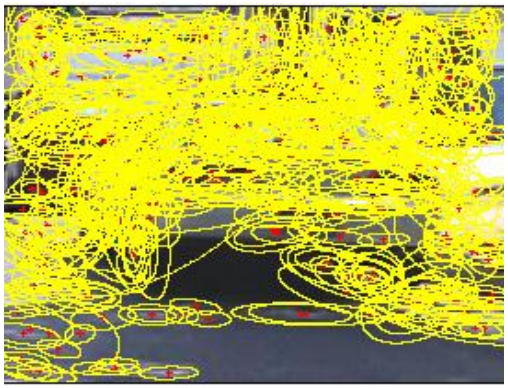
Quantize to form bag of words vector for the image

Visual vocabulary formation

Issues:

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

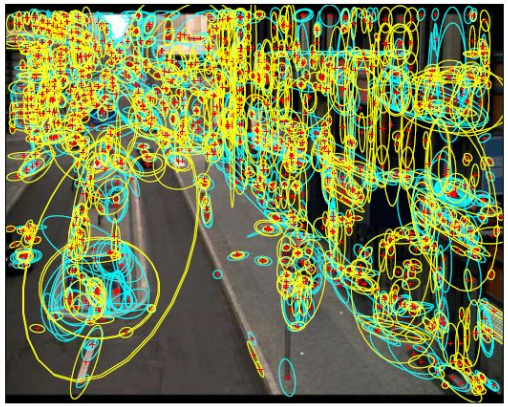
Sampling strategies



Sparse, at interest points

Dense, uniformly

Randomly



Multiple interest operators

- 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]

Image credits: F-F. Li, E. Nowak, J. Sivic

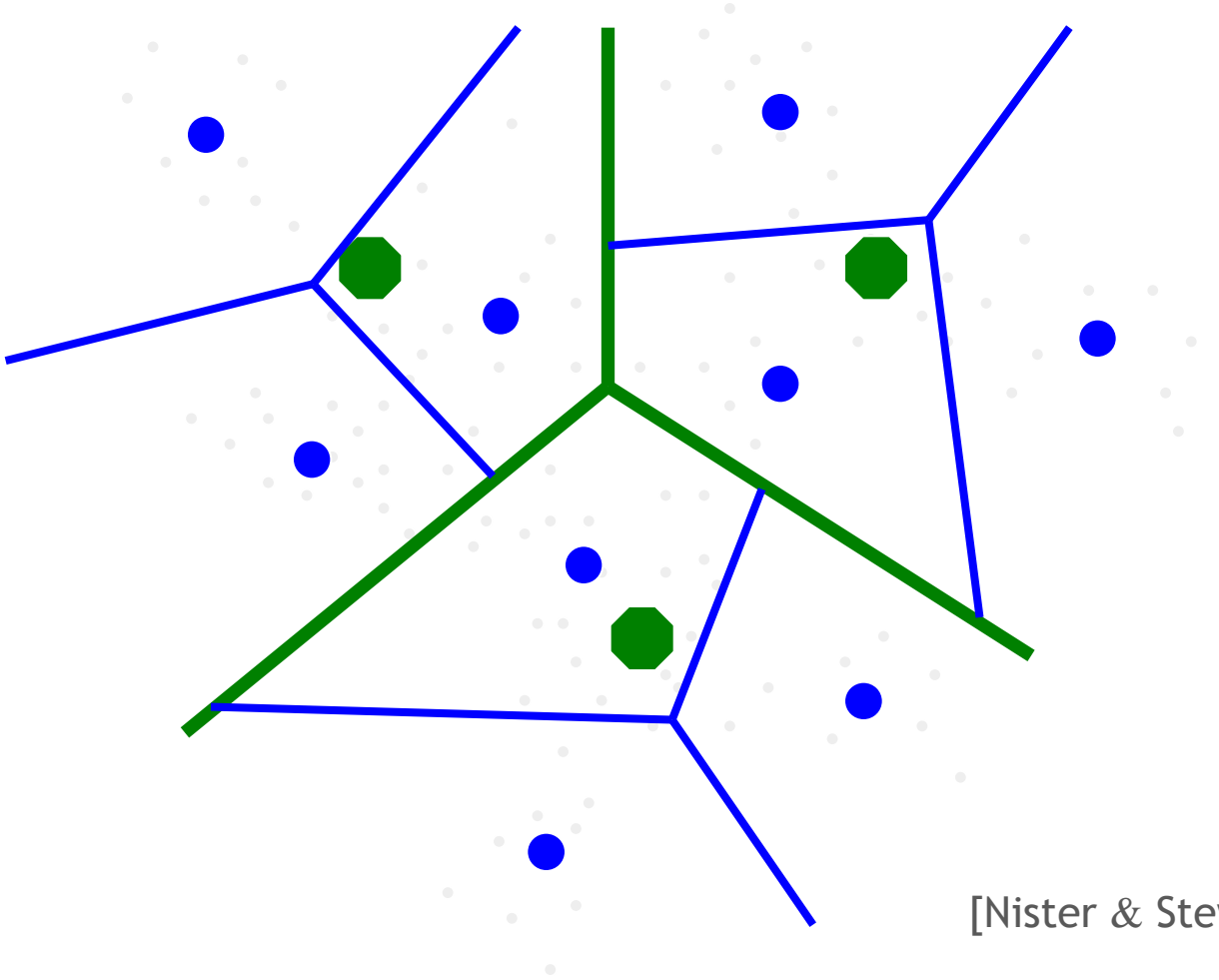
Clustering / quantization methods

- 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:**

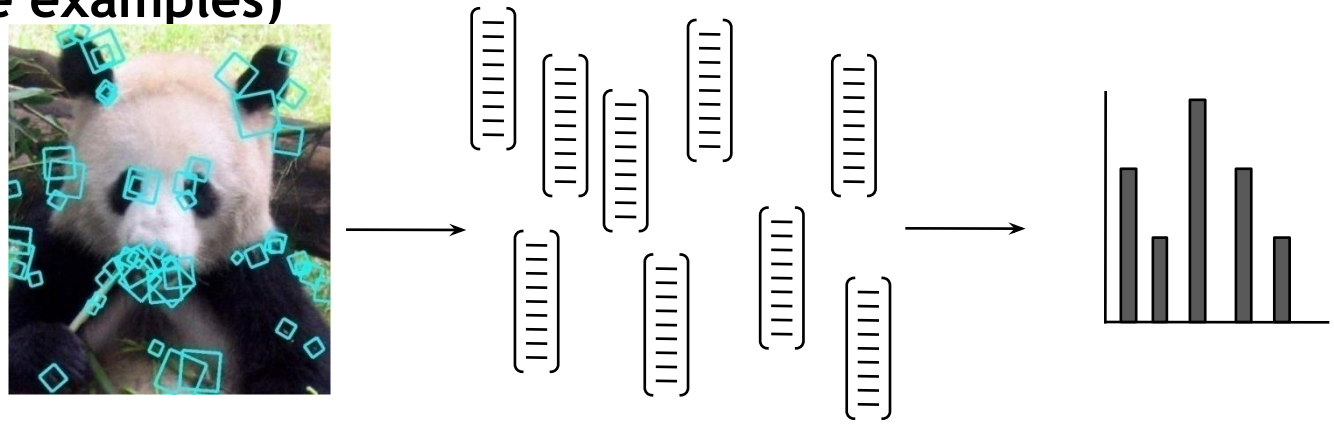


[Nister & Stewenius, CVPR'06]

Slide credit: David Nister

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.

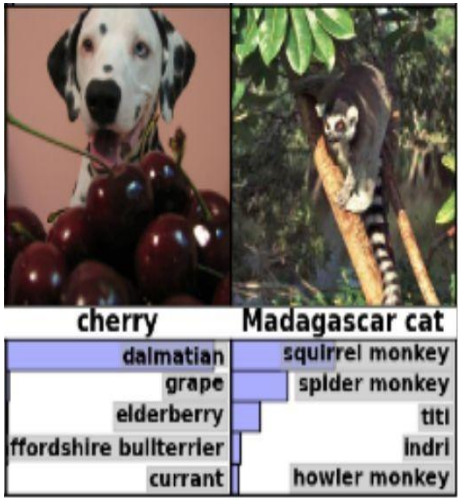
Bags of words: pros and cons

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

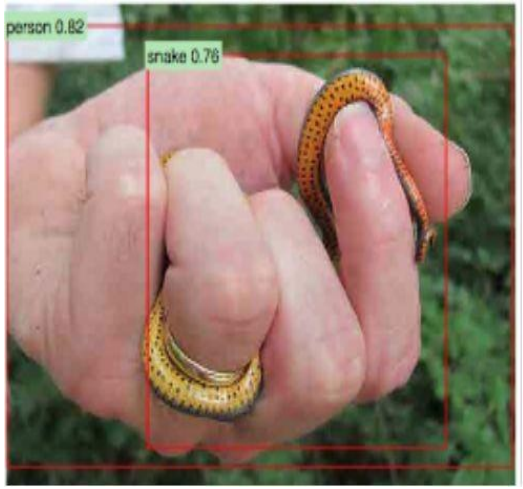
- basic model ignores geometry – must verify afterwards, or encode via features
- background and foreground mixed when bag covers whole image
- interest points or sampling: no guarantee to capture object-level parts
- optimal vocabulary formation remains unclear

Convolutional neural networks and retrieval

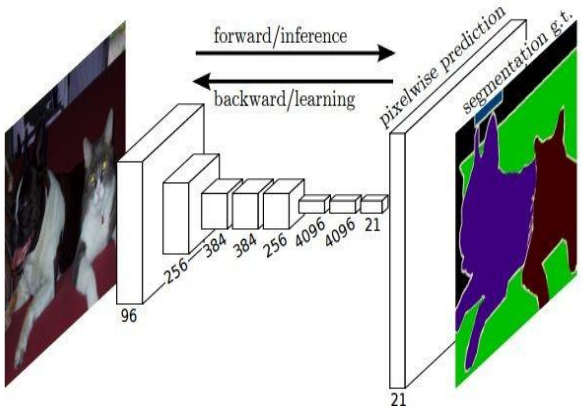
Classification



Object Detection

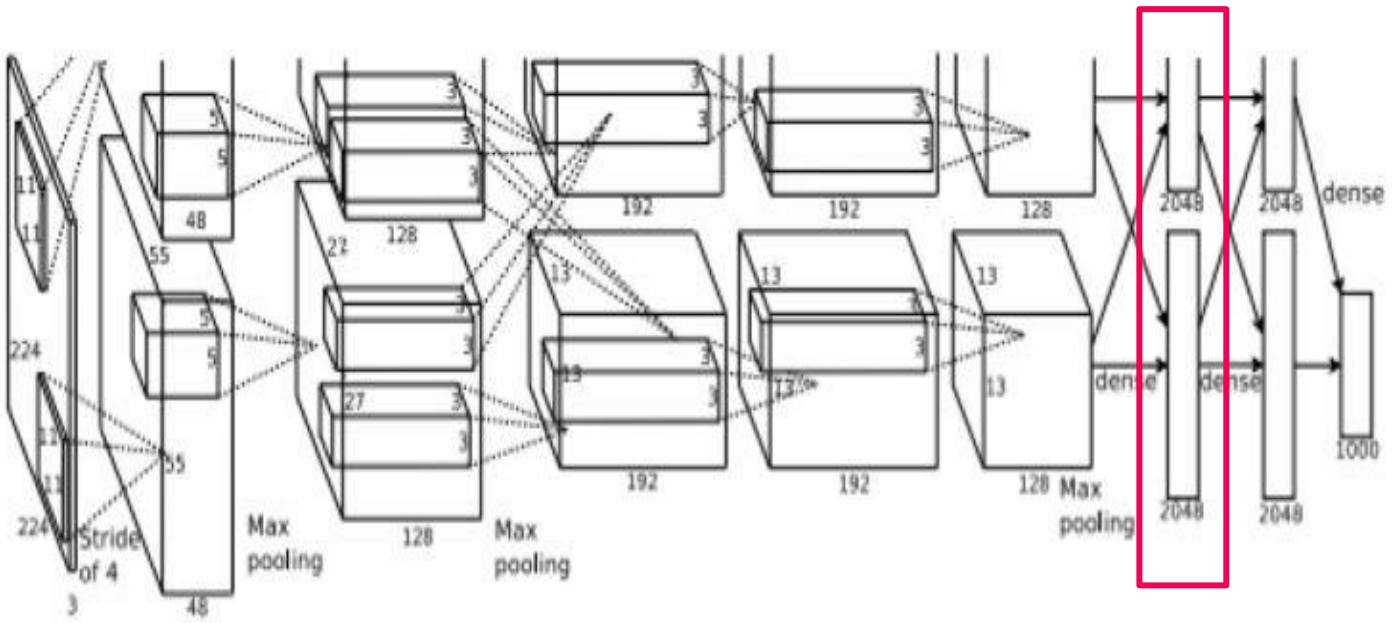


Segmentation



Off-the-shelf CNN representations

FC layers as global feature representation



Razavian et al. CNN features off-the-shelf: an astounding baseline for recognition. In CVPR

Babenko et al. Neural codes for image retrieval. ECCV 2014

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]

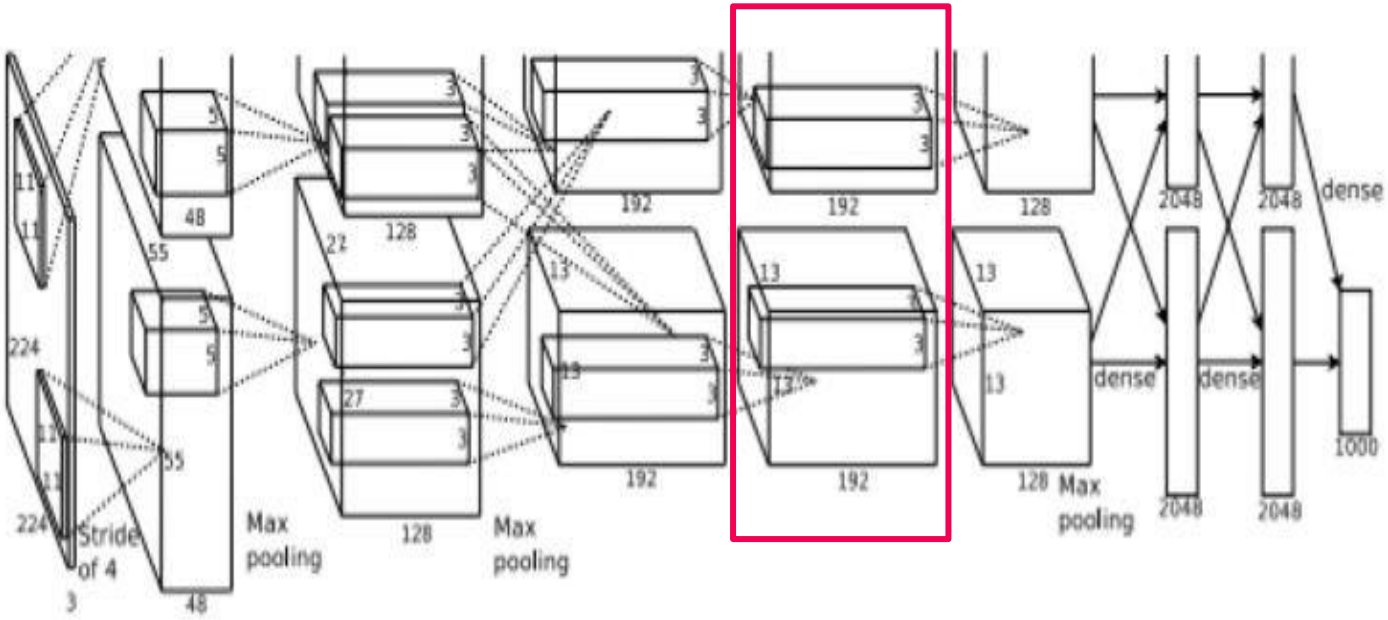
- Extending Babenko's approach with spatial search
- Several features extracted by image (sliding window approach)
- Really good results but too computationally expensive for practical situations

[1] Babenko et al, Neural codes for image retrieval CVPR 2014

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

Off-the-shelf CNN representations

sum/max pool conv features across filters



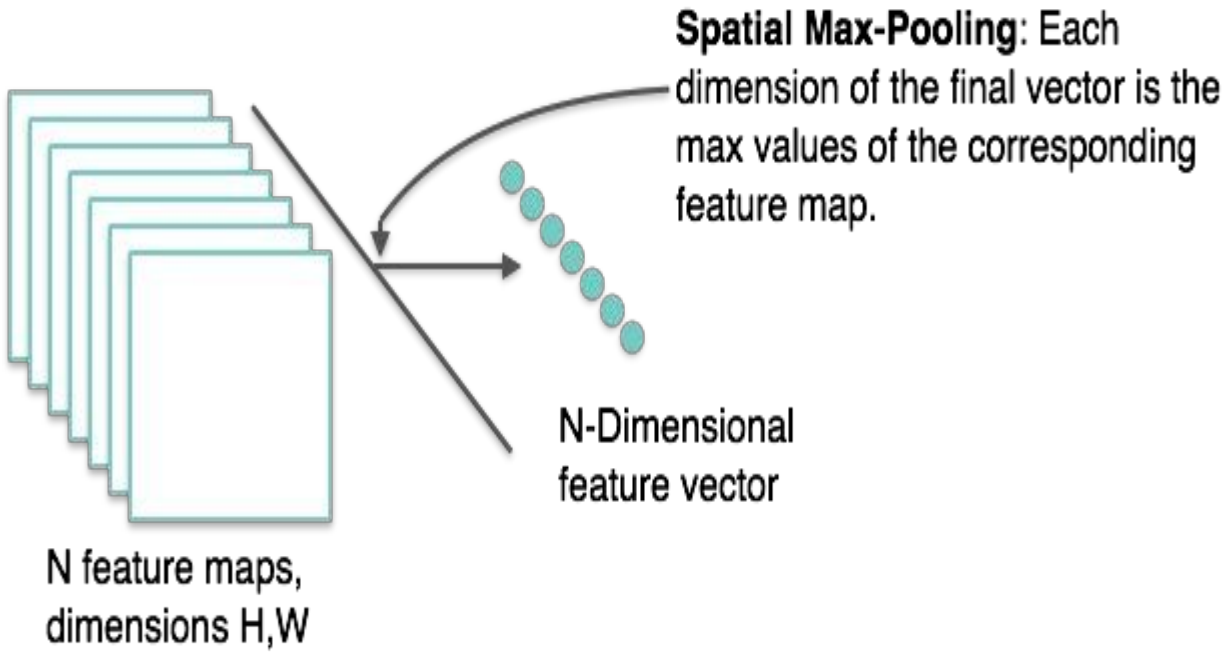
Babenko and Lempitsky, Aggregating local deep features for image retrieval. ICCV 2015

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*.

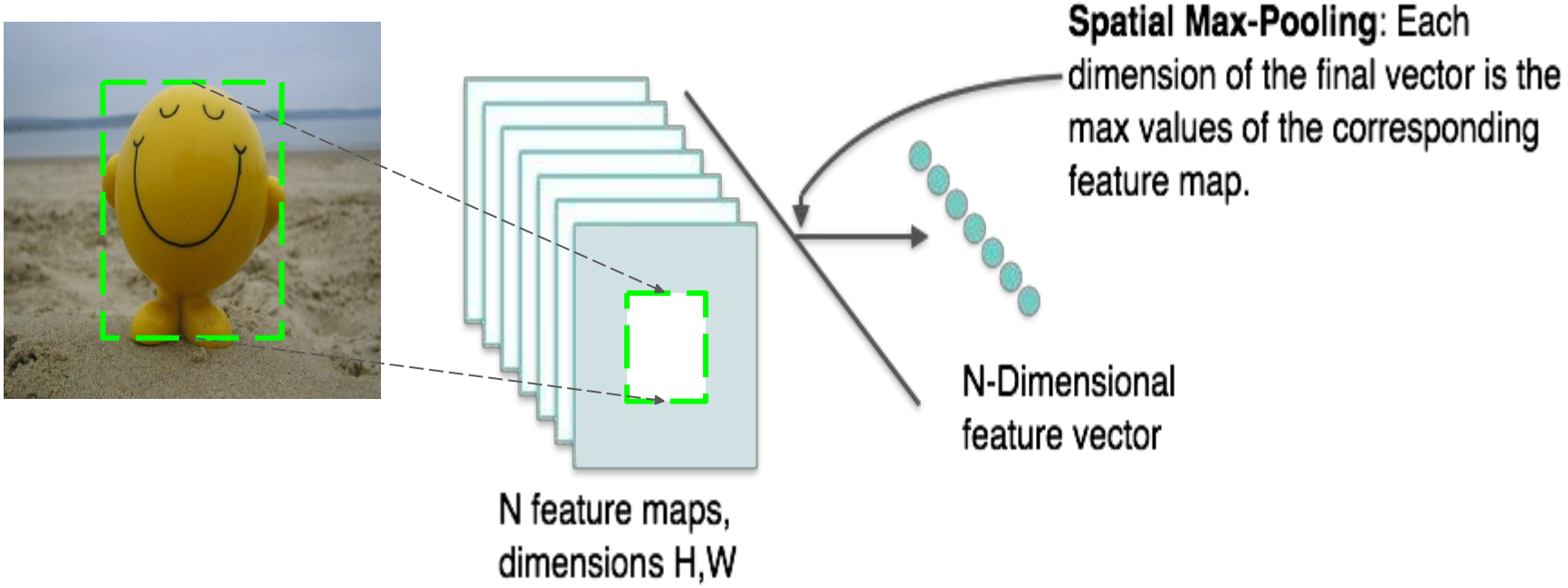
Off-the-shelf CNN representations

Descriptors from convolutional layers



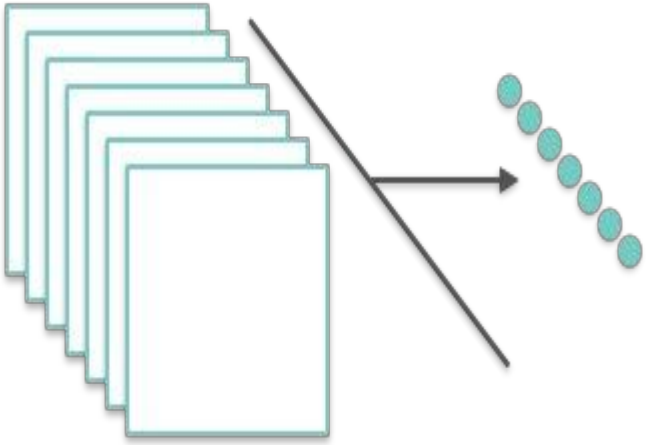
Off-the-shelf CNN representations

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



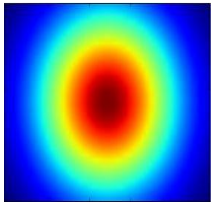
Off-the-shelf CNN representations

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



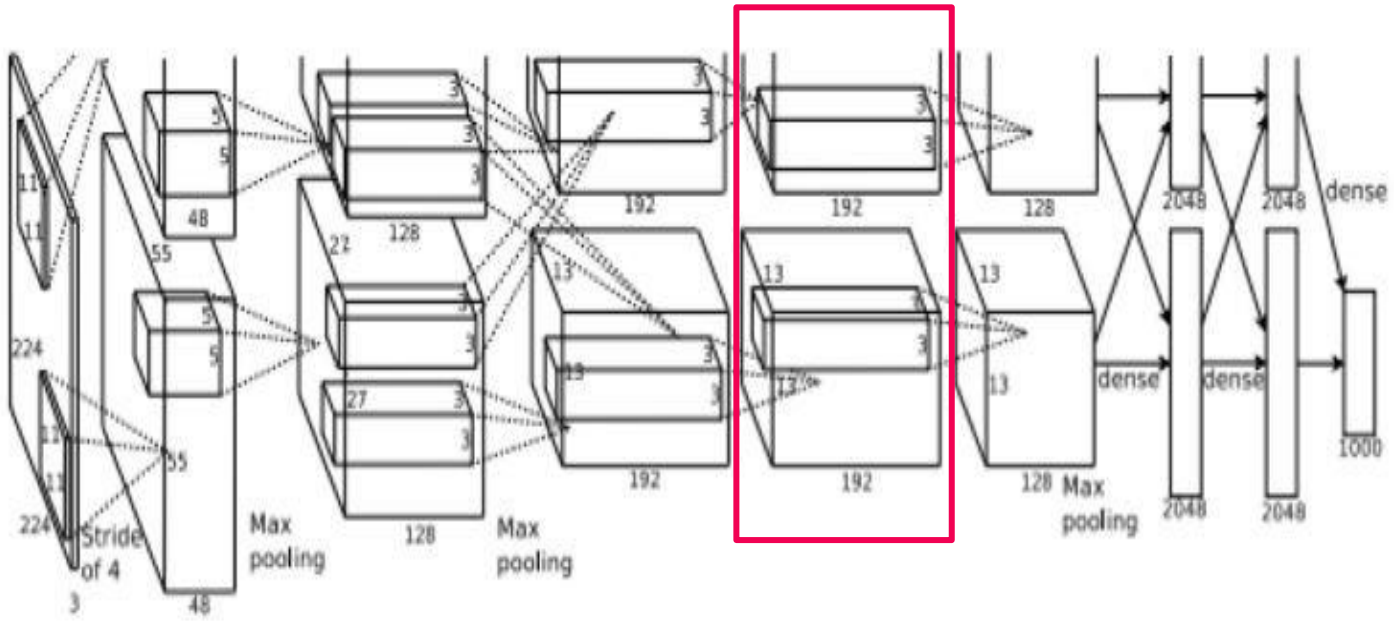
[2] weighting based on the distance to the center of the image

[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

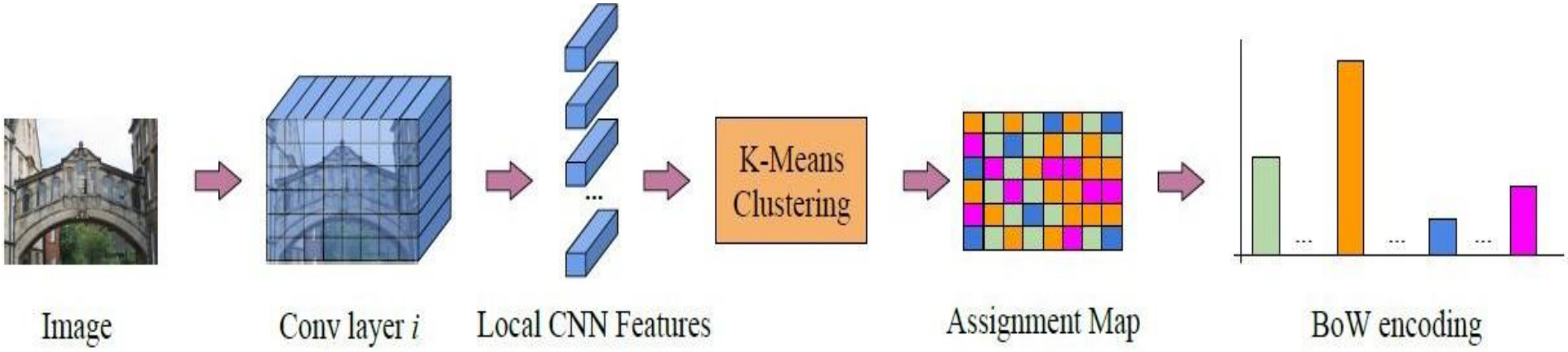


Ng et al. Exploiting local features from deep networks for image retrieval. CVPR Workshops 2015

Mohedano et al. Bags of Local Convolutional Features for Scalable Instance Search. ICMR 2016

Off-the-shelf CNN representations

Descriptors from convolutional layers



(336x256) Resolution

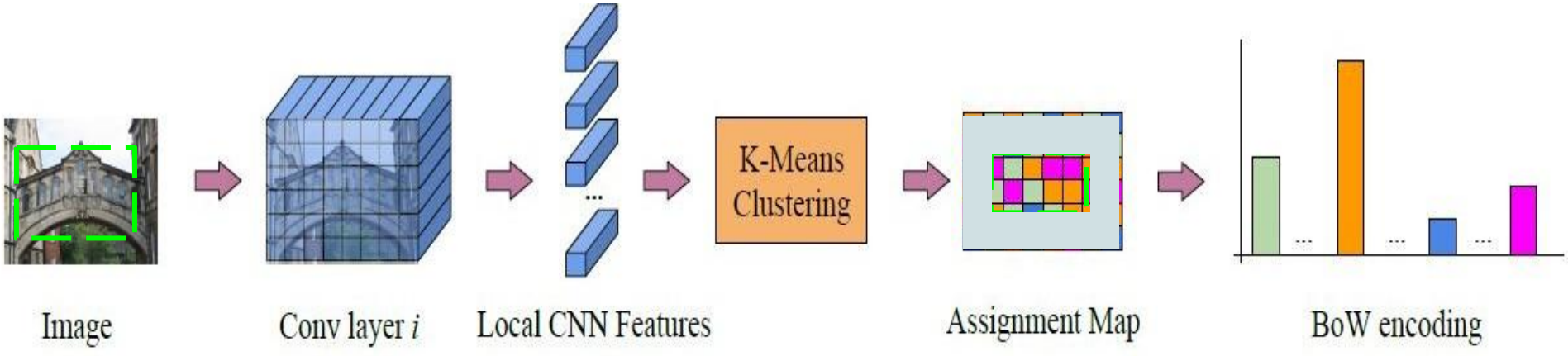
conv5_1 from VGG16 (42x32)

25K centroids

25K-D vector

Mohedano et al. Bags of Local Convolutional Features for Scalable Instance Search . ICMR 2016

Off-the-shelf CNN representations



(336x256)
Resolution

conv5_1
from
VGG16
(42x32)

25K centroids

25K-D vector

CNN as a feature extractor

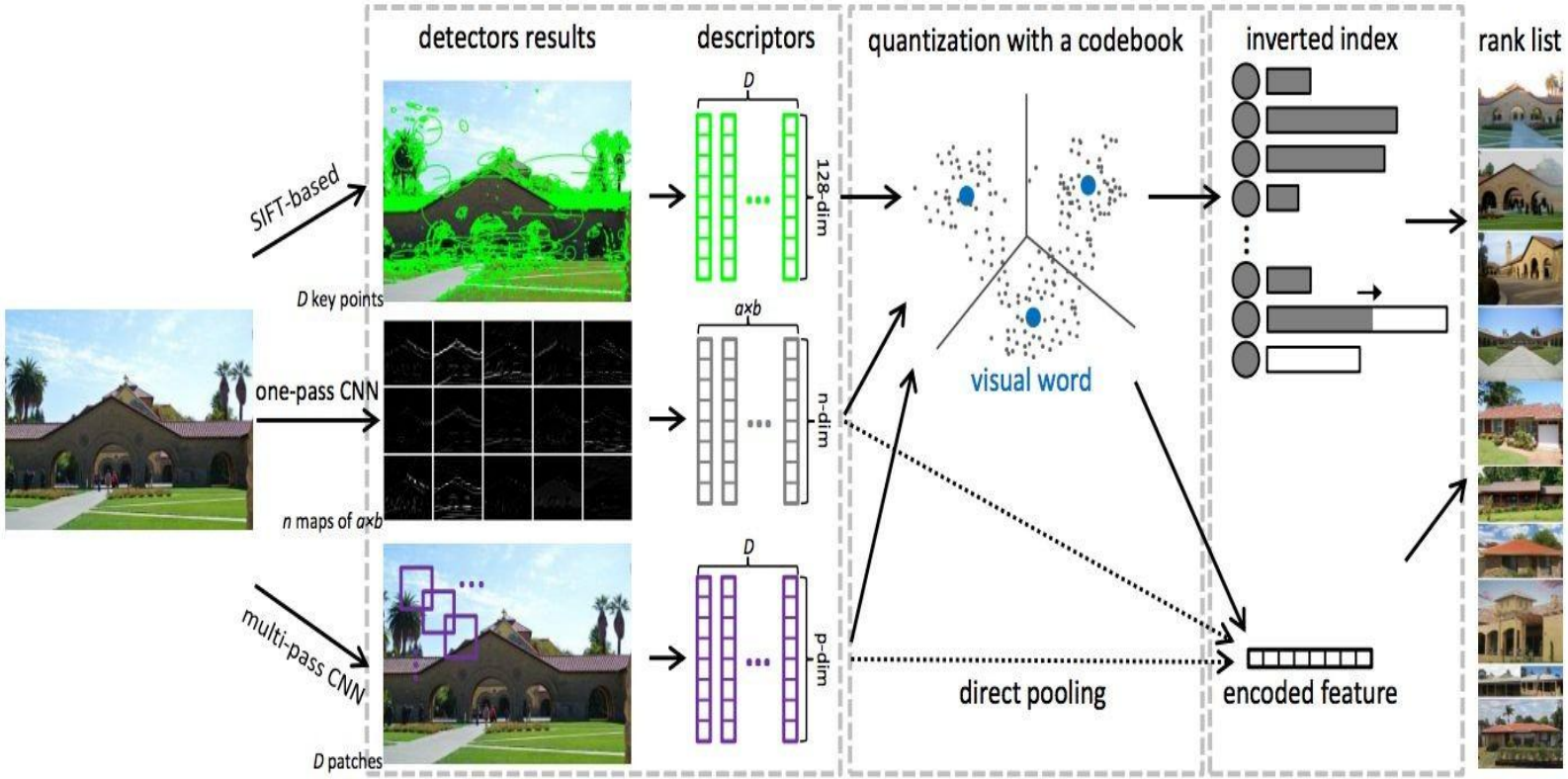


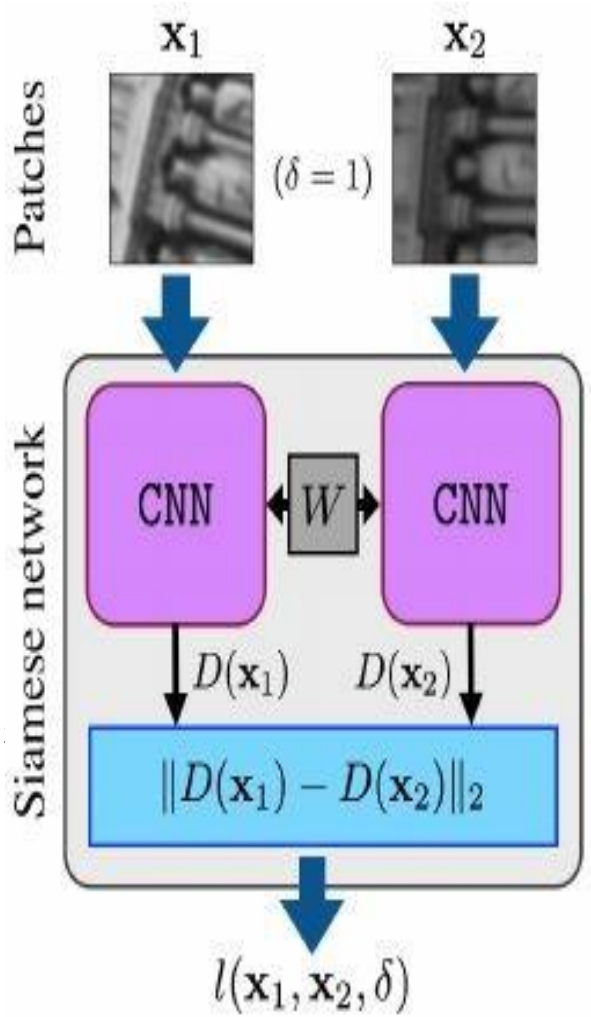
Diagram from SIFT Meets CNN: A Decade Survey of Instance Retrieval

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.

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

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

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

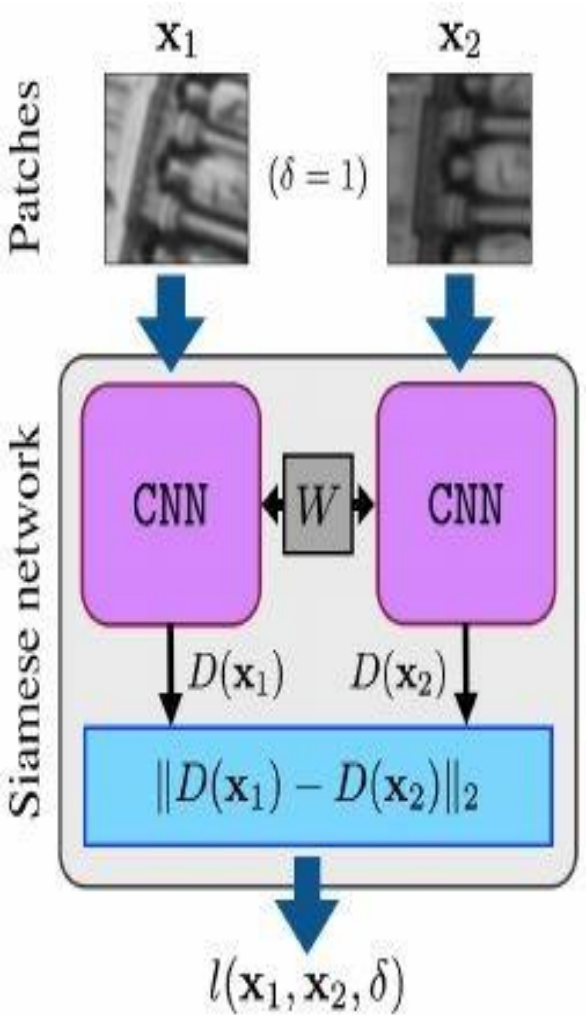
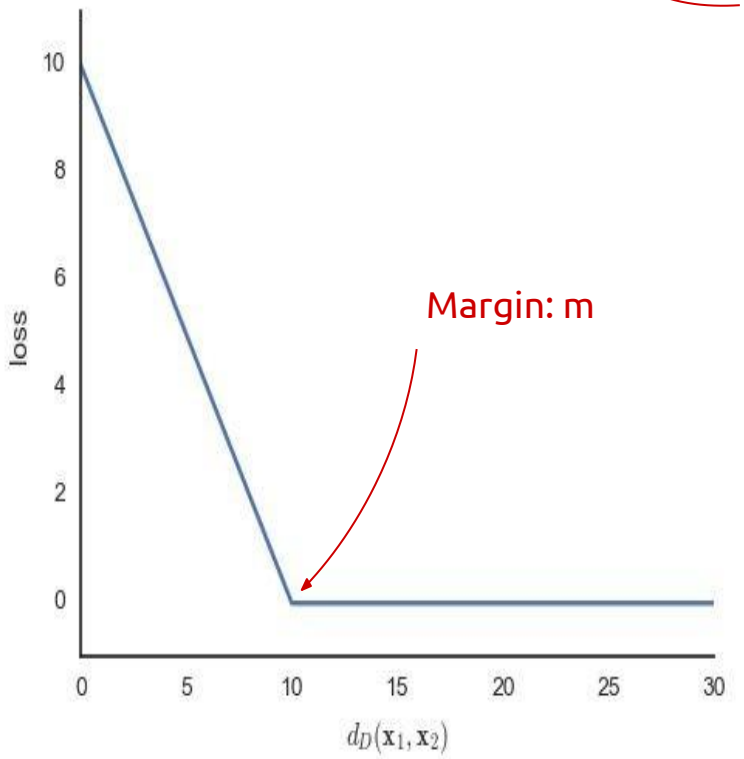


Image credit: Simo-Serra et al. Fracking deep convolutional image descriptors. CoRR 2014

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

Hinge loss

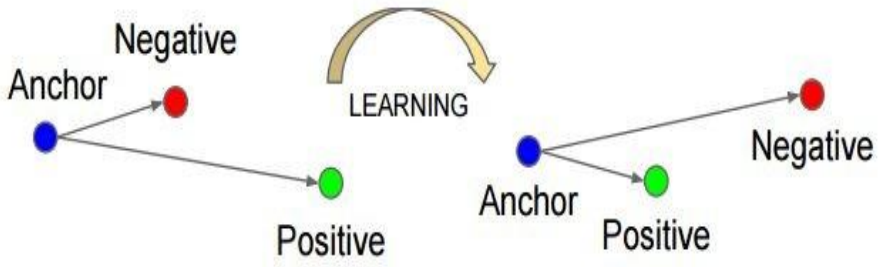
Negative pairs: if nearer than the margin, pay a linear penalty



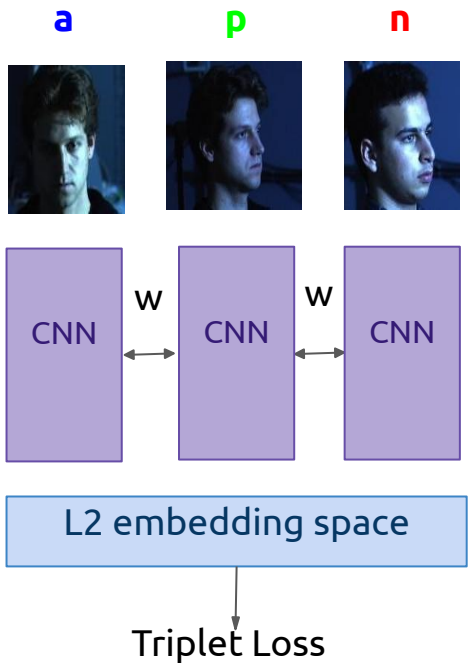
The margin is a **hyperparameter**

Learning representations for retrieval

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



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

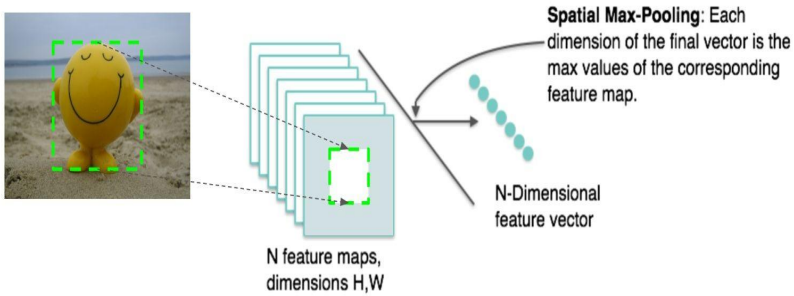


Schroff et al. FaceNet: A Unified Embedding for Face Recognition and Clustering, CVPR 2015

Learning representations for retrieval

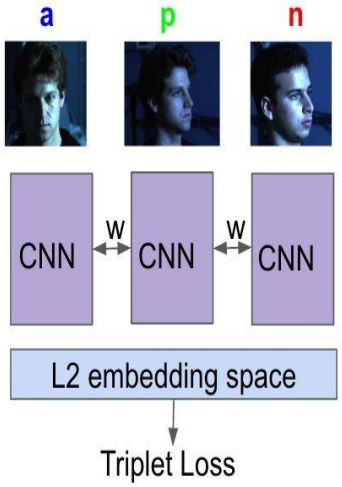
Deep Image Retrieval: Learning global representations for image search, Gordo A. et al. Xerox Research Centre, 2016

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

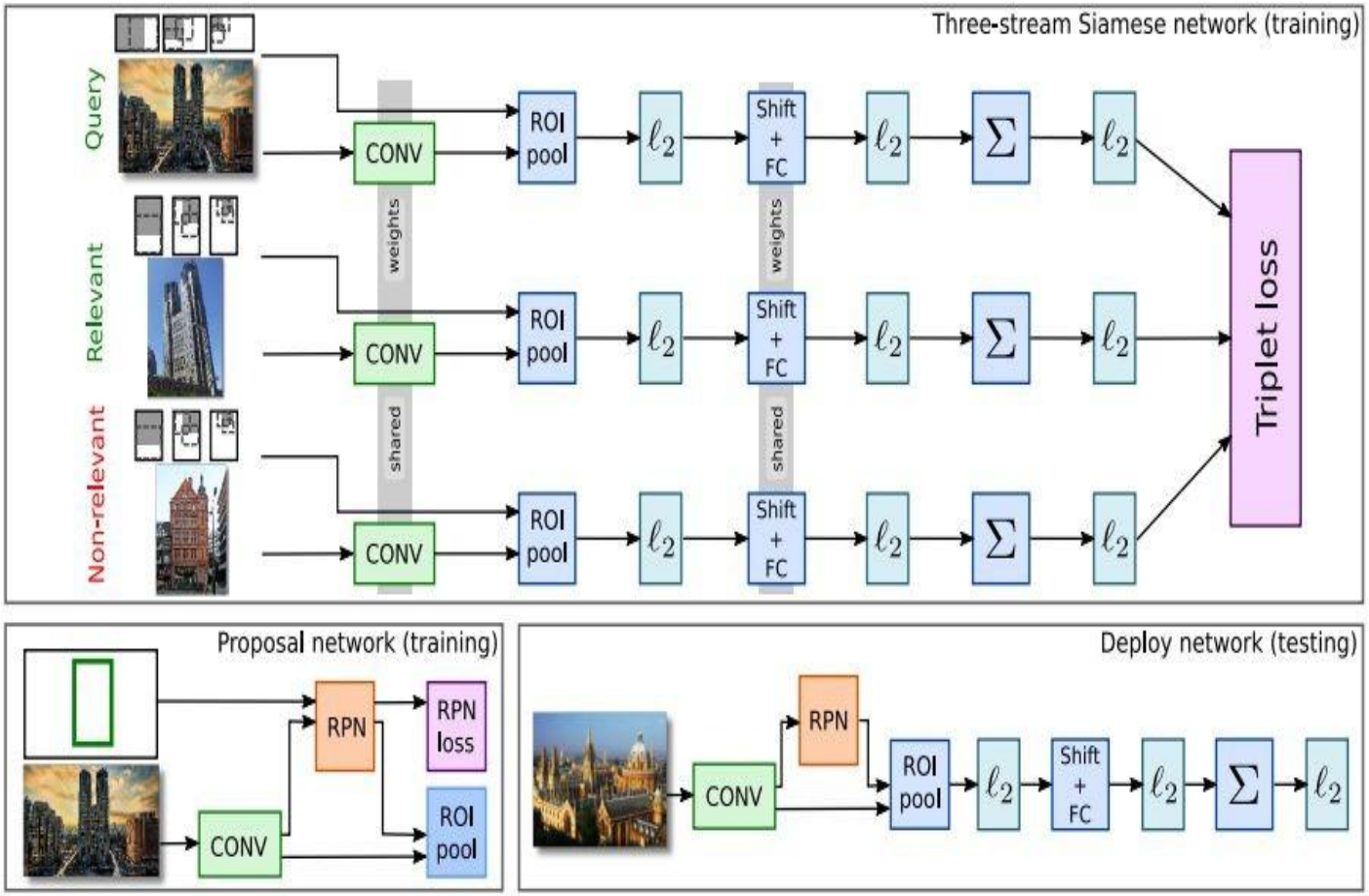


$$L(I_q, I^+, I^-) = \max(0, m + q^T d^- - q^T d^+)$$

- 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?