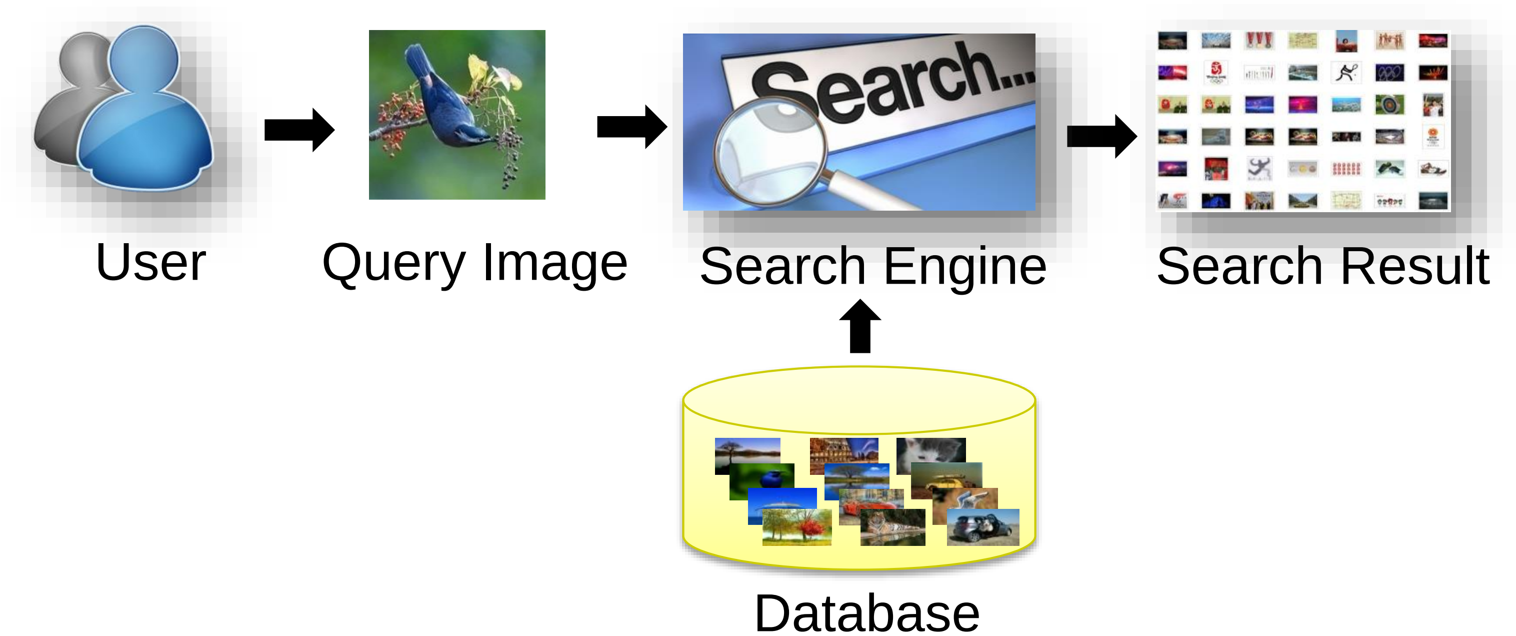
kNN Search and Advanced Visual Features

Contents

- kNN Image Search
- Intro to Advanced Visual Features
  - Visual Keywords
  - Face Detector
  - Visual Concept Detector
- Summary

Introduction

- Pipeline of Content-based Image Retrieval

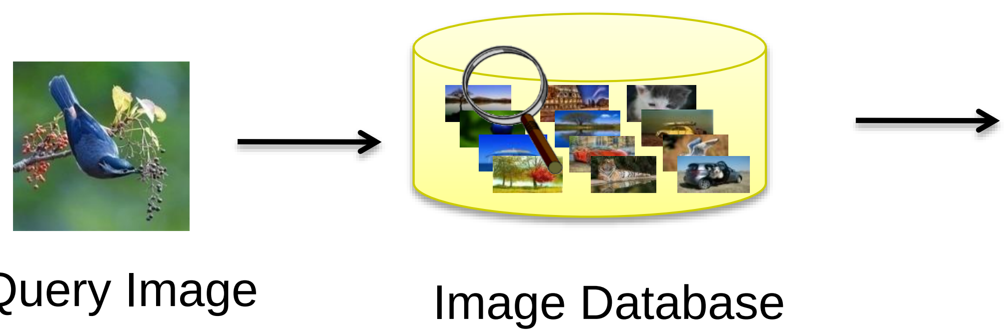


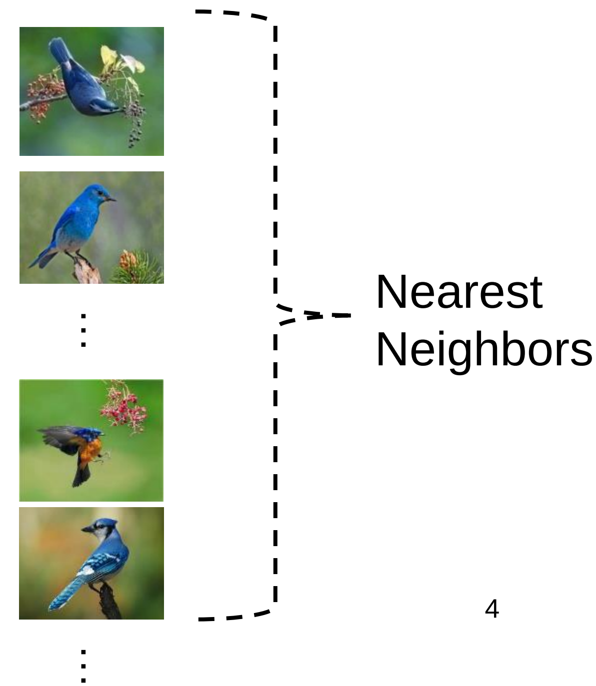
Content-based Image Retrieval

- Given an image query:
  - Feature extraction: color, edge direction, feature point etc
  - Similarity matching:

$$Sim(\underline{Q},\underline{D}_i)=\sum_{j}^{m}\alpha_j Sim(\underline{Q}^j,\underline{D}_i^j),\ \ \sum_{j}\alpha_j=1;\ \ j=1,..m\ features$$

- Return the top k matches (or k nearest neighbors):





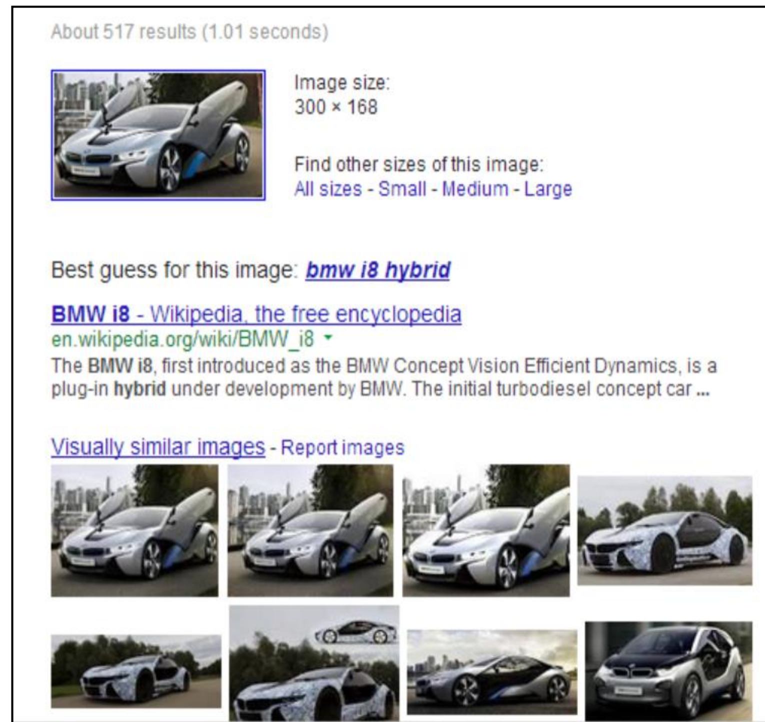
kNN Image Classification

- Image classification task is:
  given an image, does it have label y?

y = “car”

y = “road”, “cityscape”??

- Approach:
  - Given that we have a large set of images with correct labels, such as cars, bags, shoes, faces,
  - Given a new image, perform CBIR to return the top k image results
  - Use the result to vote on its label
  - If kNN list contains the label “car”, then “car” is assigned..
  - We can also assign labels like “bmw i8” if it appears in top results



Nearest Neighbor Classification

- kNN is one of the simplest approach to inductive learning:
  - Save each training example as a point in feature space
  - Then classify a new example, q, by giving it the same classification (+ or -) as its nearest neighbor in Feature Space

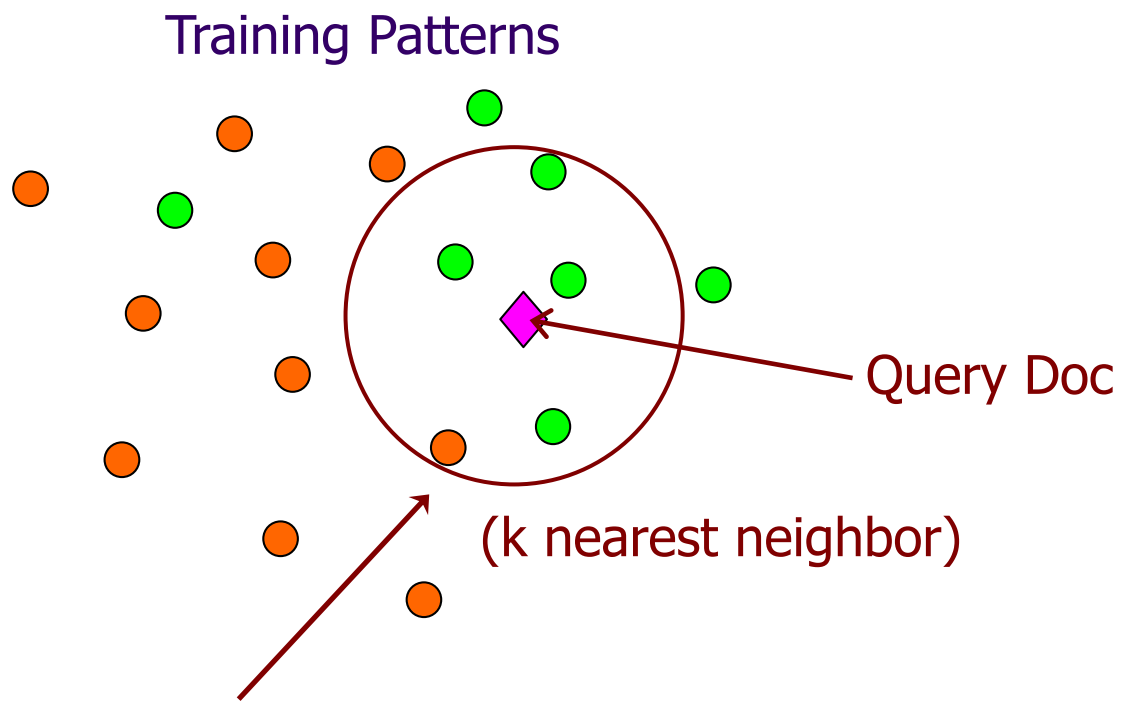
  or

$$\arg\max_i Sim(\underline{q}, \underline{x}_i), i = 1, \ldots k_{samples}$$

  then classification of q is the same as that of $\underline{x}_i$

kNN Classifier -1

- How it works graphically (for two class problem):



Basic idea: Use the types and their similarities to Q as basis for classification

Hence: New doc belongs to green category since majority of its NNs is of type green

kNN Classifier -2
The Algorithm

- The decision rule for kNN can be written as:

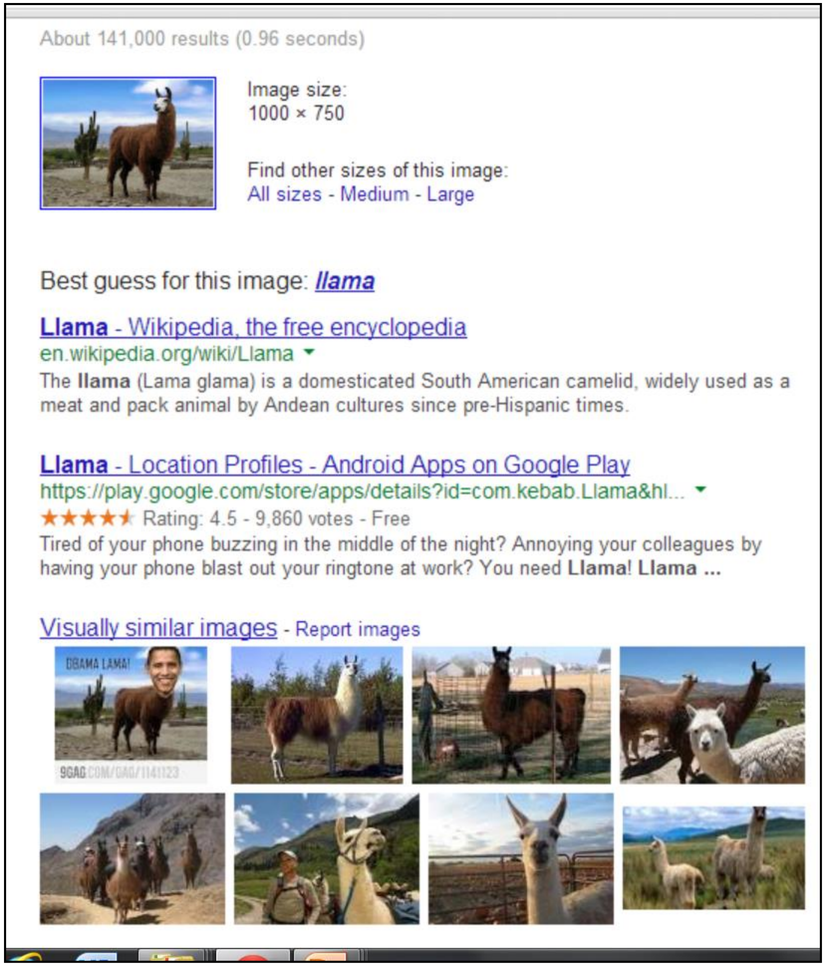
$$y(\underline{x}, c_j) = \sum_{d_i \in kNN} \{Sim(\underline{x}, \underline{d}_i) * y(\underline{d}_i, c_j)\} - b_j$$

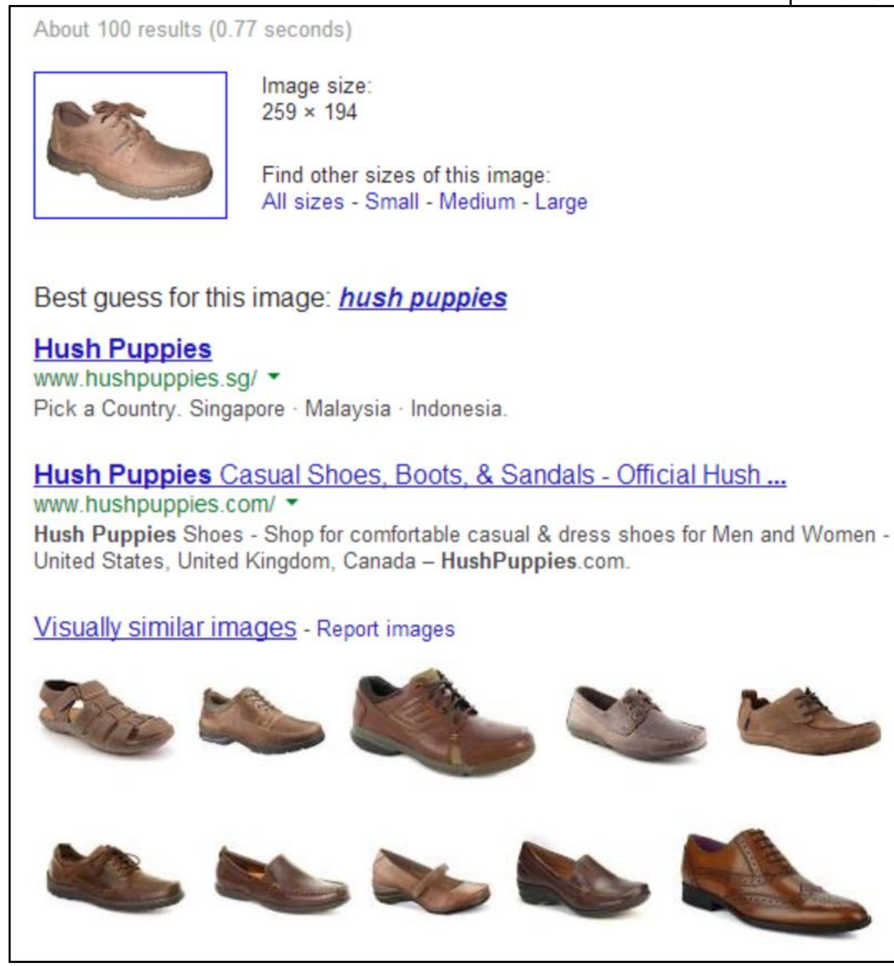
where b_j is the category-specific threshold for n-category classification problem

- kNN is an online classifier:
  - Does not perform off-line learning
  - “Remembers” every training samples
  - Hence it is also called Instance-based Learning
  - Can be inefficient during classification
  - Tend to be very effective for large training samples – provided we can tackle efficiency problem. Reasons??

Image Classification -2

- Other examples:





kNN Image Search & Classification

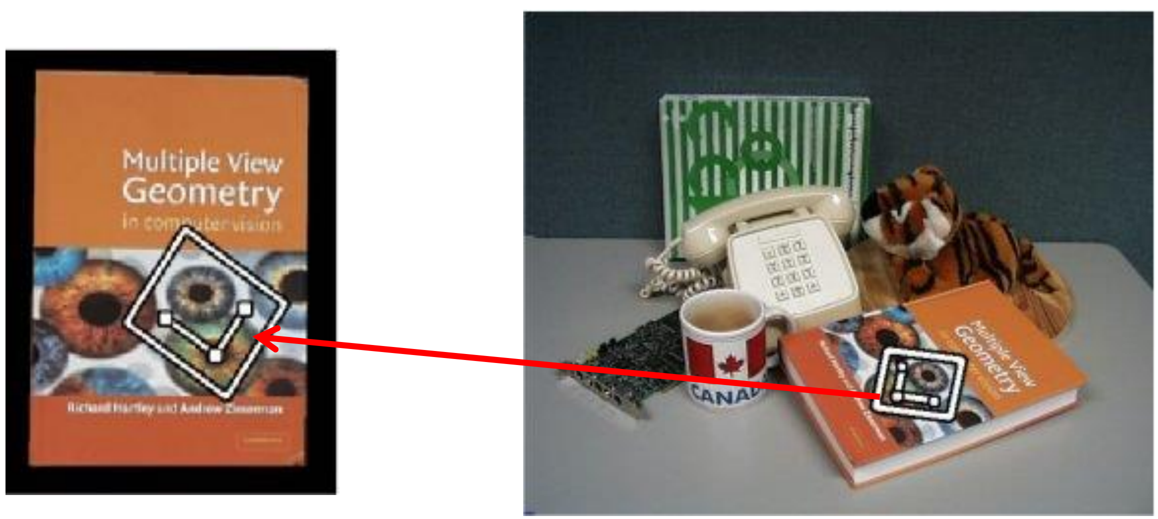
- Effective for large-scale image dataset with correct labels
- Out-performs almost all sophisticated machine learning methods
- Key Lesson:
  - At large scale (at 100s of million to billion size), simple robust methods work best
  - Also these results are achievable with the use of more advanced visual features, like local interest points (visual keywords), specialized detectors like face, and visual concept detector.
  - We will (briefly) introduce these advanced features next

Contents

- kNN Image Search
- Intro to Advanced Visual Features
  - Visual Keywords
  - Face Detector
  - Visual Concept Detector
- Summary

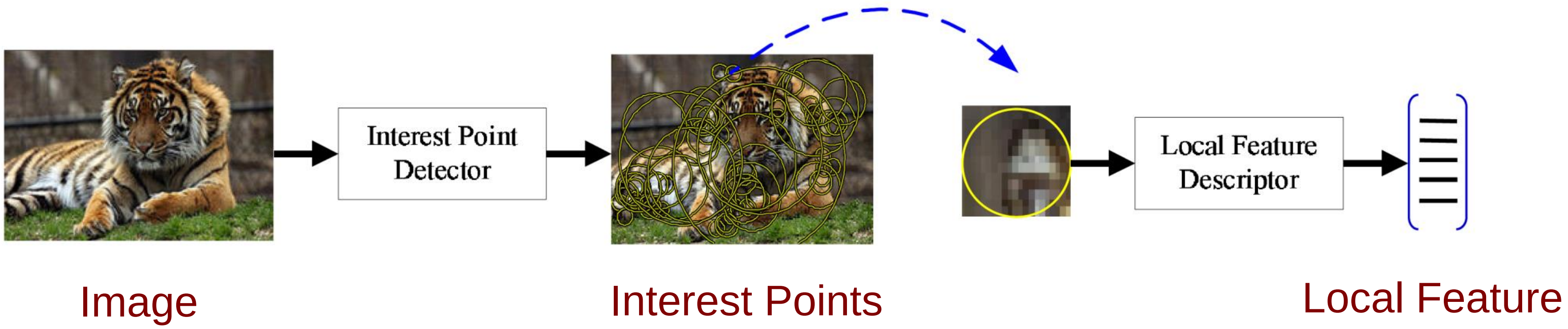
Local Interest Points -1

- What is a good local interest point?
  - Must be highly distinctive
  - Should be Invariant to:
    - Uniform scaling
    - Rotation
    - Changes in illumination
    - Minor changes in view direction
    - Image noise



Local Interest Points -2

- Main Components:
  1) Detection of local interest points
  2) Local Feature Descriptor (descriptor for effective matching)



Examples of Local Interest Points -1

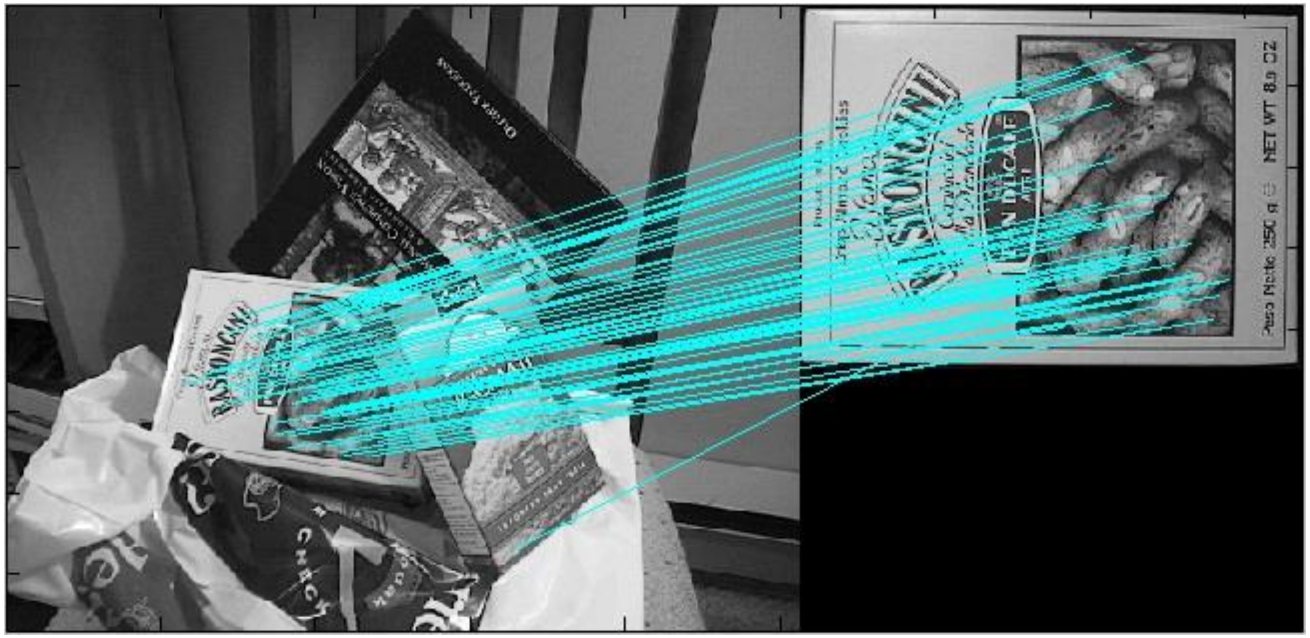
1021 interest points

Examples of Local Interest Points -2

638 interest points

Examples of Local Interest Points -3

Examples of Local Interest Points -4



80 matches

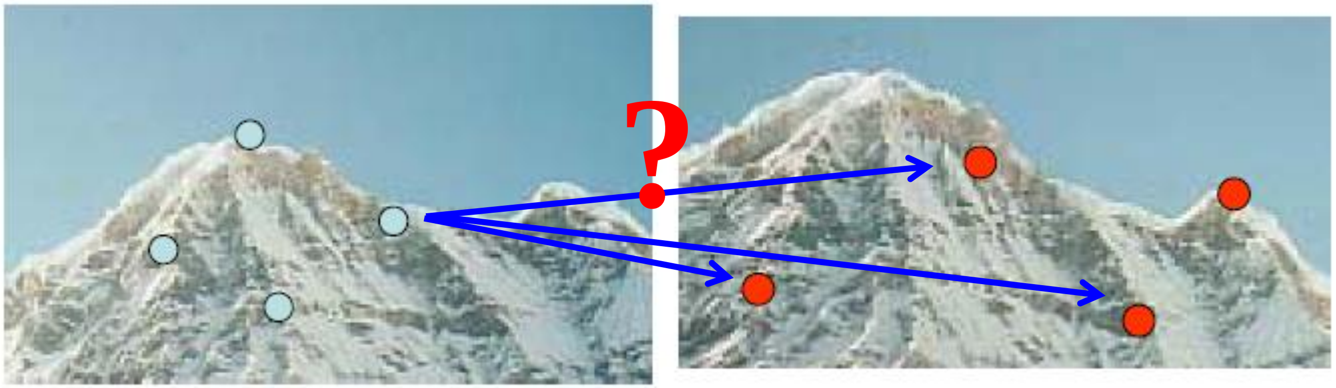
Local Interest Point Detection -1

- Goal: Repeatability of interest operator
  - We want to be able to detect corresponding points in both images

- Yet we have to be able to run the detection procedure *independently* on each image to locate corresponding points

Local Interest Point Detection -2

- Goal: Distinctiveness of descriptor
  - We want to be able to reliably match which point goes with which.

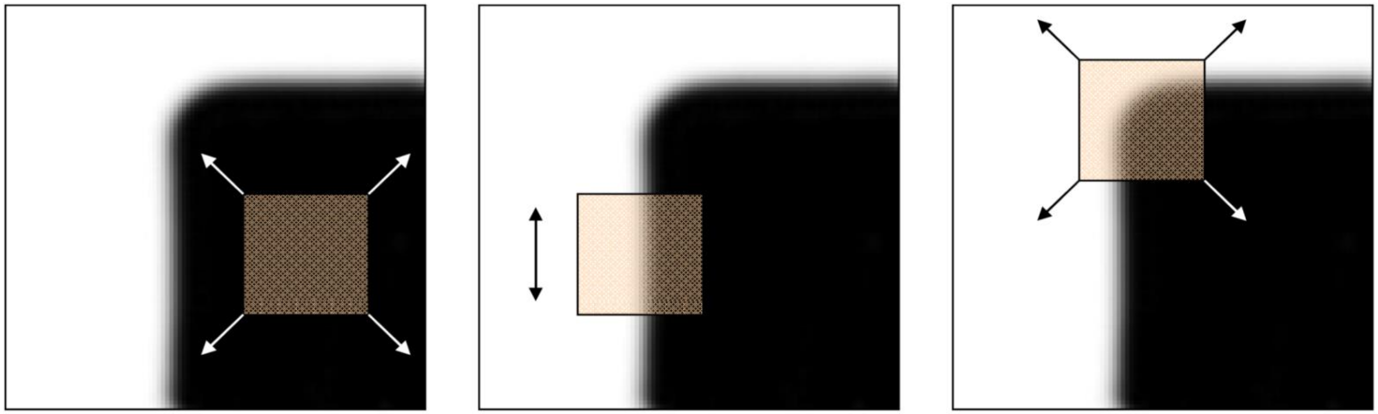


- We need a reliable and distinctive descriptor
  - Must provide some invariance to geometric and photometric differences between the two views

Local Interest Point Detection -3

Characteristics of Interest Points

- Idea: Choose distinctive (unique) interest points that have variation in multiple directions:
  - We should easily recognize the point by looking through a small window
  - Shift a window in any direction should give a large change in intensity



"Flat" Region:
No change in all directions

"Edge":
No change along the edge direction

"Corner":
Significant Change in several directions

Local Interest Point Detection -4
Harris Corner Detector

- It is rotation invariant, but not scale invariant

$$M = \sum w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

Window Function
of 8x8 or 16x16

Equivalent to looking for dominant gradient directions or Eigen vectors

Note: The tops of the horns are detected in both images (Harris Corners are shown in red)

Local Interest Point Detection -5
Scale Invariant Blob Detection

- A popular signature function used is Laplacian-of-Gaussian (LoG) function

$$LoG = \Delta G_{\sigma(x,y)} = x = \frac{\partial^2}{\partial x^2} G_{\sigma(x,y)} + \frac{\partial^2}{\partial y^2} G_{\sigma(x,y)}$$

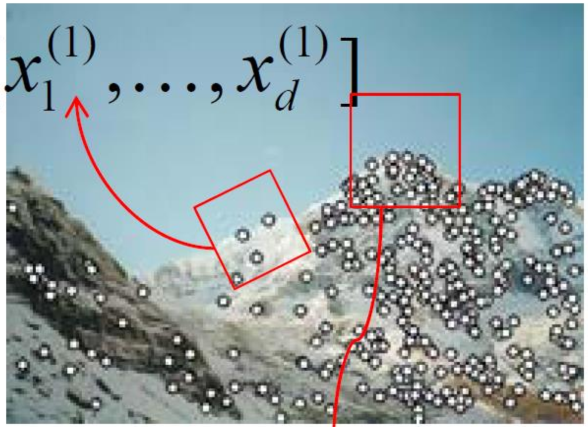
$$G_\sigma(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp[\frac{x^2+y^2}{2\sigma^2}]$$

Interest points can be defined as the centers of blobs.

Local Feature Descriptor

- Next problem: How to extract appropriate vector of feature descriptor to represent each interest point.

$$\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]$$

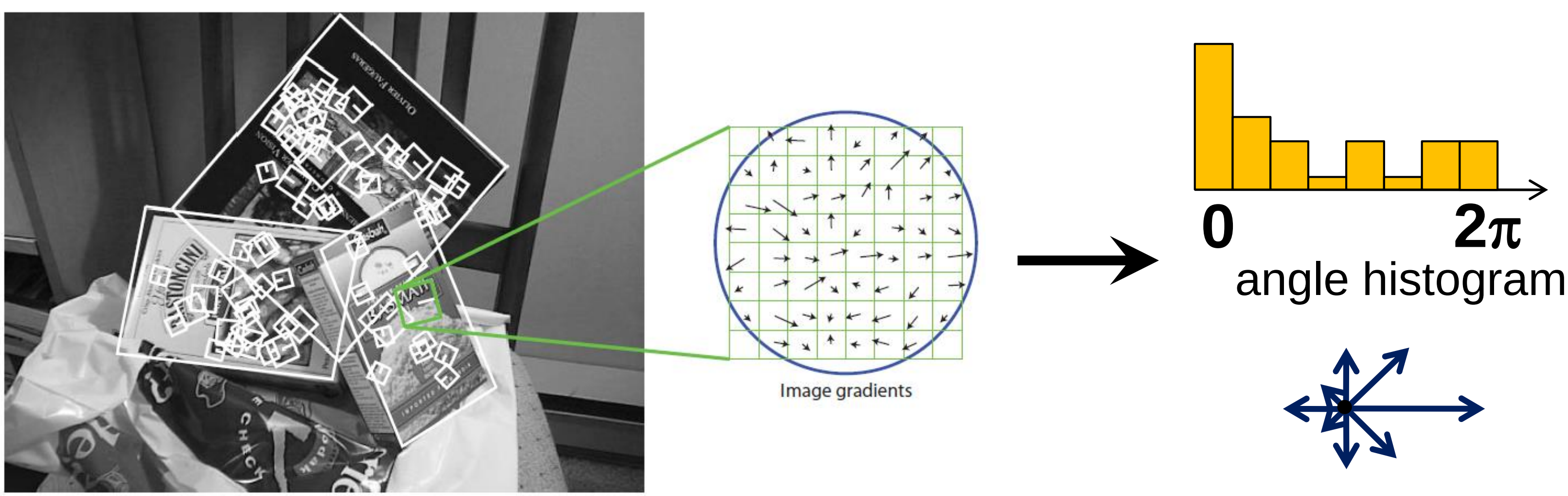


$$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$$

- Simple option: raw patches as local descriptors:
  - The simplest way to describe the neighborhood around an interest point is to write down the list of intensities to form a feature vector
  - But this is very sensitive to even small shifts, rotations

Scale Invariant Feature Transform (SIFT) descriptor -1

- Basic idea: use edge orientation representation
  - Take 16x16 square window around detected feature
  - Compute edge orientation for each pixel
  - Throw out weak edges (threshold gradient magnitude)
  - Create histogram of surviving edge orientations



Scale Invariant Feature Transform (SIFT) descriptor -2

- A popular descriptor:
  - Divide the 16x16 window into a 4x4 grid of cells (we show the 2x2 case below for simplicity)
  - Compute an orientation histogram for each cell
  - 16 cells X 8 orientations = 128 dimensional descriptor

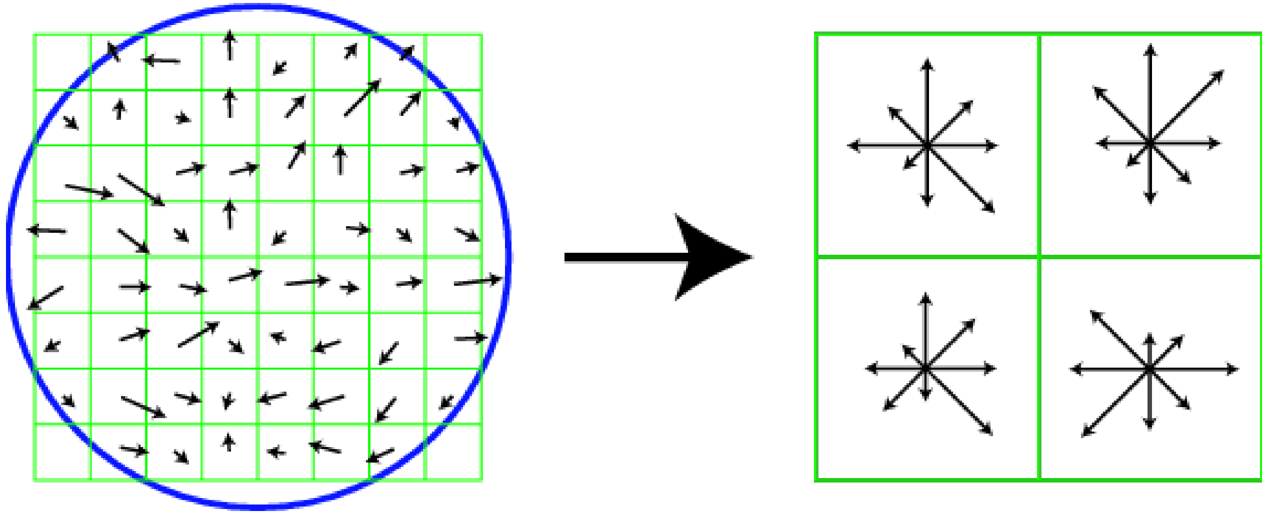


Image gradients

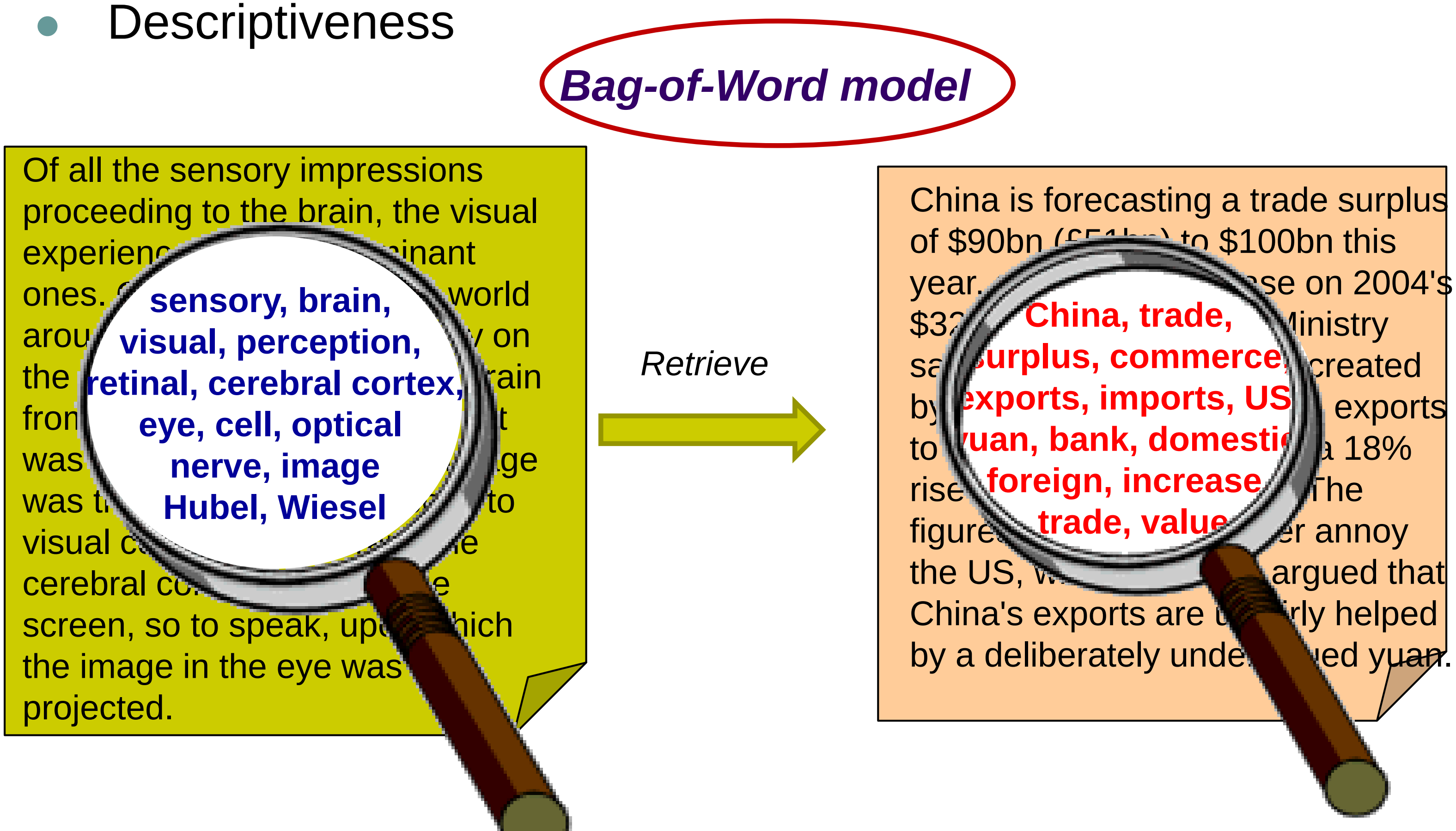
Keypoint descriptor

Scale Invariant Feature Transform (SIFT) descriptor -3

- Invariant to
  - Scale
  - Rotation
- Partially invariant to
  - Illumination changes
  - Camera viewpoint
  - Occlusion, clutter

Overall Representation: as Bag of Visual Words -1

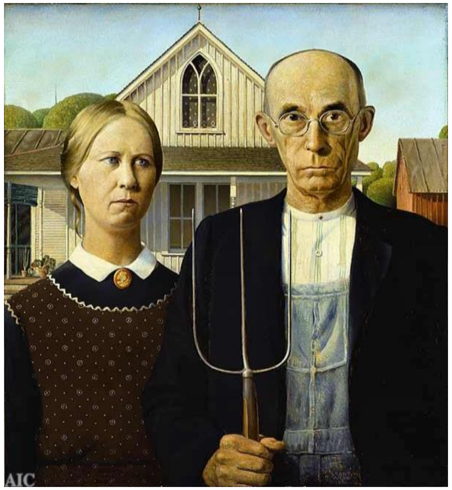
- Text Words in Information Retrieval (IR)
  - Compactness
  - Descriptiveness



Bag of Visual Words -2

- Can images be represented as Bag-of-Visual Words?

Image ⇒ **Bag of 'visual words'**

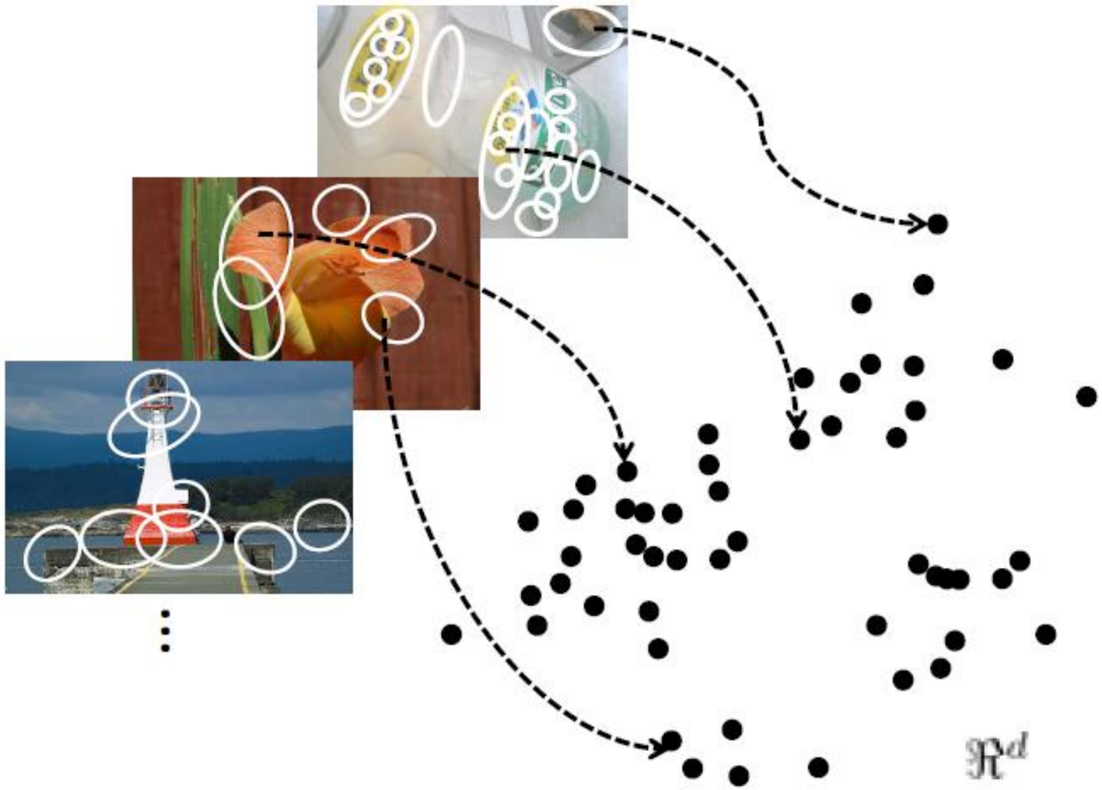


?

- Idea: quantize SIFT descriptors of all training images to extract representative visual words!

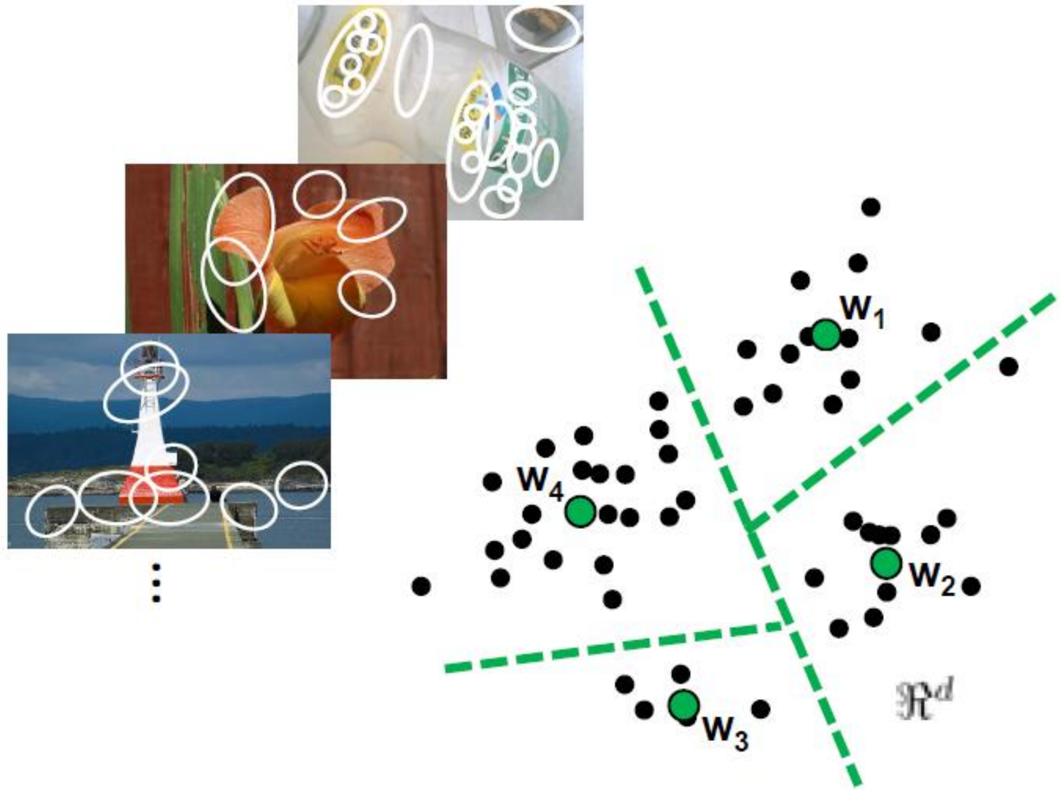
Bag of Visual Words -3

Step 1: Extract interest points of all training images

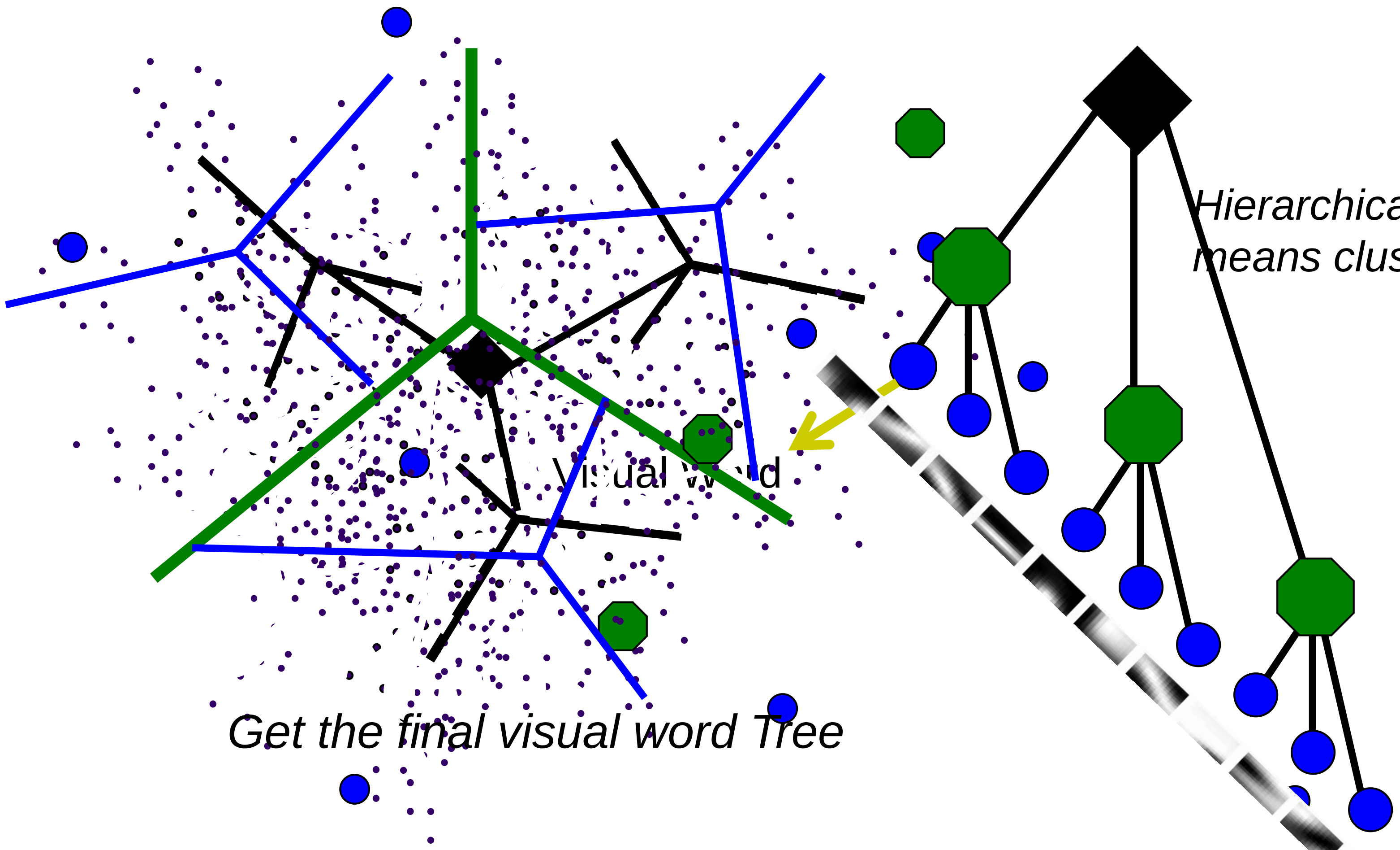


Bag of Visual Words -4

Step 2: Features are clustered to quantize the space into a discrete number of visual words.

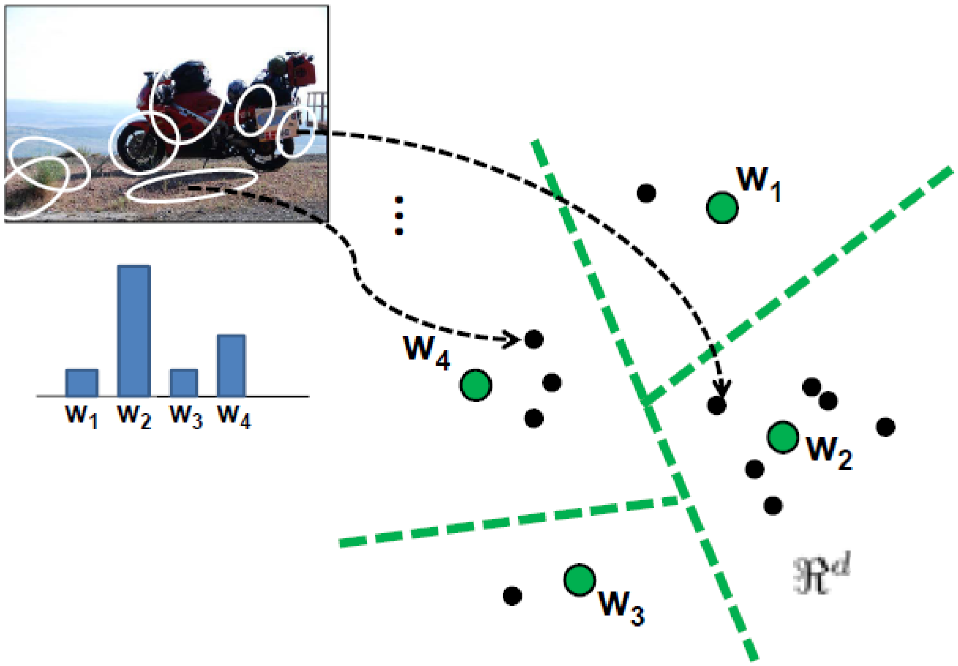


Bag of Visual Words -5



Bag of Visual Words -6

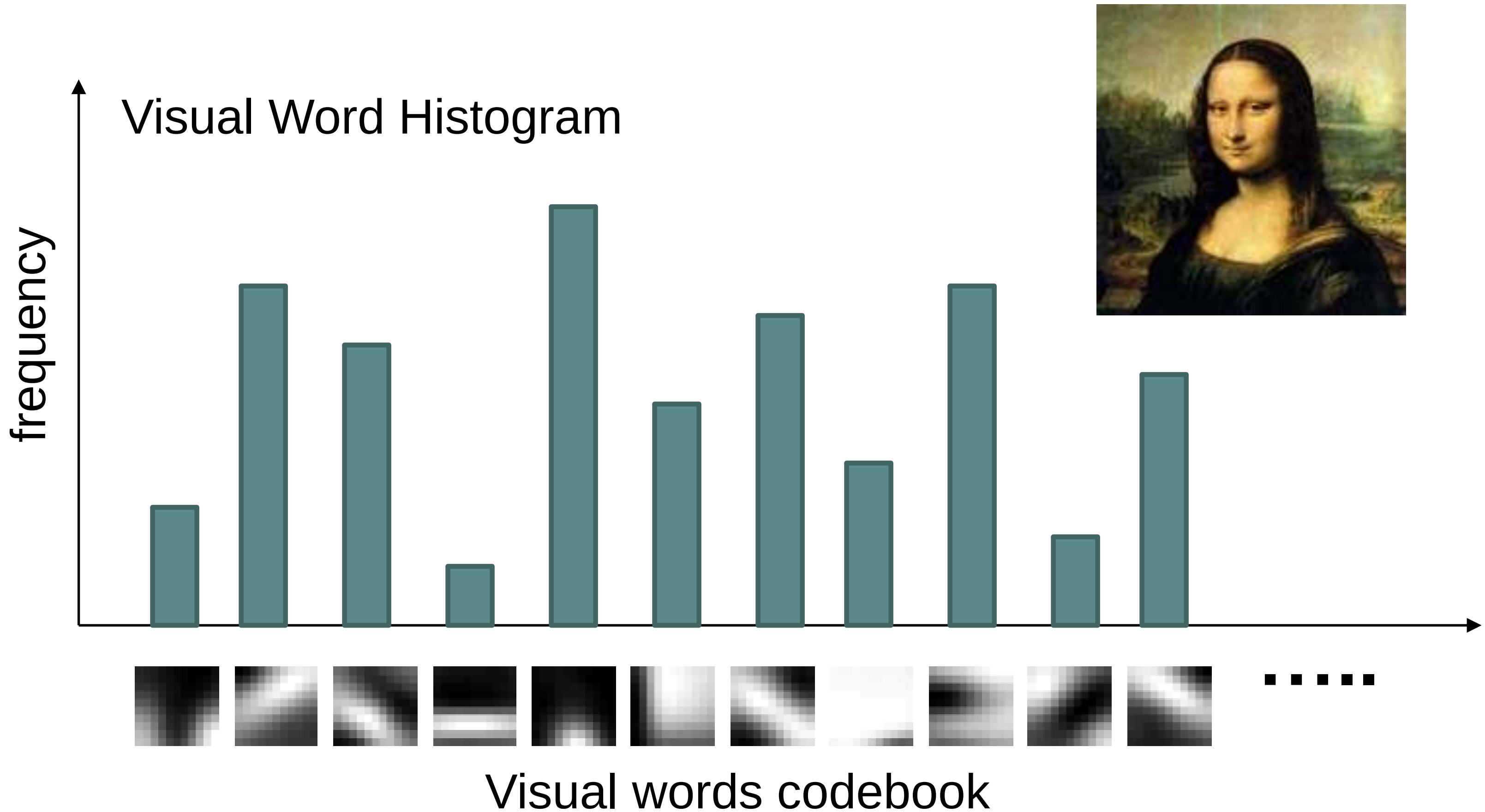
Step 3: Summarize (represent) each image as histogram of visual words



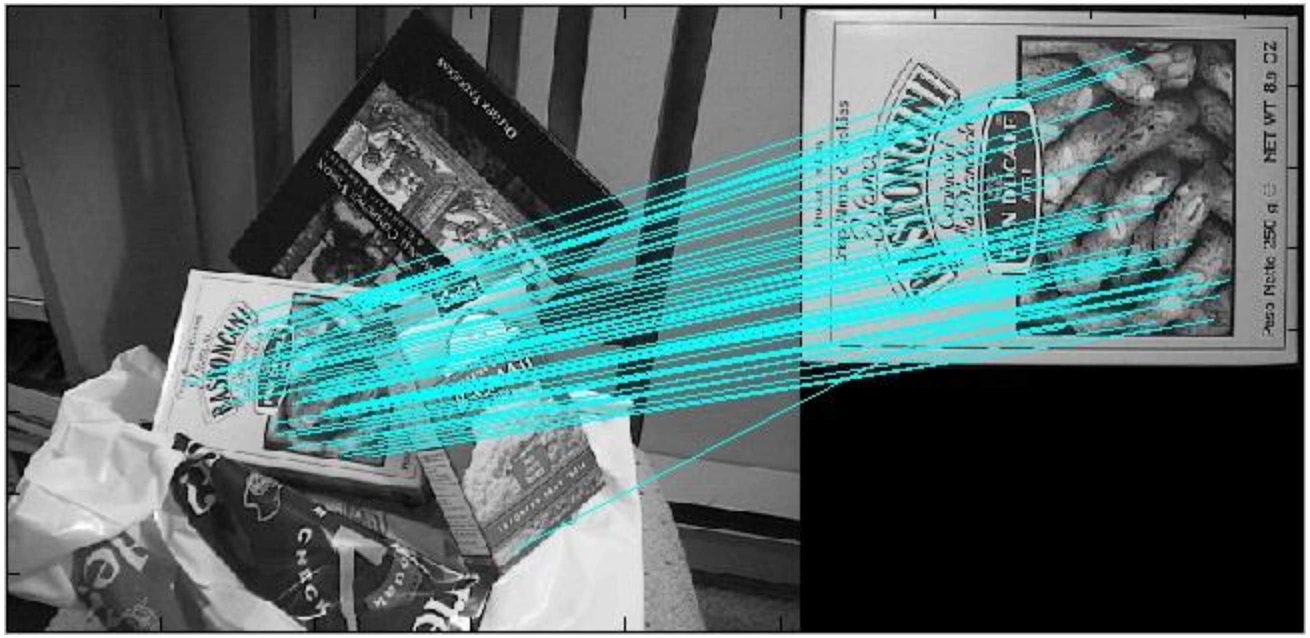
and use as basis for matching and retrieval!

Bag of Visual Words -7

- Another example:

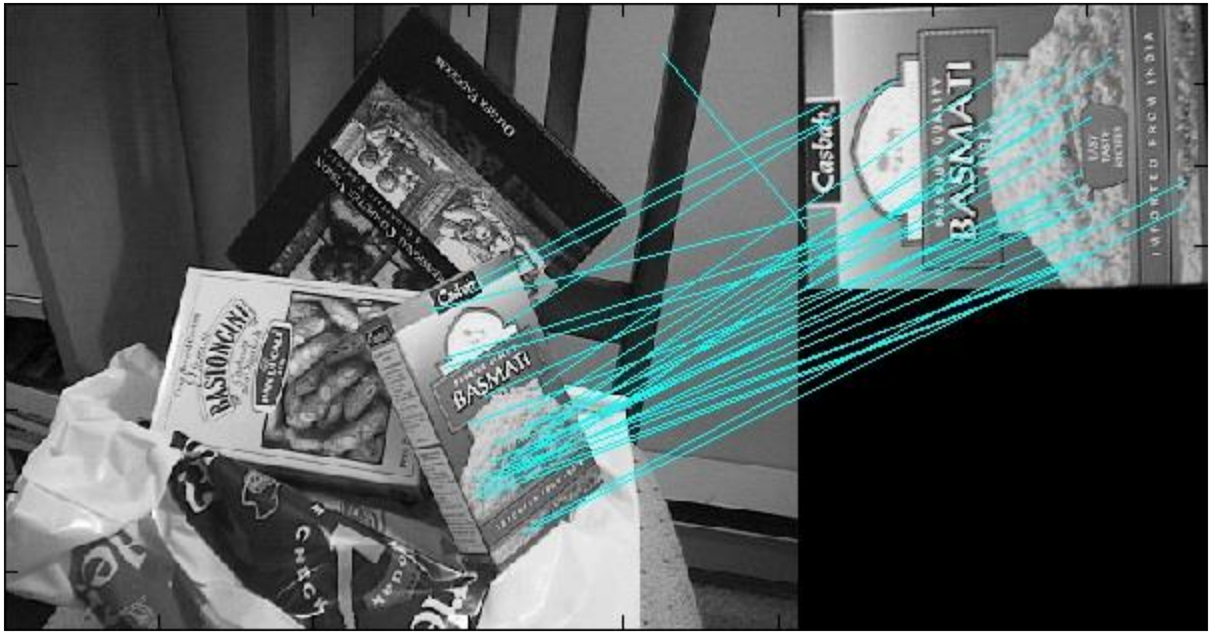


More Examples of Local Interest Point Extraction & Matching -1



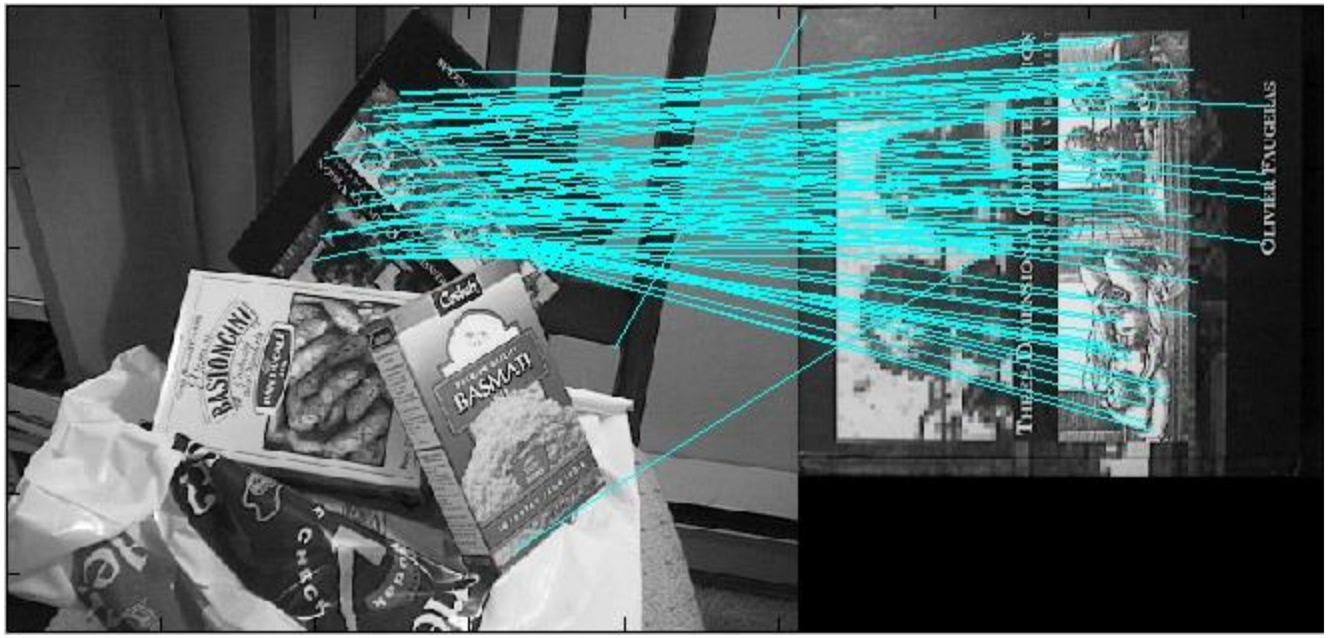
80 matches

More Examples of Local Interest Point Extraction & Matching -2



34 matches

More Examples of Local Interest Point Extraction & Matching -3



98 matches

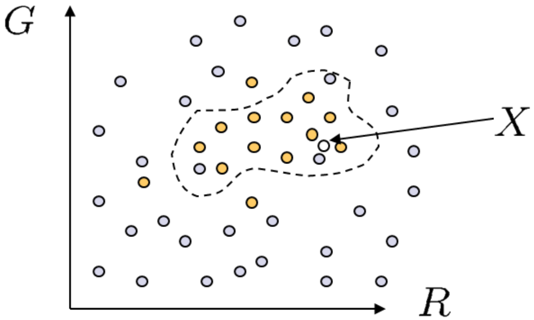
Contents

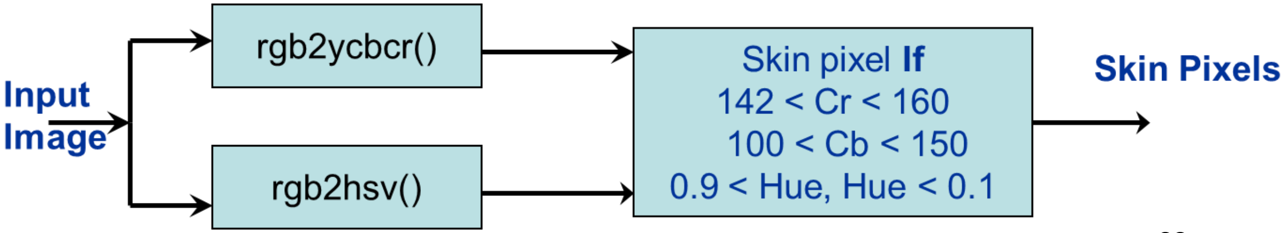
- kNN Image Search
- Intro to Advanced Visual Features
  - Visual Keywords
  - Face Detector
  - Visual Concept Detector
- Summary

Face Detection

- Basic idea: slide a window across image and evaluate a face model at every location
  - Limitation: Sliding window detector must evaluate tens of thousands of location/scale combinations

- Face is skin, why not detect skin first?
  - Skin pixels have a distinctive range of colors
  - Threshold based method to detect skin regions.





Face Detection

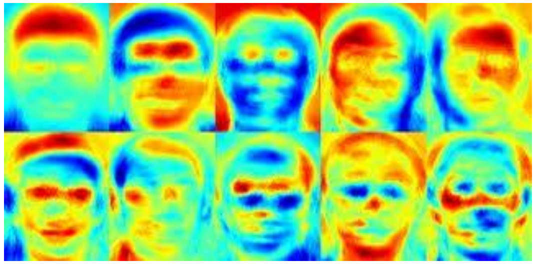
- Next, perform morphological processing, such as reject blobs of small sizes, perform closing and remove holes.

Results of Skin Region Segmentation

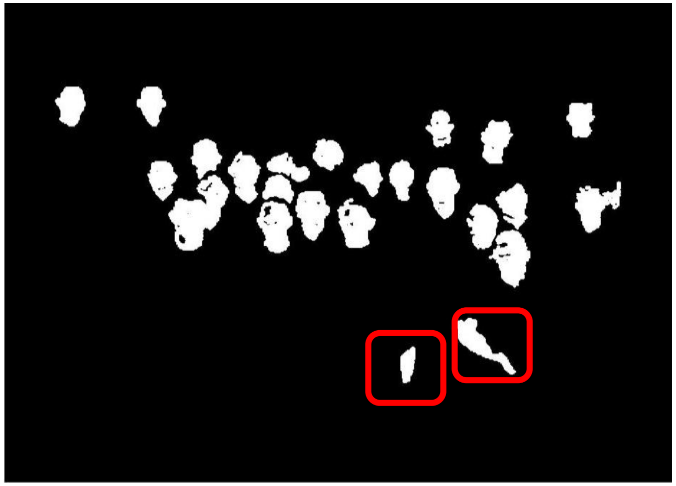
Results of Morphological Processing

Face Detection

- Non-face Object Removal (may be legs, arms or hands)
  - Use information about shape and location of skin region-based objects in conjunction to reject non-face objects, while minimizing rejection of faces.
  - Objects characterized by face-like templates



Before



After

Face Detection

- Detection rate
  - Total number of faces that are correctly detected / total number of faces actually exist in the picture.
- False positive rate
  - The detector output is positive but it is false (there is actually no face).

- Detection rate=(5/7)*100%
- False positive rate = (1/6)*100%, since 6 windows are reported to have faces but 1 of them is not a face.

False positive result

Contents

- kNN Image Search
- Intro to Advanced Visual Features
  - Visual Keywords
  - Face Detector
  - Visual Concept Detector
- Summary

What do we mean by "Concept Recognition"

Verification:
Is that a statue of rabbit?

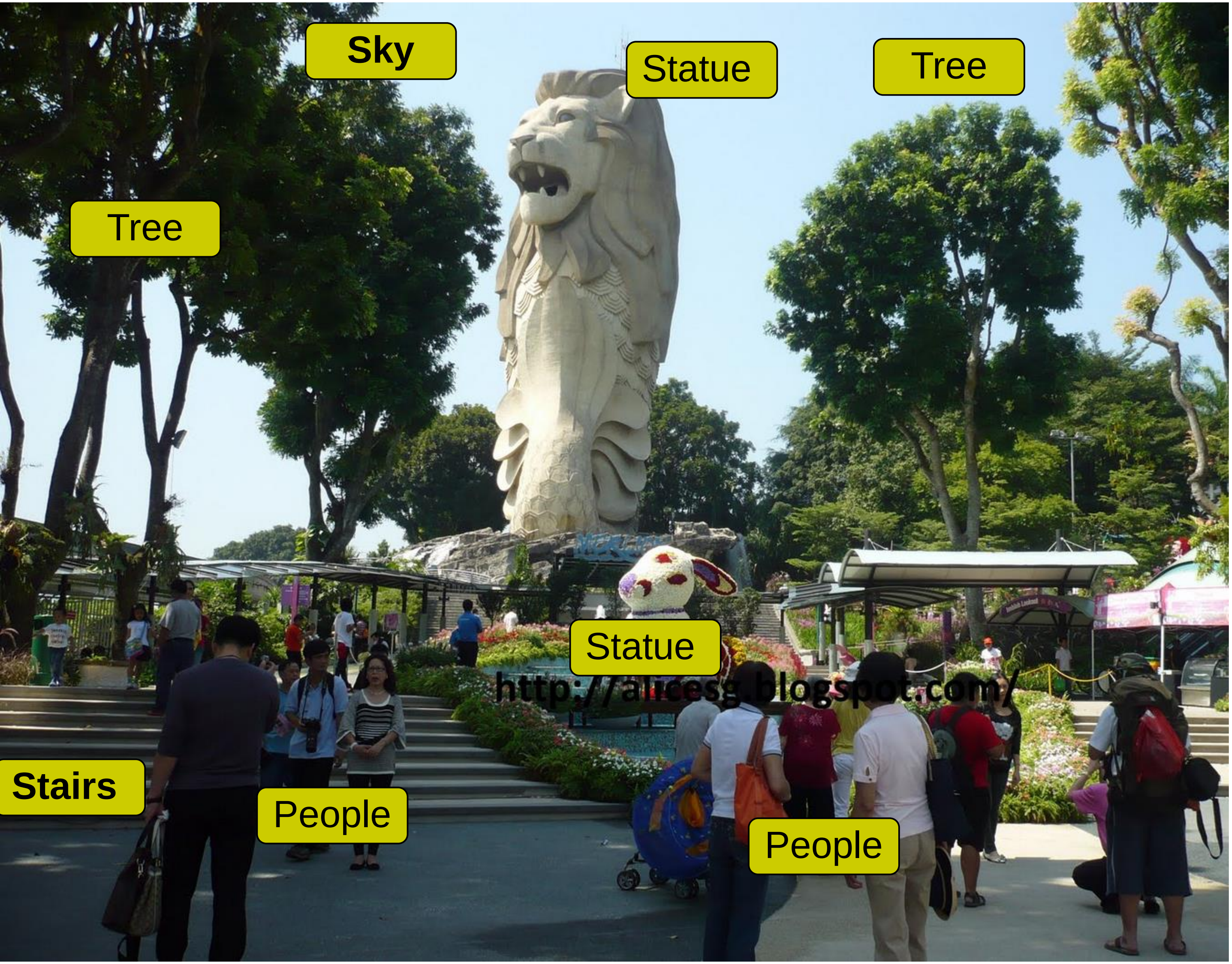
What does Concept Recognition involve?

Detection:
Are there trees?

What does Concept Recognition involve?

Identification:
Is that the merlion, Singaore’s landmark?

What does Concept Recognition involve?



Object Categorization

What does Concept Recognition involve?

- Outdoor
- Tourist Spots
- …

Scene
and Context
Categorization


~10,000 to 30,000

Concept Recognition: Challenges

- View point variation
- Illumination
- Occlusion
- Scale
- Deformation
- Background clutter



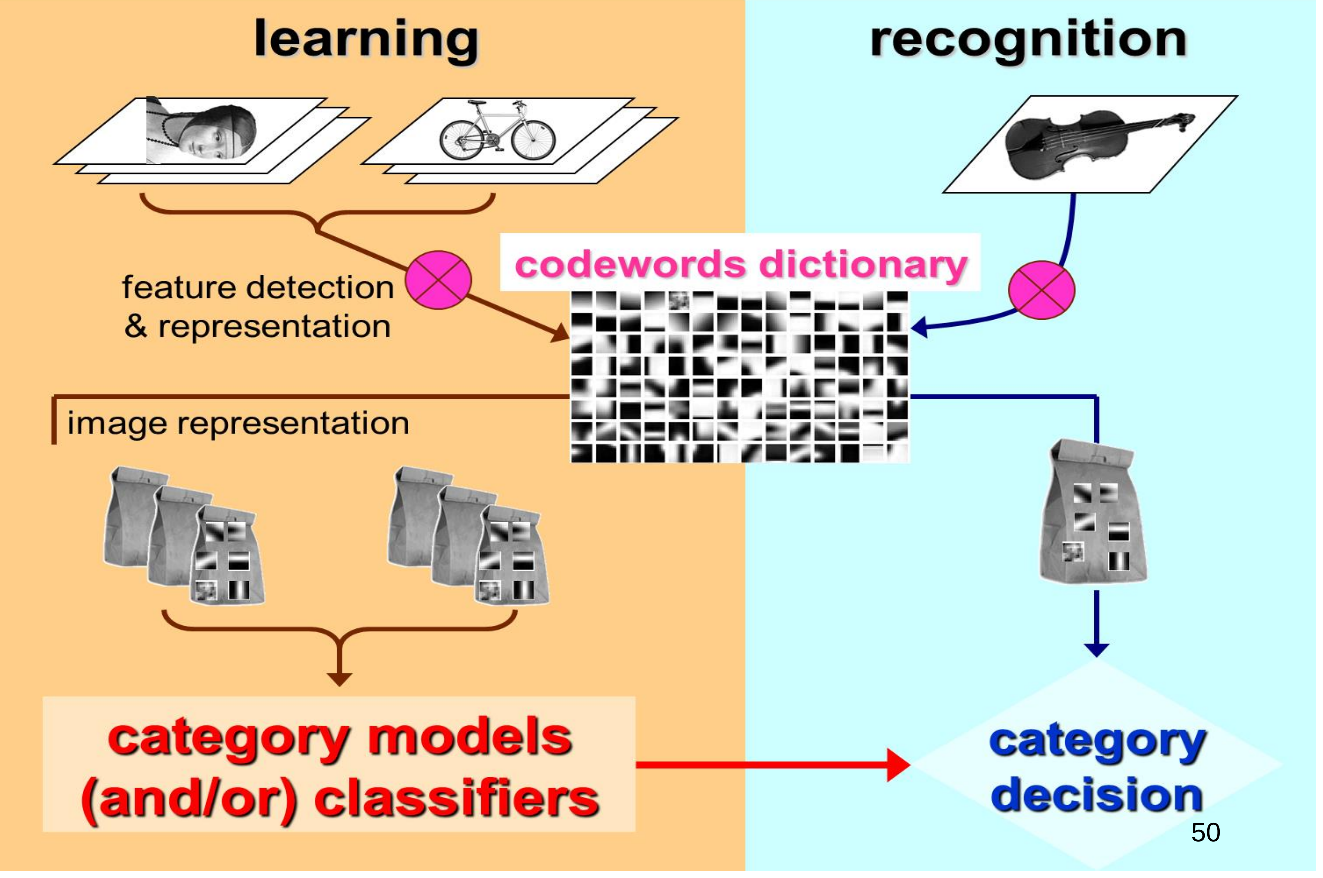




Concept Recognition: Bag-of-Word Model

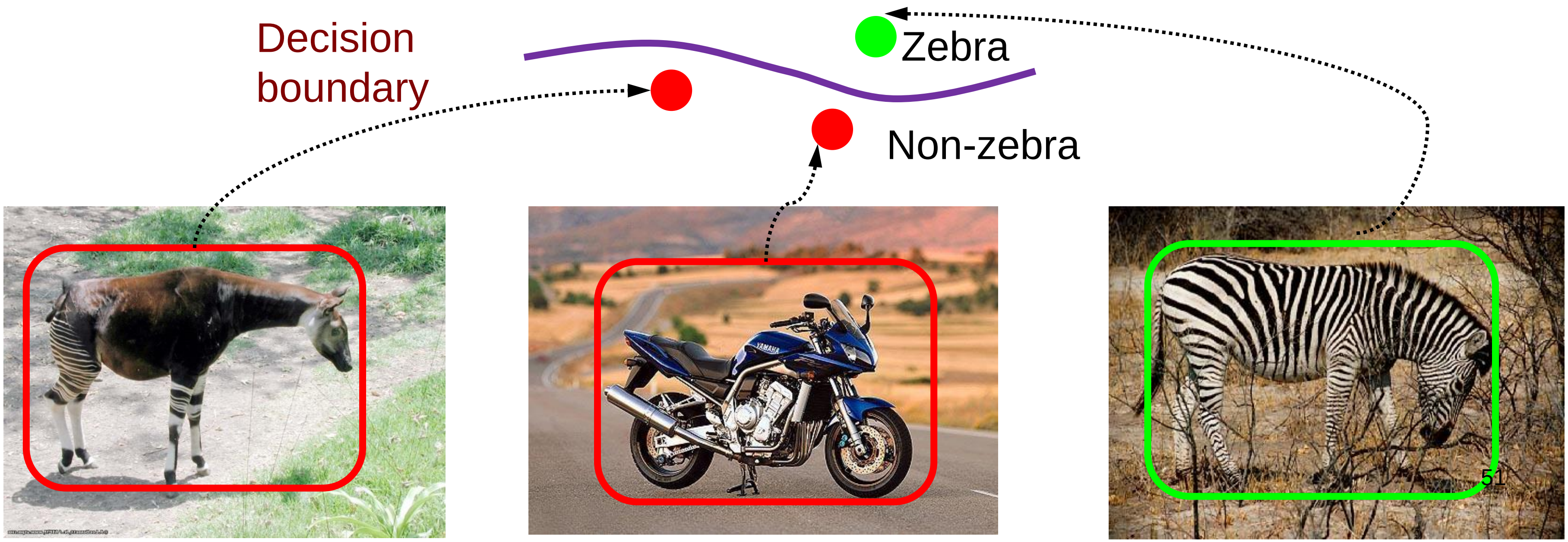
BASIC IDEA:

- Representative set of images in each category is collected
- An image is represented by a collection of "visual words"
- Object categories are modeled by the distributions of these visual words



Concept Recognition: Discriminative Model

- Object detection and recognition is formulated as a classification problem. The image is partitioned into a set of overlapping windows, and a decision is taken at each window about if it contains a target object or not.
- Each window is represented by a large number of features that encode info such as boundaries, textures, color, spatial structure.
- The classification function, that maps an image window into a binary decision, is learnt using methods such as SVMs or neural networks



Contents

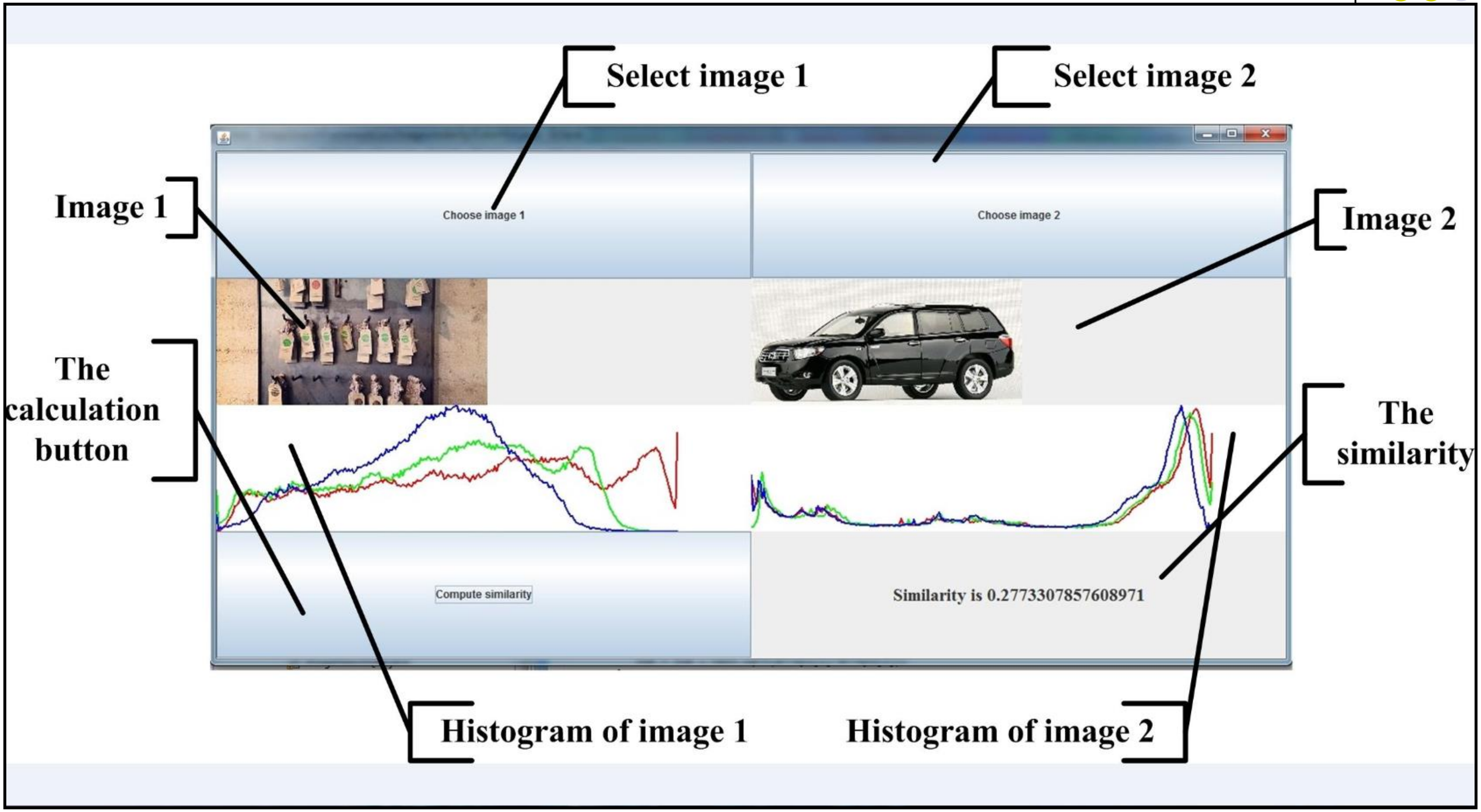
- kNN Image Search
- Intro to Advanced Visual Features
  - Visual Keywords
  - Face Detector
  - Visual Concept Detector
- Summary

Summary

- We have discussed a simple search/ classification method and ideas using several advanced visual features
  - Discussed kNN method, which has been effective for large-scale image search and classification

    (provided we tacked the indexing problem)
- Discussed 3 more advanced feature representation methods (non-examinable)
  - Bag-of-Visual-Words: basis for object level matching
  - Face detection: an example of robust specialized detector
  - Visual Concept Detector: identify concepts in images and move towards concept-based image search
  - They represent current state-of-the-arts approaches: try that in your assignment

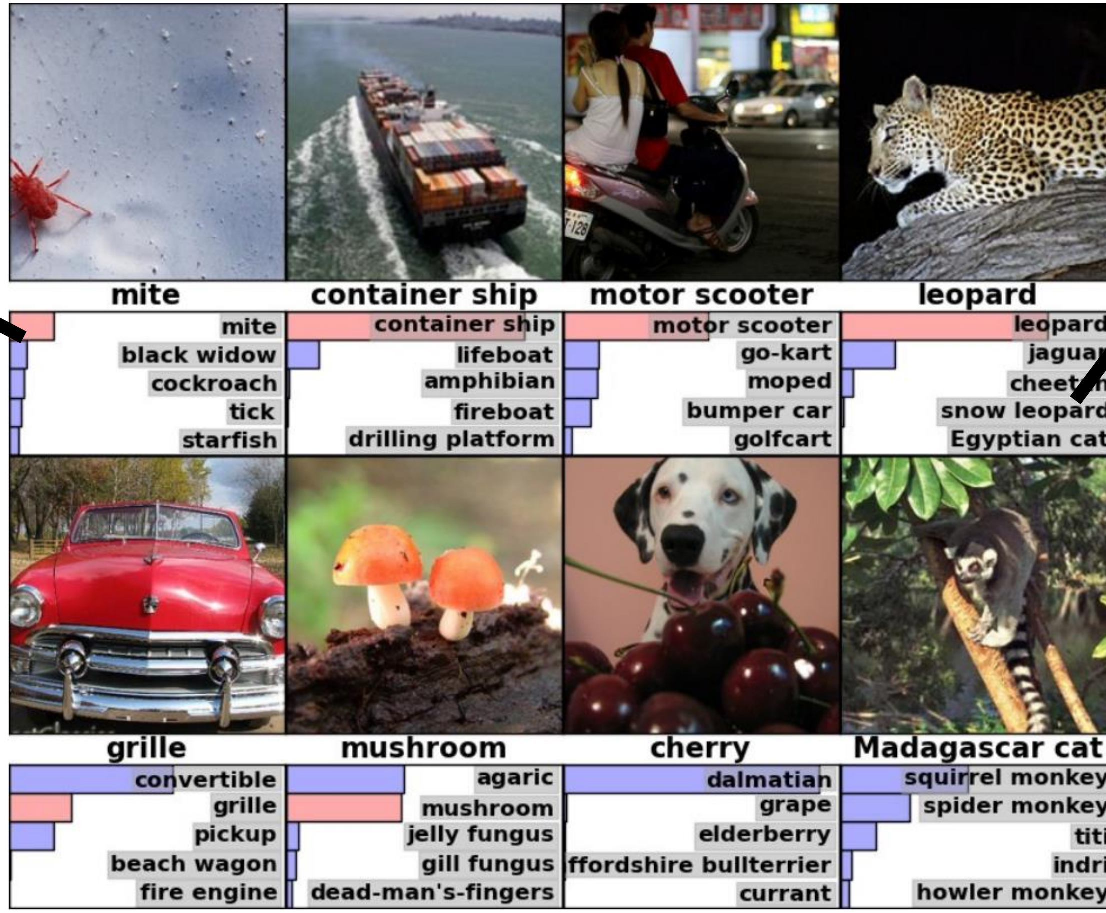
What you are given for your assignment: Color Histogram Tool



- Given an image, obtain the color histogram of the image.
- Based on the histogram, compute the similarities between two images.

Semantic Concept Features

The possibility that the image belongs to the category.



Top 5 Categories

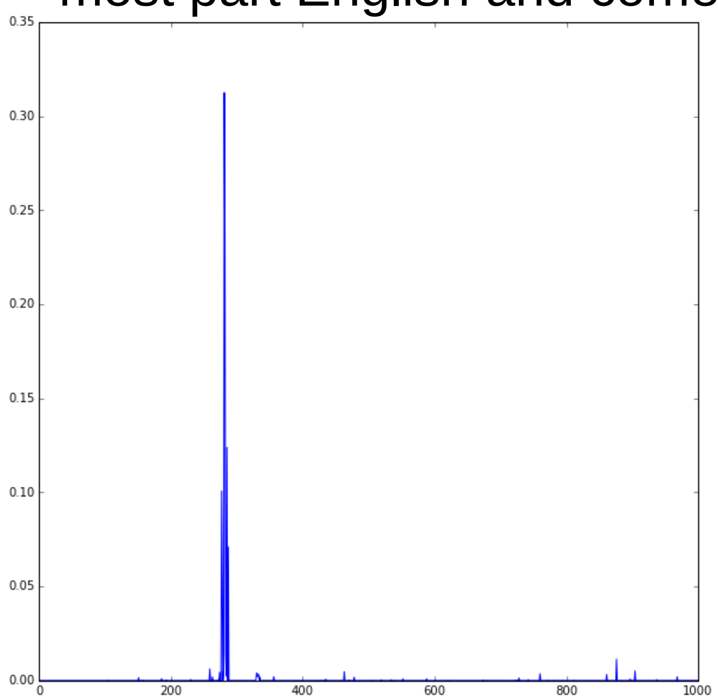
- Given an image, obtain a 1000-D semantic concept feature vector.
- Each element value of this vector shows the possibility of the image belongs to a category. We have 1000 preliminary categories.

Concept Recognition: Tools

- Many tools available
  - 1000-Class Image Classification Tool : This tool follows the traditional concept recognition framework. They extract different image features (HOG, SIFT, LBP, color histogram, color moment, etc.) and then use SVM classifier to predict concept contained in images. The tool is included in the assignment folder.
  - BVLC CaffeNet : This tool is developed based on Caffe deep learning framework. The model obtains a top-1 accuracy 57.4% and a top-5 accuracy 80.4% on the validation set. (http://caffe.berkeleyvision.org/gathered/examples/imagenet.html)
  - Clarifai's Deep Learning Systems: The system provides a high performance deep learning API which pushing the state of the art in large scale object recognition. (Clarifai Java Client: https://github.com/Clarifai/clarifai-java)

Concept Recognition: Tools

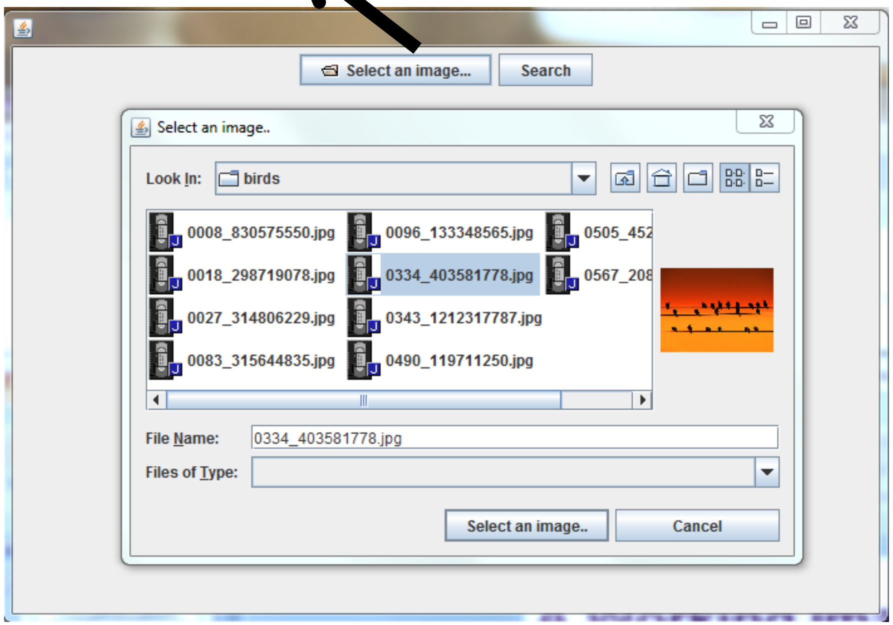
- Result Format
  - The first two tools will return a vector of size 1000 as the scores for 1000 predefined classes (http://www.image-net.org/).
  - The third tool will return a list of “classes” and a corresponding list of probabilities. The “classes” are for the most part English and come from a large vocabulary.



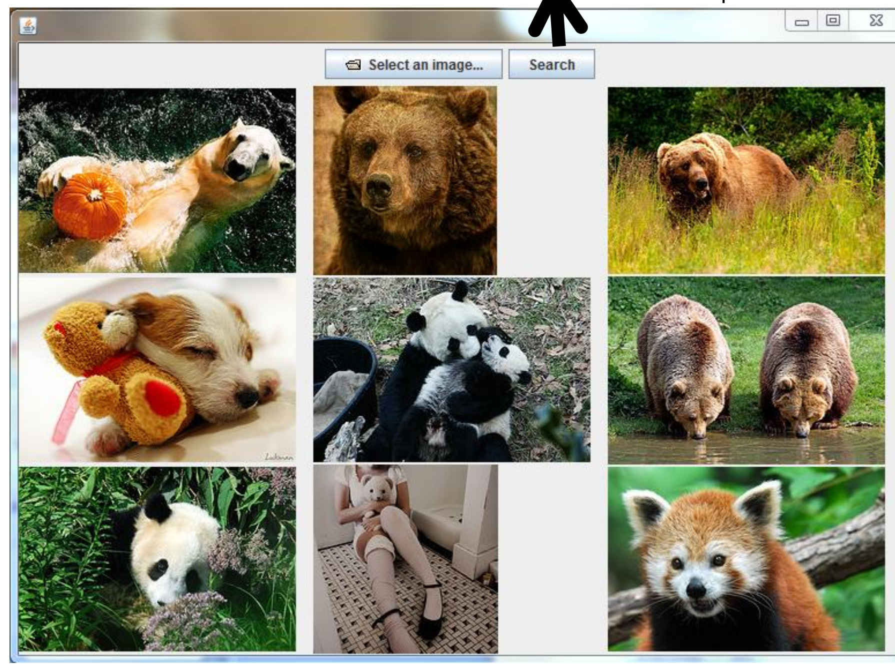

```
'result':
{'tag': {'classes': ['train', 'probs': [0.1910400390625,
   'subway',                      0.10223388671875,
   'railroad',                    0.09765625,
   'railway',                     0.093994140625,
   'station',                     0.06884765625,
   'rail',                        0.0295257568359375,
   'transportation',              0.0292816162109375,
   'urban',                       0.021987915039062 5,
   'underground',                 0.0218353271484375,
   'night',                       0.0210418701171875,
   'traffic',                     0.016265869140625,
   'commuter',                    0.0162506103515625,
   'bridge',                      0.01376342773 4375,
   'city',                        0.0127792358398437 5,
   'road',                        0.012603759765625,
   'national park',               0.0120697021484375,
   'public',                      0.0113449096679687 5,
   'platform',                    0.010833740234375,
   'street',                      0.010269165039062 5,
   'blur'],                       0.00933837890625]}},
```

A Working Image Search System

Select a query image



Search results from database



- A working image search system (including a simple UI) that uses only color histogram feature. Given an image, the system will present the ranked list of searching results.
- You need to incorporate and combine other features.

Next Lesson

- We will look into search engine architecture
- Start with basics of visual search index:
  - From Inverted Index to Hash-based Index
- Google Search Architecture