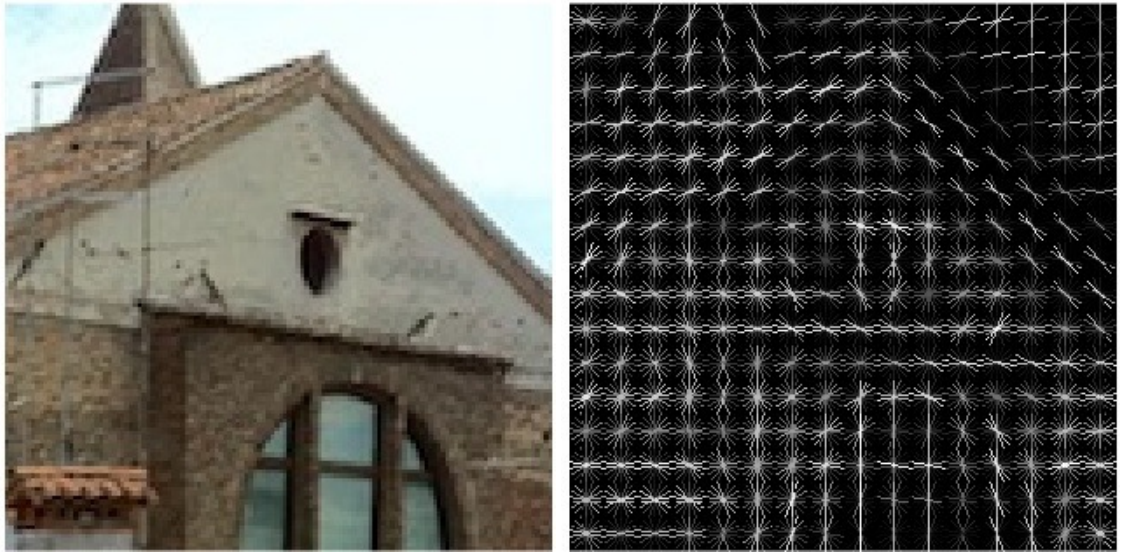
COMP9517

Computer Vision

2025 Term 3 Week 3

Dr Sonit Singh

UNSW SYDNEY



Feature Representation

Part 2

Different types of features (recap)

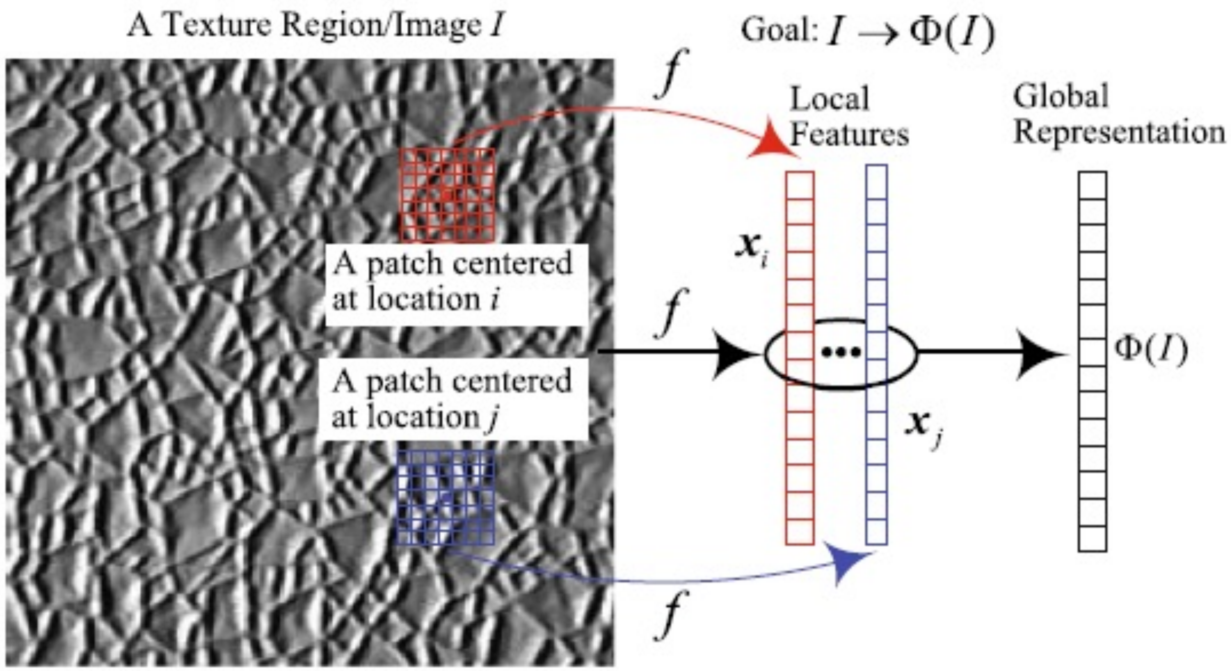
- Colour features (Part 1)
  - Colour moments
  - Colour histogram
- Texture features (Part 1)
  - Haralick texture features
  - Local binary patterns (LBP)
  - Scale-invariant feature transform (SIFT) … One more example application today
- Shape features (Part 2)
  - Basic shape features
  - Shape context
  - Histogram of oriented gradients (HOG)

Another example application of SIFT

- Classifying images based on texture
  - Classification requires an overall feature representation of an image
  - SIFT features are keypoint based
  - How do we represent an entire image using a set of SIFT features?

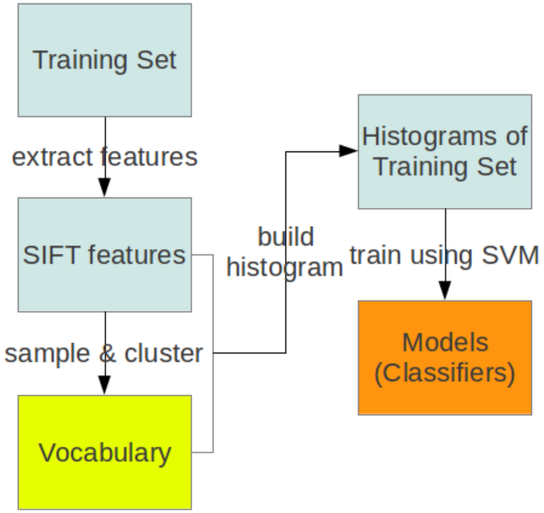
Feature encoding

- Global encoding of local SIFT features
  - Combine local SIFT keypoint descriptors of an image into one global vector

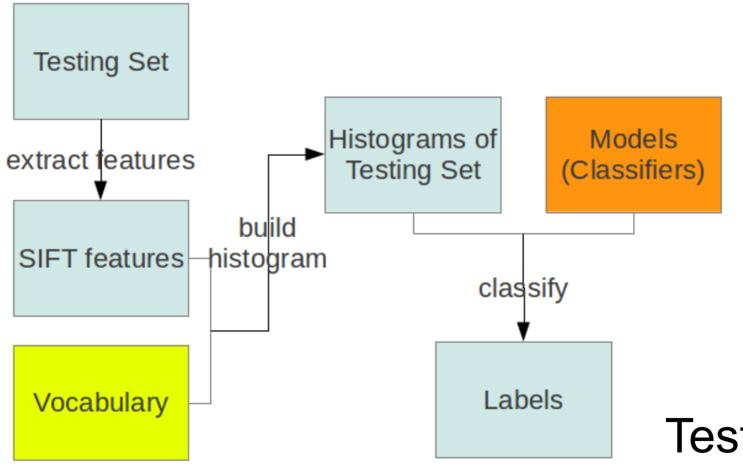


Feature encoding

- Most popular method: Bag-of-Words (BoW)
  - Variable number of local image features
  - Encoded into a fixed-dimensional histogram



Training

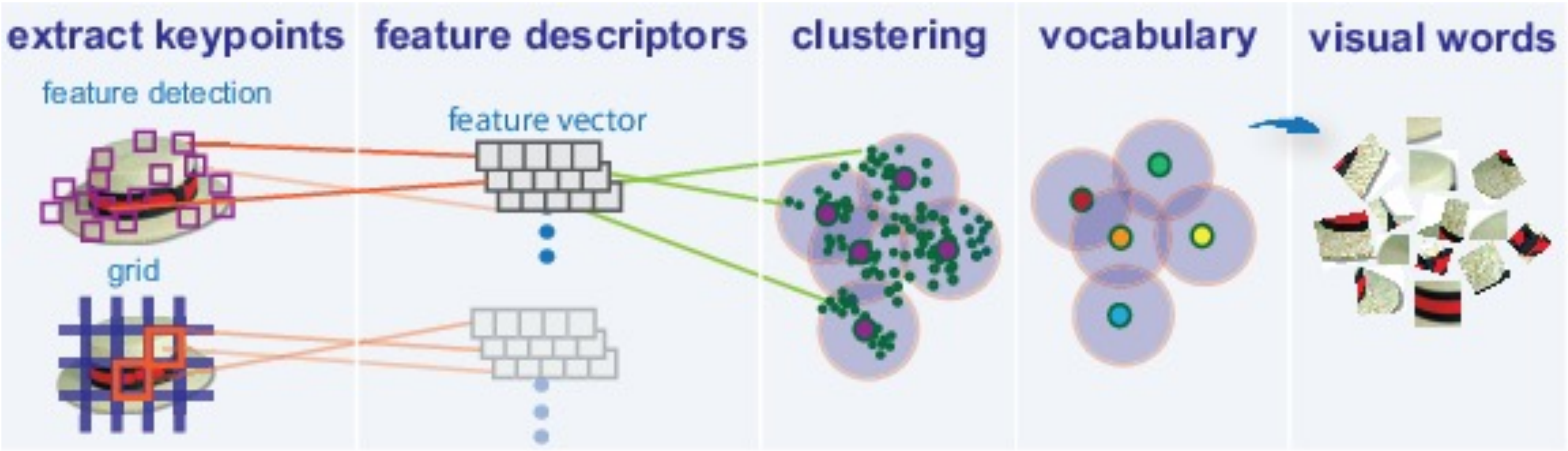


Testing

http://cs.brown.edu/courses/cs143/2011/results/proj3/hangsu/

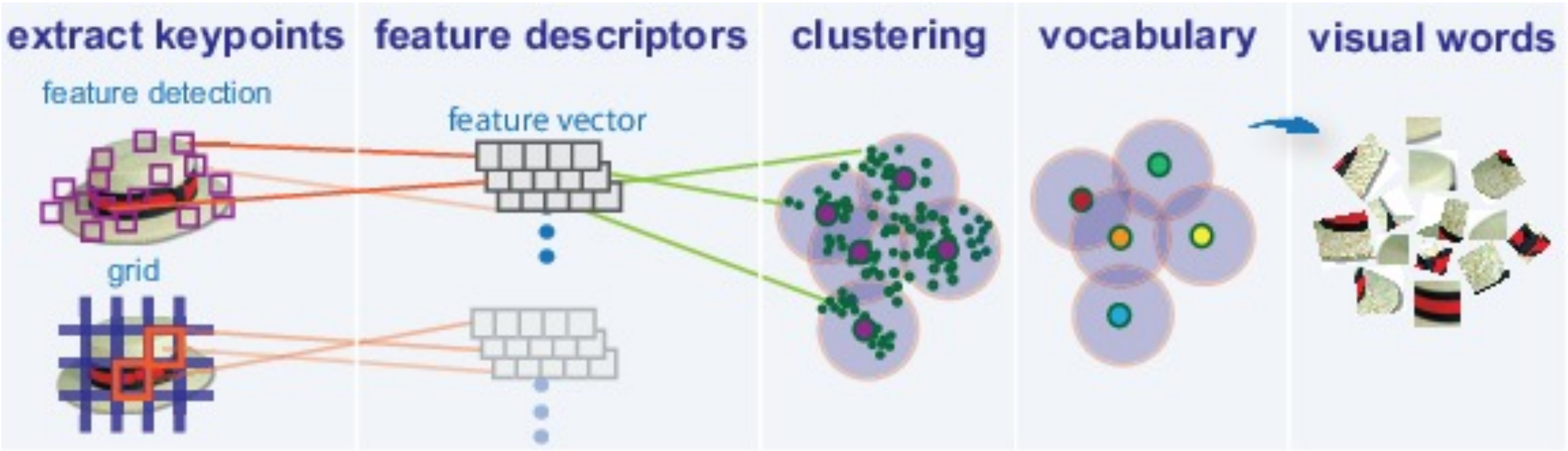
Feature encoding

- Bag-of-Words (BoW): Step 1
  - Extract local SIFT keypoint descriptors from training images
  - Create the “*vocabulary*” from the set of SIFT keypoint descriptors
  - This vocabulary represents the categories of local descriptors



Feature encoding

- Bag-of-Words (BoW): Step 1
  - Main technique used to create the vocabulary is *k*-means clustering
  - One of the simplest and most popular unsupervised learning approaches
  - Performs automatic clustering (partitioning) of the training data into *k* categories



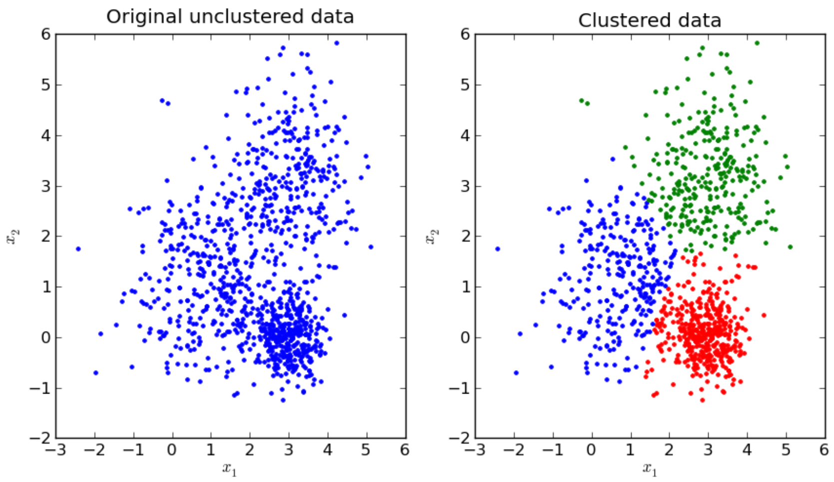
k-means clustering

Initialize: *k* cluster centres (typically randomly)

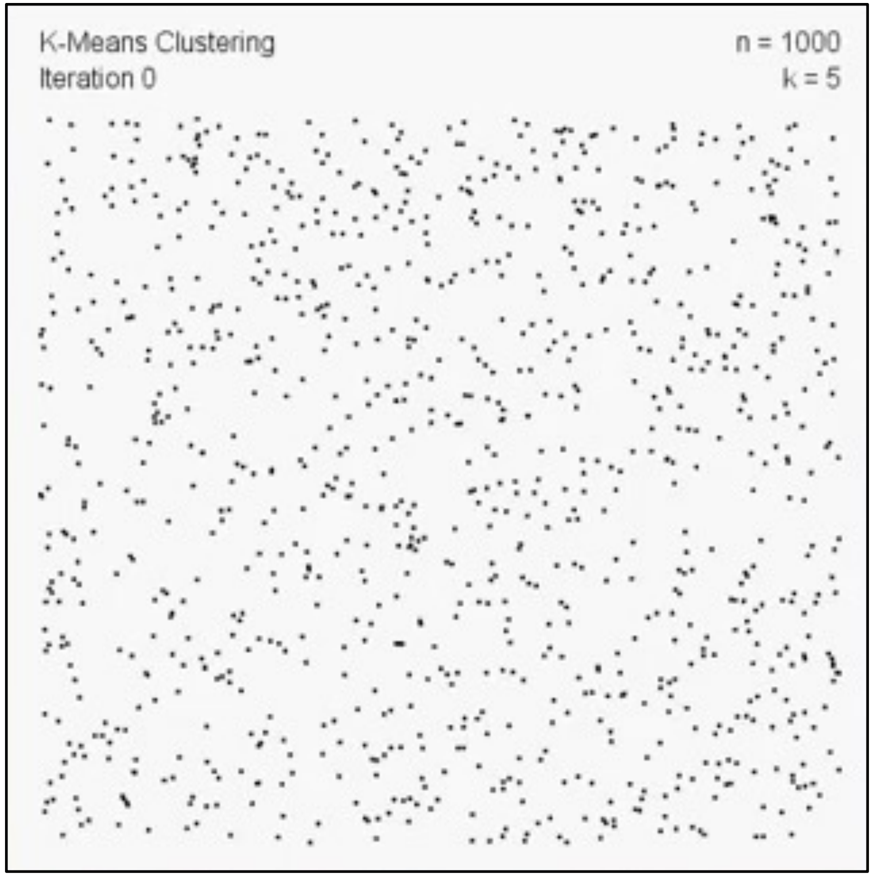
Iterate: 1. Assign data (feature vectors) to the closest cluster (Euclidean distance)

2. Update cluster centres as the mean of the data samples in each cluster

Terminate: When converged or the number of iterations reaches the maximum



Demonstration of *k*-means clustering



Examples:

Points	Clusters	Iterations
1,000	5	30
1,000	10	36
1,000	20	26
5,000	5	33
5,000	10	42
5,000	20	37
10,000	5	30
10,000	10	38
10,000	20	89
10,000	30	68
20,000	30	87

https://www.youtube.com/watch?v=BVFG7fd1H30

Iterations may vary depending on:

- Number of points
- Number of clusters
- Cluster initialization

Feature encoding

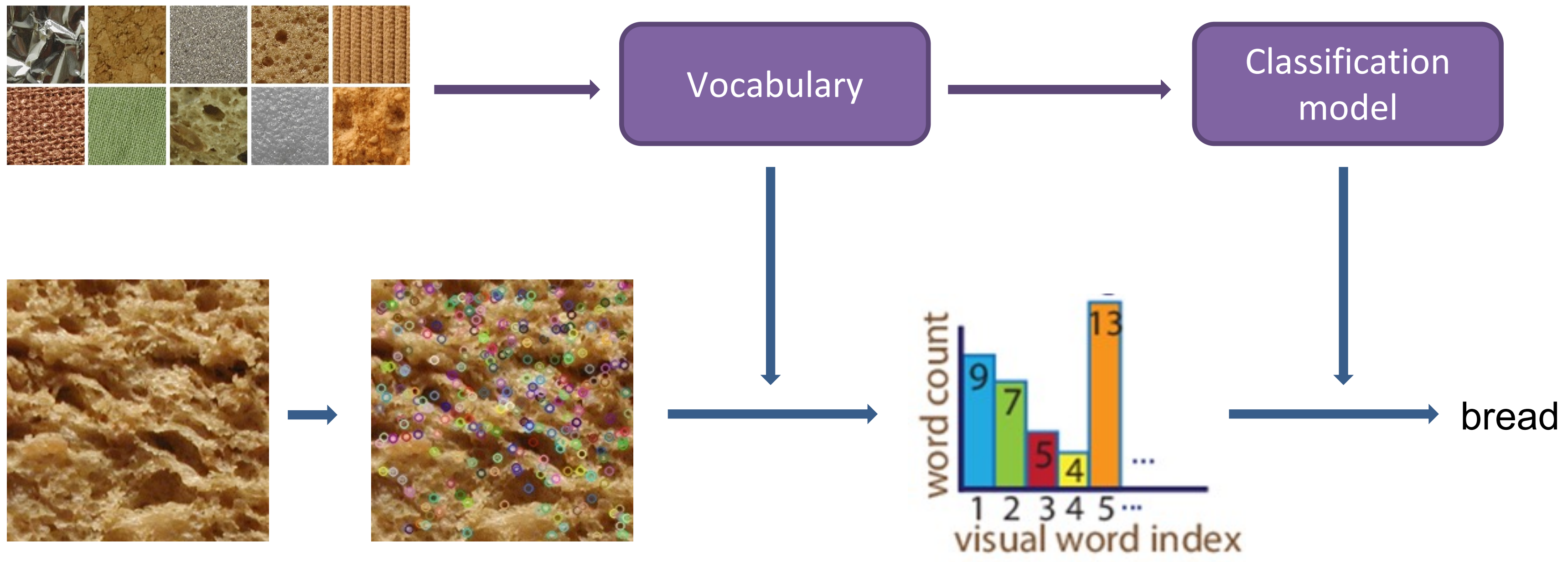
- Bag-of-Words (BoW): Step 2
  - Cluster centres are the "*visual words*" in this "vocabulary" used to represent an image
  - Each local feature descriptor is assigned to one visual word with the smallest distance

Feature encoding

- Bag-of-Words (BoW): Step 2
  - Compute the number of local image feature descriptors assigned to each visual word
  - Concatenate the numbers into a vector which is the "*BoW*" representation of the image

Example application of feature encoding

- SIFT-based texture classification



1. SIFT feature extraction

2. BoW encoding

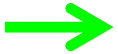
3. Classification

Example application of feature encoding

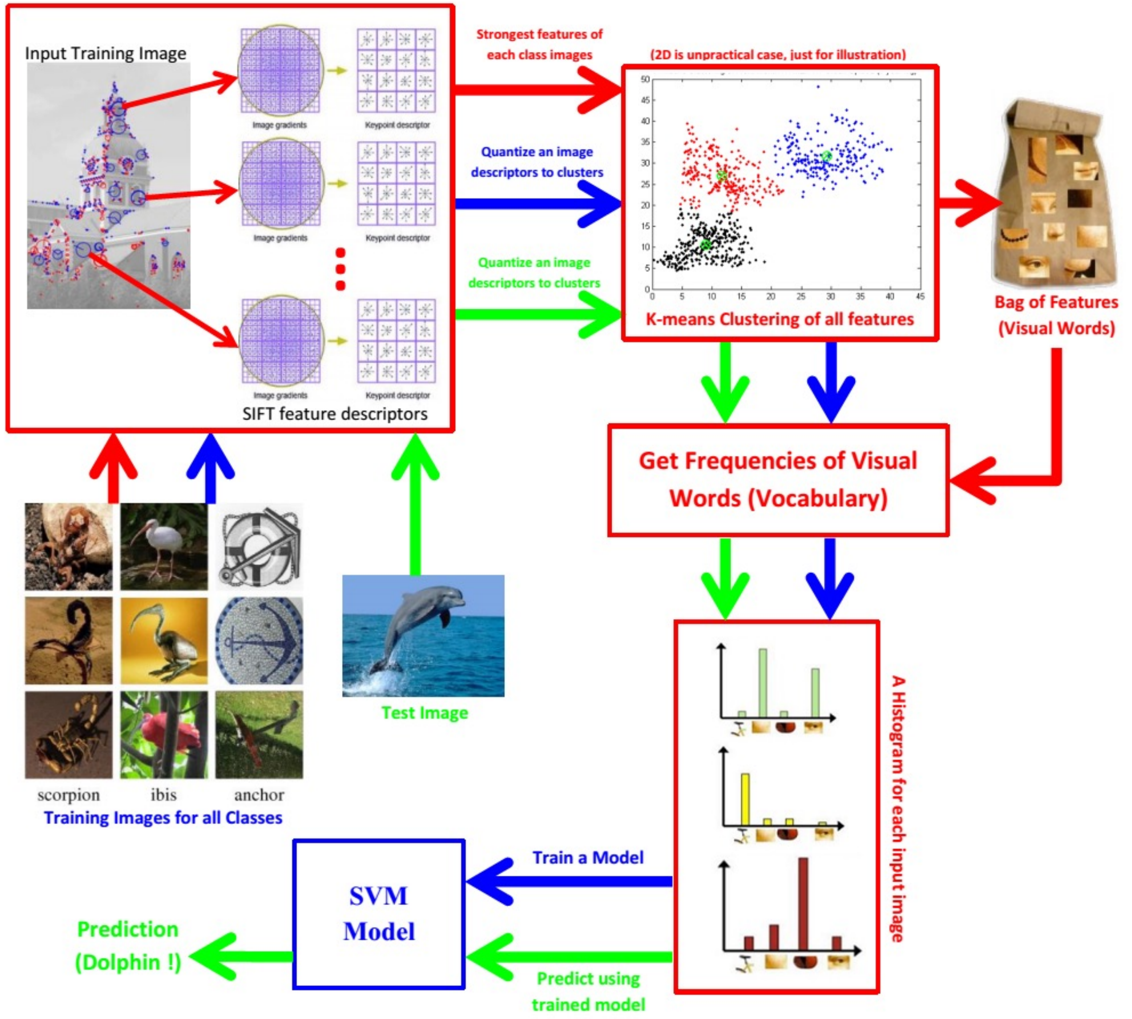
- SIFT-based texture classification

 Build vocabulary

 Train classifier

 Classify image

http://heraqi.blogspot.com/2017/03/BoW.html

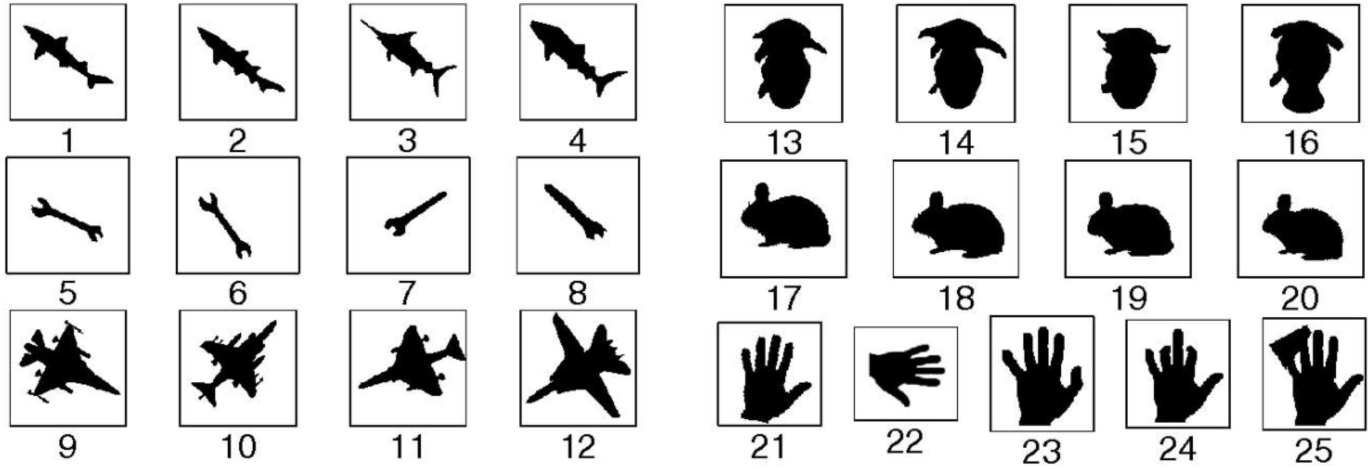


Feature encoding final notes

- Local features can be other types of features (not just SIFT)
  - LBP, SURF, BRIEF, ORB
- There are also more advanced techniques than BoW
  - VLAD, Fisher Vector
- A very good source of additional information is VLFeat.org
  - http://www.vlfeat.org/

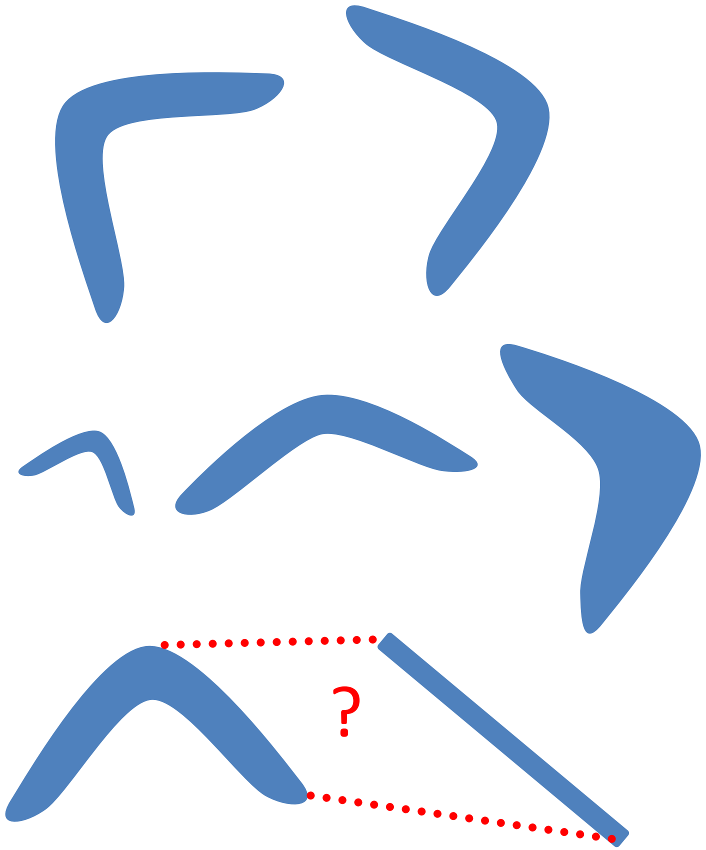
Shape features

- Shape is an essential characteristic of material objects
- Shape features are typically extracted after image segmentation
- They can be used to identify and classify objects
- Example: object recognition



Shape features

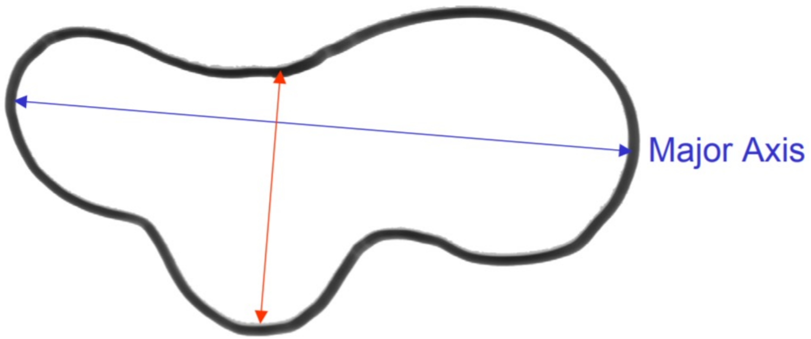
- Challenges in defining shape features
  - Invariant to rigid transformations
  - Tolerant to non-rigid deformations
  - Unknown correspondence



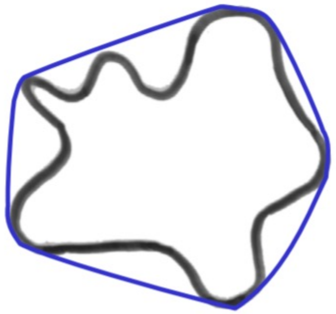
Basic shape features

- Simple geometrical shape descriptors

Net Area



Principal Axes

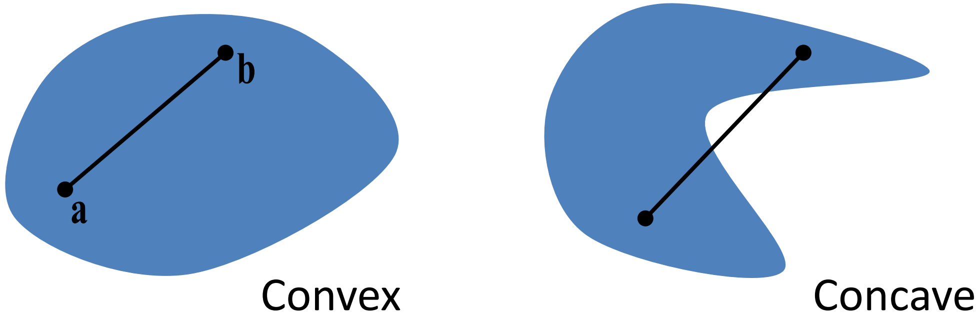


Convex Area = Area of the convex hull that encloses the object

Basic shape features

- Convexity versus concavity of an object

  An object is called convex (or concave) if the straight line between any two points in the object is (or is not) contained in the object

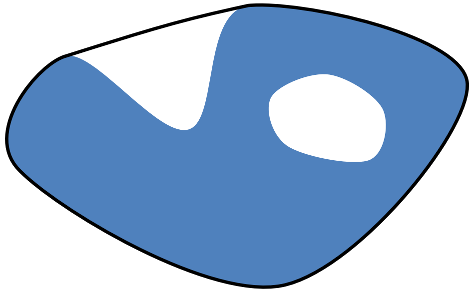


- Convex hull of an object

  The smallest convex set that contains the object

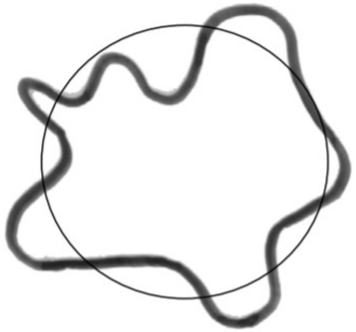
- Convex deficiency of an object

  Set difference between the convex hull and the object



Basic shape features

- Simple geometrical shape descriptors



Compactness: $\xleftrightarrow{\text{inversely related}}$ Circularity:

Ratio of the area of a circle with the same perimeter as the object to the area of the object

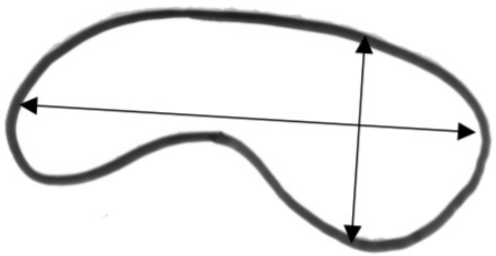
Ratio of 4π times the area of an object to the second power of its perimeter ($4\pi A/P^2 = 1$ for a circle)

Basic shape features

- Simple geometrical shape descriptors

Elongation:

Ratio between the length and width of the object's bounding box

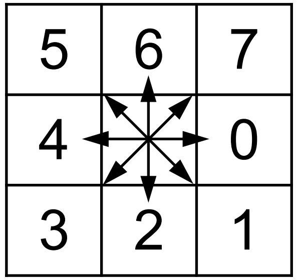


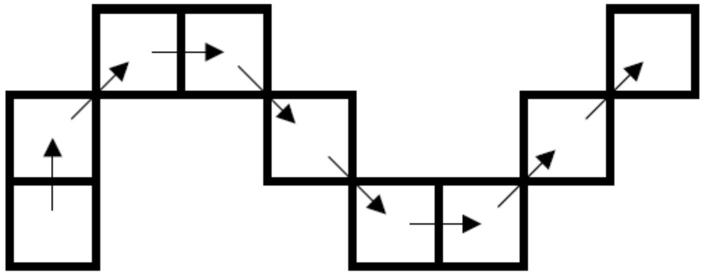
Eccentricity:

Ratio of the length of the minor axis to the length of the major axis

Boundary descriptors

- Chain code descriptor
  - Represents object shape by the relative positions of consecutive boundary points
  - Consists of a list of directions from a starting point
  - Provides a compact boundary representation



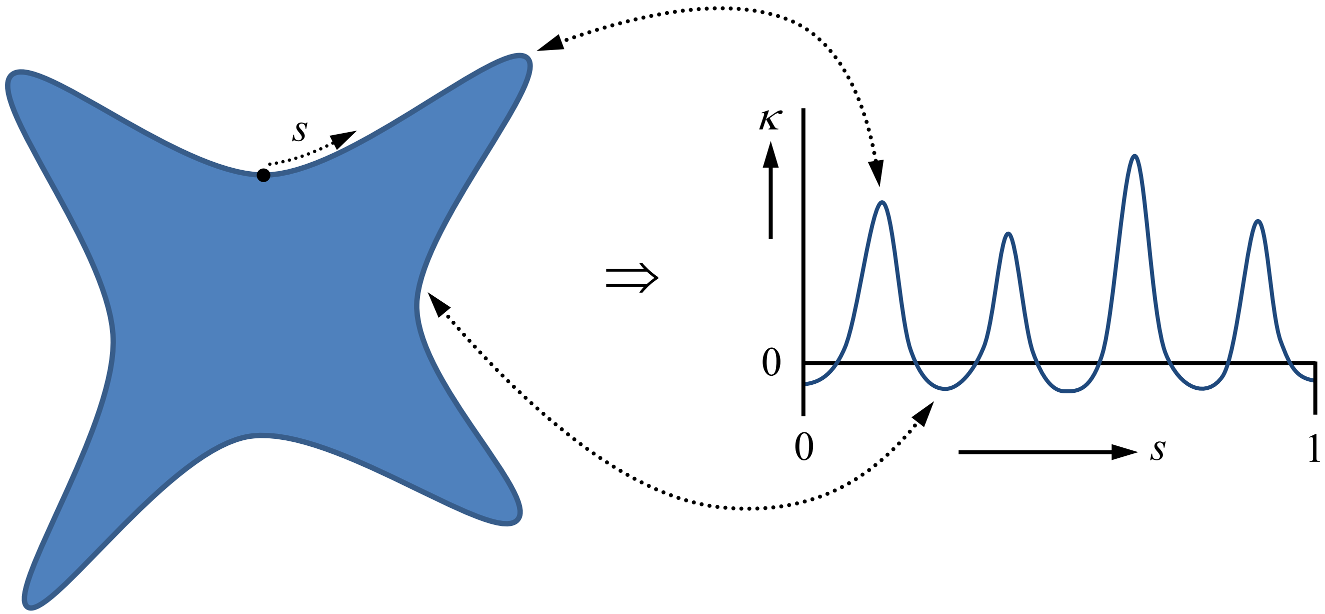


Example:

6,7,0,1,1,0,7,7

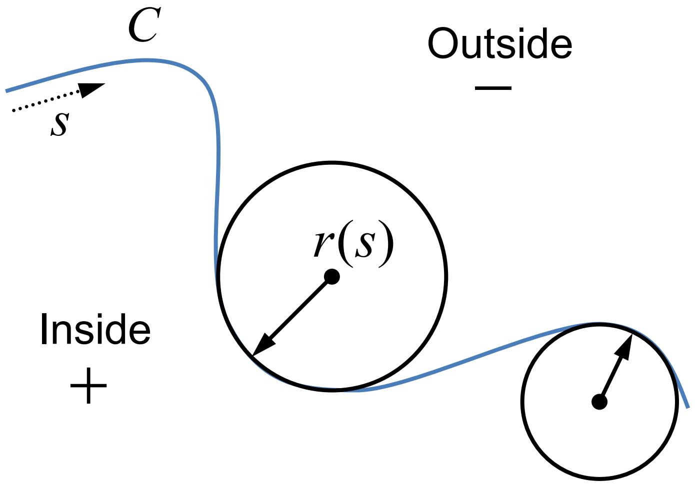
Boundary descriptors

- Local curvature descriptor
  - The curvature of an object is a local shape attribute
  - Convex (versus concave) parts have positive (versus negative) curvature



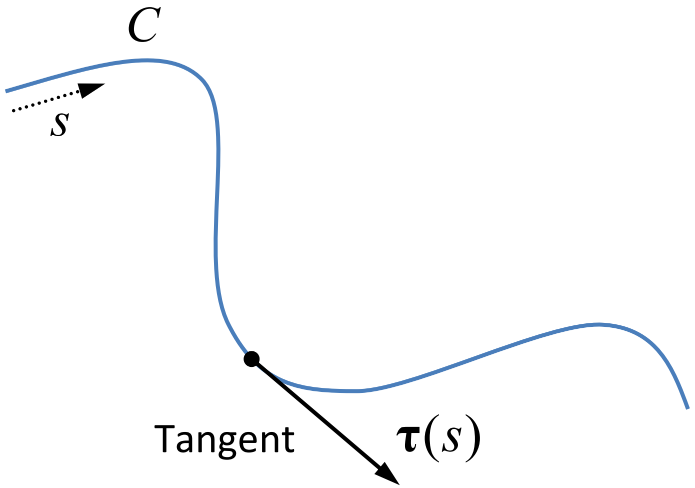
Boundary descriptors

- Two interpretations of local curvature



Geometrical interpretation

$$\kappa(s) = \pm \frac{1}{r(s)}$$



Physical interpretation

$$\kappa(s) = \pm \left\| \frac{d\boldsymbol{\tau}}{ds}(s) \right\|$$

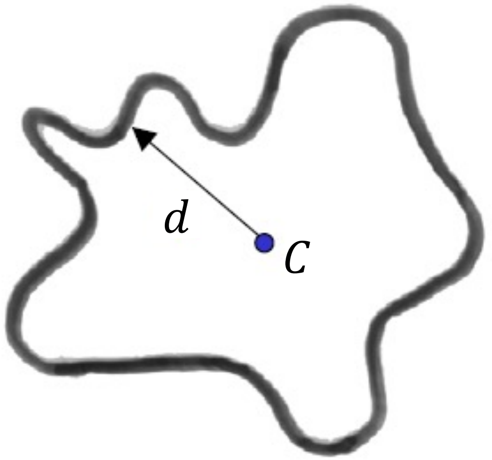
Boundary descriptors

- Global curvature descriptors
  - Total bending energy: $B = \oint_C \kappa^2(s)ds$
    > Amount of physical energy stored in a rod bent to the contour
    > Circular objects have the smallest contour bending energy $B = 2\pi/r$
  - Total absolute curvature: $K = \oint_C |\kappa(s)|ds$
    > Absolute value of the curvature integrated along the object contour
    > Convex objects have the smallest total absolute curvature $K = 2\pi$

Boundary descriptors

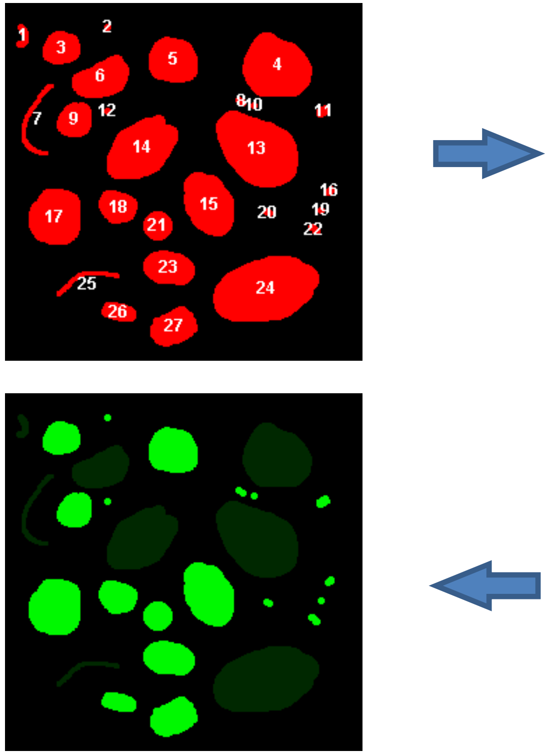
- Radial distance descriptor
  - Use the centroid C of the shape as the reference point and compute the radial distance d for all N pixels i along its boundary
  - $d(i) = \sqrt{\left((x(i) - \bar{\mathrm{x}})^2 + (y(i) - \bar{\mathrm{y}})^2\right)}$

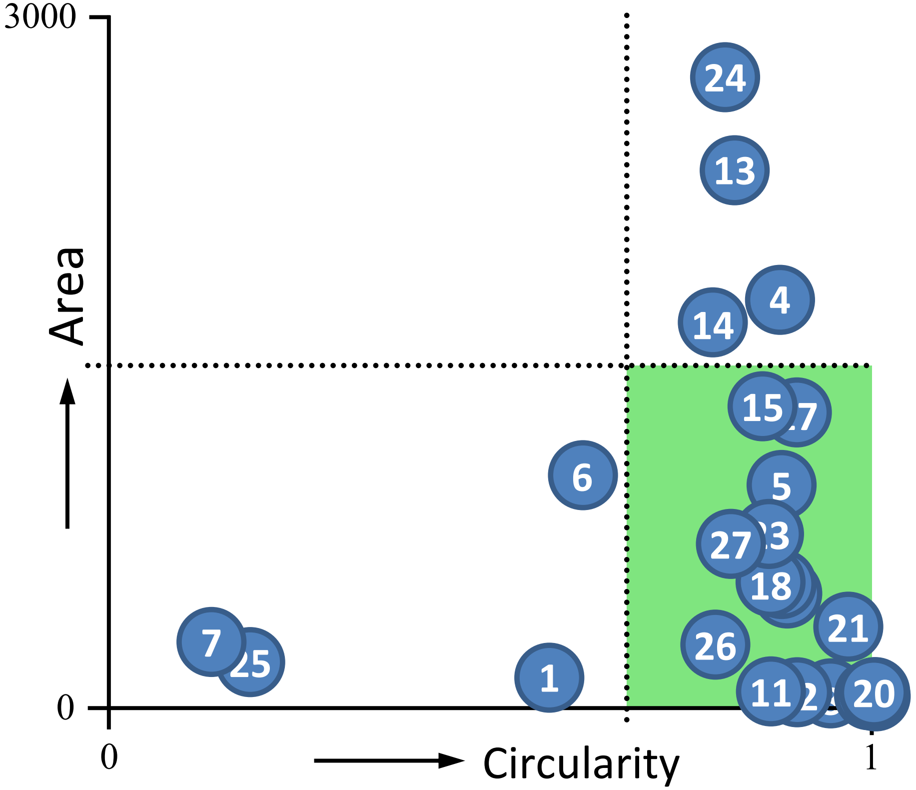
    $\text{for i} = 0, 1, 2, \ldots, \text{N-1}$
  - Scale invariance is achieved by normalizing $d(i)$ by the maximum distance to obtain the radial distance $r(i)$
  - The number of times the signal $r(i)$ crosses its mean can be used as a measure of boundary roughness



Example application of shape features

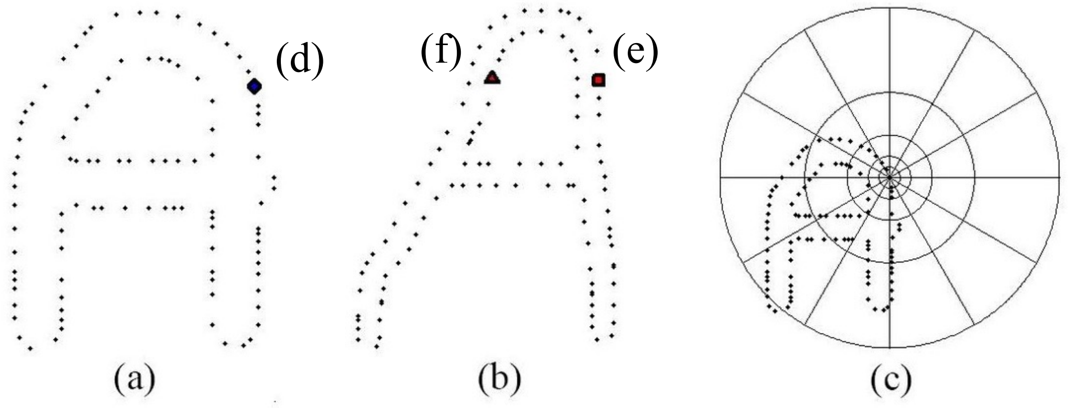
- Combining feature descriptors to classify objects



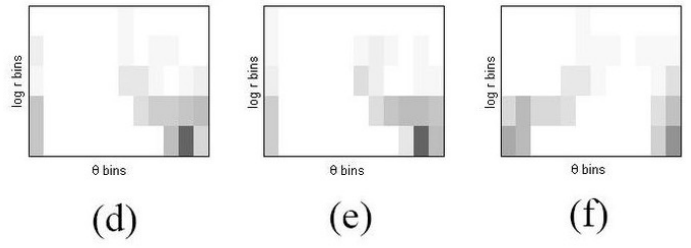


Shape context

- Shape context is a point-wise local feature descriptor
  - Pick n points p_i on the contour of a shape
  - For each point, create a radial coordinate system centred at this point and compute a histogram h_i based on the relative coordinates of the other $n - 1$ points
  - This is the shape context of p_i



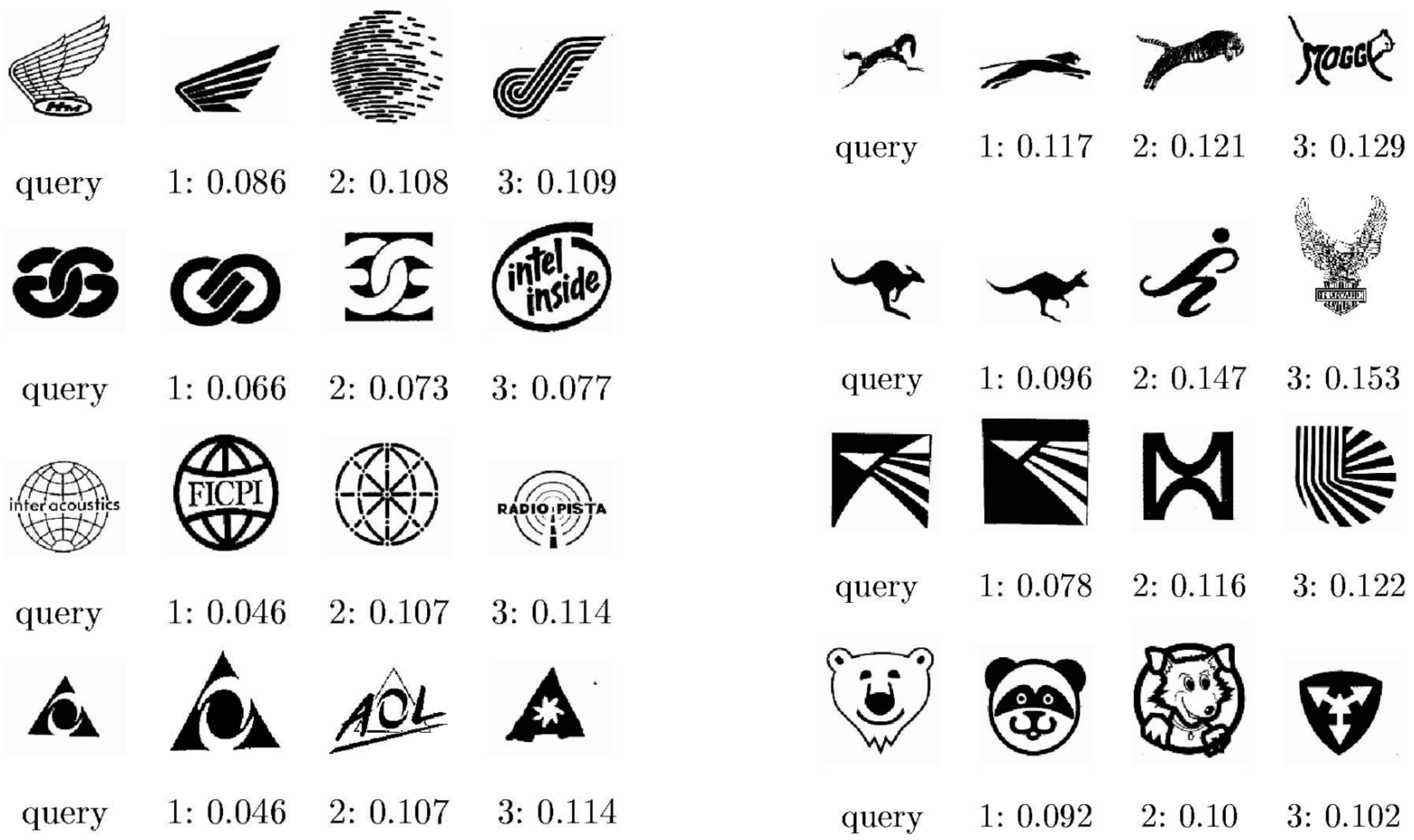
$$h_i(k) = \#\{q \neq p_i : (q - p_i) \in \text{bin}(k)\}$$



Belongie et al. (2002). *Shape matching and object recognition using shape contexts.* IEEE TPAMI 24(4):509-522. https://doi.org/10.1109/34.993558

Example application of shape context

- Shape matching



Example application of shape context

- Shape matching
  - Step 1: Sample a list of points on shape edges

    For example from Canny edge detector:

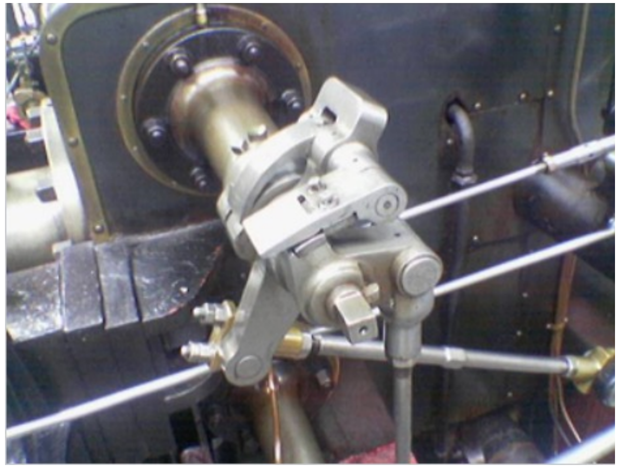
    > Gaussian filtering

    > Intensity gradient

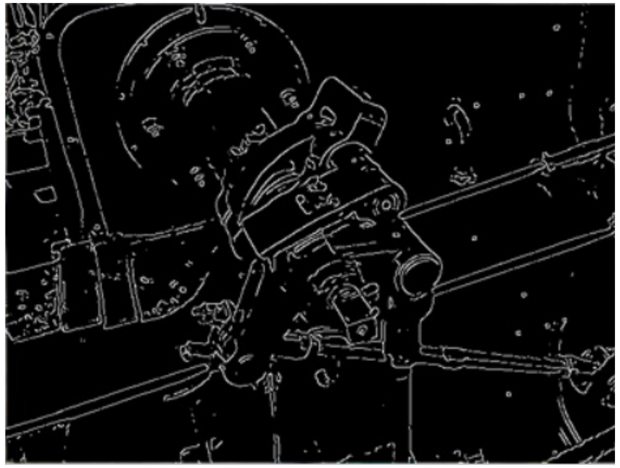
    > Non-maximum suppression

    > Hysteresis thresholding

    > Edge tracking



$\Rightarrow$



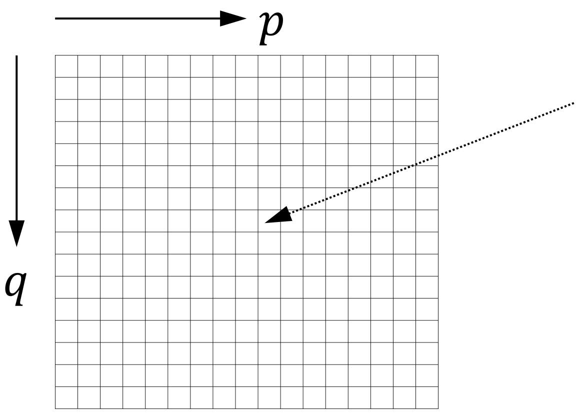
J. Canny (1986). *A computational approach to edge detection*. IEEE TPAMI 8(6):679-698. https://doi.org/10.1109/TPAMI.1986.4767851

Example application of shape context

- Shape matching
  - Step 2: Compute the shape context for each point

$$h_i(k) = \#\{q \neq p_i : (q - p_i) \in \mathrm{bin}(k)\}$$

  - Step 3: Compute the cost matrix between two shapes P and Q



$$C(p_i, q_j) = \frac{1}{2} \sum_{k=1}^{K} \frac{\left[h_i(k) - h_j(k)\right]^2}{h_i(k) + h_j(k)}$$

h_i is the shape context of $p_i \in P$

h_j is the shape context of $q_j \in Q$

Example application of shape context

- Shape matching
  - Step 4: Find the one-to-one matching minimising the total cost between point pairs

$$H(\pi) = \sum_i C\left(p_i, q_{\pi(i)}\right)$$

  - Step 5: Transform one shape to the other based on the one-to-one point matching
    - Choose the desired transformation (for example affine)
    - Apply least-squares or RANSAC fitting
    - This yields the optimal transformation T

Example application of shape context

- Shape matching
  - Step 6: Compute the shape distance

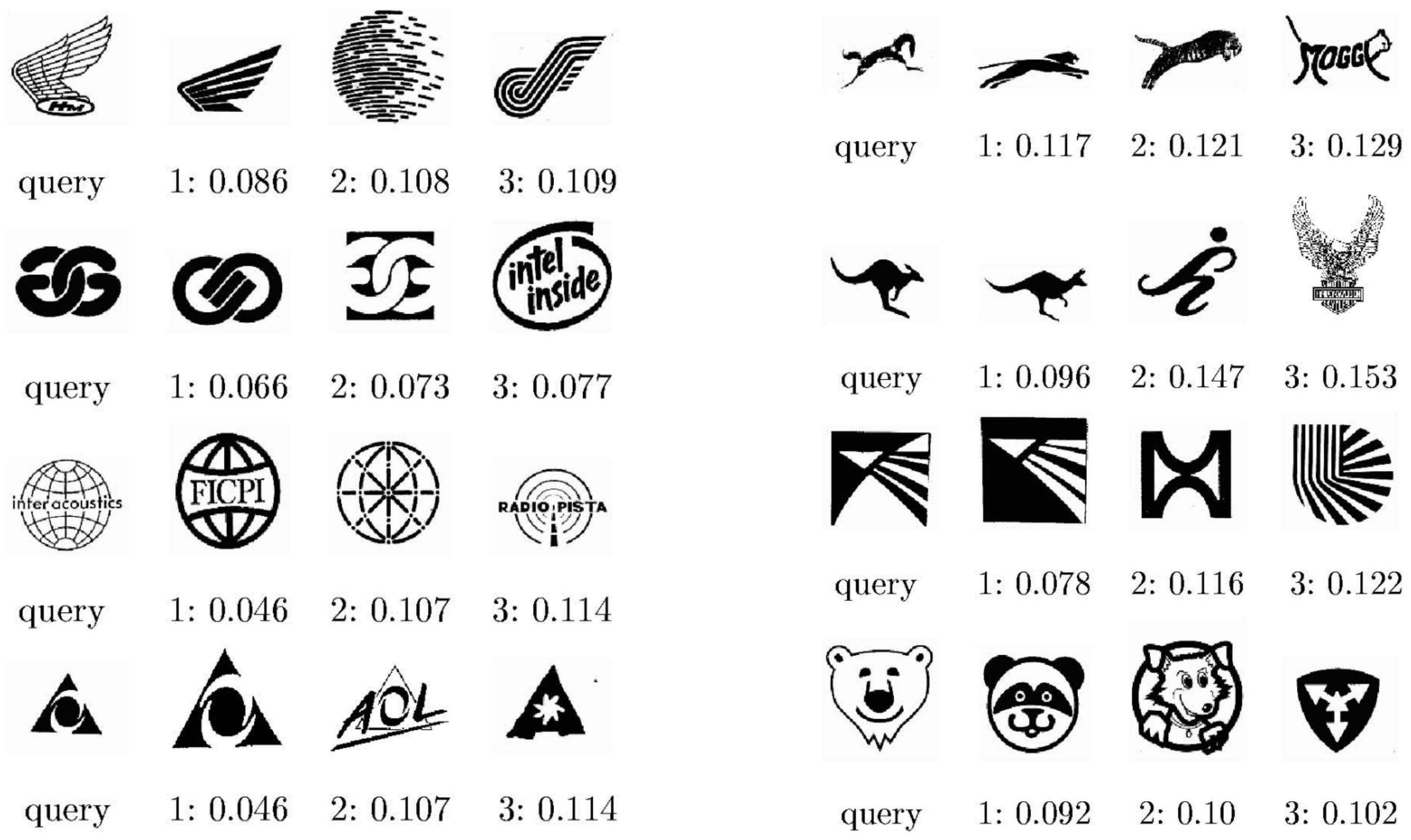
$$D(P,Q) = \frac{1}{n}\sum_{p\in P}\min_{q\in Q} C(p,T(q)) + \frac{1}{m}\sum_{q\in Q}\min_{p\in P} C(p,T(q))$$

  - Other costs may also be taken into consideration
    > Appearance of the image at the points
    > Bending energy of the transformation

Example application of shape context

- Shape matching

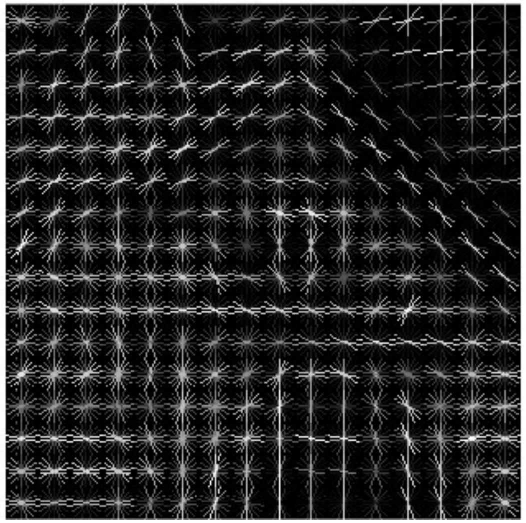
  1. Sample points
  2. Compute shape context
  3. Compute cost matrix
  4. Find point matching
  5. Perform transformation
  6. Compute distance



Histogram of oriented gradients

- Histogram of oriented gradients popularly referred to as HOG
- Describes the distributions of gradient orientations in localized areas
- Does not require initial segmentation

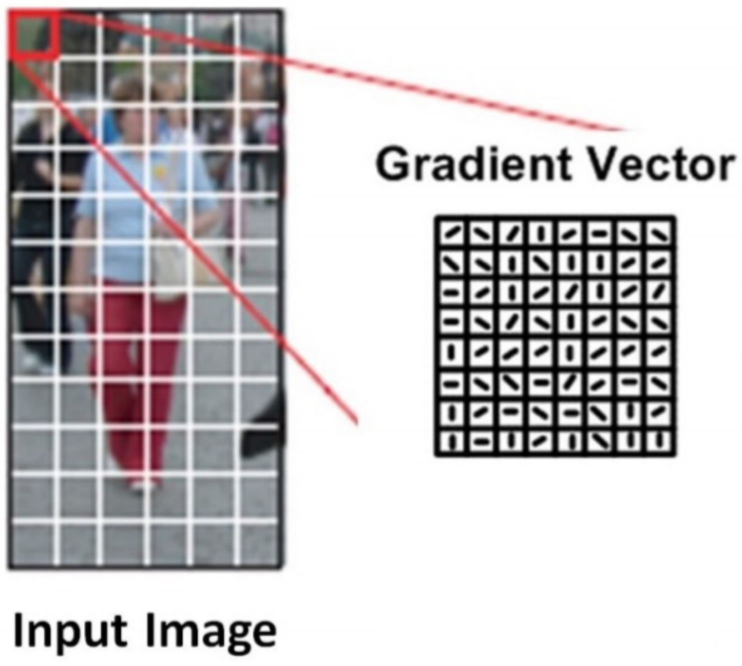
$\Rightarrow$



N. Dalal and B. Triggs
Histograms of oriented gradients for human detection
Computer Vision and Pattern Recognition 2005
https://doi.org/10.1109/CVPR.2005.177

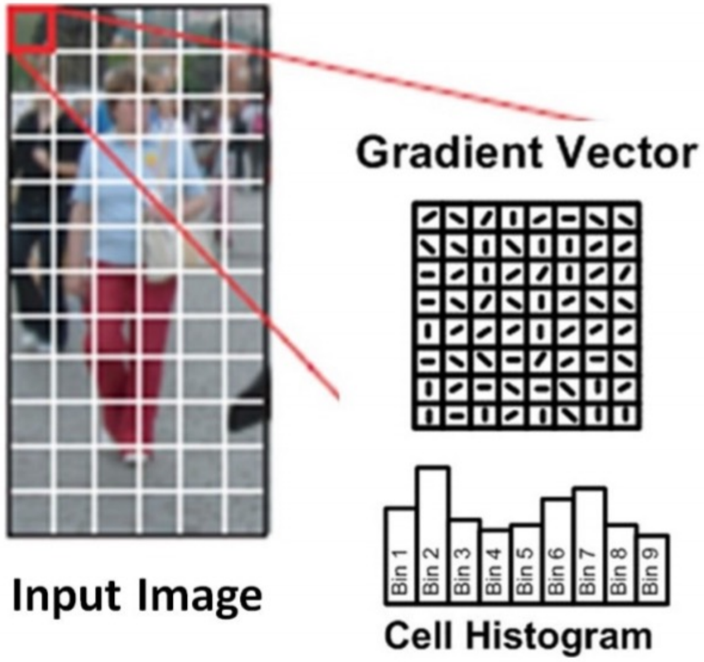
Histogram of oriented gradients

- Step 1: Calculate the gradient vector at each pixel
  - Gradient magnitude
  - Gradient orientation



Histogram of oriented gradients

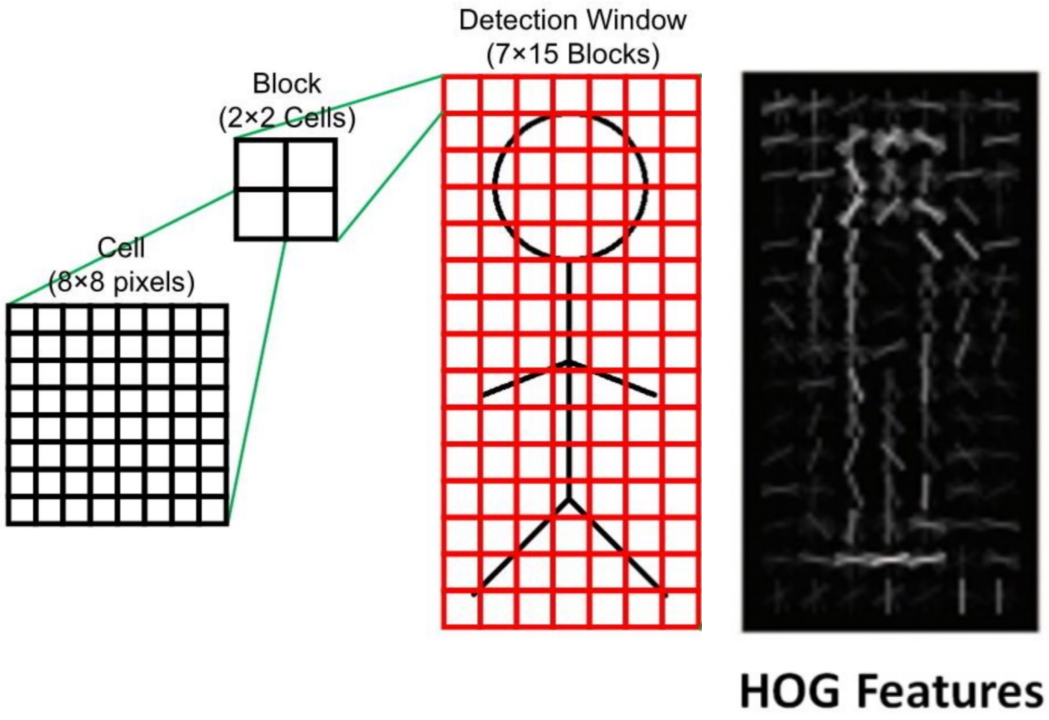
- Step 2: Construct the gradient histogram of all pixels in a cell
  - Divide orientations into N bins (typically $N = 9$ bins evenly splitting 180 degrees)
  - Assign the gradient magnitude of each pixel to the bin corresponding to its orientation



Histogram of oriented gradients

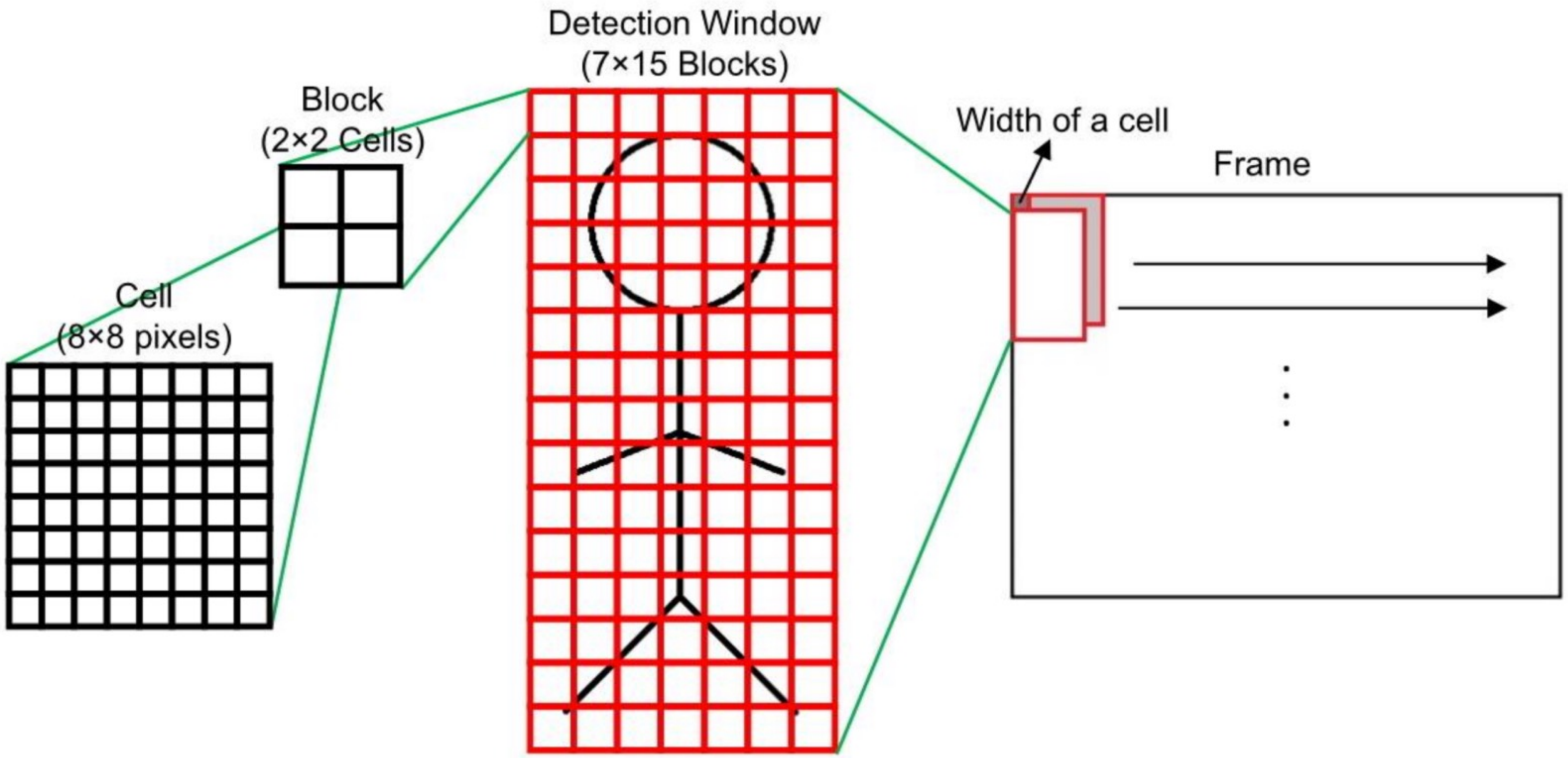
- Step 3: Generate detection-window level HOG descriptor
  - Concatenate cell histograms
  - Block-normalise cell histograms

$$\# \text{features} = \underbrace{(7 \times 15)}_{\#\text{ blocks}} \times \overbrace{9}^{\#\text{ orientations/cell}} \times \underbrace{4}_{\#\text{ cells/block}} = 3{,}780$$



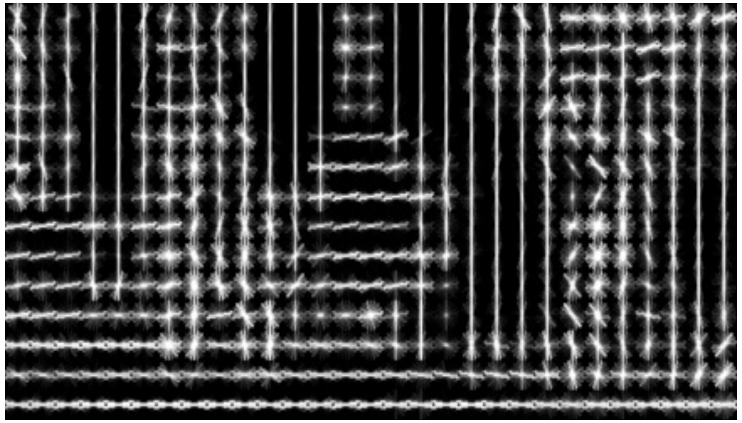
Histogram of oriented gradients

- Detection via sliding window on the image

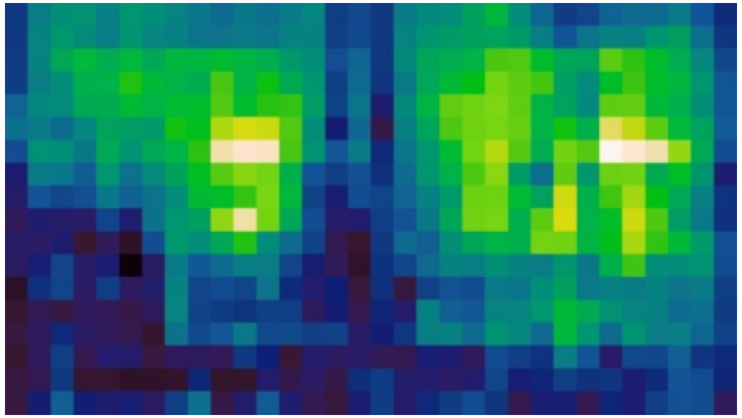


Histogram of oriented gradients

- Detection via sliding window on the image
  - Compute the HOG descriptor for many example windows from a training dataset
  - Manually label each example window as either “person” or “background”
  - Train a classifier (such as SVM) from these example windows and labels
  - For each new (test) image predict the label of each window using this classifier



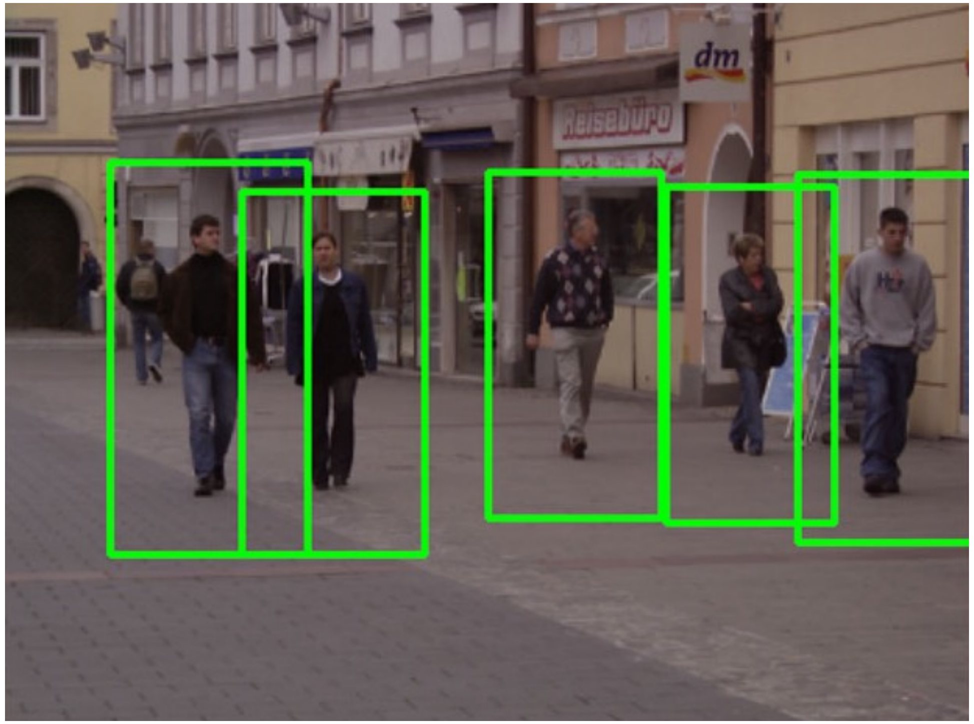
HOG feature map



Detector response map

Example application of HOG

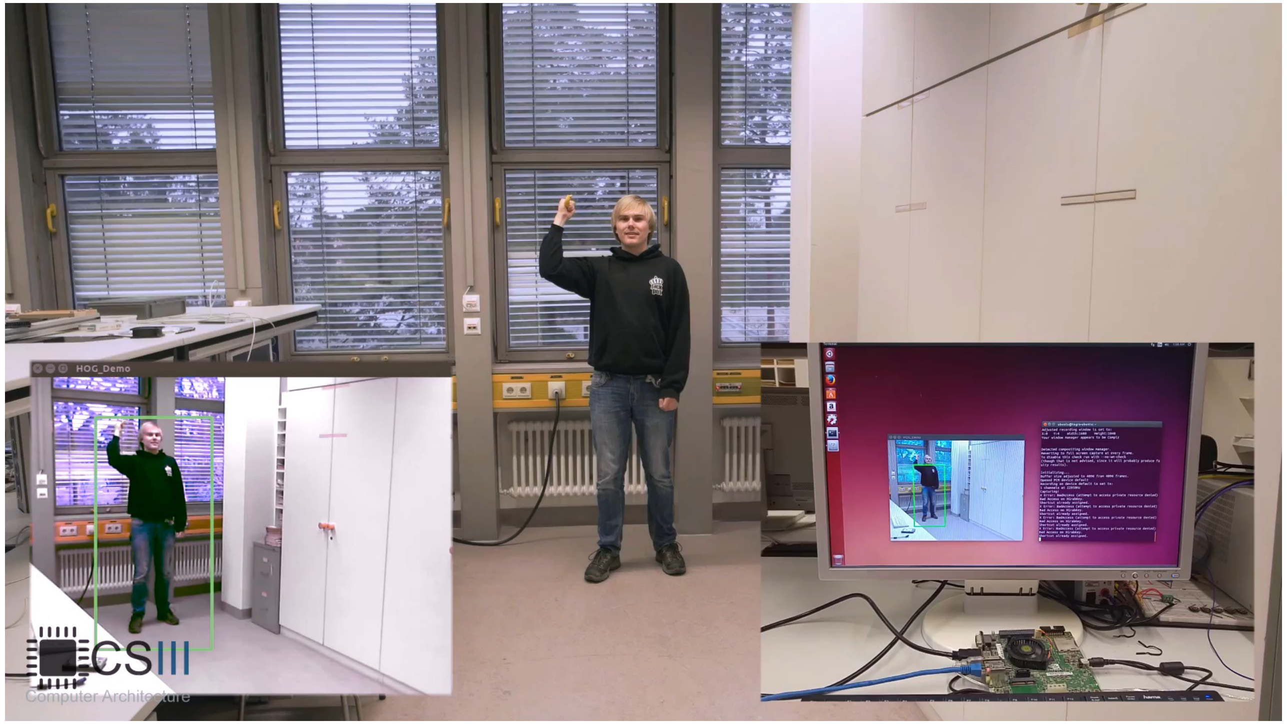
- Detecting humans in images



https://www.pyimagesearch.com/2015/11/09/pedestrian-detection-opencv/

Example application of HOG

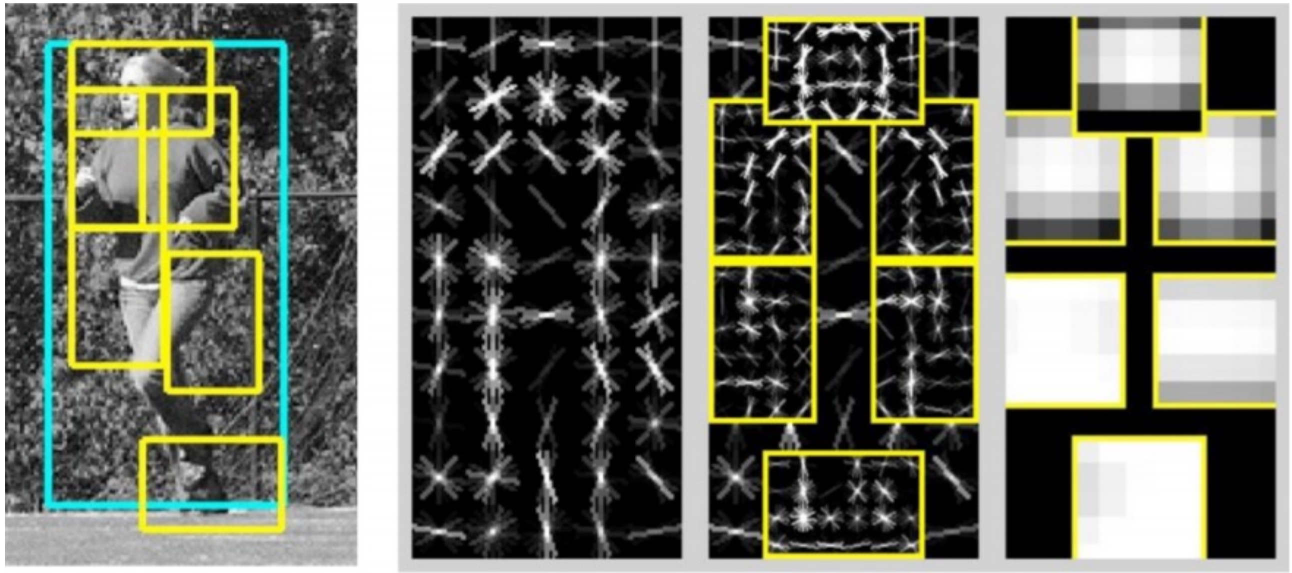
- Detecting and tracking humans in videos



https://www.youtube.com/watch?v=0hMMRIB9DUc

Example application of HOG

- Fine-grained detection using deformable parts model



detection root filter part filters deformation models

https://doi.org/10.1109/CVPR.2008.4587597

Summary

- Feature representation is essential in solving many computer vision problems
- Most commonly used image features:
  - Colour features (Part 1)

    Colour moments and histogram
  - Texture features (Part 1)

    Haralick, LBP, SIFT
  - Shape features (Part 2)

    Basic, shape context, HOG

Summary

- Other techniques discussed (Part 1)
  - Descriptor matching
  - Least squares and RANSAC
  - Spatial transformations
  - Feature encoding (BoW)
  - K-means clustering
  - Shape matching
  - Sliding window detection

Further reading on discussed topics

- Chapters 4 and 6 of Szeliski

Acknowledgements

- Some content from slides of James Hays, Michael A. Wirth, Cordelia Schmit
- From BoW to CNN: Two decades of texture representation for texture classification
- And other resources as indicated by the hyperlinks

Example exam question

Given the image on the right showing the result of a segmentation of various objects and the desired classification of these objects. The two different colours (red and green) indicate the two different classes which the objects are to be assigned to. A straightforward way to perform classification is by computing the value of a quantitative shape measure for each object and then thresholding those values. Suppose we compute the circularity and the eccentricity. Which of these two measures can be used to produce the shown classification?

A. Only circularity

B. Only eccentricity

C. Both circularity and eccentricity

D. Neither circularity nor eccentricity