Computer vision

Object detection/ recognition

Doc. Ing. Vanda Benešová, PhD.

Object detection,

object recognition

Methods of:

Pattern recognition

- Rule based pattern recognition
- Statistical pattern recognition
- Fuzzy pattern recognition

Artificial intelligence

- Feature detection + classification

Neural networks

- Methods of Deep learning
- Convolutional neural networks CNN

Object category vs. object instance

object category detection / recognition :

variation in a category is typically large

generalisation is important

object instance detection / recognition :

the necessity of distinguishing between similar objects

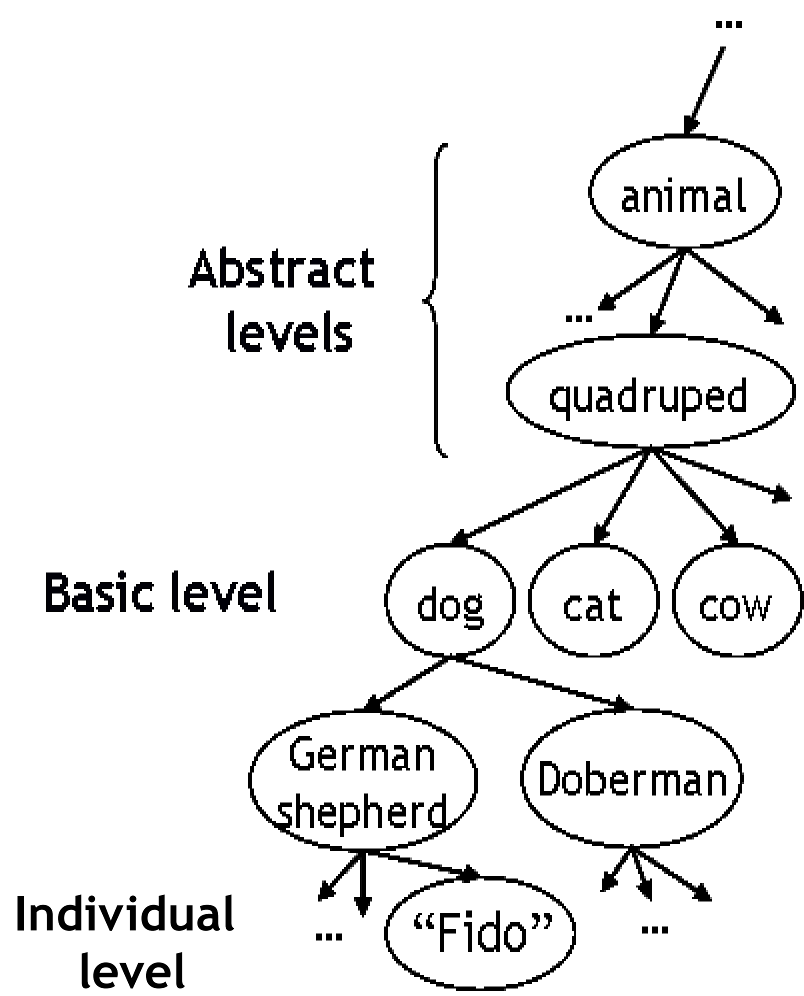
Visual Object Categories

Basic-level categories in humans seem to be defined

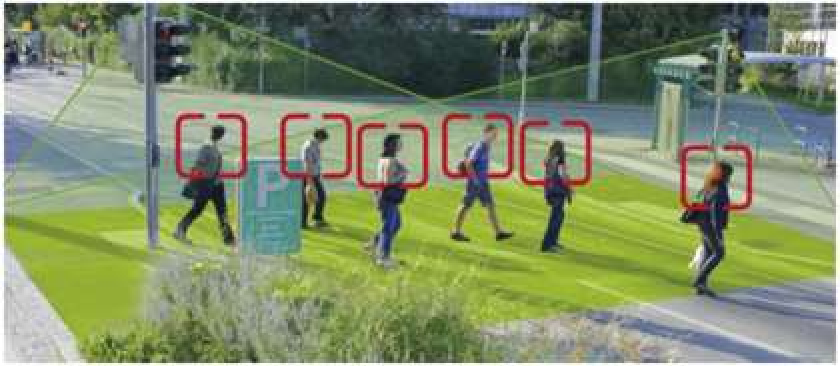
There is evidence that humans (usually) start with basic-level categorization before doing identification.

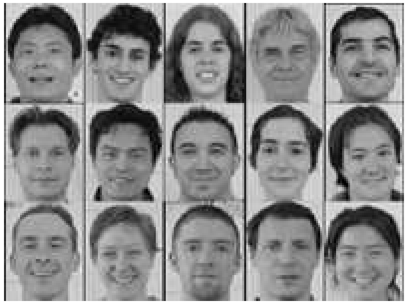
Basic-level categorization is easier and faster for humans than object identification!

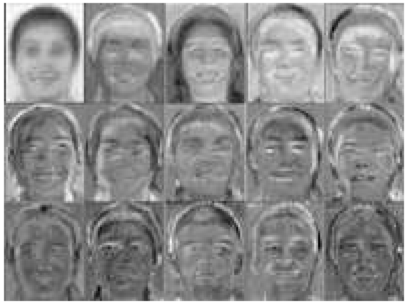
...promising starting point for visual classification



People detection vs. people recognition







Challenges

Invariant to changes in:

Illumination,
camera viewpoint,
occlusion,
object pose,
intra-class variations..

(scale, orientation invariance)

Basic approaches

- Bottom-up approach

part-based representations

Local features detection + recognition

- Top-down approach

Segmentation + object recognition

Global appearance recognition - sliding window (object hypotheses)

Deep learning + Convolutional neural networks CNN

Segmentation + object recognition (intro)

Segmentation + object recognition

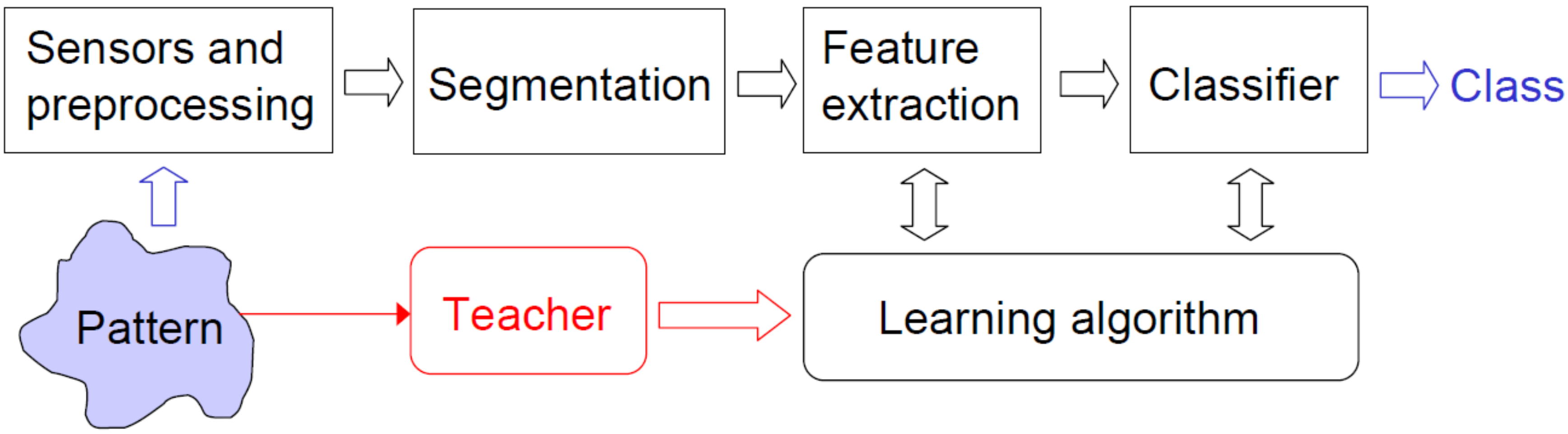
Robust segmentation

- Geometric object (road signs)
- Color dominant object (road signs)

Examples:

Road signs detection, OCR……
(road signs)

Object recognition using segmentation and classification



Global Appearance & Sliding Windows (intro)

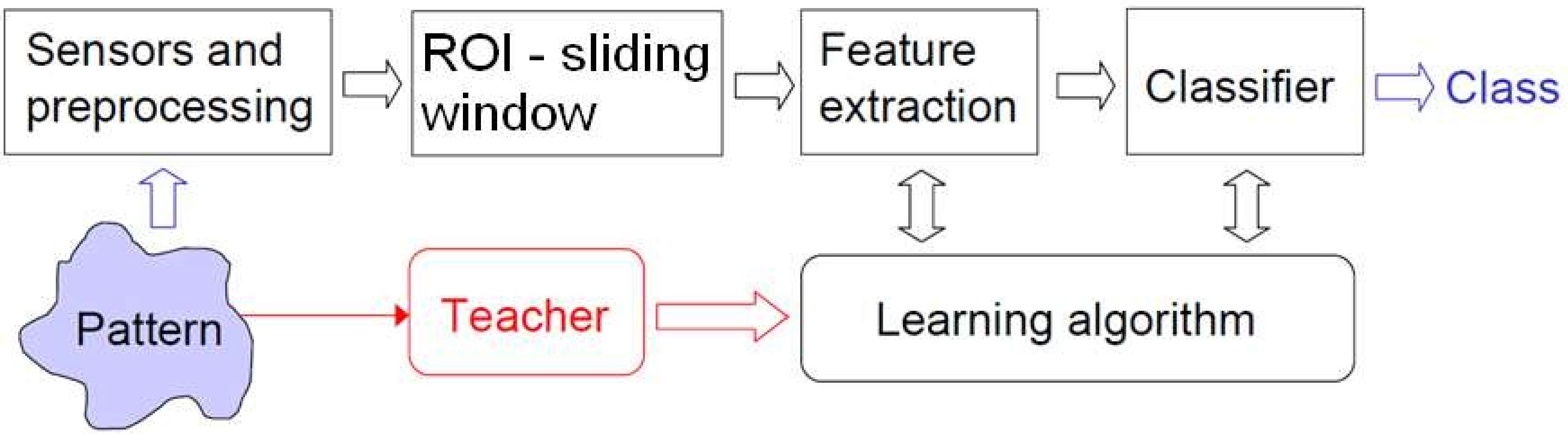
Global Appearance & Sliding Windows

Sliding window

Examples: Face detection,

People detection, ….

Global Appearance & Sliding Windows



ROI: Region of Interest

Global Appearance & Sliding Windows

Binary classification task

The question that answers classifier:

Is in the given window the object? (yes or no?)

Features?

Classificator?

Local descriptors (intro)

Local desriptors

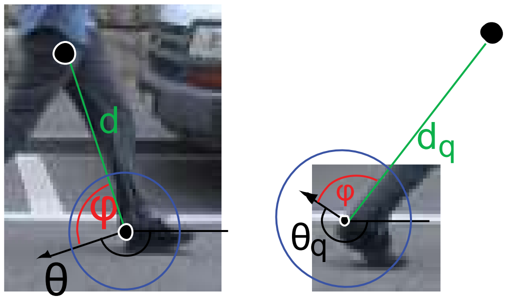
More robust

Occlusions of objects

Changes of camera view position

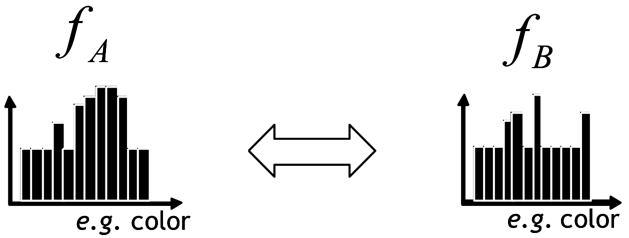
Rotation, scale invariance

Intra category variations



Object detection using local descriptors

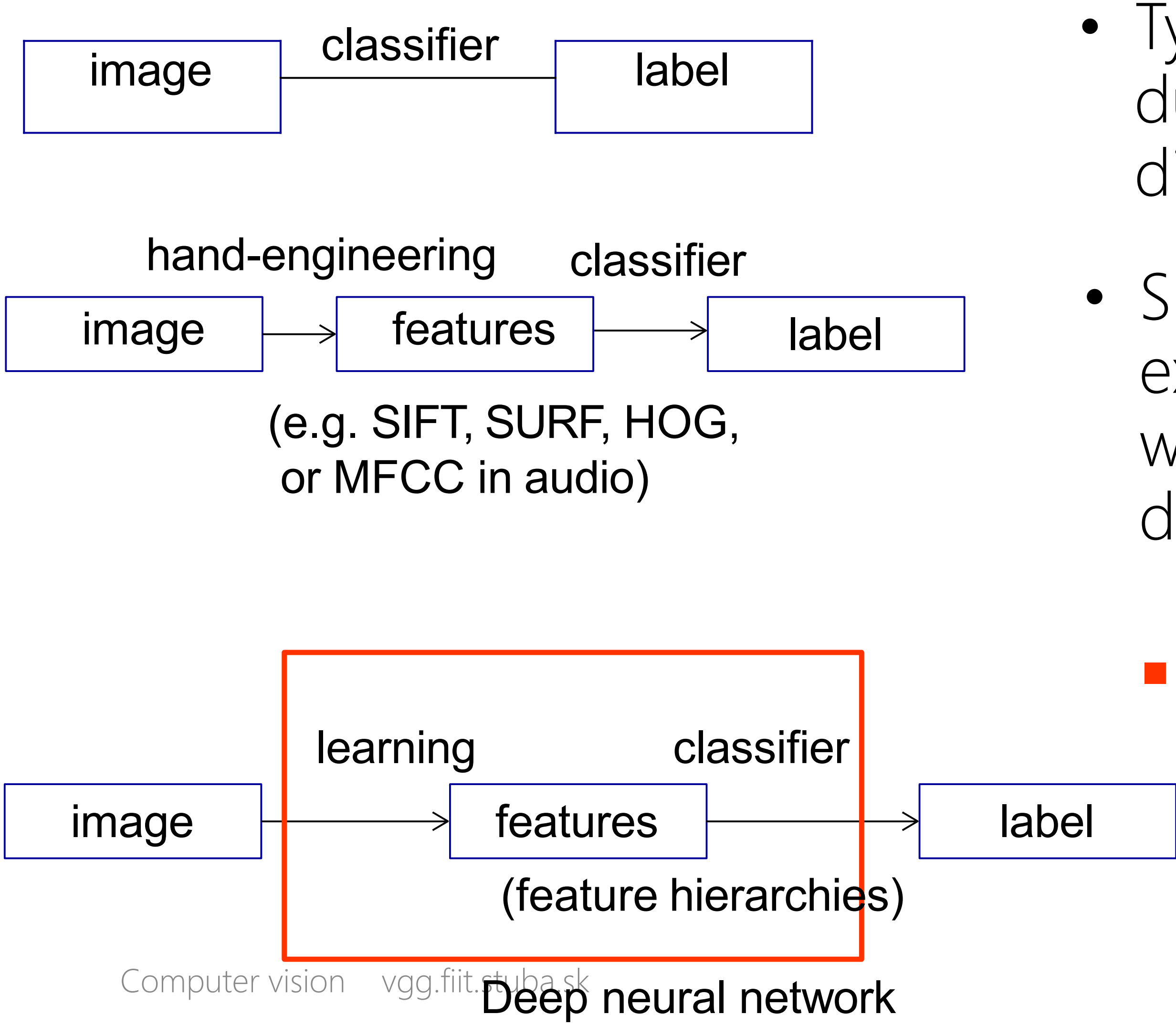
1. Find a set of distinctive key-points
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors



$$d(f_A, f_B) < T$$

Deep Learning (intro)

Classification task and Deep Learning

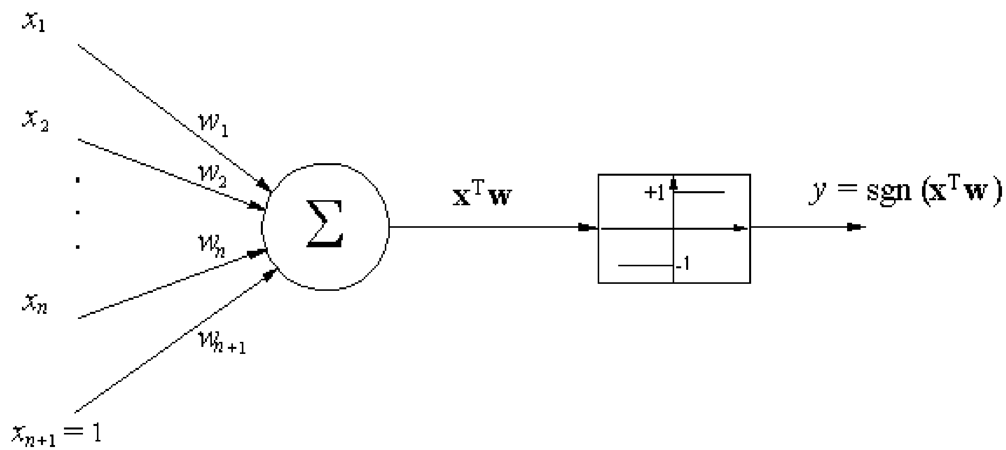


- Typically not feasible, due to high dimensionality
- Suboptimal, requires expert knowledge, works in specific domain only

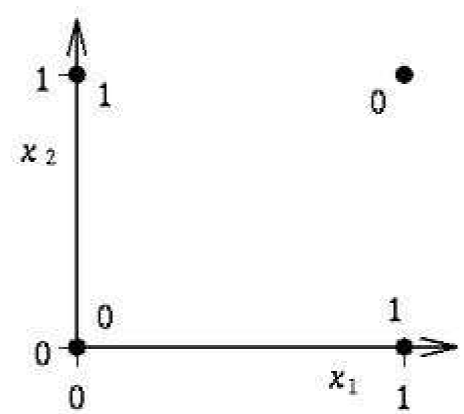
- Deep learning
  = both the classifiers and the features are learned automatically

(Artificial) Neural Networks

- Neural networks are here for more than 50 years
  - Rosenblatt-1956 (perceptron)



Minsky-1969 (xor issue, => skepticism)



Neural Networks

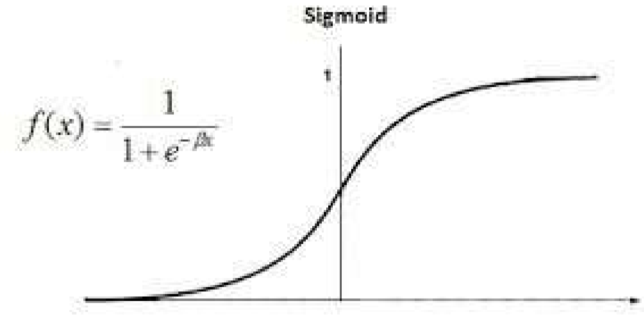
Rumelhart and McClelland – 1986:

Multi-layer perceptron,

Back-propagation (supervised training)

Differentiable activation function

Stochastic gradient descent

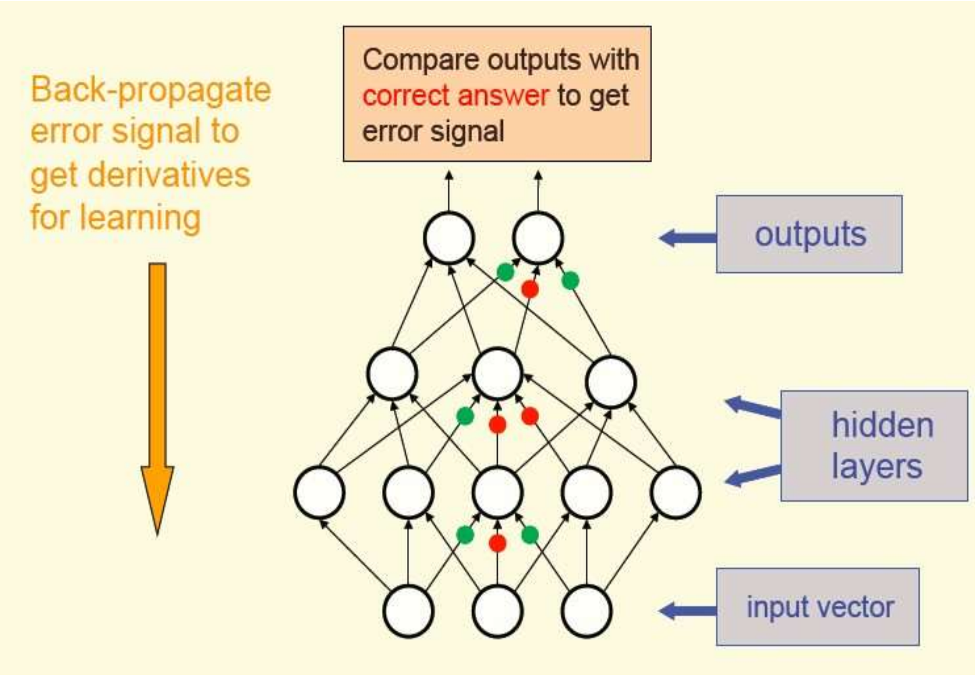


Empirical risk

$$Q(w) = \sum_{i=1}^{n} Q_i(w),$$

Update weights:

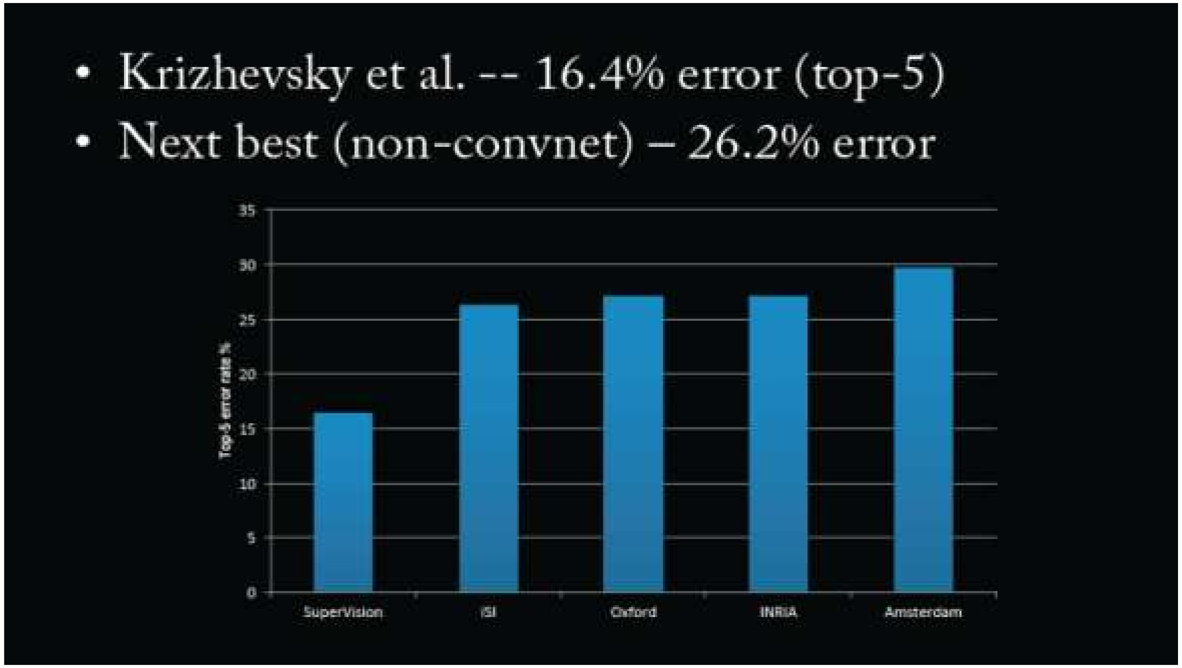
$$w := w - \alpha \nabla Q_i(w).$$



Deep convolutional neural networks

Deep Learning – is a set of machine learning algorithms based on multi-layer networks

Deep convolutional neural networks

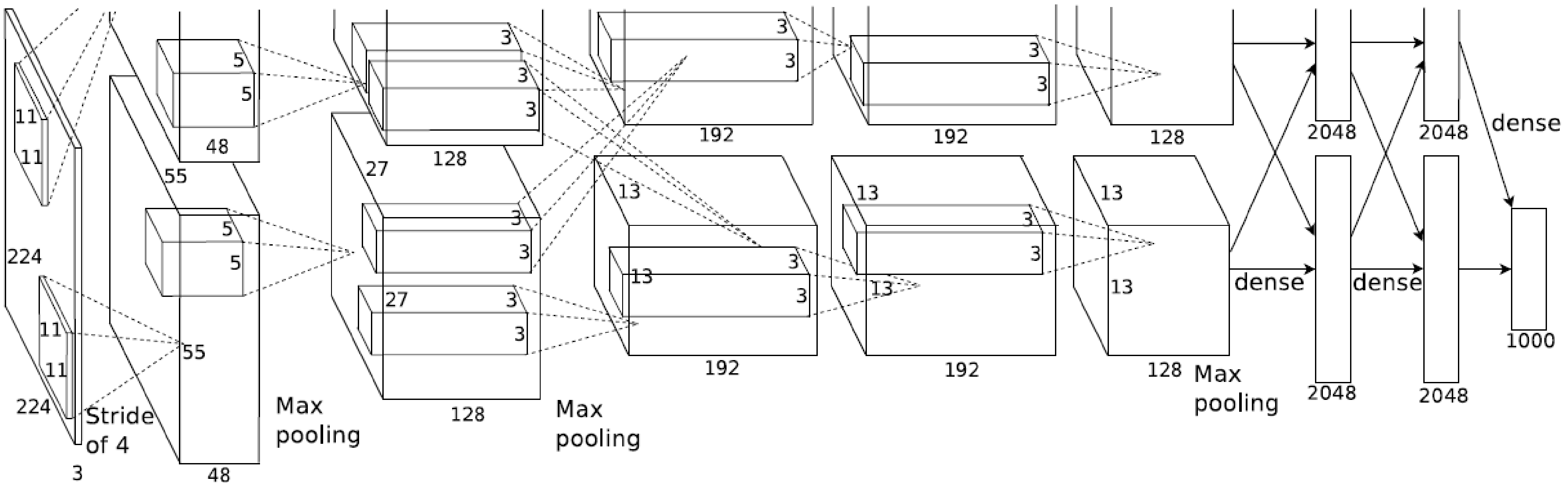


Krizhevsky, Sutskever, Hinton: ImageNet classification with deep convolutional neural networks. NIPS, 2012.

Recognizes 1000 categories from ImageNet

Outperforms state-of-the-art by significant margin (ILSVRC 2012)

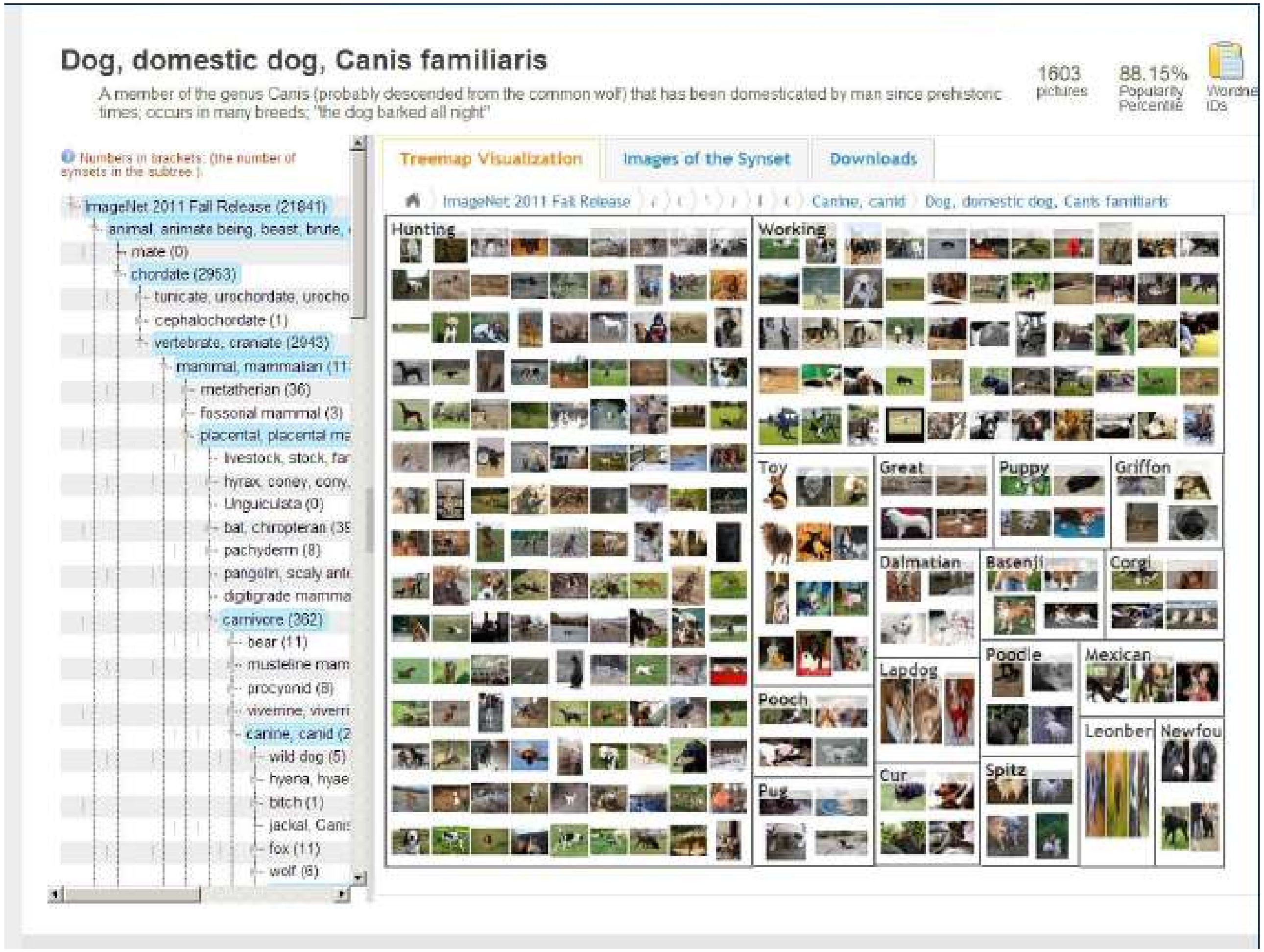
Deep convolutional neural networks



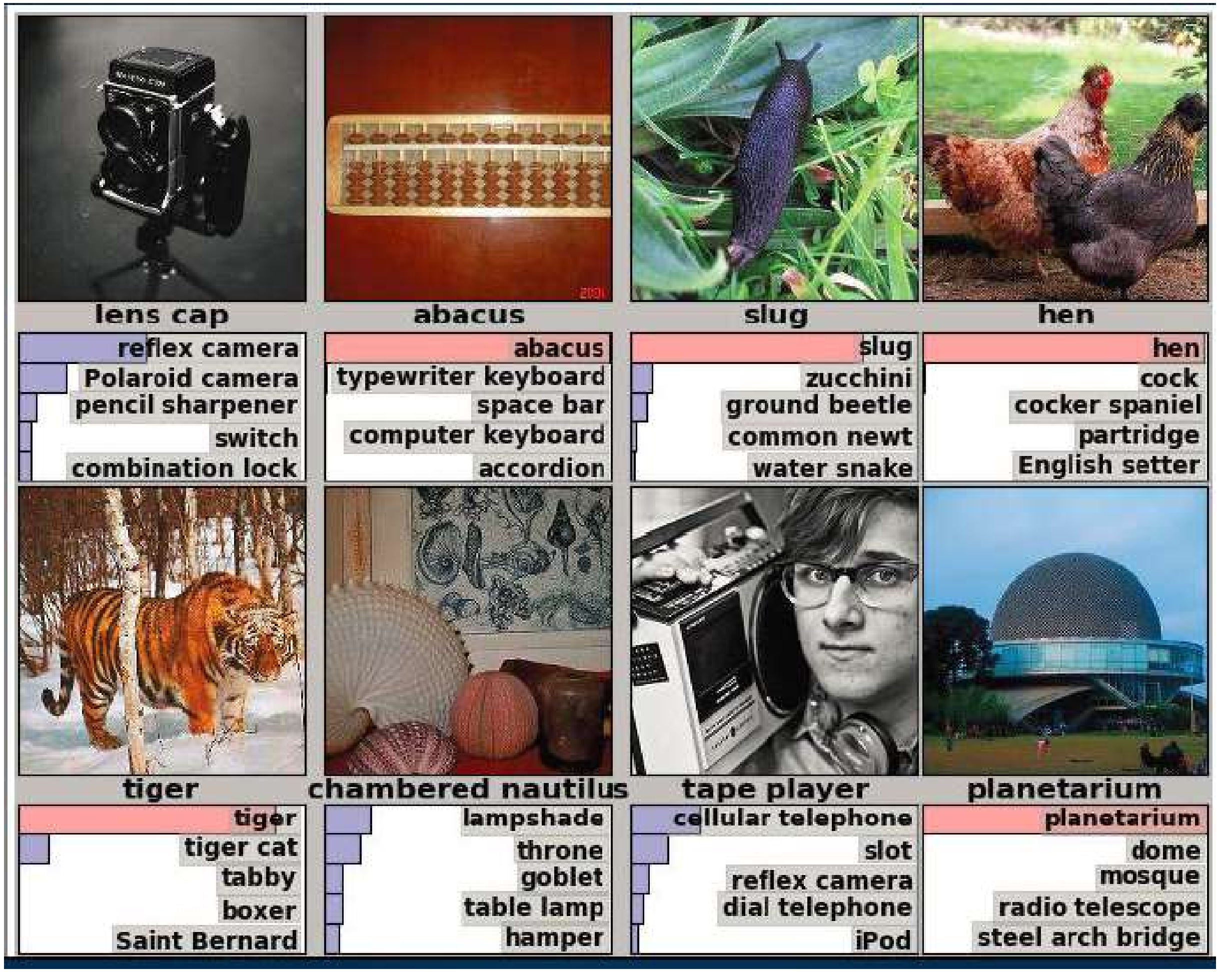
- 5 convolutional layers, 3 fully connected layers
- 60M parameters, trained on 1.2M images (~1000 examples for each category)

CNN story: 2012 - ILSVRC

Imagenet data base: 14 mln labeled images, 20K categories



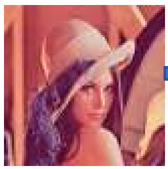
ILSVRC: Classification



Features

Feature vector

Basic Concept of Classification



Feature extraction → Classification → Class label

Features for object detection / recognition

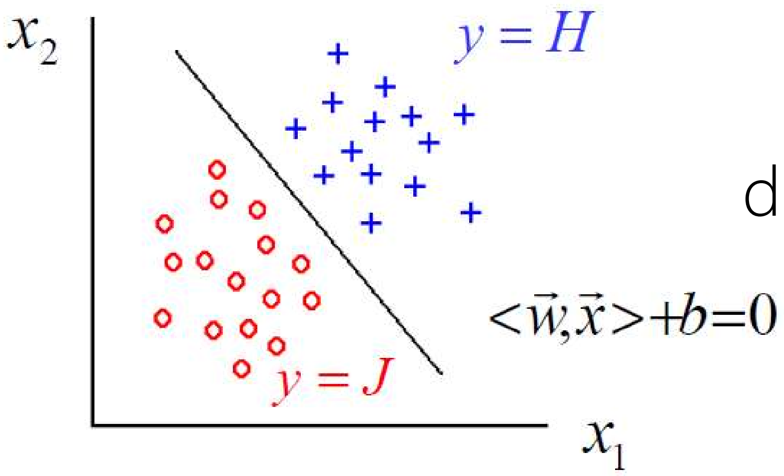
- Colour features
- Shape features
- Texture features
    - Edge features

...others

Feature vector

Feature vector:

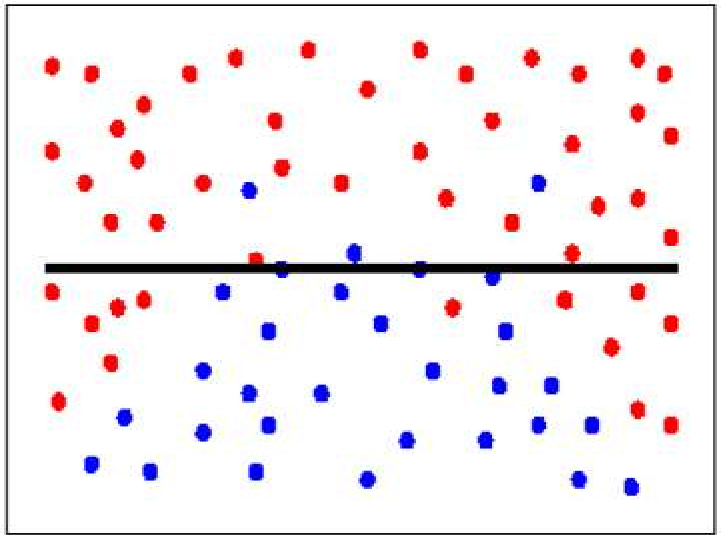
$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

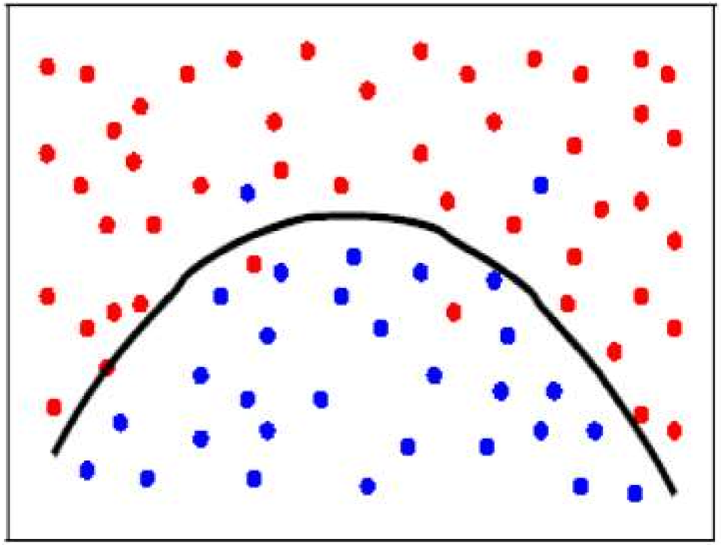


decision boundary - line

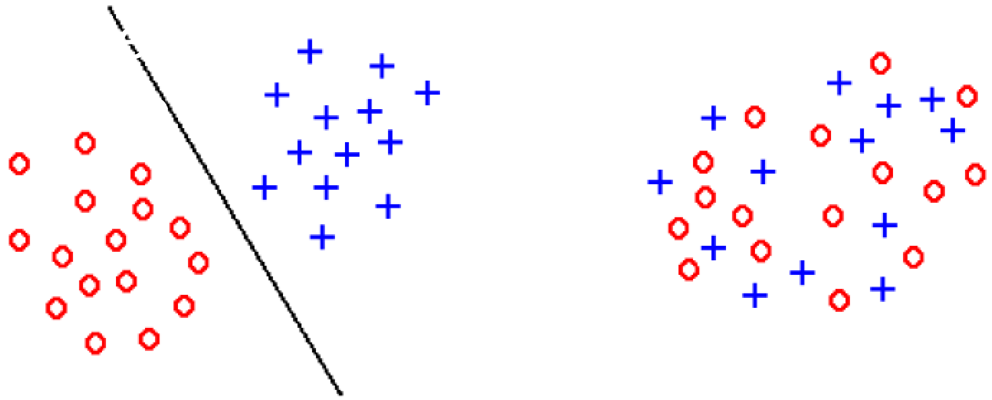
Features

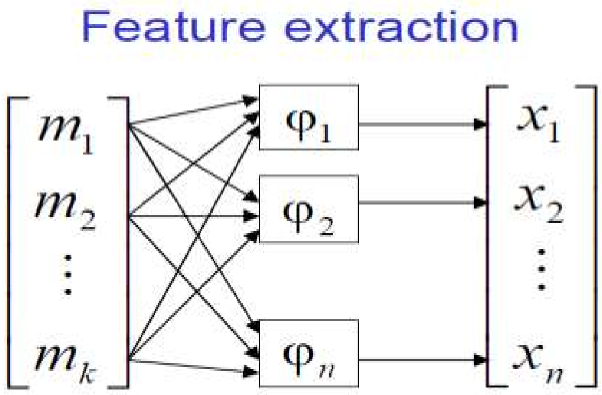
Linear / non linear separable classes

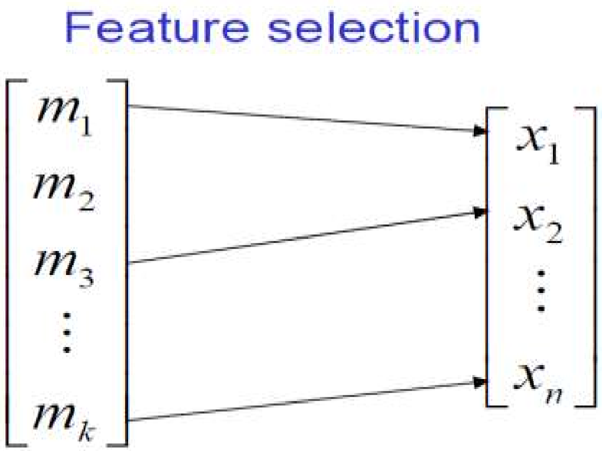




Feature extraction / feature selection



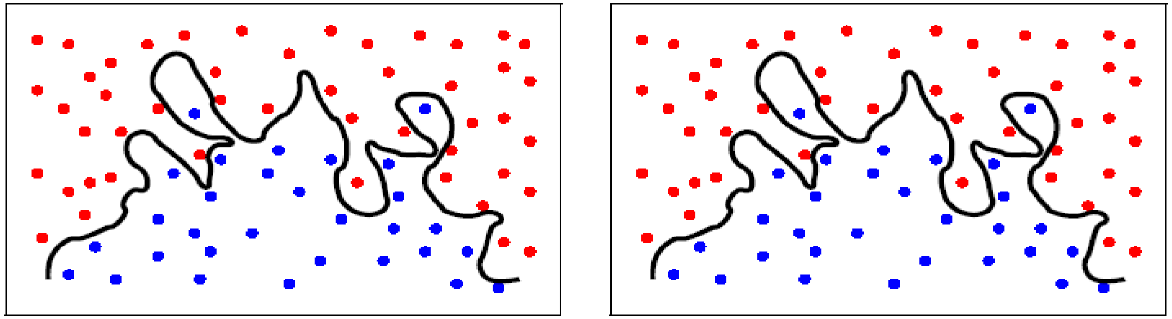




Problem - Overfitting

Generalization !important!

Cross-validation



Colour features

Dominant colour/colours

The simplest description of the colour in the image

The dominant colour covers a large part of the picture

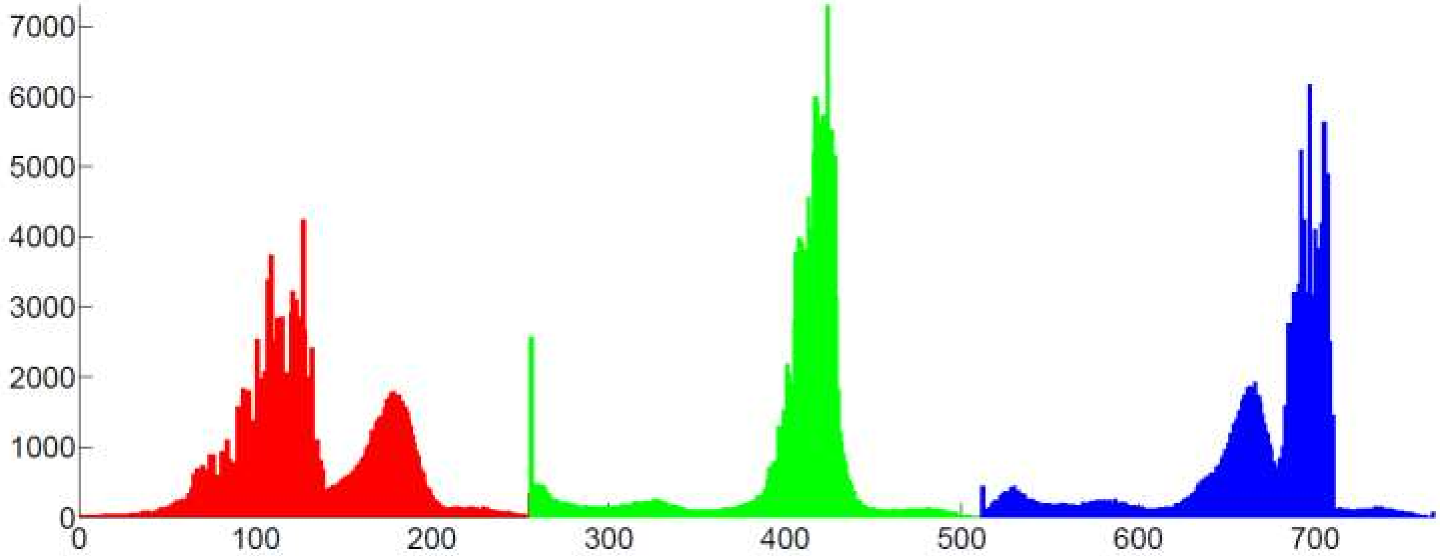
One dominant colour or more dominat colours

Descriptors dominant colour is generally a set of pairs:

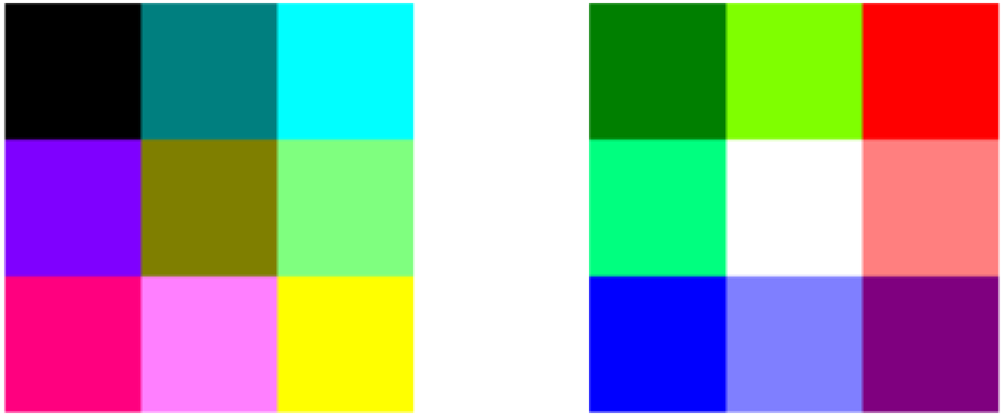
colour, percentage

The problem: space information is not included

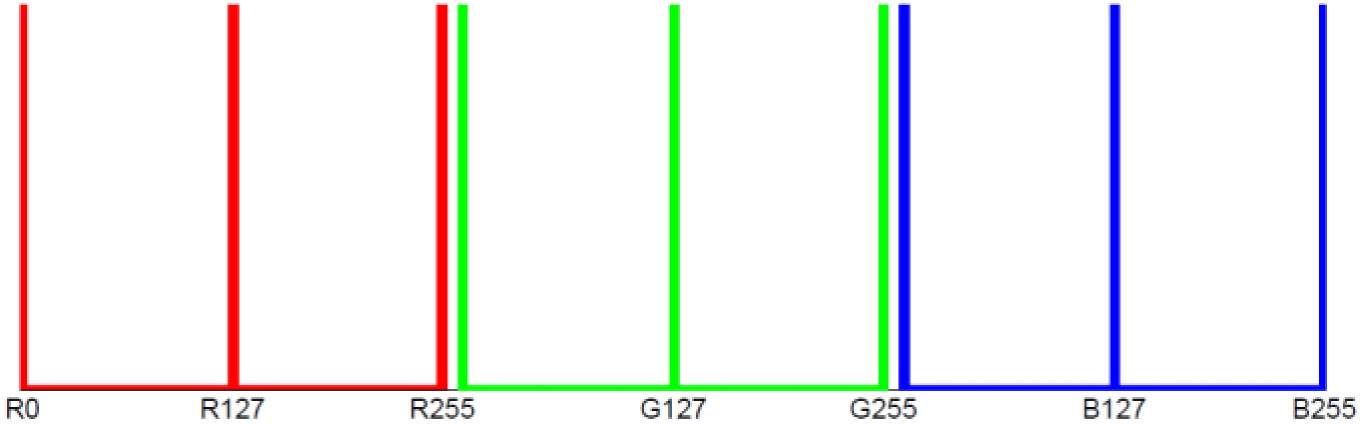
Colour 1D histogram



Colour 1D histogram



Two images containing various colors.



The same histogram of the two images.

The Scalable Color Descriptor (MPEG7)

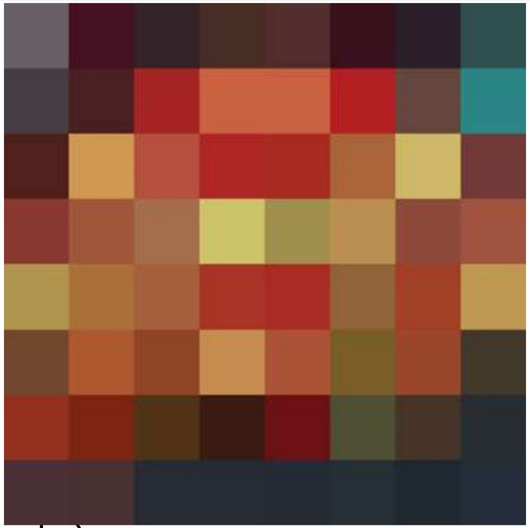
is derived from a colour histogram defined in the HSV colour space with fixed colour space quantization.

It uses a Haar transform (HT) coefficient encoding, allowing scalable representation of description, as well as complexity scalability of feature extraction and matching procedures.

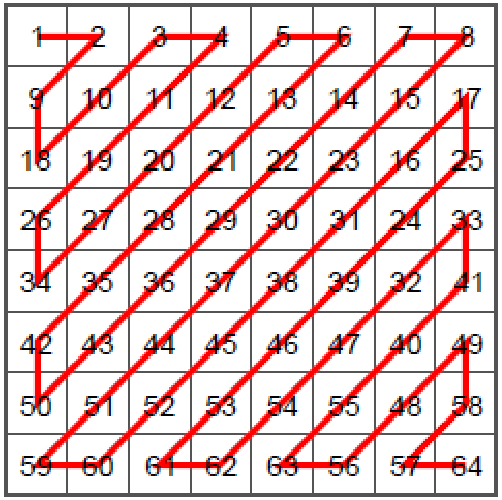
HT represents histograms with a different number of classes

The Scalable Colour Descriptor (MPEG7)

a)



b)



a) the image is divided into 8 × 8 blocks

b) the average colour of blocks.

c) Zig-zag ordering of coefficients in the descriptor distribution of colours

Colour descriptor based on spatial distribution

Include spatial information

we recorded an average location (x and y-coordinates of points with a given colour) and standard deviation

$$\bar{x}_i = \frac{1}{N.A_i} \sum_{c(\mathbf{p})=C_i} x,$$

$$\bar{y}_i = \frac{1}{M.A_i} \sum_{c(\mathbf{p})=C_i} y,$$

$$\sigma_i = \sqrt{\frac{1}{A_i} \sum_{c(\mathbf{p})=C_i} d(\mathbf{p}, \mathbf{b}_i),}$$

where:

A_i is the area having the colour content of C,

P = (x, y) is the image of a point

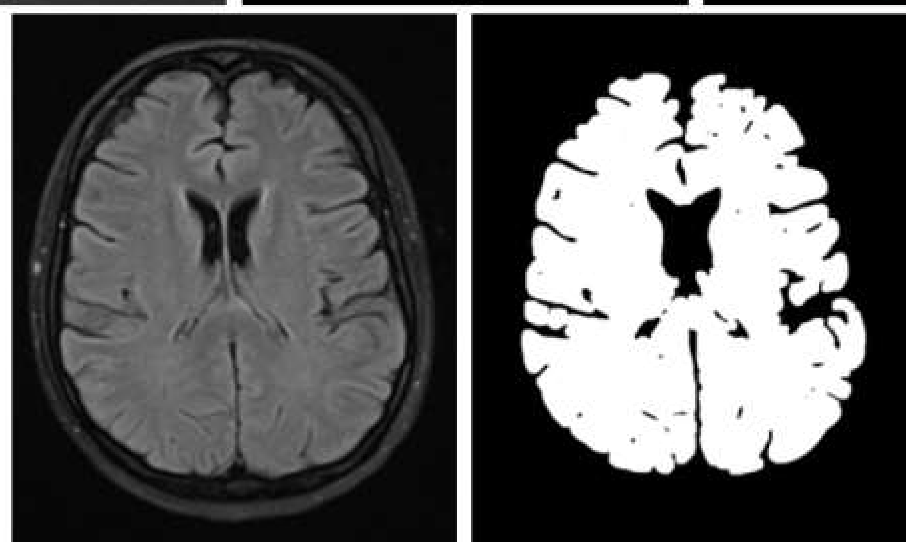
M × N is the image size.

If the standard deviation is small, we know that colour is concentrated in a small region of the picture. If the standard deviation is large, colour is deployed around the image.

Shape features

Shape features – binary image

Shape features are typically used for binary image that we get after image segmentation



Shape Representation

Chain codes

Signatures

Skeleton of region

Shape Representation
Chain codes

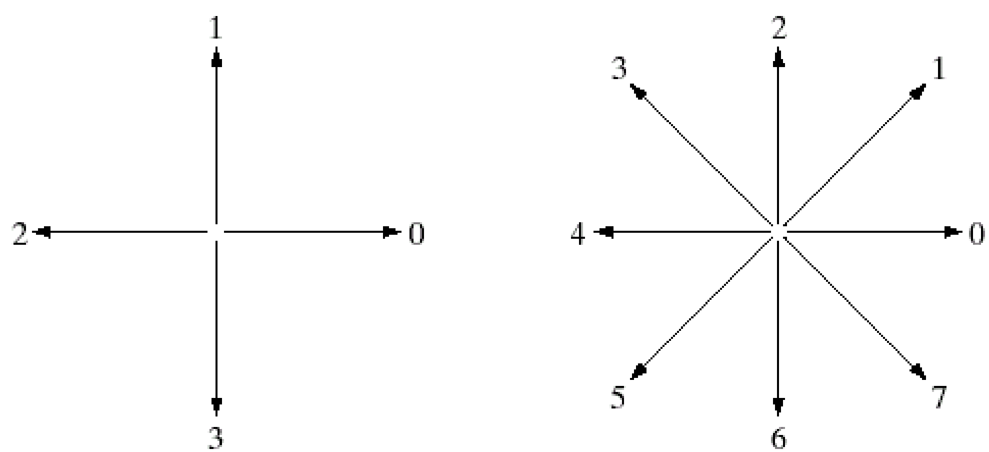
Represent a boundary by a connected sequence of straight-line segments of specified length and direction

4-directional chain codes

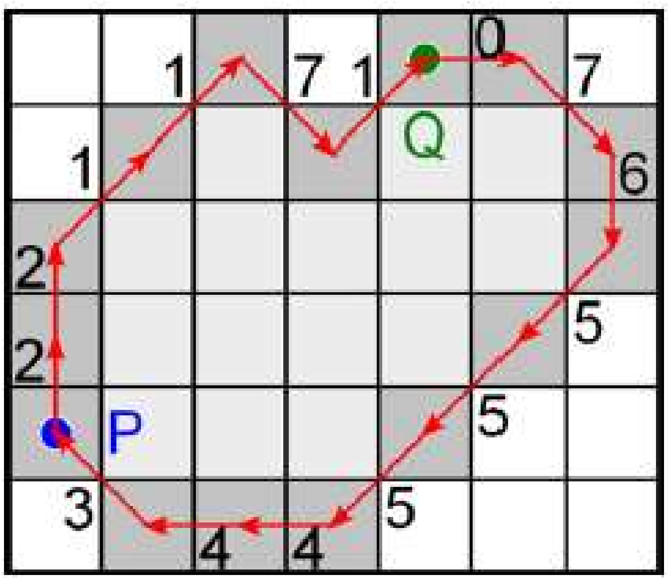
8-directional chain codes

a b

FIGURE 11.1 Direction numbers for (a) 4-directional chain code, and (b) 8-directional chain code.



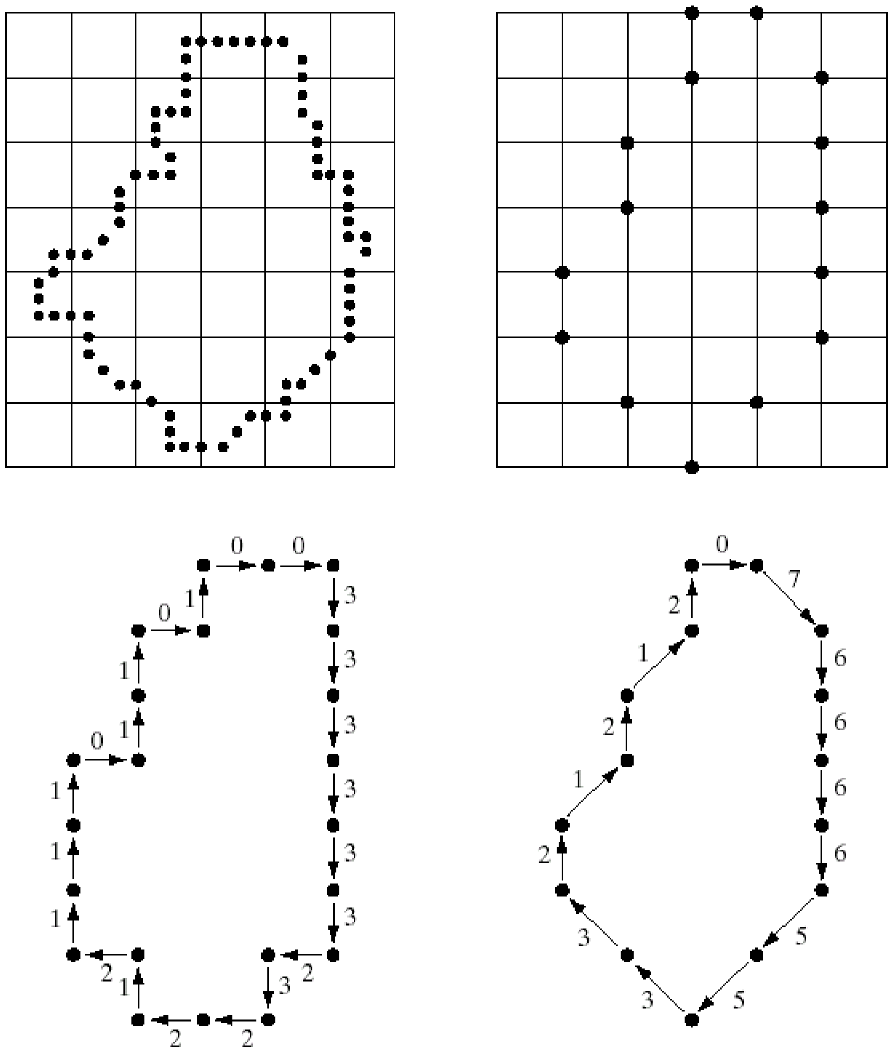
Freeman Chain code



For 8-neighbor code a limit of 8-neighborhood of the point 0,. . .7 as shown in Figure

The chain (Freeman) code boundary object is then a sequence of numbers that contain information that direction limit from the point continues.

Shape Representation
Chain codes



a b
c d

FIGURE 11.2
(a) Digital boundary with resampling grid superimposed.
(b) Result of resampling.
(c) 4-directional chain code.
(d) 8-directional chain code.

Shape Representation
Chain codes

Normalization for rotation – first difference

Counting (counterclockwise) the number of direction changes that separate two adjacent element of the code

Normalization for starting position – shape number

The first difference of smallest magnitude

Normalization for size

Multi-scaling resampling

4-directional chain code:	**01100010303332322221211**
First difference:	**310300133130003130003130**
Shape number:	**000313000313031030013313**

Shape Representation
Signatures

A 1-D functional representation of a boundary

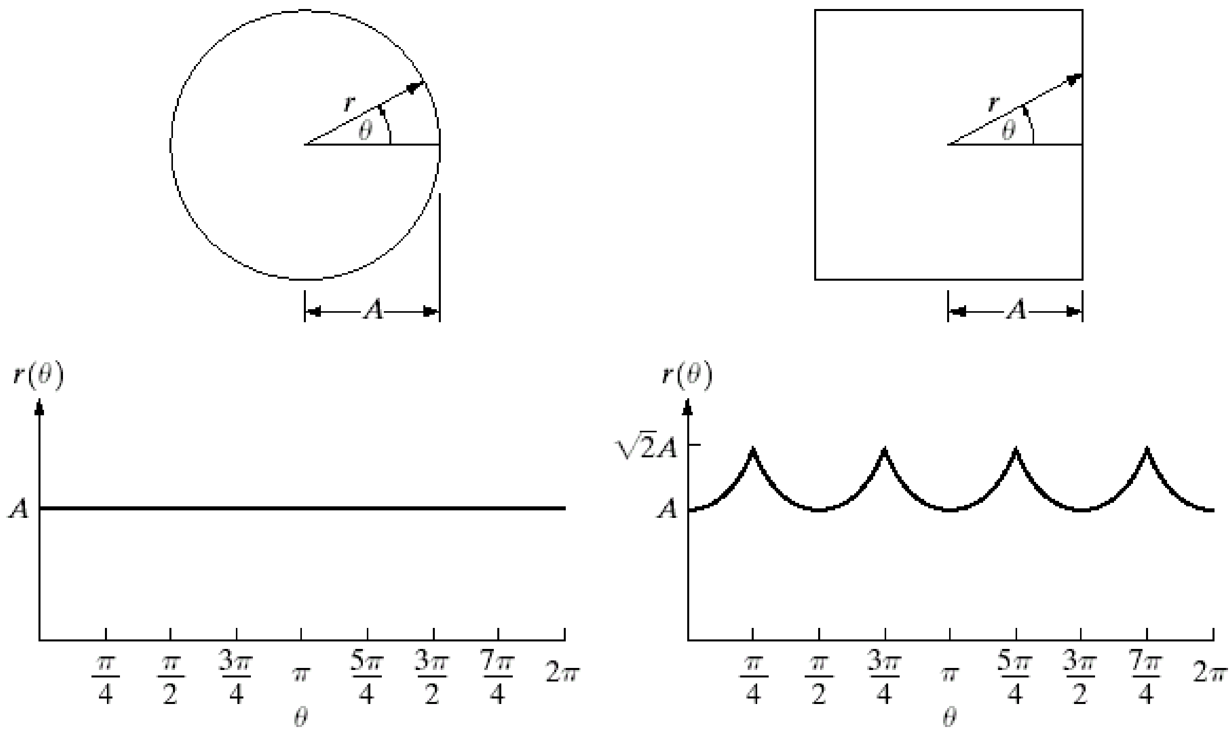
Basic idea : reduce the boundary representation to a 1-D function, which might be easier to describe than a 2-D boundary

One simple approach : use the distance from the centroid to the boundary as a function of angle. It is invariant to translation, but not to rotation and scaling.

Rotation : select the farthest point from the centroid as the starting point

Scaling : normalize the function by variance

Shape Representation Signatures



a b

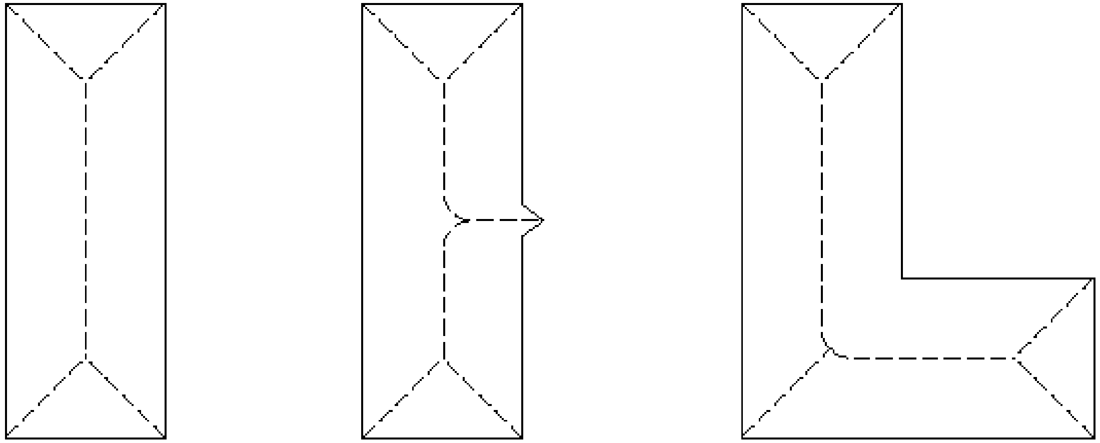
FIGURE 11.5 Distance-versus-angle signatures. In (a) $r(\theta)$ is constant. In (b), the signature consists of repetitions of the pattern $r(\theta) = A\sec\theta$ for $0 \leq \theta \leq \pi/4$ and $r(\theta) = A\csc\theta$ for $\pi/4 < \theta \leq \pi/2$.

Skeleton of a region

Use skeleton to represent a region

Skeletonizing (thinning) a region

Computationally expensive



a b c

FIGURE 11.7 Medial axes (dashed) of three simple regions.

Texture features

Texture features

Structural vs. Statistical Approaches

Edge-Based Measures

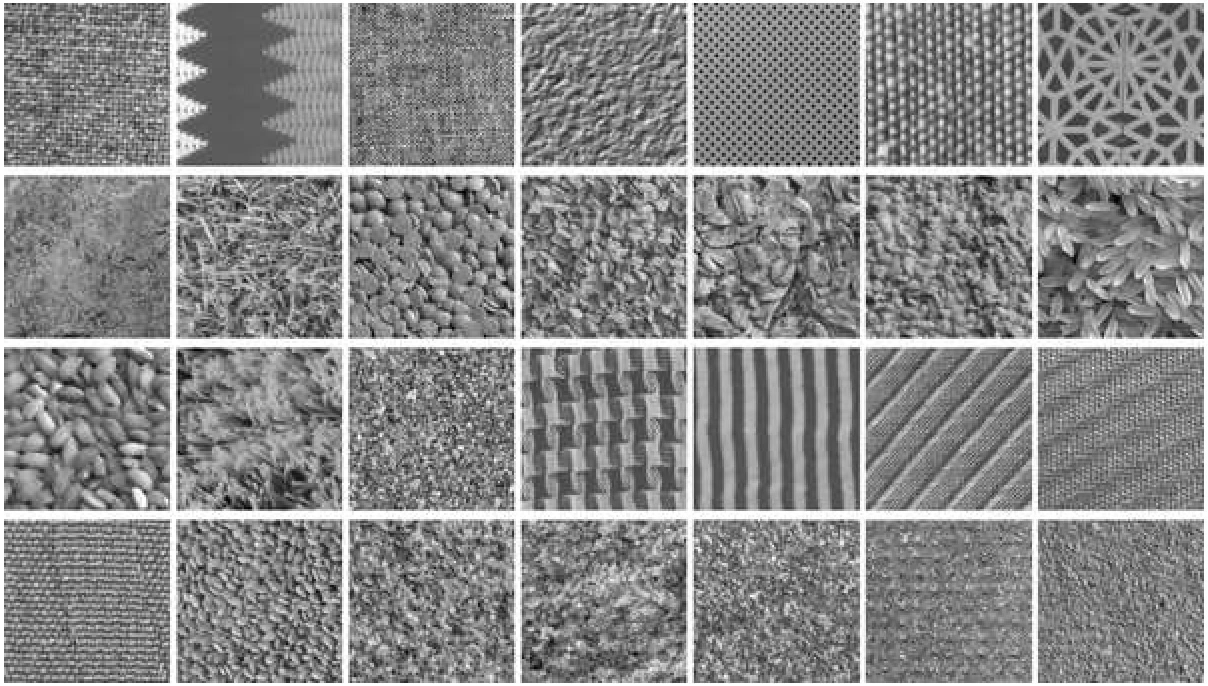
Local Binary Patterns

Co-occurence Matrices

Gabor Filters

Texture features

Texture is a description of the spatial arrangement of colour or intensities in an image or a selected region of an image.



Statistical Texture Measures

Segmenting out textons

Numeric quantities or statistics that describe a texture can be computed from the grey tones (or colours) alone.

This approach is less intuitive, but is computationally efficient.
It can be used for both classification and segmentation.

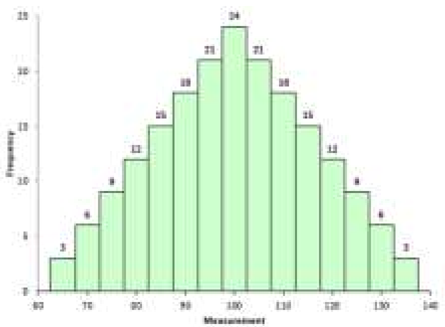
Simple Statistical Texture Measures
Statistical moments

2. Standard deviation

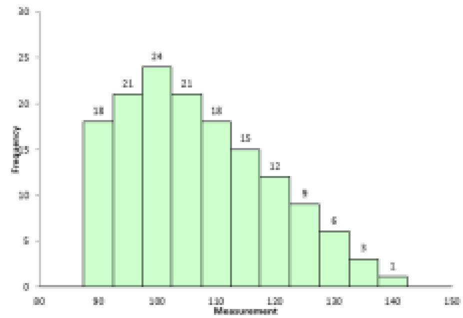
$$s_x = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$$

3. Skewness – degree of symetry in the distributio

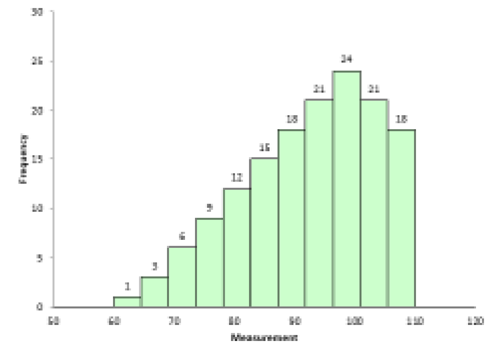
$$Skewness = \frac{n}{(n-1)(n-2)}\sum\frac{(X_i - \bar{X})^3}{s^3}$$



Symmetrical Dataset with Skewness = 0



Dataset with Positive Skewness



Dataset with Negative Skewness

4. Kurtosis – peakedness of the distribution

$$Kurtosis = \left\{\frac{n(n+1)}{(n-1)(n-2)(n-3)}\sum\frac{(X_i - \bar{X})^4}{s^4}\right\}$$

Local Binary Pattern - LBP

For each pixel p, create an 8-bit number

b1 b2 b3 b4 b5 b6 b7 b8,

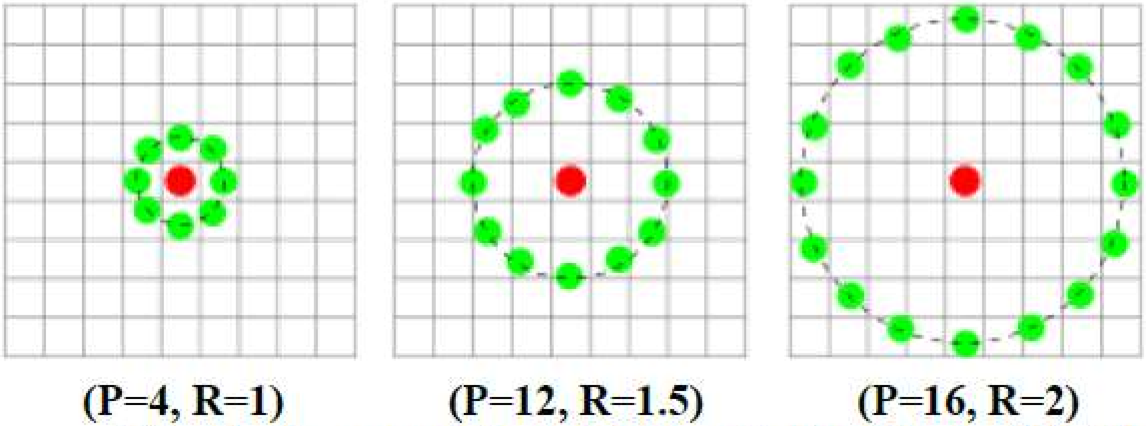
where bi = 0 if neighbour i has value less than or equal to p's value and 1 otherwise.

Represent the texture in the image (or a region) by the histogram of these numbers.

	1	2	3	
	100	101	103	
8	40	50	80	4
	50	60	90	5
	7	6		

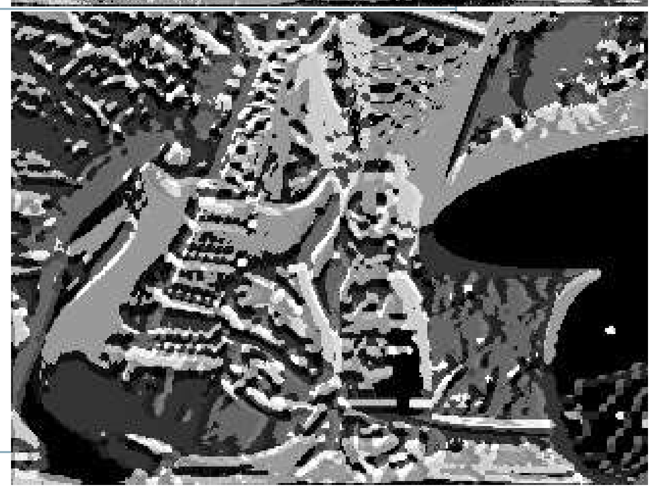
$\longrightarrow$ $\boxed{1\ 1\ 1\ 1\ 1\ 1\ 0\ 0}$

Local Binary Pattern - LBP



Circularly symmetric neighbor sets for different (P, R).

Local Binary Pattern - LBP

Radius	Sampling Points	
1	4	
4	4	

- https://github.com/bytefish/libfacerec/blob/master/src/lbp.cpp
- Examples: http://bytefish.de/blog/local_binary_patterns/

Co-occurrence Matrix Features

A co-occurrence matrix is a 2D array C in which

Both the rows and columns represent a set of possible image values.

$C_d(i,j)$ indicates how many times value i co-occurs with value j in a particular spatial relationship d.

The spatial relationship is specified by a vector d = (dr,dc).

Co-occurrence Example

I

1	1	5	6	8
2	3	5	7	1
4	5	7	1	2
8	5	1	2	5

GLCM

	1	2	3	4	5	6	7	8
1	1	2	0	0	1	0	0	0
2	0	0	1	0	1	0	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0	0
5	1	0	0	0	0	1	2	0
6	0	0	0	0	0	0	0	1
7	2	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0

co-occurrence matrix

From C_d we can compute
N- the normalized co-occurrence matrix,
where each value is divided by the sum of all the values.

Co-occurrence Features

Energy measures uniformity of the normalized matrix.

$$Energy = \sum_i \sum_j N_d^2(i,j) \tag{7.7}$$

$$Entropy = -\sum_i \sum_j N_d(i,j) log_2 N_d(i,j) \tag{7.8}$$

$$Contrast = \sum_i \sum_j (i-j)^2 N_d(i,j) \tag{7.9}$$

$$Homogeneity = \sum_i \sum_j \frac{N_d(i,j)}{1+ \mid i-j \mid} \tag{7.10}$$

$$Correlation = \frac{\sum_i \sum_j (i-\mu_i)(j-\mu_j) N_d(i,j)}{\sigma_i \sigma_j} \tag{7.11}$$

where μ_i, μ_j are the means and σ_i, σ_j are the standard deviations of the row and column

Gabor Filters

Gabor wavelets

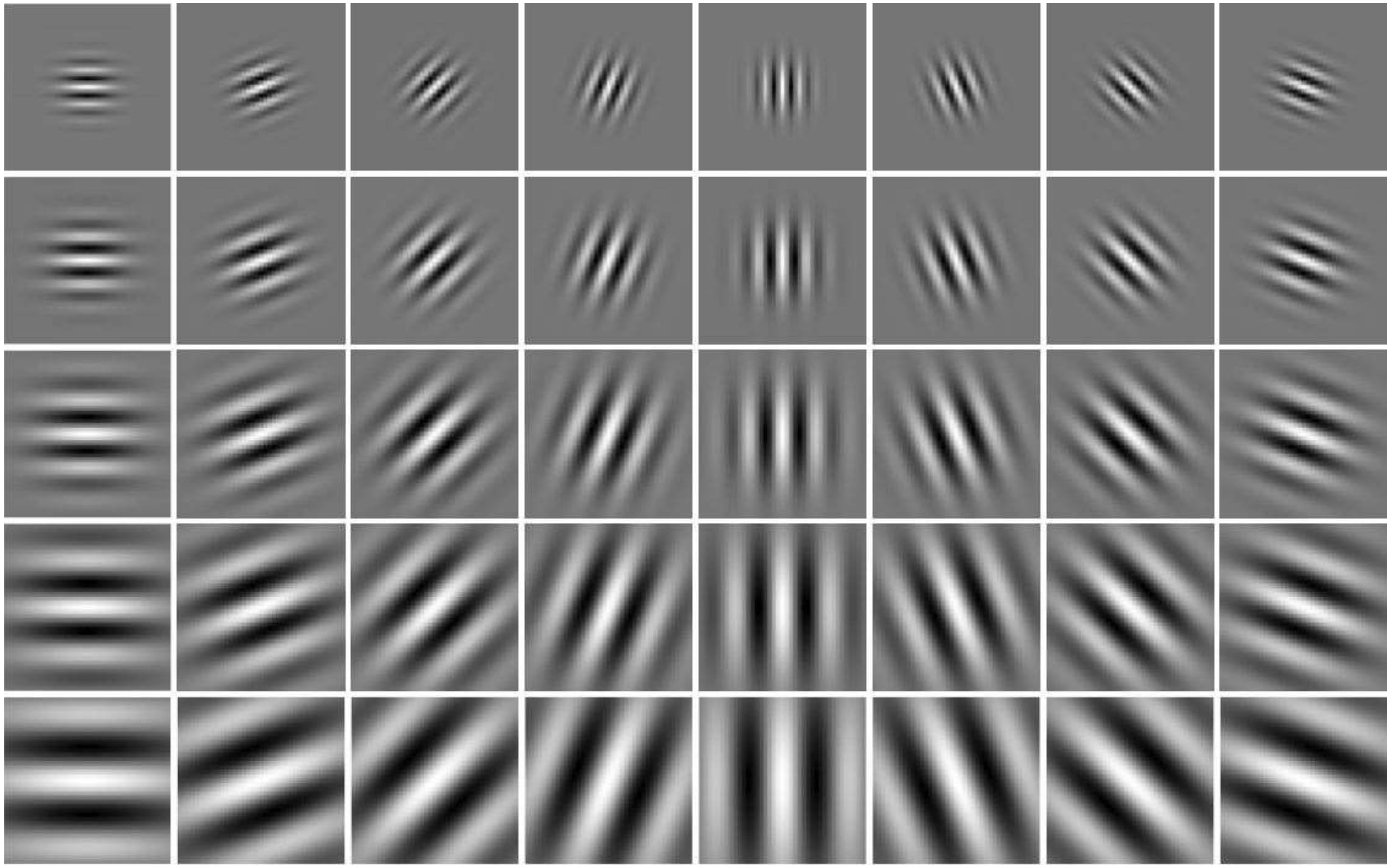
Wavelets at different frequencies and different orientations

Generalised Gabor functions :

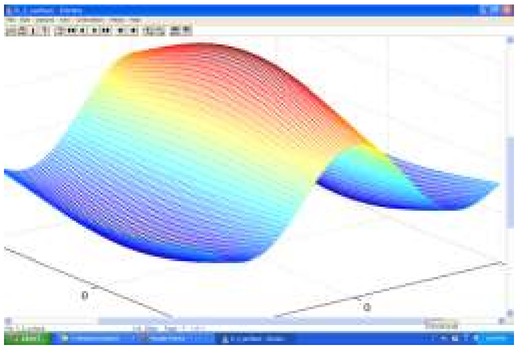
$$\gamma(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left[-\frac{1}{2}\left(\frac{(x-x_0)^2}{\sigma_x^2}+\frac{(y-y_0)^2}{\sigma_y^2}\right)+2\pi\sqrt{-1}\left(u_0x+v_0y\right)\right]$$

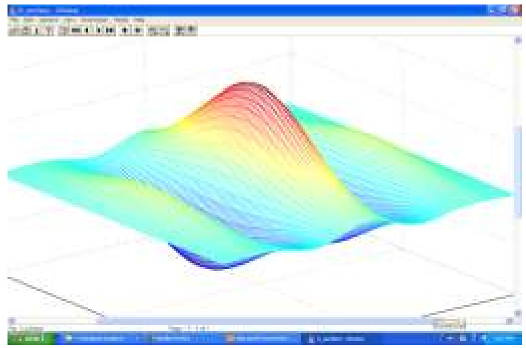
Gabor Filters
Set of convolution kernels

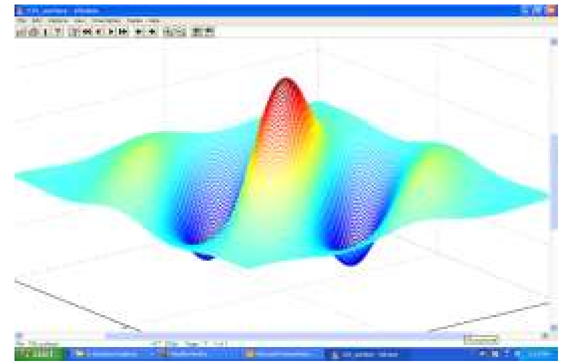
Different frequences and orientations

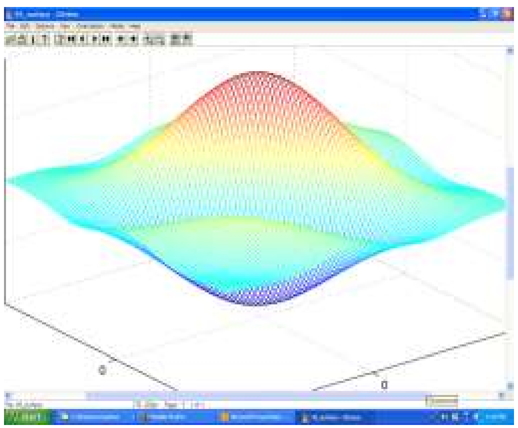


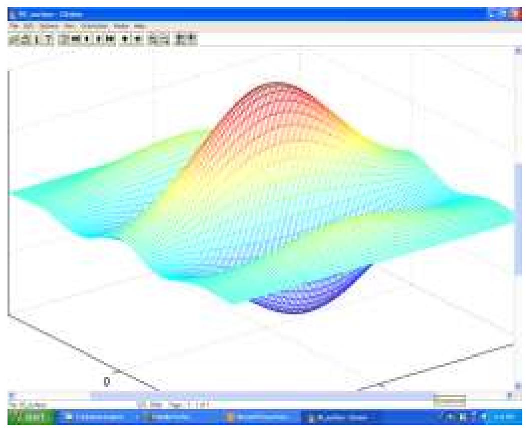
Gabor Filters
Convolution kernels – examples in 3D viz.



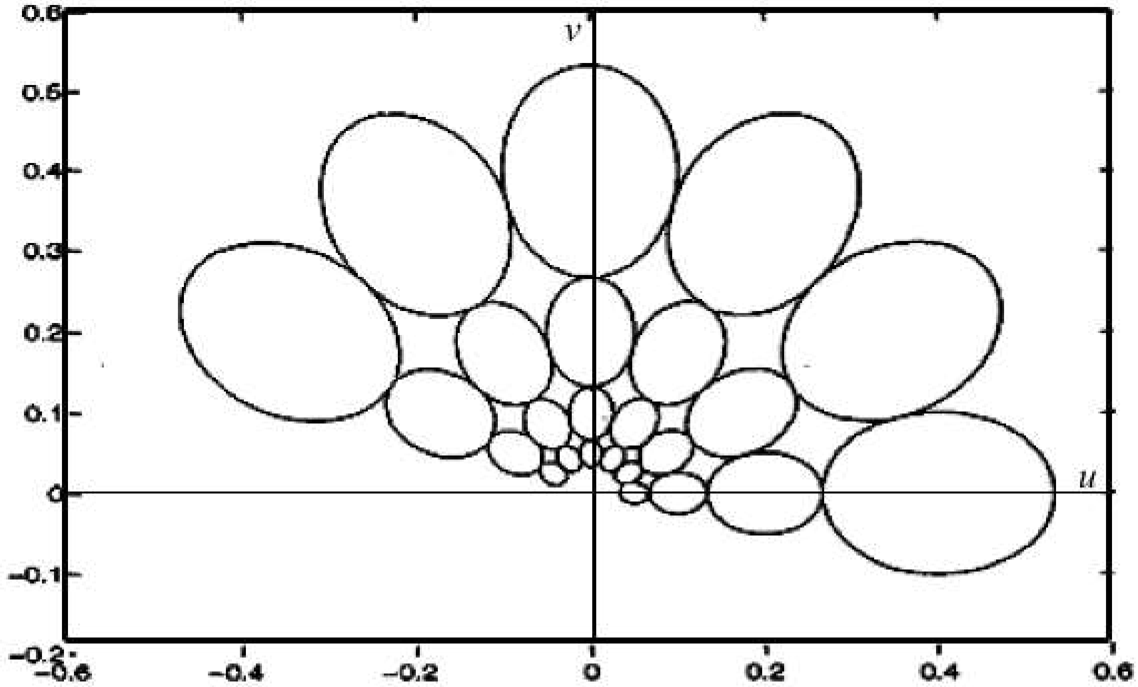








Gabor Filters

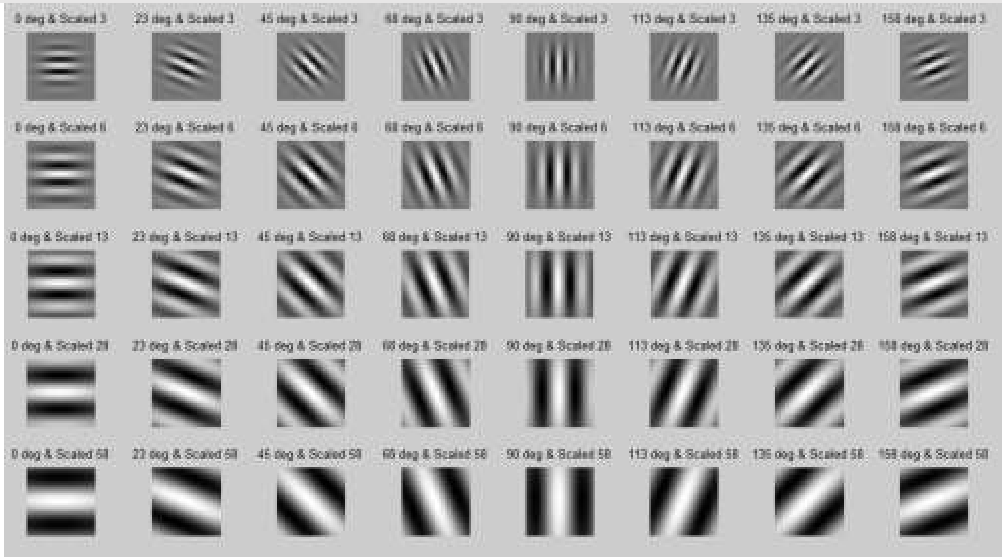


Gabor Filters – segmentation example

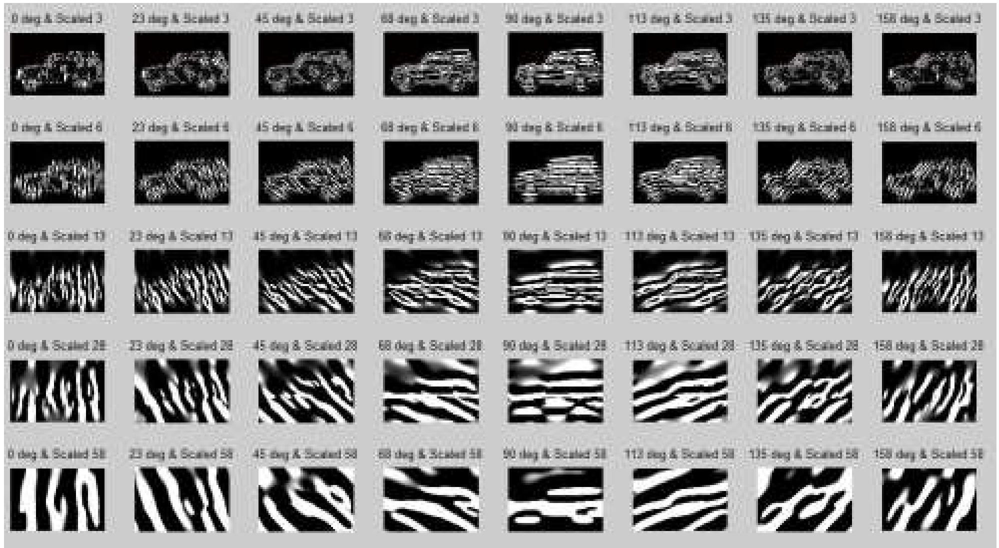
Gabor Filters – segmentation example

Input image

Gabor kernels



Convolution output



Zheng, Danian, Yannan Zhao, and Jiaxin Wang. "Features extraction using a gabor filter family." *Proceedings of the sixth Lasted International conference, Signal and Image processing, Hawaii.* 2004.

- Texture features

Edge features

Edge-based Texture Measures

1. edgeness per unit area

$$F_{edgeness} = |\{ p \mid gradient\_magnitude(p) \geq threshold\}| / N$$

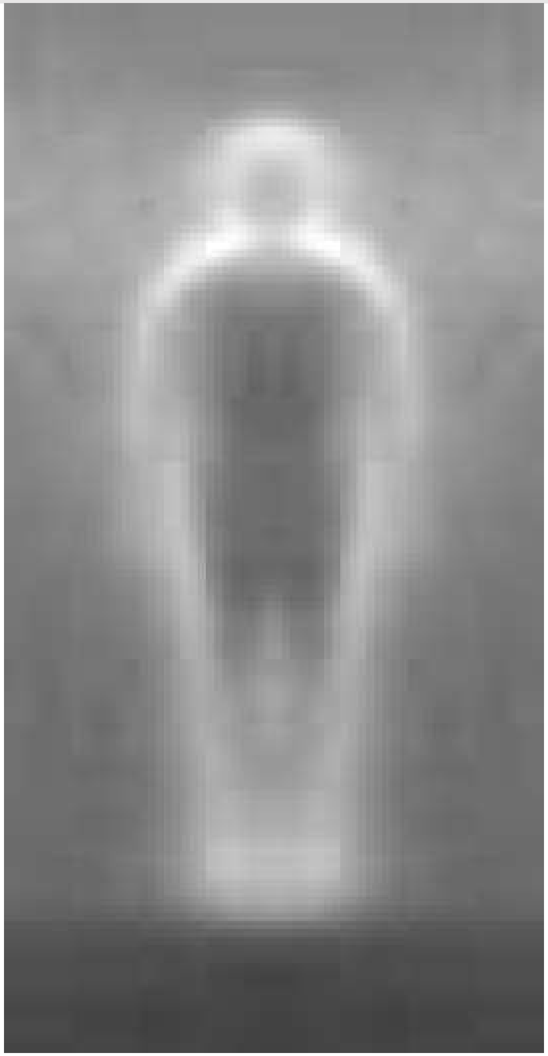
where N is the size of the unit area

2. edge magnitude and direction histograms

$$F_{magdir} = (H_{magnitude}, H_{direction})$$

where these are the normalized histograms of gradient magnitudes and gradient directions, respectively.

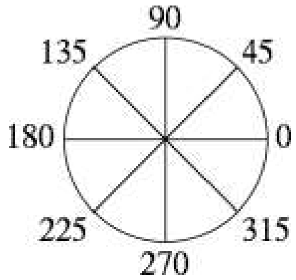
Histogram of gradient orientations HOG

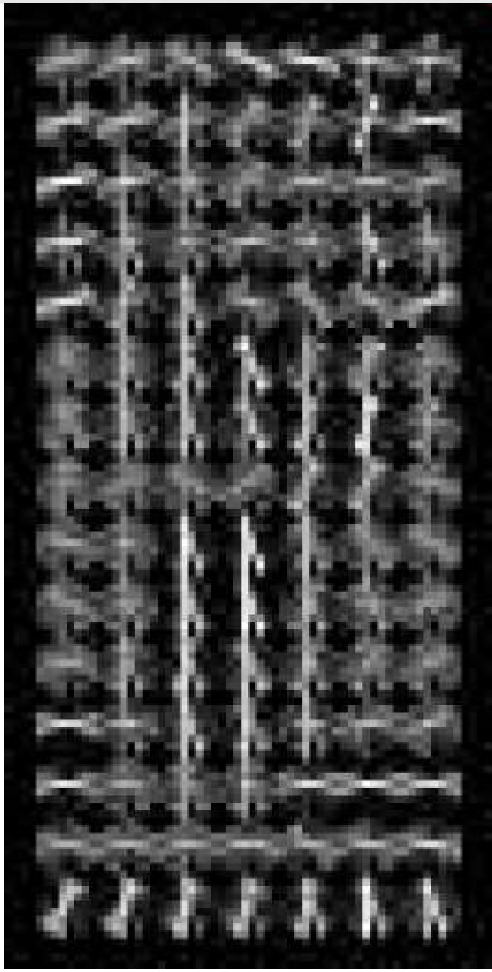


Edge vertical filtration
+
Edge horizontal filtration

-> edge gradient
Magnitude + angle (orientation)

Histogram of gradients:
Angle weighted by magnitude





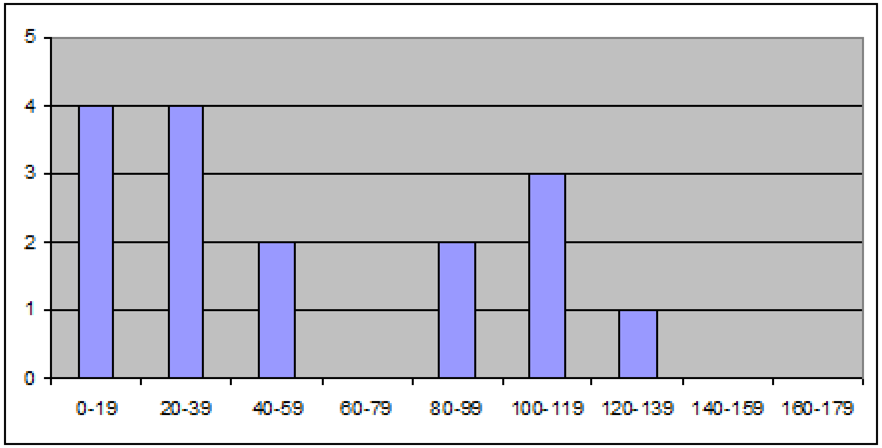
Histogram of gradient orientations HOG

Edge vertical filtration
+
Edge horizontal filtration
(Sobel)

Using Convolution

Histogram of gradients:

Angle weighted by magnitude



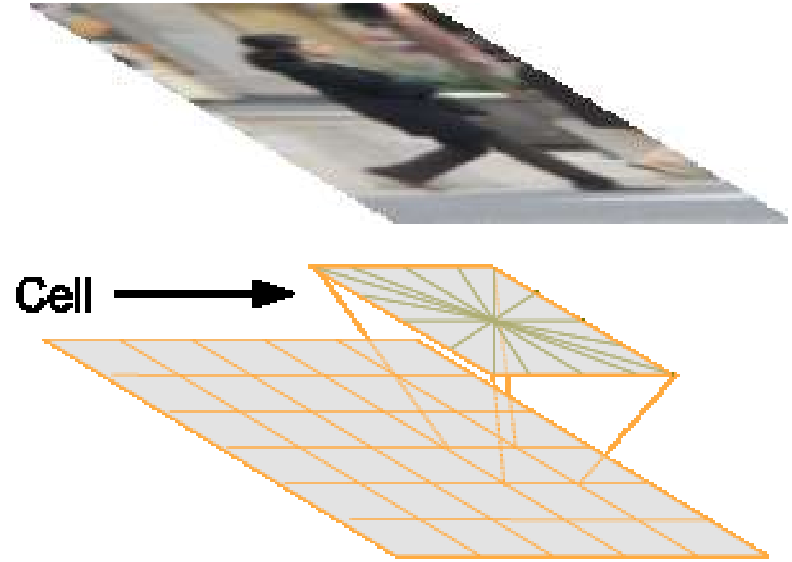
Histogram of gradient orientations HOG - example

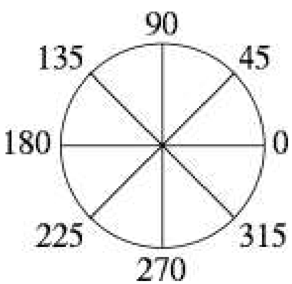
Cell histograms

typically

8 (or 9) bins for gradient orientations (0-180 degrees)

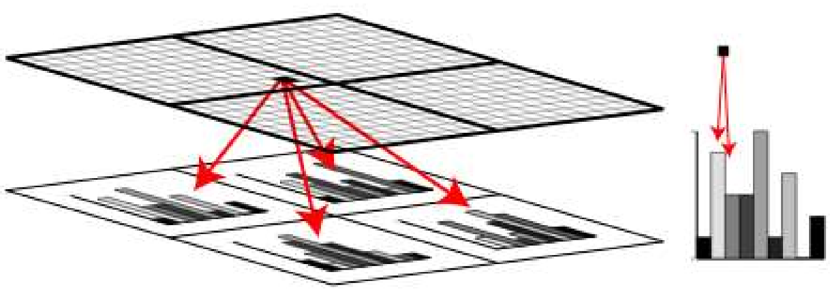
Filled with magnitudes





HOG feature: chain of data

4 cells



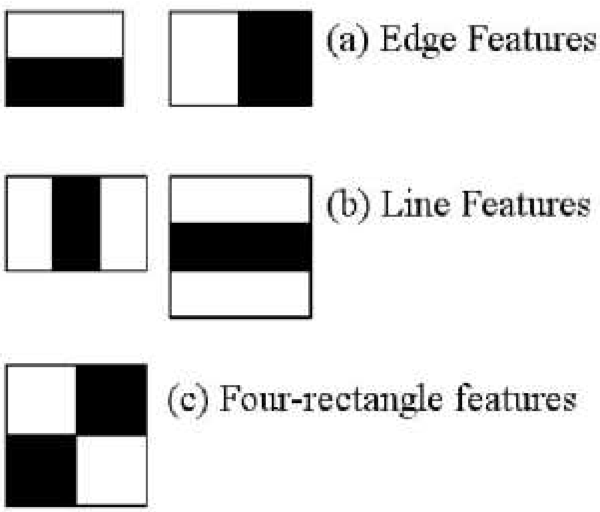
$$f = \left(h_1^1, \ldots, h_9^1,\; h_1^2, \ldots, h_9^2,\; h_1^3, \ldots, h_9^3,\; h_1^4, \ldots, h_9^4\right)$$

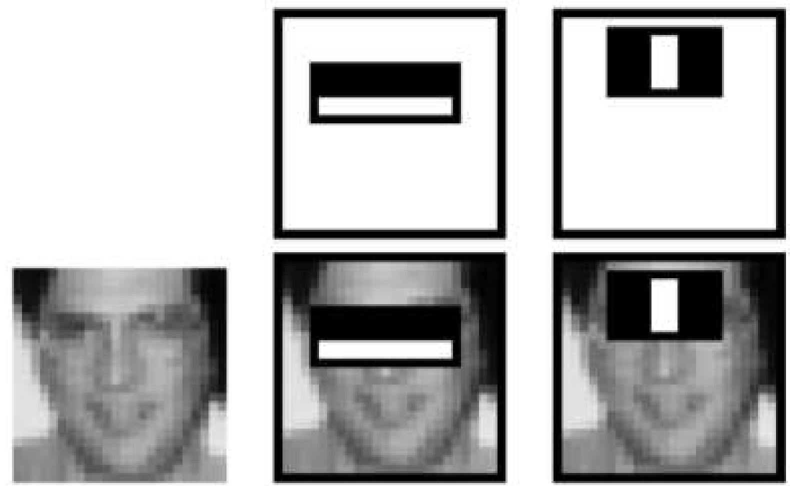
Haar-like features

The sum of pixels which lie within the white rectangles are subtracted from the sum of pixels in the grey rectangles ->

Compute differences between sums of pixels in rectangles

Similar to Haar wavelets, efficient to compute using integral image





Haar-like features
Viola & Jones, CVPR 2001

Considering all possible filter parameters: position, scale, and type:

180,000+ possible features associated with each of sliding window (24x24)

Use AdaBoost both to select the informative features and to form the classifier

Viola & Jones, CVPR 2001