University of Technology
الجامعة التكنولوجية

Computer Science Department
قسم علوم الحاسوب

Pattern Recognition
تمييز الانماط

Prof. Dr. Shaimaa Hameed Shaker
أ.د. شيماء حميد شاكر

cs.uotechnology.edu.iq

PATTERN RECOGNITION

Fourth Class,

M M branch, Second Semester, 2024-2025.

Pattern recognition

- Introduction of pattern recognition
- Basic Concepts of pattern recognition
- Optical Pattern Recognition
- Object Description and Representation
- Feature Selection and Generation
- SIFT and SIRF
- Harris Corner Detection
- Template Matching
- Clustering Techniques
- Clustering Algorithms
- Classification
- ID3 Algorithm
- OCR
- Pattern recognition Applications

References:

1- pattern recognition. Sergios Th., second edition.
2- Supervised and Unsupervised Pattern Recognition J. David Irwin, *Auburn University*

Lecture one

Pattern Recognition Introduction

Pattern is everything around in this digital world. A pattern can either be seen physically or it can be observed mathematically by applying algorithms .

Example: The colors on the clothes, speech pattern, etc. In computer science, a pattern is represented using vector feature values .

What is Pattern Recognition?

Pattern recognition is the process of recognizing patterns by using a machine learning algorithm. Pattern recognition can be defined as the classification of data based on knowledge already gained or on statistical information extracted from patterns and/or their representation. One of the important aspects of pattern recognition is its application potential .

Examples: Speech recognition, speaker identification, multimedia document recognition (MDR), automatic medical diagnosis . Given a pattern, its recognition and classification can consist of one of the following two tasks:

- Supervised classification identifies the input pattern as a member of a predefined class.
- Unsupervised classification assigns the input pattern to a hitherto undefined class.

In a typical pattern recognition application, the raw data is processed and converted into a form that is amenable for a machine to use. Pattern recognition involves the classification and cluster of patterns .

In classification, an appropriate class label is assigned to a pattern based on an abstraction that is generated using a set of training patterns or domain knowledge. Classification is used in supervised learning.

Clustering generated a partition of the data which helps decision making, the specific decision-making activity of interest to us. Clustering is used in unsupervised learning.

Features may be represented as continuous, discrete, or discrete binary variables. A feature is a function of one or more measurements, computed so that it quantifies some significant characteristics of the object .

Example: consider our face then eyes, ears, nose, etc are features of the face . A set of features that are taken together, forms the features vector .

Example: In the above example of a face, if all the features (eyes, ears, nose, etc) are taken together then the sequence is a feature vector([eyes, ears, nose]). The feature vector is the sequence of a feature represented as a d-dimensional column vector. In the case of speech, MFCC (Mel-frequency Cepstral Coefficient) is the spectral feature of the speech. The sequence of the first 13 features forms a feature vector . Pattern recognition possesses the following features :

- Pattern recognition system should recognize familiar patterns quickly and accurate.
- Recognize and classify unfamiliar objects.
  - Accurately recognize shapes and objects from different angles.
  - Identify patterns and objects even when partly hidden.
  - Recognize patterns quickly with ease, and with automaticity.

What Is the Goal of Pattern Recognition?

The goal of pattern recognition is based on the idea that the decision-making process of a human being is somewhat related to the recognition of patterns. For example, the next move in a chess game, buying or selling stocks is decided by a complex pattern of financial information. Therefore, the goal of pattern recognition is to clarify these complicated mechanisms of decision-making processes and to automate these e functions using computers .

Pseudotsuga menziesii
Arbutus menziesii
Acer macrophyllum
06419446010
handwriting
福

Training and Learning in Pattern Recognition

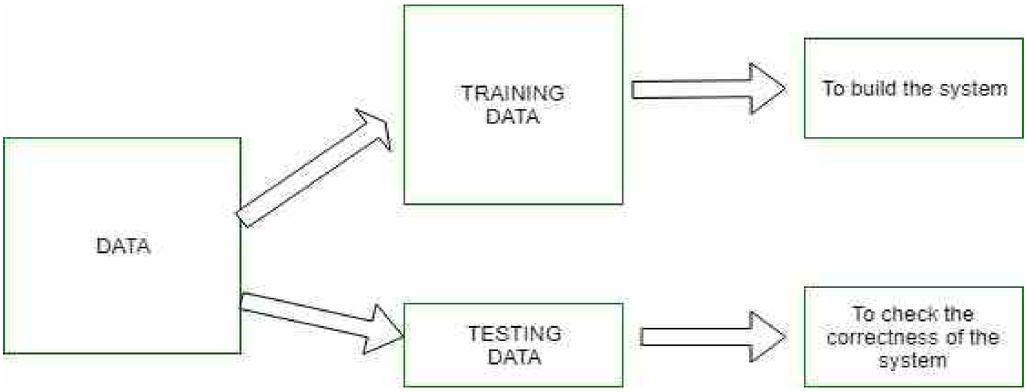
Learning is a phenomenon through which a system gets trained and becomes adaptable to give results in an accurate manner. Learning is the most important phase as to how well the system performs on the data provided to the system depends on which algorithms are used on the data. The entire dataset is divided into two categories, one which is used in training the model i.e. Training set, and the other that is used in testing the model after training, i.e. Testing set.

- **Training set**

  The training set is used to build a model. It consists of the set of images that are used to train the system. Training rules and algorithms are used to give relevant information on how to associate input data with output decisions. The system is trained by applying these algorithms to the dataset, all the relevant information is extracted from the data, and results are obtained. Generally, 80% of the data of the dataset is taken for training data.

- **Testing set**

  Testing data is used to test the system. It is the set of data that is used to verify whether the system is producing the correct output after being trained or not. Generally, 20% of the data of the dataset is used for testing. Testing data is used to measure the accuracy of the system. For example, a system that identifies which category a particular flower belongs to is able to identify seven categories of flowers correctly out of ten and the rest of others wrong, then the accuracy is 70 %



Real-time Examples and Explanations

A pattern is a physical object or an abstract notion. While talking about the classes of animals, a description of an animal would be a pattern. While talking about various types of balls, then a description of a ball is a pattern. In the case balls considered as pattern, the classes could be football, cricket ball, table tennis ball, etc. Given a new pattern, the class of the pattern is to be determined. The choice of attributes and representation of patterns is a very important step in pattern classification. A good representation is one that makes use of discriminating attributes and also reduces the computational burden in pattern classification.

An obvious representation of a pattern will be a **vector**. Each element of the vector can represent one attribute of the pattern. The first element of the vector will contain the value of the first attribute for the pattern being considered.

Example: While representing spherical objects, (25, 1) may be represented as a spherical object with 25 units of weight and 1 unit diameter. The class label can form a part of the vector. If spherical objects belong to class 1, the vector would be (25, 1, 1), where the first element represents the weight of the object, the second element, the diameter of the object and the third element represents the class of the object.

Advantages:

1. Pattern recognition solves classification problems
2. Pattern recognition solves the problem of fake biometric detection.
3. It is useful for cloth pattern recognition for visually impaired blind people.
4. It helps in speaker diarization.
5. We can recognize particular objects from different angles.

Disadvantages

1. The syntactic pattern recognition approach is complex to implement and it is a very slow process.
2. Sometimes to get better accuracy, a larger dataset is required.
3. It cannot explain why a particular object is recognized.

Applications

- **Image processing, segmentation, and analysis**

Pattern recognition is used to give human recognition intelligence to machines that are required in image processing.

- **Computer vision**
  Pattern recognition is used to extract meaningful features from given image/video samples and is used in computer vision for various applications like biological and biomedical imaging.

- **Seismic analysis**
  The pattern recognition approach is used for the discovery, imaging, and interpretation of temporal patterns in seismic array recordings. Statistical pattern recognition is implemented and used in different types of seismic analysis models.

- **Radar signal classification/analysis**
  Pattern recognition and signal processing methods are used in various applications of radar signal classifications like AP mine detection and identification.

- **Speech recognition**
  The greatest success in speech recognition has been obtained using pattern recognition paradigms. It is used in various algorithms of speech recognition which tries to avoid the problems of using a phoneme level of description and treats larger units such as words as pattern
- **Fingerprint identification**
  Fingerprint recognition technology is a dominant technology in the biometric market. A number of recognition methods have been used to perform fingerprint matching out of which pattern recognition approaches are widely used.

Lecture 2

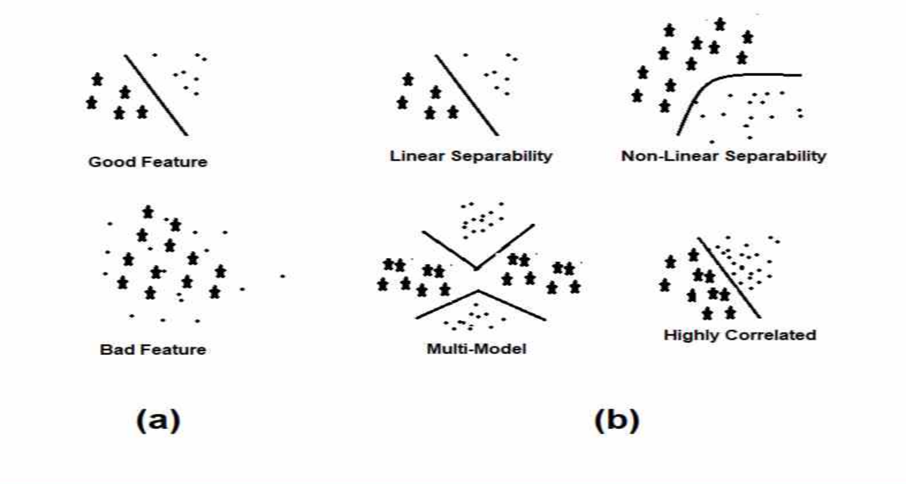
Pattern Recognition System: Basic Concepts

Pattern is everything around in this digital world. A pattern can either be seen physically or it can be observed mathematically by applying algorithms. In Pattern Recognition, pattern is comprises of the following two fundamental things:

1. Collection of observations.
2. The concept behind the observation.

Feature Vector

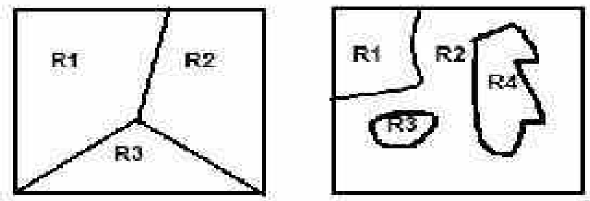
The collection of observations is also known as a feature vector. A feature is a distinctive characteristic of a good or service that sets it apart from similar items. ***Feature vector*** is the combination of n features in n-dimensional column vector.The different classes may have different features values but the same class always has the same features values.



Classifier and Decision Boundaries

In a statistical-classification problem, a decision boundary is a hypersurface that partitions the underlying vector space into *two sets*. A decision boundary is the region of a problem space in which the output label of a classifier is ambiguous. Classifier is a hypothesis or discretevalued function that is used to assign (categorical) class labels to particular data points.

Classifier is used to partition the feature space into class-labeled decision regions. While Decision Boundaries are the borders between decision regions.



Classifier and decision boundaries

Components in Pattern Recognition System

A ***pattern recognition systems*** can be partitioned into basic five components for various pattern recognition systems , these are as following:

- ***A Sensor :*** A sensor is a device used to measure a property, such as pressure, position, temperature, or acceleration, and respond with feedback.
- ***A Preprocessing Mechanism*** **:** Segmentation is used and it is the process of partitioning a data into multiple segments. It can also be defined as the technique of dividing or partitioning an data into parts called segments.
- ***A Feature Extraction Mechanism*** **:** feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. It can be manual or automated.
- ***A Description Algorithm*** **:** Pattern recognition algorithms generally aim to provide a reasonable answer for all possible inputs and to perform "most likely" matching of the inputs, taking into account their statistical variation.

- ***A Training Set*** **:** Training data is a certain percentage of an overall dataset along with testing set. As a rule, the better the training data, the better the algorithm or classifier performs.

Design Principles of Pattern Recognition

In pattern recognition system, for recognizing the pattern or structure two basic approaches are used which can be implemented in different techniques these are :

1-Statistical Approach

Statistical methods are mathematical formulas, models, and techniques that are used in the statistical analysis of raw research data. The application of statistical methods extracts information from research data and provides different ways to assess the robustness of research outputs. Two main statistical methods are used :

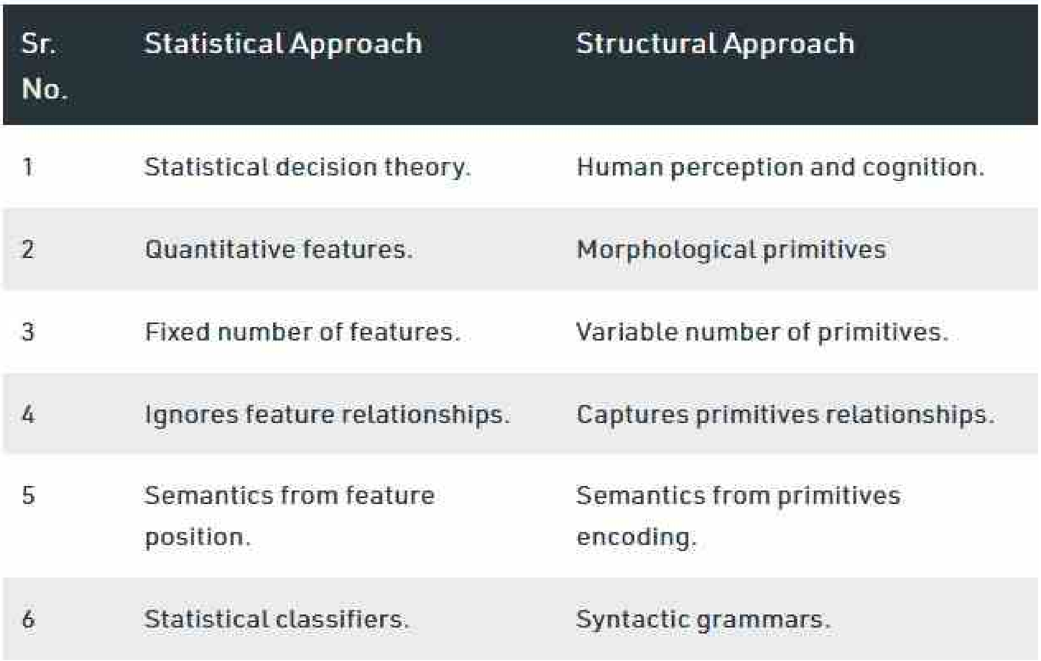
- **Descriptive Statistics:** It summarizes data from a sample using indexes such as the mean or standard deviation.
- **Inferential Statistics:** It draw conclusions from data that are subject to random variation.

2-Structural Approach

The Structural Approach is a technique wherein the learner masters the pattern of sentence. Structures are the different arrangements of words in one accepted style or the other. Types of structural approach:

- Sentence Patterns
- Phrase Patterns
- Formulas
- Idioms

Q: A comparison between statistical and structural approaches .

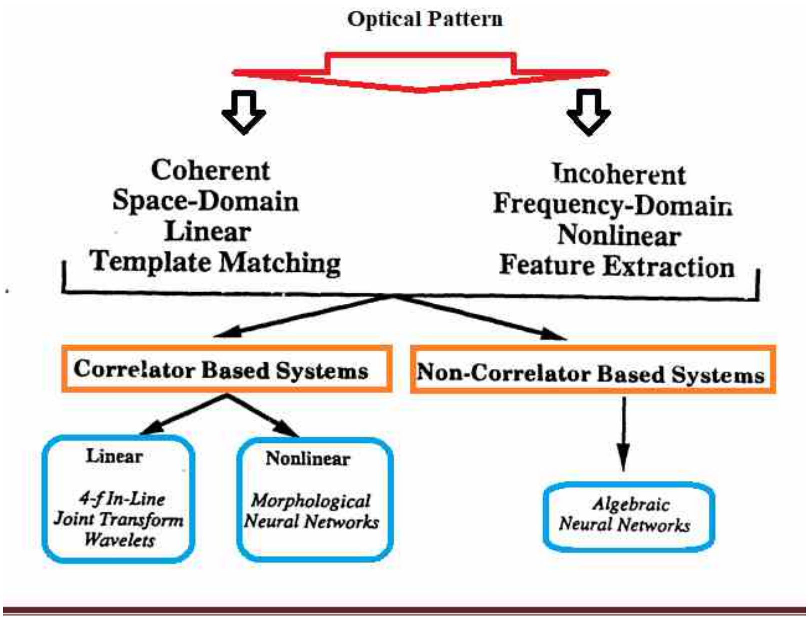


Sr. No.	Statistical Approach	Structural Approach
1	Statistical decision theory.	Human perception and cognition.
2	Quantitative features.	Morphological primitives
3	Fixed number of features.	Variable number of primitives.
4	Ignores feature relationships.	Captures primitives relationships.
5	Semantics from feature position.	Semantics from primitives encoding.
6	Statistical classifiers.	Syntactic grammars.

Lecture 3

OPTICAL PATTERN RECOGNITION DEFINITIONS

Pattern recognition is a field of great interest with a wide range of applications. It involves systems that are capable of recognizing patterns. These patterns can be as simple as an alphabet character or a complex image such as a human face. The application of machine vision to sorting products on industrial assembly lines; recognizing fingerprints; and, on the battlefield, recognizing mine fields or track enemy tanks and airplanes are all of great importance. Pattern recognition is an intriguing problem that has been of interest to researchers over the last three decades. Recognition needs to be performed at high speed with a high confidence level. Discrimination between similar objects is very crucial. The object to be recognized can be distorted and/or corrupted by noise. It has been realized for a long time that recognizing objects with arbitrary aspect projection, scale, and rotation is a computationally intensive problem.



COHERENT

Coherent optical processors are based on using coherent light illumination. The system is analyzed using Fourier optics. The system is linear space-invariant in terms of the amplitude of the electric field. The output is the convolution of the input function with the impulse response of the system.

INCOHERENT

Incoherent optical processors are based on the use of partially coherent light for illumination. This system is linear in intensity. The output intensity of the system is the convolution of the input intensity with the modules square of the impulse response. Incoherent processors do not suffer from coherent artifact noise. Also, the input does not need to be displayed on a SLM, which eliminates the need for incoherent-to-coherent converters. Color image processing can be done using incoherent systems.

SPACE- AND FREQUENCY-DOMAIN PROCESSING

Data manipulation in both coherent and incoherent processors can be done in either the spatial domain (object space) or frequency domain (Fourier or other transform space). In the object space processing case, object and reference are both processed without transforming. In the frequency domain processing case, transforms of the object and reference are computed first, then processed.

SPACE-DOMAIN PROCESSORS

There are a number of techniques that are used in space domain optical processors; we describe a few of these techniques here.

1-Shadow Casting

In this technique the object and reference are imposed on top of each other to perform the desired operation. In the case of correlators, the object and the reference patterns are scanned either optically (by using a collimated beam followed by a ground glass) or mechanically (by moving one of the patterns with respect to the other).

2-Algebraic

Many optical processing operations can be achieved using linear algebra manipulations. These can be implemented by using vector-matrix and matrix-matrix multiplication type operations. Vector matrix multiplication is performed by expanding each element of the input vector and casting each on a row of the matrix which is displayed on either a fixed mask or a dynamic SLM. The output from each column is focused to form an element of the output vector.

FREQUENCY-DOMAIN COHERENT PROCESSORS

Frequency-domain processing in a coherent system is mainly dominated by correlator.

a-Correlator

Optical coherent correlators are the most widely used systems in pattern recognition. Correlation is achieved in frequency domain processing by superimposing the Fourier transform (FT) of the input function and the complex conjugate of the FT of the reference function on each other (to multiply the two functions). The inverse FT of the product results in the correlation of the two functions. The FT of the reference function is referred to as the filter. There are a wide variety of filter designs. We will describe filter designs later in this document. The following is a discussion of some of the systems and applications based on coherent correlators.

b-4-f In-Line Correlator System

This is the basic building block for optical data processing systems. The input function is placed a distance F, focal length, in front of a Fourier transforming lens and illuminated with collimated coherent light. The filter is plqced in the back focal plane of this Fourier transforming lens. The back focal plane of the lens is referred to as the Fourier plane or the frequency domain. Another Fourier transforming lens is placed a distance V: behind the filter plane and in the back focal plane of this lens 's the output plane. The first lens performs the FT and the second lens performs the inverse Fourier transform (IFT) operation. The output will depend on the filter pattern. If it is the transform of the reference function, the output will be the convolution of the input function and the reference function. If the filter is the complex conjugate of the reference function, the output will be the correlation of the input and reference function.

c-Joint-Transform Correlator

In this system both input and reference functions are placed in the input plane, displayed on the same SLM, in the front focal plane of a FT lens. FT of both functions is formed in the back-focal plane of the lens. The pattern resulting from the interference. -f both transforms is detected and displayed on another SLM. A FT of the interference pattern is produced by another FT lens and the correlation of the two functions will be formed in the back focal plane of this lens. Also in this system, the convolution of the two functions will be present at the output plane. In both the in-line and the joint-transform corrclators, the convolution and correlation are displayed at different locations in the output plane, so they can be detected separately.

d-Wavelelts

WTs are multiple correlations of the signal with a wavelet function which has a variable scale. Wavelets can provide position and scale invariant detection and classification of images with low signal to-noise ratios. Optical correlators can be used to implement WTs in parallel. Wavelets are used in time-frequency and multiresolution analysis. There are a wide range of wavelet functions that can result in a wealth of applications.

NONLINEAR PROCESSORS

a-Monhologjcal Processors

Optical morphological processors are based on specific mathematical operations performed on an image. There are two main operations: dilation and erosion, on which morphological operations are based. Using these two basic operations, opening, closing, segmentation, skeletonization, noise suppression, edge detection and pattern recognition are realized. Optically these operations are implemented using space- (shadow-casting and defocused imaging) and frequency-domain correlators. The correlation is performed between the input image and a structuring element. In pattern recognition the hit-or-miss transform (HMT) is used. The HMT detects specific features from the input image and leaves a single spot in its place with all other features suppressed.

Morphological systems are also used very efficiently in preprocessing systems for image enhancement, the output of which can be processed using other pattern recognition systems, such as an optical correlator.

b-Neural Networks

Artificial NNs are nonlinear systems. They are a distributed system of interconnected nodes and nonlinear processing elements, mimicking the brain. The high connectivity of NNs provides a high capability of feature extraction and classification. The interconnections between the nodes are on the order of N2, where N is the number of nodes. This high interconnection density makes it extremely difficult to hardwire electronically. The inherent parallelism of optics and high connectivity makes it the natural choice for such an application. The challenging part of the system is the non linearity needed for implementing the neuron. This can be achieved using optoelectronic devices. Artificial NNs can be implemented either using algebraic (matrix-vector multiplier) or correlator based systems. Also, depending on either recording the interconnection weights off- or in-line, non-adaptive and adaptive NNs, respectively, can be realized.

Lecture 4

Object Description and Representation

What is an image?

An image is defined as a two-dimensional function,F(x,y), where x and y are spatial coordinates, and the amplitude of F at any pair of coordinates (x,y) is called the intensity of that image at that point. When x,y, and amplitude values of F are finite, we call it a digital image.

Image analysis is used as a fundamental tool for recognizing, differentiating, and quantifying diverse types of images, including grayscale and color images, multispectral images for a few discrete spectral channels or wavebands, and hyperspectral images with a sequence of contiguous wavebands covering a specific spectral region (e.g., visible and near-infrared).

The ***Computer Vision Image Analysis*** service can extract a wide variety of visual features from your images. For example, it can determine whether an image contains adult content, find specific brands or objects, or find human faces.

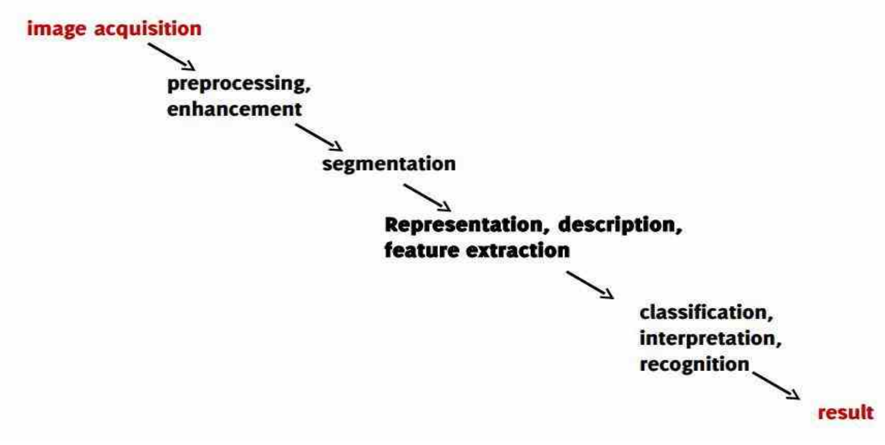


Image analysis fundamental steps.

Representation and description

Commonly after segmentation one needs to represent objects in order to describe them. There are two types of boundary:

❖ **External (boundary):**
- Representation: Polygon of the boundary
- Description: The circumference

❖ **Internal (regional) :**
- Representation: Pixels inside the object
- Description: The average color

The Representation of the Object include :

•An encoding of the object

•Truthful but possibly approximate

The Descriptor of the Object include :

•Only an aspect of the object

•Suitable for classification

• Consider invariance to e.g. noise, translation

Sometimes necessary/desirable to represent an object in a less complicated or more intuitive way . Simple descriptions like enclosing circle, enclosing rectangle, inscribed circle etc. . The boundary or boundary segments . Divide an object into regions or parts . After the segmentation of an image, its regions or edges are represented and described in a manner appropriate for further processing. The image objects can be represented as:

- Whole regions

  – grey level or color image

  – compressed image

  – binary image

- Contours (region boundaries)

  – in Cartesian coordinates

  – in polar coordinates in polar coordinates
  – in some other coordinates

  – through a chain code / transform

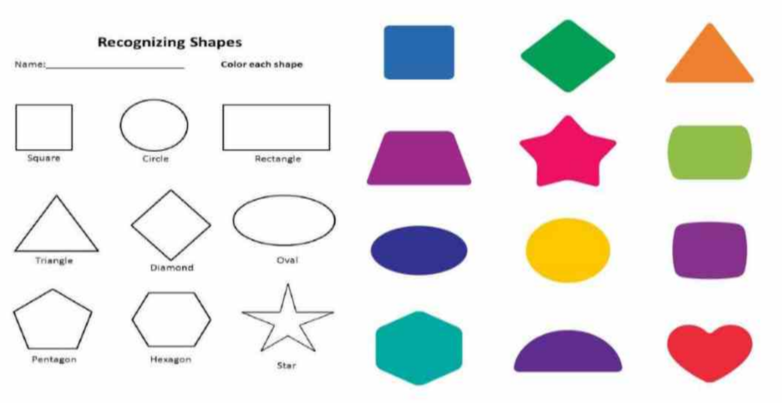
  – as coefficients of some transform (e.g. Fourier)

  – through a run length code / transform

What is ”shape

- A numerical description of the spatial configurations in the image.
- There is no generally accepted methodology of shape description.
- Location and description of high curvature points give essential information about the shape.
- Location and description of high curvature points give essential information.
- Invariance is an important issue.

- Shape is often defined in a 2D image, but its usefulness in a 3D world depends on 3D -> 2D mapping
- Location and description of high curvature points give essential information



Shape invariance

- Shape descriptors depend on viewpoint, object recognition may often be impossible if object or observer changes position.
- Shape description invariance is important , shape invariants represent properties which remain unchanged under an appropriate class of transforms.
- Stability of invariants is a crucial property which affects their applicability affects their applicability .
- The robustness of invariants to image noise and errors introduced by image sensors is of prime errors introduced by image sensors is of prime importance.

Lecture 5

Feature Selection and Generation

What is Feature Selection?

In machine learning and statistics , feature selection, also known as *variable selection*, attribute selection or variable subset selection, is the *process of selecting a subset of relevant features (variables, predictors) for use in model construction.* Feature selection techniques are used for several reasons:

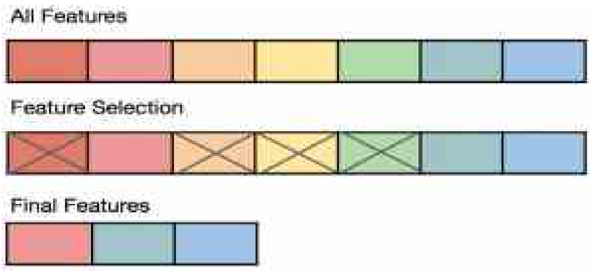
- simplification of models to make them easier to interpret by researchers/users.
- shorter training times.

□ to avoid the curse of dimensionality.

- improve data's compatibility with a learning model class.

□ encode inherent symmetries present in the input space.

The central premise when using a feature selection technique is that the data contains some features that are either *redundant* or *irrelevant*, and can thus be removed without incurring much loss of information. *Redundant* and *irrelevant* are two_distinct notions, since one relevant feature may be redundant in the presence of another relevant feature with which it is strongly correlated.



Feature selection techniques should be distinguished from feature extraction. Feature extraction creates new features from functions of the original features, whereas feature selection returns a subset of the features. Feature selection techniques are often used in domains where_there are many features and comparatively few samples (or data points).

Feature selection is also called variable selection or attribute selection. It is the automatic selection of attributes in your data that are most relevant to the predictive modeling problem you are working on.

feature selection is the process of selecting a subset of relevant features for use in model construction. *Feature selection* is itself useful, but it mostly acts as a filter, muting out features that aren't useful in addition to your existing features.

Selecting the most predictive features from a large space is tricky the more training examples you have, the better you can perform, but the computation time will increase. Several overarching methods exist which fall into one of two categories:

- Supervised Selection

  This type of method involves examining features in conjunction with a trained model where performance can be computed. Since features are selected based on the model's actual performance, these strategies tend to work well. However, their downside is the exorbitant amount of time they take to run.

- Unsupervised Selection

  These methods perform statistical tests on features to determine which are similar or which don't convey much information. Some, like the Variance (or CoVariance) Selector, keep an original subset of features intact, and thus are interpretable.

The Problem The Feature Selection Solves

Feature selection methods create an accurate predictive model. They help you by choosing features that will give you as good or better accuracy while its requiring less data. Feature selection methods can be used to identify and remove unneeded, irrelevant and redundant attributes from data that do not contribute to the accuracy of a predictive model or may in fact decrease the accuracy of the model. Fewer attributes is_desirable because it reduces the complexity of the model, and a simpler model is simpler to understand and explain. The objective of variable selection are :

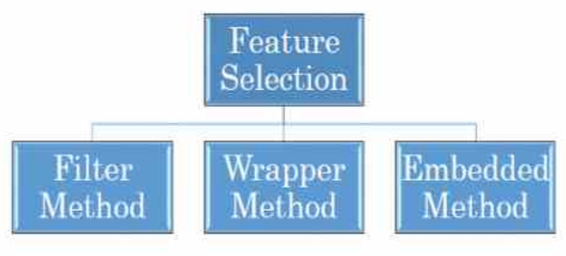
1. improving the prediction performance of the predictors.
2. providing faster and more cost-effective predictors.

3. providing a better understanding of the underlying process that generated the data.

Feature Selection Algorithms

There are three general classes of feature selection algorithms:

- Filter methods.
- Wrapper methods .
- Embedded methods.



Filter Methods

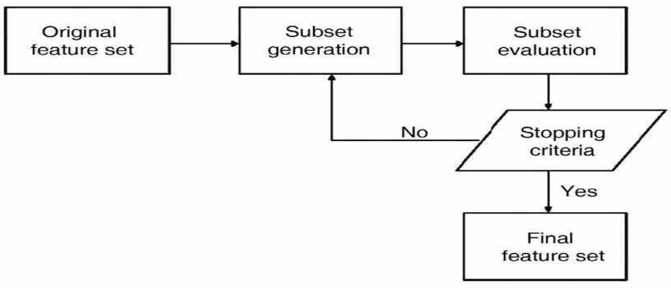
Filter feature selection methods apply a statistical measure to assign a scoring to each feature. The features are ranked by the score and either_selected to be kept or removed from the dataset. Some examples of some filter methods include the Chi squared test, information gain and correlation coefficient scores. This method is generally used as preprossessing step.

Advantages:

- Computationally very fast.
- Avoids overfitting.
- Do not depend on the models, but only features.
- Based on different statistical methods.

Disadvantages:

- Do not remove multicollinearity.
- Sometimes may fail in selection.

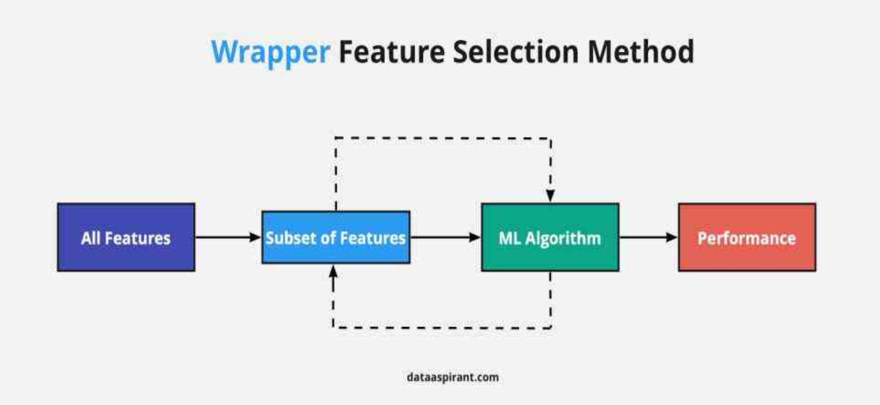


Wrapper Methods

Wrapper methods consider the selection of a set of features as a search problem, where different combinations are prepared, evaluated and compared to other combinations. A predictive model is used to evaluate a_combination of features and assign a score based on model accuracy.

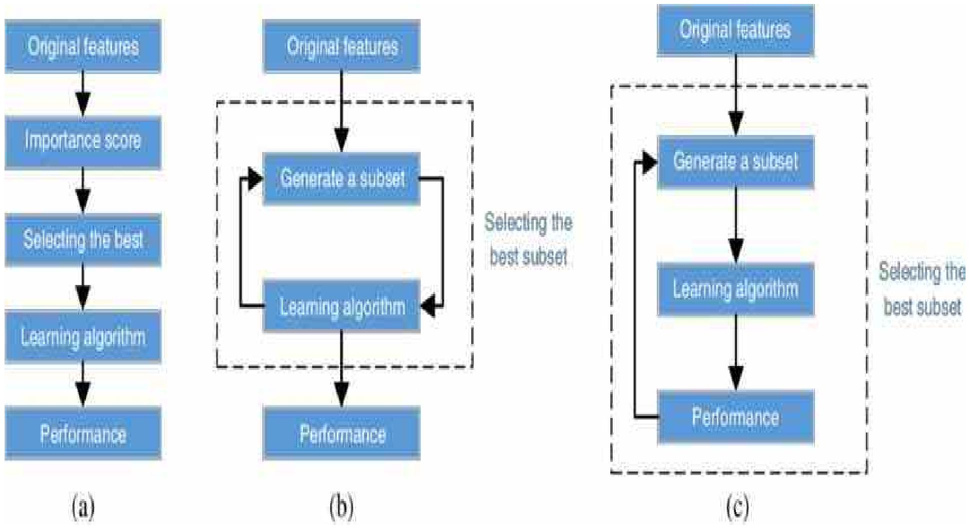
The search process may be methodical such as a best-first search, it may stochastic such as a random hill-climbing algorithm, or it may use heuristics, like forward and backward passes to add and remove features. An example if a wrapper method is the recursive feature elimination algorithm. Generally, three directions of procedures are possible:

- Forward selection : starts with one predictor and adds more iteratively. At each subsequent iteration, the best of the remaining original predictors are added based on performance criteria.
- Backward elimination : starts with all predictors and eliminates one-by-one iteratively. One of the most popular algorithms is Recursive Feature Elimination (RFE) which eliminates less important predictors based on feature importance ranking.
- Step-wise selection : bi-directional, based on a combination of forward selection and backward elimination. It is considered less greedy than the previous two procedures since it does reconsider adding predictors back into the model that has been removed (and vice versa). Nonetheless, the considerations are still made based on local optimisation at any given iteration.



Embedded Methods

Embedded methods learn which features best contribute to the accuracy of the model while the model is being created. The most common type of embedded feature selection methods are regularization methods. Regularization methods are also called penalization methods that introduce additional constraints into the optimization of a predictive algorithm (such as a regression algorithm) that bias the model toward lower complexity (fewer coefficients). Examples of regularization algorithms are the LASSO, Elastic Net and Ridge Regression.



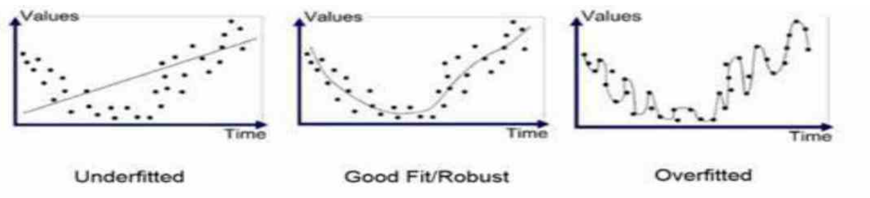
Three types of feature selection. (a) Filter. (b) Wrapper. (c) Embedded.

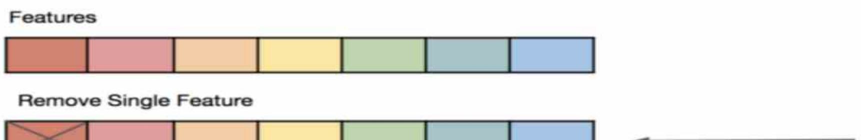
A Trap When Selecting Features

Feature selection is another key part of the applied machine learning process, like model selection. It is important to consider feature selection a part of the model selection process. If you do not, you may inadvertently introduce bias into your models which can result in over fitting.

Feature Generation

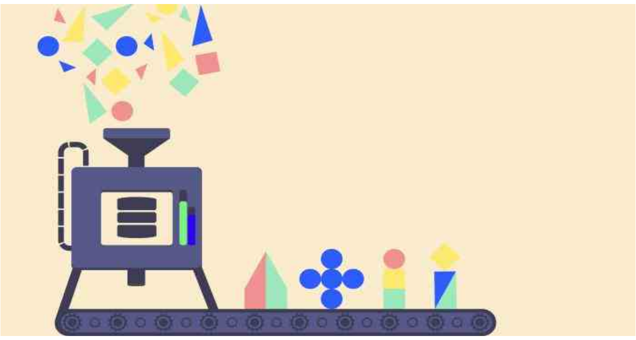
In the real-world, dataset collection is loosely controlled, noisy, unreliable, redundant, and incomplete. This makes data pre-processing an integral stage in the machine learning pipeline.





A feature (or column) represents a measurable piece of data like name, age or gender. It is the basic building block of a dataset. The quality of a feature can vary significantly and has an immense effect on model performance. We can improve the quality of a dataset's features in the pre-processing stage using processes like Feature Generation and Feature Selection.

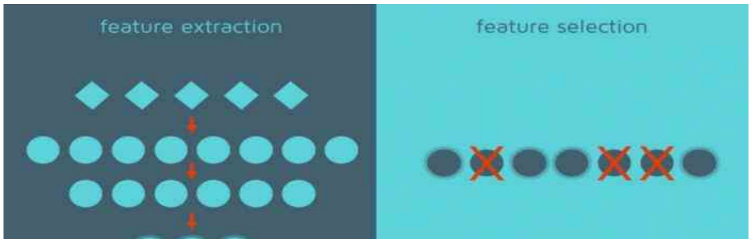
Feature Generation (also known as feature construction, feature extraction or feature engineering) is the process of transforming features_into new features that better relate to the target. This can involve mapping a feature into a new feature using a function like log, or creating a new feature from one or multiple features using multiplication or addition.



Feature Generation .

Feature Generation can improve model performance when there is a feature interaction. Two or more features interact if the combined effect is (greater or less) than the sum of their individual effects. It is possible to make interactions with three or more features, but this tends to result in diminishing returns.

Feature Generation is often overlooked as it is assumed that the model will learn any relevant relationships between features to predict the target variable. However, the generation of new flexible features is important as it allows us to use less complex models that are faster to run and easier to understand and maintain.



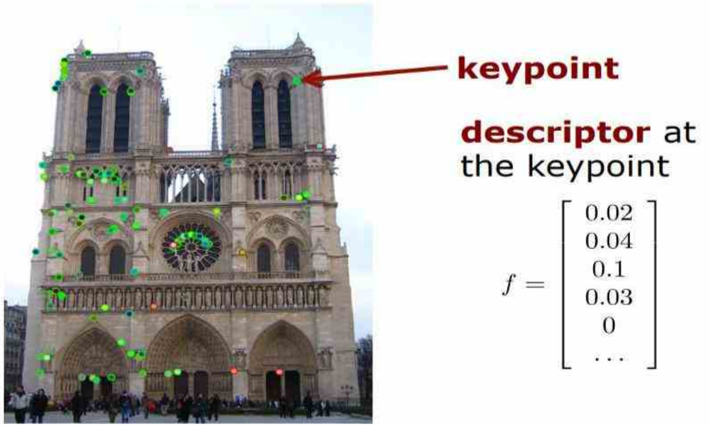
Difference between feature selection and feature extraction .

Lecture 6

SIFT and SIRF

Visual Features Keypoints and Descriptors

- Keypoint is a (locally) distinct location in an image
- The feature descriptor summarizes the local structure around the keypoint

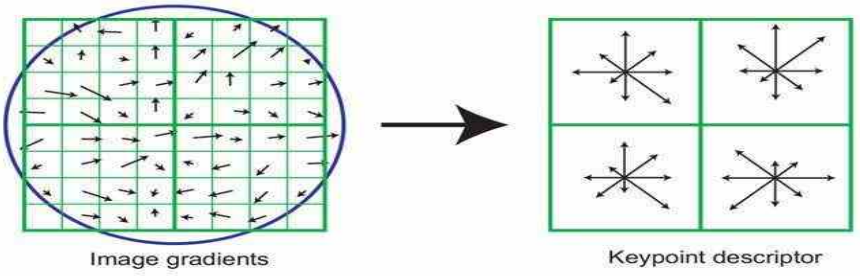


SIFT (Scale Invariant Feature Transform)

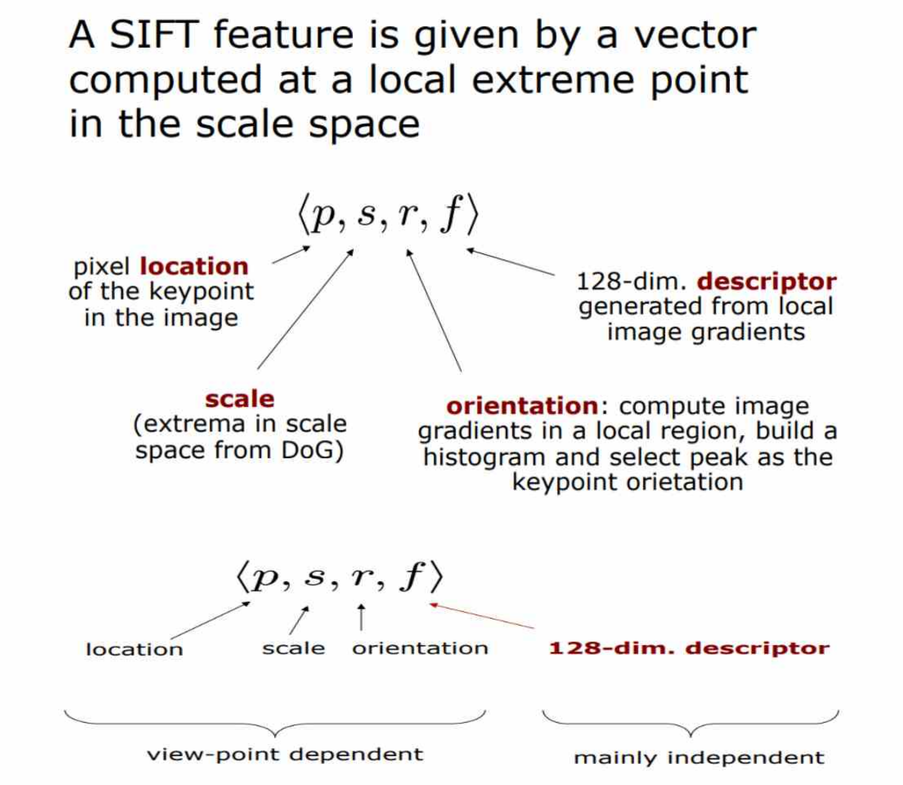
The scale-invariant feature transform (SIFT) is an algorithm used to detect and describe local features in digital images. It locates certain key points and then furnishes them with quantitative information (so-called descriptors) which can for example be used for object recognition. The descriptors are supposed to be invariant against various transformations which might make images look different although they represent the same object.

SIFT Descriptor in Sum

1. Compute image gradients in local 16x16 area at the selected scale
2. Create an array of orientation histograms
3. 8 orientations x 4x4 histogram array = 128 dimensions (yields best results)



A SIFT feature is a selected image region (also called keypoint) with an associated descriptor. Keypoints are extracted by the SIFT detector and their descriptors are computed by the SIFT descriptor. It is also common to use independently the SIFT detector (i.e. computing the keypoints without descriptors) or the SIFT descriptor (i.e. computing descriptors of custom keypoints.

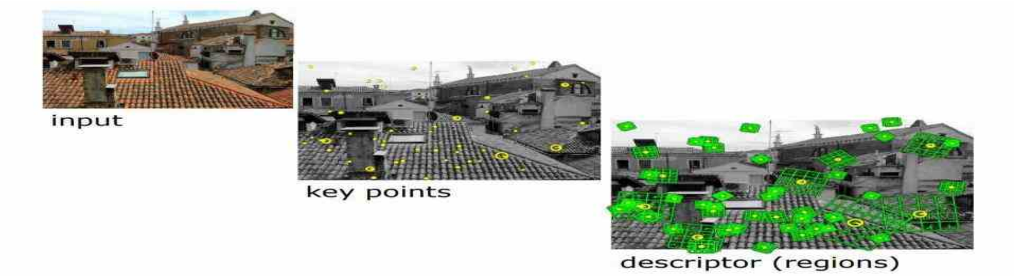

A SIFT feature is given by a vector computed at a local extreme point in the scale space

$\langle p, s, r, f \rangle$

pixel **location** of the keypoint in the image
scale (extrema in scale space from DoG)
orientation: compute image gradients in a local region, build a histogram and select peak as the keypoint orietation
128-dim. **descriptor** generated from local image gradients

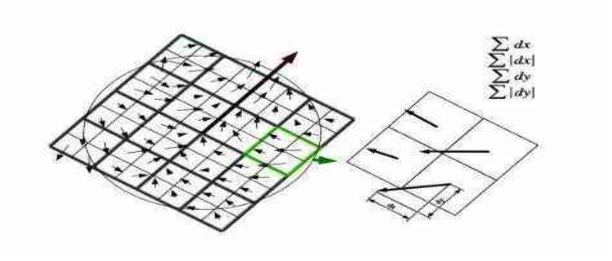
$\langle p, s, r, f \rangle$

location
scale
orientation
128-dim. descriptor
view-point dependent
mainly independent



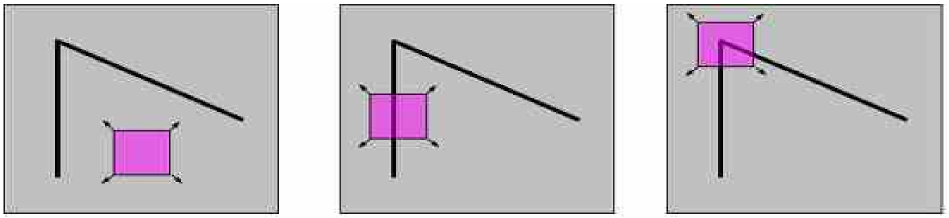
SURF

SURF is the speed up version of SIFT. In SIFT, Lowe approximated Laplacian of Gaussian with Difference of Gaussian for finding scale-space. SURF goes a little further and approximates LoG with Box Filter. One big advantage of this approximation is that, convolution with box filter can be easily calculated with the help of integral images. And it can be done in parallel for different scales. Also, the SURF rely on determinant of Hessian matrix for both scale and location. For orientation assignment, SURF uses wavelet responses in horizontal and vertical direction for a neighborhood of size 6s. Adequate guassian weights are also applied to it. The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window of angle 60 degrees. wavelet response can be found out using integral images very easily at any scale. SURF provides such a functionality called Upright-SURF or U-SURF. It improves speed and is robust upto . OpenCV supports both, depending upon the flag, upright. If it is 0, orientation is calculated. If it is 1, orientation is not calculated and it is faster.



Lecture7

Corner detection is an approach used within computer vision systems to extract certain kinds of features and infer the contents of an image. Corner detection is frequently used in motion detection, image registration, video tracking, image mosaicing, panorama stitching, 3D reconstruction and object recognition. Corner detection overlaps with the topic of interest point detection.



"flat" region:
no change in all
directions

"edge":
no change along the
edge direction

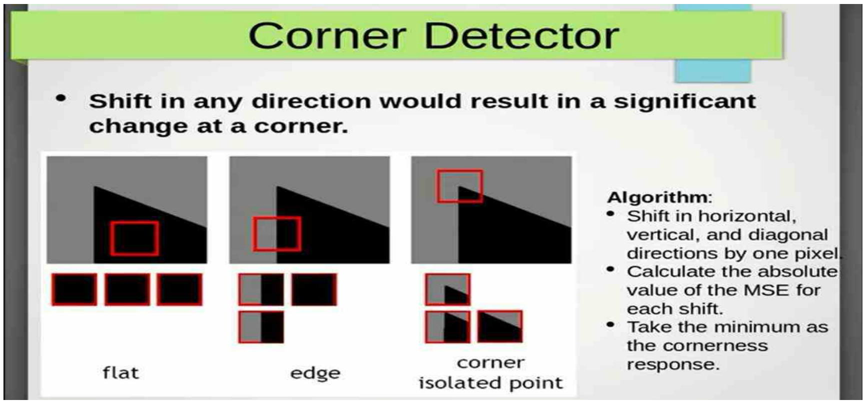
"corner":
significant change in
all directions

A corner can be defined as the intersection of two edges. A corner can also be defined as a point for which there are two dominant and different edge directions in a local neighbourhood of the point.

An interest point is a point in an image which has a well-defined position and can be robustly detected. This means that an interest point can be a corner but it can also be, for example, an isolated point of local intensity maximum or minimum, line endings, or a point on a curve where the curvature is locally maximal.

As a consequence, if only corners are to be detected it is necessary to do a local analysis of detected interest points to determine which of these are real corners. Examples of edge detection that can be used with post-

processing to detect corners are the Kirsch operator and the Frei-Chen masking set.



Corner Detector

- Shift in any direction would result in a significant change at a corner.

Algorithm:
- Shift in horizontal, vertical, and diagonal directions by one pixel.
- Calculate the absolute value of the MSE for each shift.
- Take the minimum as the cornerness response.

flat
edge
corner
isolated point

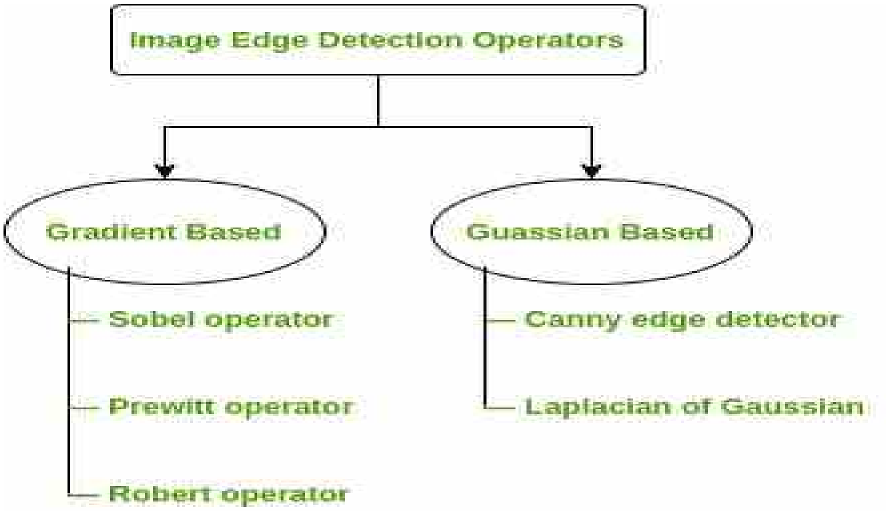
Edges are significant local changes of intensity in a digital image. An edge can be defined as a set of connected pixels that forms a boundary

: between two disjoint regions. There are three types of edges

- ○ Horizontal edges
- ○ Vertical edges
- ○ Diagonal edges

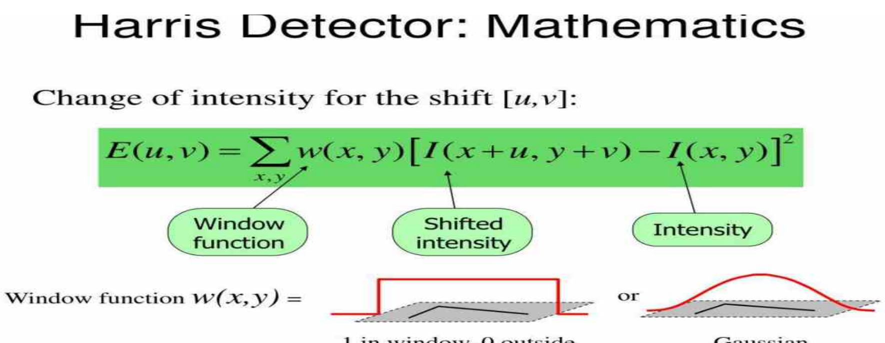
Edge Detection is a method of segmenting an image into regions of discontinuity. It is a widely used technique in digital image processing like

- pattern recognition
- image morphology
- feature extraction



Harris Corner Detection

A corner is a point whose local neighborhood is characterized by large intensity variation in all directions. Corners are important features in computer vision because they are points stable over changes of viewpoint and illumination. The so-called Harris Corner Detector was introduced by Chris Harris and Mike Stephens in 1988 in the paper “A Combined Corner and Edge Detector.”


Harris Detector: Mathematics
Change of intensity for the shift [u,v]:
$E(u,v)=\sum_{x,y} w(x,y)\left[I(x+u,y+v)-I(x,y)\right]^2$
Window function
Shifted intensity
Intensity
Window function $w(x,y)$ =
or
1 in window, 0 outside
Gaussian

E *-The difference between the original and the next window.*

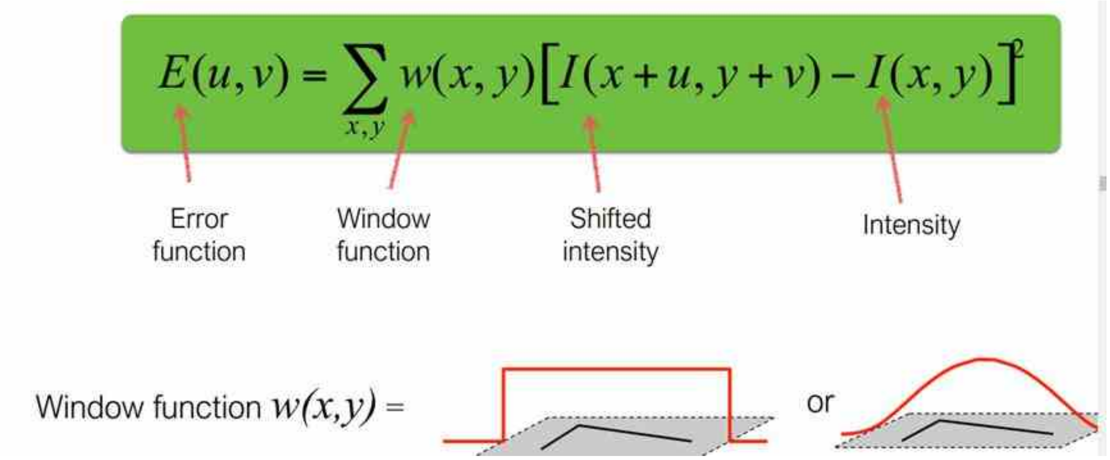
u *— The window's displacement in the x-direction* ***v*** *— The window's displacement in the y-direction* ***w(x, y)*** *— Current position of the window.*

I *— The intensity of the image at a position (x, y).*

I(x, y) *— The intensity of the original window*

I(x+u, y+v) *— The intensity of the next window.*

Taylor series Expansion is used to compute *E*.


$E(u,v)=\sum_{x,y} w(x,y)\left[I(x+u,y+v)-I(x,y)\right]^2$
Error function
Window function
Shifted intensity
Intensity
Window function $w(x,y)$ =
or

Let's consider a two-dimensional image I and a patch W of size m*m centered in (x_0,y_0). We want to evaluate the intensity variation occurred if the window W is shifted by a small amount (u, v). Such variation can be estimated by computing the Sum of Squared Differences (SSD):

$$\mathrm{SSD}(u,v)=\sum_{(x,y)\in W} g(x,y)\left[I(x,y)-I(x+u,y+v)\right]^2$$

where g(x,y) [Current position of the window]is a window function that can be a rectangular or a Gaussian function. We need to maximize the function the SSD(u,v) [The difference between the original and the next window] for corner detection. Since u and v are small, the shifted

intensity I(x+u, y+v)[I=The intensity of the image at a position (x, y)] can be approximated by the following first-order Taylor expansion[***u*** *= The window's displacement in the x-direction* ***,v*** *= The window's displacement in the y-direction*].

$$I(x+u,y+v)\approx I(x,y)+uI_x(x,y)+vI_y(x,y)$$

where I_x and I_y are partial derivatives of I in x and y direction, respectively. By substituting (2) in (1) we obtain:

$$\mathrm{SSD}(u,v) \approx \sum_{(x,y)\in W} g(x,y)\left[u^2 I_x^2 + 2uvI_xI_y + v^2I_y^2\right].$$

The equation (3) can be expressed in the following matrix form:

$$\mathrm{SSD}(u,v) \approx \left[\begin{array}{cc} u & v \end{array}\right] M \left[\begin{array}{c} u \\ v \end{array}\right]$$

where M is a 2 * 2 matrix computed from image derivatives:

$$M = \sum_{(x,y)\in W} g(x,y)\left[\begin{array}{cc} I_xI_x & I_xI_y \\ I_xI_y & I_yI_y \end{array}\right]$$

The matrix M is called structure tensor . The Harris detector uses the following response function that scores the presence of a corner within the patch:

$$R = \det(M) - k\,\mathrm{tr}(M)^2.$$

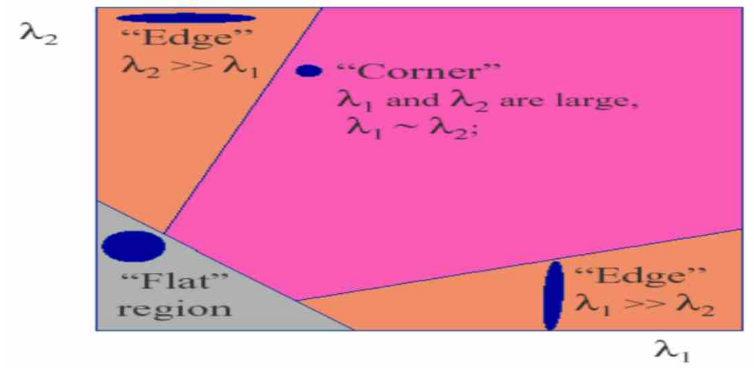
k is a constant to chose in the range [0.04, 0.06]. Since M is a symmetric matrix, det (M) = $\lambda_1\lambda_2$ and tr(M) = $\lambda_1+\lambda_2$ where λ_1 and λ_2 are the eigenvalues of M. Hence, we can express the corner response as a function of the eigenvalues of the structure tensor:

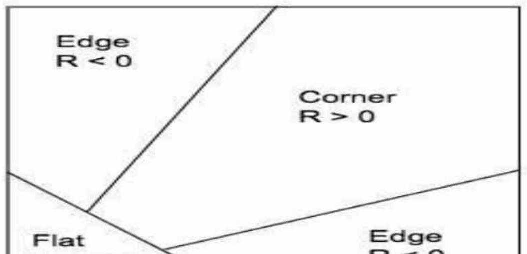
$$R = \lambda_1\lambda_2 - k\,(\lambda_1 + \lambda_2)^2.$$

So the eigenvalues determine whether a region is an edge, a corner or flat:

- if λ_1 and λ_2 are small, then |R| is small and the region is flat.
- if $\lambda_1 >> \lambda_2$ or vice versa, then R <0 and the region is an edge.
- if $\lambda 1$ $\lambda 2$ and both eigenvalues are large, then R is large and the region is a corner.

The classification of the points using the eigenvalues of the structure tensor is represented in the following figure:





Harris corner detector consists of the following steps.

1- Convert the original image into a grayscale image I. The pixel values of *I* are computed as a weighted sum of the corresponding R, G, B values: $I = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$

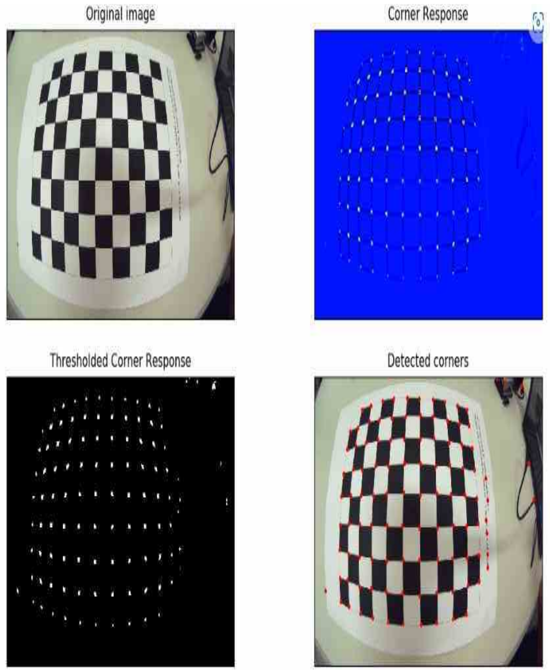
2- Compute the derivatives I_X and I_Y by convolving the image *I* with the Sobel operator:

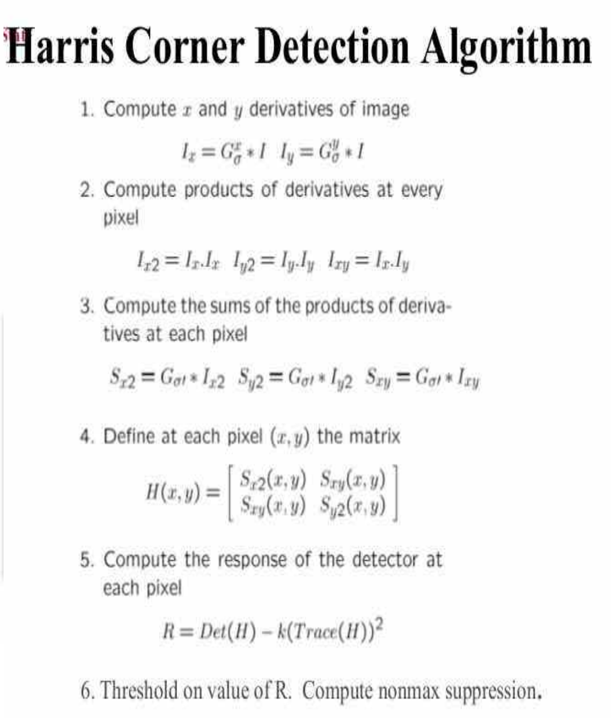
$$I_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * I \quad \text{and} \quad I_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * I$$

3- Compute the products of the derivatives I_XI_X,I_XI_Y,I_YI_Y.

4- Convolve the images I_XI_X,I_XI_Y,I_YI_Y with a Gaussian filter or a mean filter. Define the structure tensor for each pixel as expressed in eq.

5- Compute the response function for each pixel:$R = \det(M) - k\,\mathrm{tr}(M)^2$. 6- Set a threshold T on the value of R and find pixels with responses above this threshold. Finally, compute the non-max suppression in order to pick up the optimal corners.





Harris Corner Detection Algorithm

1. Compute x and y derivatives of image

$$I_x = G_\sigma^x * I \quad I_y = G_\sigma^y * I$$

2. Compute products of derivatives at every pixel

$$I_{x2} = I_x.I_x \quad I_{y2} = I_y.I_y \quad I_{xy} = I_x.I_y$$

3. Compute the sums of the products of derivatives at each pixel

$$S_{x2} = G_{\sigma'} * I_{x2} \quad S_{y2} = G_{\sigma'} * I_{y2} \quad S_{xy} = G_{\sigma'} * I_{xy}$$

4. Define at each pixel (x, y) the matrix

$$H(x,y) = \begin{bmatrix} S_{x2}(x,y) & S_{xy}(x,y) \\ S_{xy}(x,y) & S_{y2}(x,y) \end{bmatrix}$$

5. Compute the response of the detector at each pixel

$$R = Det(H) - k(Trace(H))^2$$

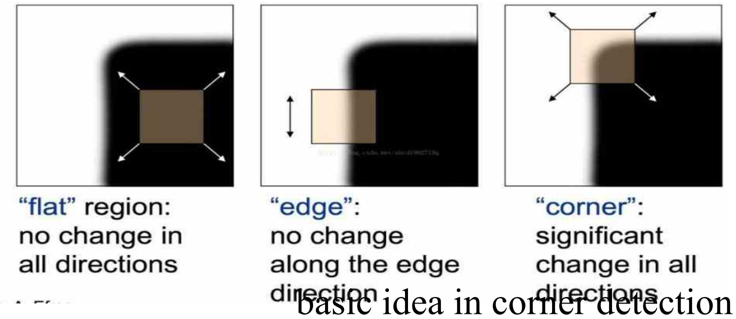
6. Threshold on value of R. Compute nonmax suppression.

Interest Point

An interest point is a point in an image which has a well defined position and can be robustly detected. A good example of interest point is a corner.

Corner Detection

- A corner can be defined as the intersection of two edges, it represents a point in which the directions of these two edges change, and it is characterized by a region with intensity change in two different directions.
- The basic idea in corner detection, at a corner shifting a window in any direction should give a large change in intensity.
- Corner detection is a popular research area in image processing and therefore many corner detectors have been presented, such as SUSAN detector Harris detector and FAST detector.



basic idea in corner detection

The Properties of good corner detectors

- Detect all (or most) true interest points .
- No false interest points
- Well localized
- Robust with respect to noise.
- Efficient detection.

Lecture 8

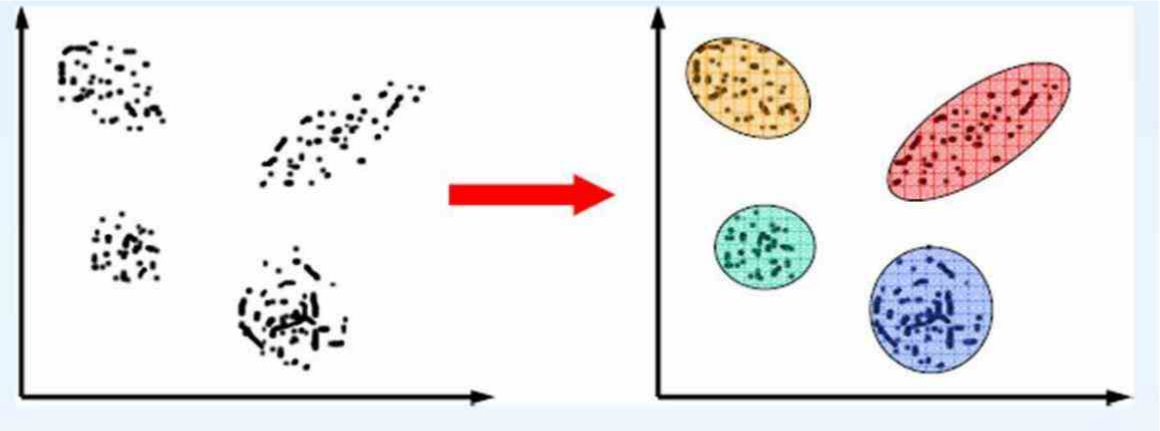
Clustering Techniques Algorithms

1. Introduction

Clustering is *unsupervised Learning* technique (learning without a priori knowledge about the classification of samples; learning without a teacher) which aims at grouping a set of objects into clusters so that objects in the same clusters should be similar as possible, whereas objects in one cluster should be as dissimilar as possible from objects in other clusters. It can be defined in different ways :-

- Clustering is *"the process of organizing objects into groups whose members are similar in some way".*
- Grouping a set of data objects into clusters.
- Clustering is the classification of similar objects into separated groups.
- Clustering is a basic tool used in data analysis, pattern recognition and data mining for finding unknown groups in data.

A cluster is therefore *a collection of objects which are "**similar**" between them and are "dissimilar" to the objects belonging to other clusters*. Similar objects (same cluster) should be ***close*** to one another (short distance).



There are two types of clustering which are hard and soft clustering. Object in hard clustering either belong to cluster or not while in soft clustering the objects add to cluster depending on specific probability.

What Is Good Clustering?

- A good clustering method will produce **high quality** clusters in which:
  - ✓ The intra-class similarity is high.

✓ The inter-class similarity is low, as shown in the above figure.

- The quality of a clustering result also depends on both the similarity measure used by the method and its implementation.
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns

2. General Applications of Clustering

- Pattern Recognition
- Spatial Data Analysis
- social media analyzing
- Image Processing
- Economic Science (especially market research)
- Document classification
- Cluster Weblog data to discover groups of similar access patterns

3. The basic steps to develop a clustering task

a. **Feature selection**:- Features must be *properly selected* so as to encode as much information as possible concerning the task of interest.

b. **Proximity measure**:- This is a measure that quantifies *how "similar" or "dissimilar*" two feature vectors are.

c. **Clustering criterion**:- This depends on the *interpretation the expert gives to the term "sensible"*, based on the type of clusters that are expected to underlie the data set. The clustering criterion may be expressed via a cost function or some other types of rules.

d. **Clustering algorithms**:- This step refers to the choice of a specific algorithmic scheme that unravels the clustering structure of the data set.

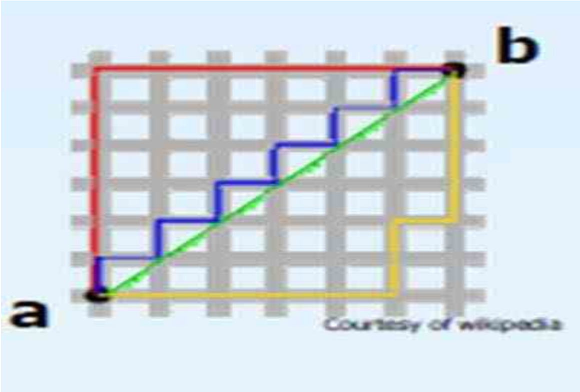
e. **Validation the results**:- Once the results of the clustering algorithm have been obtained, we have to verify their correctness. This is usually carried out using appropriate tests.

f. **Interpretation of the results**:-In many cases, the expert in the application field must integrate the results of clustering with other experimental evidence and analysis in order to draw the right conclusions.

4. Similarity and Dissimilarity Between Objects

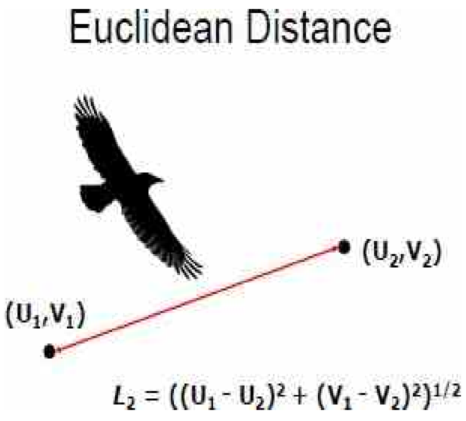
- Similarity between people, units, or objects can be quantified using either *correlation metrics or distance metrics.*
- Distances are normally used to measure the similarity or dissimilarity between two data objects.
- Items within a cluster are similar, and/or the distance between them is small.
- Items in different clusters are dissimilar, and/or the distance between them is large.
- Similarity is expressed in terms of a distance function, which is typically metric: d (i, j).
- The definitions of distance functions are usually very different for interval-scaled, Boolean, Categorical, ordinal, and ratio variables.

4.1 Distance Measures for Metric Variables



$$D(a,b)=\left(\sum_{k=1}^{d}(a_k-b_k)^2\right)^{1/2}$$

1. Euclidean distance (green path):



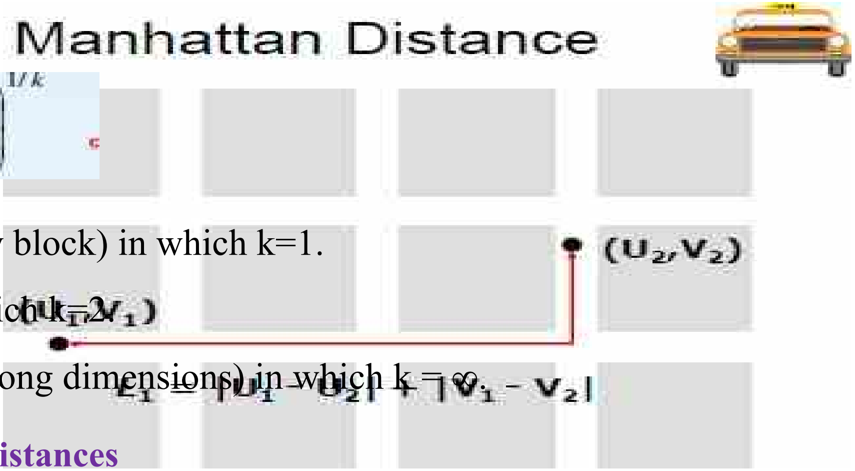
2. Manhattan distance or City Block distance :

(Yellow, red and blue paths give same distance)

$$D(a,b) = \sum_{k=1}^{d} |a_k - b_k|$$

3. General Minkowski Metrics:-

$$L_k(a,b) = \left(\sum_{i=1}^{d} |a_i - b_i|^k \right)^{1/k}$$



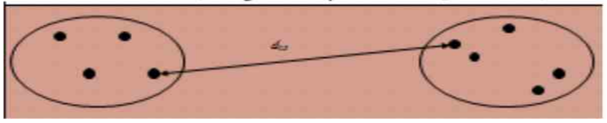
- L1 (Manhattan or city block) in which k=1.
- L2 (Euclidean) in which k=2.
- L∞ (max distance among dimensions) in which k = ∞.

4.2 Inter/Intra Cluster Distances

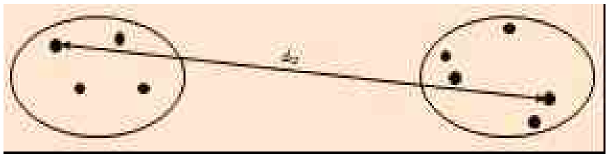
- **Inter-cluster distance**:- Methods to define a distance between clusters:-

1- single linkage: Similarity of two clusters is based on the two most similar (closest) points in the different clusters:-

$$d_{IJ} = \min \{ d_{ij} : i \in I \text{ and } j \in J \}$$

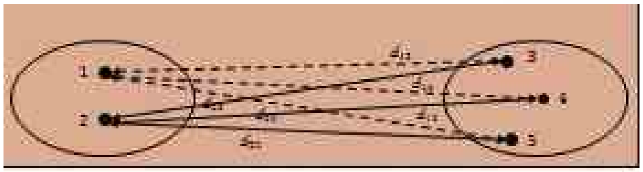


2- Complete linkage: Similarity of two clusters is based on the two least similar (farthest distant) points in the different clusters:- $d_{IJ} = \max \{ d_{ij} : i \in I \text{ and } j \in J \}$



Group average: Proximity of two clusters is the average of pair wise:-

It can be measured by calculating the sum of the distance between all pairs of clusters divided by the number of points. $dIJ = \sum_{i=1}^{I} \sum_{j=1}^{J} \frac{dij}{(NI\ NJ)}$



Where N is the number of members in a each cluster.

3- Centroid linkage: is the distance between the clusters centers.

$$dIJ = d(\bar{x}I, \bar{x}j) \text{ where } \bar{x}J = \frac{1}{NJ} \sum_{j=1}^{NJ} xj$$

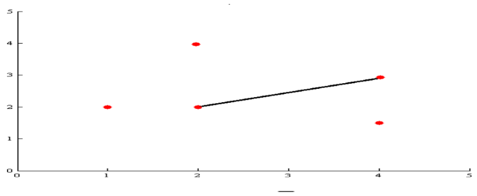
- **Intra cluster distance**: is the distance between the cluster members.

This distance is used to ***evaluate*** the clustering methods under consideration, so the clustering method which gives ***minimum*** intra cluster distance, can be considered as the best method.

The Sum of Squared Error (SSE) and the Standard Deviation (Sd) can be used to calculate the intra cluster distance.

Hence, A good clustering is one where (Intra-cluster distance) the sums of distances between objects in the same cluster are minimized. (Inter-cluster distance) the distances between different clusters are maximized.

Example 1. Find the Euclidean distance and the City Block distance between the pair (2,2) , (4,3).



Euclidean distance = $((2 – 4)^2 + (2-3)^2)^{½} = \sqrt{5} = 2.23$ City Block distance = $| 2 – 4 | + | 2- 3 | = 3$

What makes a Good Similarity Metric

- Symmetry: **d(x,y) = d(y,x)**.
- Triangular inequality: **d(x,y) ≤ d(x,z) + d(z,y).**
- Non identical distinguishability: **if d(x,y) ≠ 0 then x ≠ y.**
- Identical nondistinguishability: **if x = y, then d(x,y) = 0,**In other words, distance to itself is zero: **(d(x,x)=0).**

Major Clustering Method

Unsupervised classification (clustering) has the advantage of no extensive prior knowledge of the region is required. Clustering algorithms may be classified as listed below:

1) Partitioning Clustering:

It is simply a division of the set of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset.

1- K-means Clustering:-

- Partitioning clustering approach.
- Each cluster is associated with a centroid (center point).
- Each point is assigned to the cluster with the closest centroid.
- Number of clusters, K, must be specified.
- The basic algorithm is very simple.

How K-means does works

1- Partition the objects in k clusters (can be done by random partitioning or by arbitrarily clustering around two or more objects).

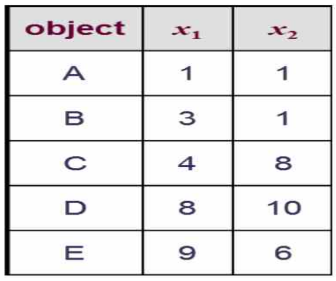
2- Calculate the centroids of the clusters.

3- Assign or reassign each object to that cluster whose centroid is closest (distance is calculated as Euclidean distance).

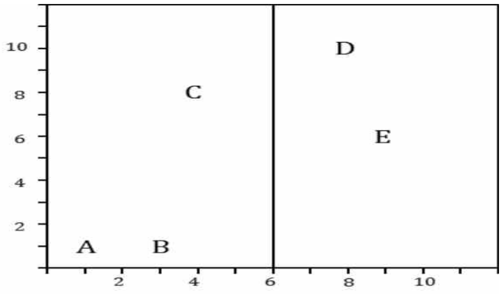
4- Recalculate the centroids of the new clusters formed after the gain or loss of objects to or from the previous clusters.

5- Repeat steps 3 and 4 for a predetermined number of iterations **or** until membership of the groups no longer changes.

Example 1: Given the below objects, apply k-mean algorithm to partition those objects into two clusters.



object	x_1	x_2
A	1	1
B	3	1
C	4	8
D	8	10
E	9	6



Step 1: make an arbitrary partition of the objects into clusters: e.g. objects with x1≤6 into Cluster 1, all other into Cluster 2 A,B and C in Cluster 1, and D and E in Cluster 2.

Step 2: calculate the centroids of the clusters:

Cluster 1: C1 = 2.67 , C2 = 3.33

Cluster 2: C1 = 8.50 , C2 = 8.00

Step 3: calculate the Euclidean distance between each object and each of the two clusters centroids:

Objects	d (x, centriod 1)	d (x, centriod 2)
A	2.87	10.26
B	2.35	8.90
C	4.86	4.50
D	8.54	2.06
E	6.87	2.06

Step 4: C turns out to be closer to Cluster 2 and has to be reassigned cluster 2.

Repeat step2 and step3.

Step 2-repeated . calculate the centroids of the clusters:

Cluster 1: C1 = 2.00 , C2 = 1.00

Cluster 2: C1 = 7.00 , C2 = 8.00

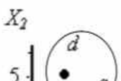
Step 3-repeated: calculate the Euclidean distance between each object and each of the two clusters centroids:

Objects	d (x, centriod 1)	d (x, centriod 2)
A	1.00	9.22
B	1.00	8.06
C	7.28	3.00
D	10.82	2.24
E	8.60	2.83

No further reassigning is necessary.

<u>Example2</u>: The daily expenditures on food (X1) and clothing (X2) of five persons are shown below. Apply the k-mean algorithm using : k = 2, the **first two objects** as the initial clusters, use the Manhattan distance as distance measure, give the resulted two clusters.

Person	X_1	X_2
a	2	4



Step 1: Initialization: Randomly we choose the first two persons as two centroids (k=2) for two clusters. In this case the 2 centroid are: c1=(2,4) and c2=(8,2).

	Person	Cluster center	
		X1	X2
Cluster 1	A	2	4
Cluster 2	B	8	2

Step 2: we calculate the distance between cluster centriod to each object. Using Manhattan distance then we have the following distance

Objects	d (x, centriod 1)	d (x, centriod 2)
A	✓ 0.00	8.00
B	8.00	✓ 0.00
C	8.00	✓ 2.00
D	✓ 2.00	10.00
E	9.50	✓ 1.5

Thus, we obtain two clusterscontaining: Cluster 1= {A,D} and cluster 2={B,C,E}.

- Their new centroids are:

	Person	Cluster center	
		X1	X2
Cluster 1	A, D	(2+1)/2 = 1.5	(4+5) /2 = 4.5
Cluster 2	B,C,E	(8+9+8.5)/3=8.5	(2+3+1)/3= 2

Step 3: By using these new centroids we compute the Manhattan distance of each object, as shown in table below .

Objects	d (x, centriod 1)	d (x, centriod 2)
A	✓ 1.00	8 .50

B	9.00	✓ 0.50
C	9.00	✓ 1.50
D	✓ 1.00	10.50
E	10.50	✓ 1.00

- Thus, we obtain two clusters containing: Cluster 1= {A,D} and cluster 2={B,C,E}.
- Since, there is no change in the cluster.

  Thus, the algorithm comes to a halt here and final result consist of 2 clusters Cluster 1= {A,D} and cluster 2={B,C,E}.

Evaluating K-means Clusters:-

- The most common measure is Sum of Squared Error (SSE). For each point, the **error** is the distance to the nearest cluster center (mean). To get SSE, we square these errors and sum them.

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

x is a data point in cluster Ci and ***m*** is the representative point for cluster Ci. We can show that ***m_i*** corresponds to the center (mean) of the cluster.

- Given two clusters, we can choose the one with the smaller error. One easy way to reduce SSE is to increase K, the number of clusters.
- A good clustering with smaller K can have a lower SSE than a poor clustering with higher K.

Advantages of K-means:-

1. One of the simplest unsupervised learning algorithms.
2. It is low computation and cost.
3. Requires just single input parameter i.e. K, number of clusters.
4. The objective it tries to achieve is to minimize total intra-cluster variance.

Limitations of K-means:-

1. The result is significantly sensitive to the initial randomly selected cluster centers.
2. Dead centers: are centers that have no members or associated data. They are normally located between two active centers or outside the data range.
3. Works only when mean is defined (Works with numeric data only! what about categorical data?)

4. K-means has problems when the data contains noise or outliers.
5. The number of clusters (K) is difficult to determine.

Results ***depend*** on the metric used to measure distances ***and*** on the value of k.

2- Fuzzy C-mean.

- One of the problems of the k-means algorithm is that it gives a hard partitioning of the data, that is to say that each point is attributed to one and only one cluster. But points on the edge of the cluster, or near another cluster, may not be as much in the cluster as points in the center of cluster.
- Therefore, in fuzzy clustering, each point does not pertain to a given cluster, but has a degree of belonging to a certain cluster, as in fuzzy logic.
- Fuzzy clustering is an extension of k – means clustering.
- For all objects the degrees of membership in the k clusters adds up to one.
- a fuzzy weight ω is introduced, which determines the fuzziness of the resulting clusters
  - ✓ for ω → 1, the cluster becomes a hard partition.
  - ✓ for ω → ∞, overlapping of clusters tends to be more.
  - ✓ typical values are ω = 1.25 and ω = 2.

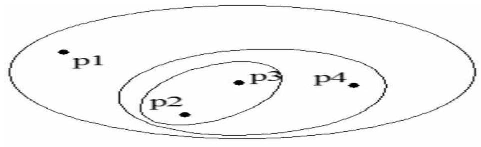
2) **Hierarchical clustering:-** (Follow the binary tree principle)

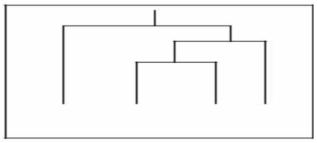
Produces a sequence of nested partitions. More specifically, these algorithms involve N steps, as many as the number of data vectors. At each step t, a new clustering is obtained based on the clustering produced at the previous step t −1. The algorithm steps are:

1- Find dis/similarity between every pair of objects in the data set by evaluating a distance measure.

2- Group the objects into a hierarchical cluster tree (dendrogram) by linking newly formed clusters.

3- Obtain a partition of the data set into clusters by selecting a suitable „cut-level‟ of the cluster tree.

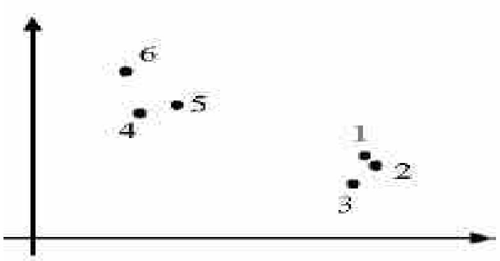


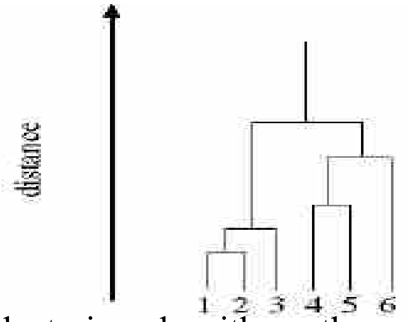


P1 P2 P3 P4

Traditional Hierarchical Clustering Traditional Dendrogram

Dendrogram is an effective means of representing the sequence of clusterings produced by an agglomerative algorithm.





There are two main categories of this clustering algorithms, the **agglomerative** and **the divisive** hierarchical algorithms**.**

Agglomerative (bottom up)	Divisive (top down)
These algorithms produce a sequence ofclustering of decreasing number of clusters at each step.	These algorithms act in the opposite direction; that is, they produce a sequence of clustering of increasing number of clusters at each step.
The clustering produced at each step results from the previous one by merging ***two*** clusters into one.	The clustering produced at each step results from the previous by splitting a single cluster into ***two***.
Stop when k number of clusters is Achieved	Stop when k number of clusters is achieved

Agglomerative Hierarchical clustering

1- Begin with the disjoint clustering having level L(0) = 0 and sequence number m = 0. 2- Find the least dissimilar pair of clusters in the current clustering, say pair (r), (s), according to d[(r),(s)] = min d[(i),(j)] where the minimum is over all pairs of clusters in the current clustering.

3- Increment the sequence number : m = m +1. Merge clusters (r) and (s) into a single cluster to form the next clustering m. Set the level of this clustering to L(m) = d[(r),(s)].

4- Update the proximity matrix, D, by deleting the rows and columns corresponding to clusters (r) and

(s) and adding a row and column corresponding to the newly formed cluster. The proximity between the new cluster, denoted (r,s) and old cluster (k) is defined in this way:

d[(k), (r,s)] = min d[(k),(r)], d[(k),(s)]

5- If all objects are in one cluster, stop. Else, go to step 2.

The proximity between the new cluster is measured using single-linkage approach in which each cluster is represented by all the objects in the cluster, and the similarity between two clusters is measured by the similarity of the closest pair of data points belonging to different clusters.

Advantages of hierarchical clustering:-

1- No apriori information about the number of clusters required, e.g. like K-mean.

2- Easy to implement and gives best result in some cases.

Limits of hierarchical clustering:-

1- Time complexity of at least O(n^2 *log n*) is required, where „*n*" is the number of data points.

2- Algorithm can never undo what was done previously. In other words There is no provision for reassigning objects that have been incorrectly grouped

3- No method of calculating inter-cluster distances is universally the best but, **single-linkage** clustering is **least successful** and, **group average** clustering tends to be **fairly well**.

4- Based on the type of distance matrix chosen for merging different algorithms can suffer with one or more of the following:

i) Sensitivity to noise and outliers

ii) Breaking large clusters

iii) Difficulty handling different sized clusters and convex shapes

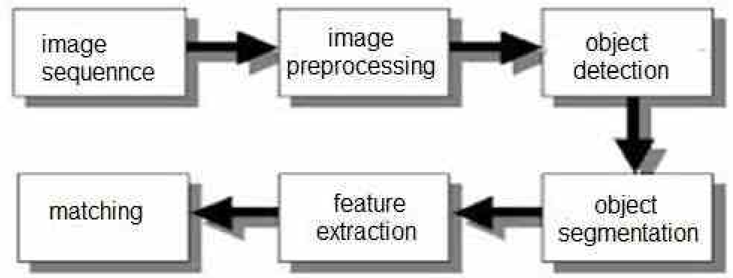
Lecture 9

Template Matching

1. Introduction:

As the simplest theoretical hypothesis in pattern recognition, the Theory of template mainly considers that people store various mini copies of exterior patterns formed in the past in the long-term memory. These copies, named templates, correspond with the exterior stimulation patterns one by one. When a simulation acts on people's sense organs, the simulating information is first coded, ***compared and matched*** with pattern stored in brain, then identified as one certain pattern in brain which ***matches best***. In daily life we can also find out some examples of template matching, comparing with template, machine can recognize the seals on paychecks rapidly.

Template Matching is a technique in digital image processing for finding small parts of an image which match a template image. It can be used in manufacturing as a part of quality control, a way to navigate a mobile robot, or as a way to detect edges in images.



2. Template Matching Approaches:

Template matching is a simple in which instances of pre-stored patterns are sought in an image. Template matching has been performed at the:

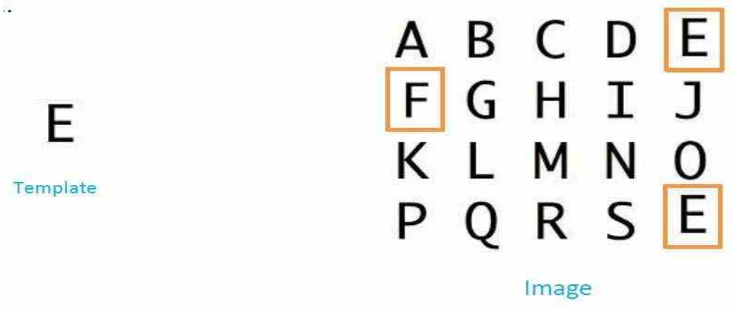
- Pixel level (Low level).
- Regions Level (Higher level).
- Feature level.

2.1 Pixel Level Matching:

There are 4 approaches for low level pixel templates:

a. Total templates: (global basis): Here an ***exact*** match is sought; the Template is the same size as the input image. There is no rotation or translation invariance.

b. Partial templates: (local basis): Here the template is *free* from the background. ***Multiple*** matches are allowed. ***Partial*** matches may also be allowed. Care must be taken in this case -- an **F** template could easily match to an **E**.

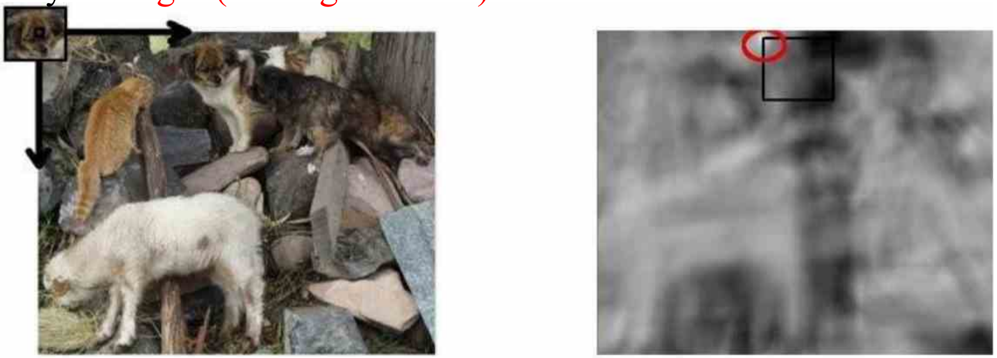


We need two primary components:

✓ **Source image (I):** The image in which we expect to find a match to the template image.

✓ **Template image (T):** The ***patch image*** which will be compared to the source image. our goal is to detect the highest matching area:

To identify the matching area, we have to *compare* the template image against the source image by sliding it (sliding window).



Sliding window

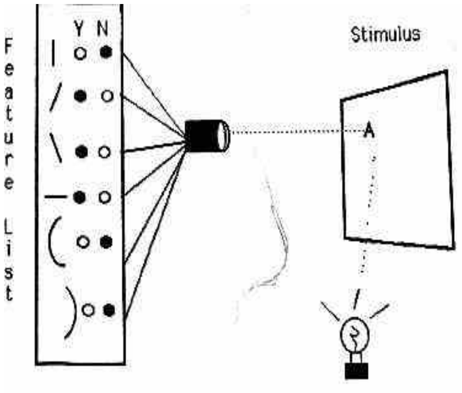
By **sliding**, we mean moving the ***patch image*** one pixel at a time left to right, up to down (convolution operation). At each location, a metric is calculated so it represents how "good" or "bad" the match at that location is (or how similar the patch is to that particular area of the source image).

Methodology

For each location of **T** over **I**, *store* the metric in the *result matrix* **(R)**. Each location in **R** contains the match metric; the brightest locations indicate the highest matches.

c. Piece templates: (local basis)

Here patterns matched are **broken** into component templates. *E.g.* The pattern **A** could be recognized by 3 templates /, \ , **-** ANDed together. Order in which component templates matched is important -- *largest* first. Storage requirements are less in this method.



d. Flexible templates

These templates can handle ***stretching, disorientation*** and other possible deviations.

1. A good prototype of a known object is first obtained and represented ***parametrically (e.g. average intensity value, variance, histogram ,...)***. As other examples are presented then the parameter are modified (*parameter adjusted learning*).
2. Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters (e.g. SIFT, SURF).

Region Level (Higher Level Matching):

A **problem** with pixel based is that although fairly cheap and simple to implement; rotation and translation is a ***problem***, also images are rarely perfect ***suffering*** from blurring, stretched and other distortions and contaminated with noise.

High level template matching methods operate on an image that has typically been ***segmented*** into regions of interest. Regions can be ***described*** in terms of *area, perimeter, and curvature* and also ***compared*** -- *bigger than, adjacent to, above, distance between.*

Templates are described in relationships between regions. Production rules and other linguistic representations have been used here. Also statistical methods (relaxation based techniques) have been applied to perform the matching.

Example face features can be represented in high level feature matching.

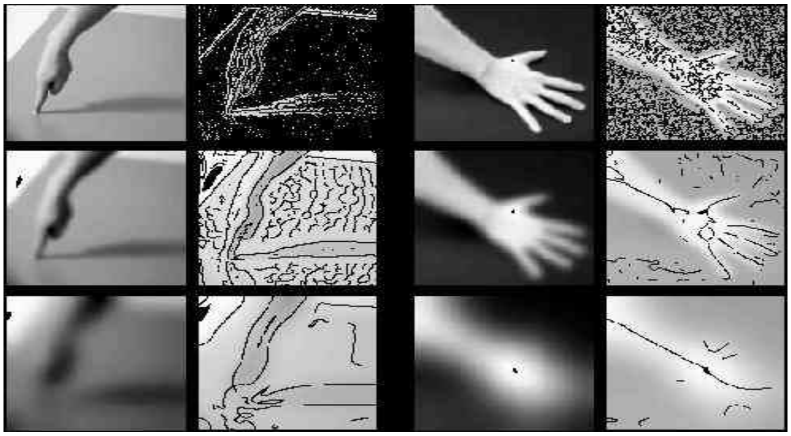
Feature-based Matching:

If the template image has strong features, a feature-based approach may be considered; the approach may prove further useful if the match in the search image might be transformed in some fashion. Since this approach does not consider the entirety of the template image, it can be more computationally efficient when working with source images of larger resolution.

- **Examples:** of feature-based approach (edges , corners):

e. Edge-based Matching

Edge-based Matching enhances the matching operation, since the shape of any object is defined mainly by the shape of its edges. Therefore, instead of matching of the whole template, we could extract its edges and match only the nearby pixels, thus avoiding some unnecessary computations. In common applications ***the achieved speed-up is usually significant***.



f. Corners : edges not oriented along only one of the two directions will be candidate corners.

3. Measures of Match and Mismatch:

Measure of match between two templates is considered to be a metric that indicate the degree of similarity or dissimilarity between them. *This metric can be increasing or decreasing with degree of similarity.* When a metric is specifically stated to be a *measure of match*, it is a quantity is *increasing* with the degree of *similarity*, on the other hand, when a metric is specifically stated to be a *measure of mismatch*, it is a quantity is *increasing* with the degree of *dissimilarity.*

a. Distance measures (difference measures):

These measures of match are based on the pixel-by-pixel intensity differences between the two images *f* and *g*.

1. **Root mean square distance (RMS)**: The RMS distance metric is a common measure of mismatch between two templates. It is given by:

$$d_{rms}(f,g) = \sqrt{\frac{1}{n}\sum_{A}(f_{ij} - g_{ij})^2} \quad \text{……………………. (1)}$$

2. **Euclidean Distance**: Let *I* be a Template image and *g* be a source image of size *n*×*m*. In this formula *(r,c)* denotes the top left corner of source image *g*.

$$d(I,g,r,c) = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m}\left(I(r+i,c+j) - g(i,j)\right)^2} \quad \text{……………………. (2)}$$

3. Sum of Absolute Differences (**SAD**): A pixel in the search image with coordinates **(x_s, y_s)** has intensity **$I_s(x_s, y_s)$** and a pixel in the template with coordinates **(x_t, y_t)** has intensity **$I_t(x_t, y_t)$**.Thus the absolute difference in the pixel intensities is defined as : **$\text{Diff}(x_s, y_s, x_t, y_t) = | I_s(x_s, y_s) - I_t(x_t, y_t) |$**.

$$SAD(x,y) = \sum_{i=0}^{T_{rows}}\sum_{j=0}^{T_{cols}} \text{Diff}(x+i, y+j, i, j) \quad \text{……………………. (3)}$$

The smaller the value of the distance metric, the better the similarity between the template image and the source image.

Note: The above three metric also used to calculate the difference between ***two feature vectors***.

b. Correlation measure:

Correlation is a measure of the degree to which two variables ***agree***, not necessary in actual value but in ***general behavior***. The two variables are the corresponding pixel values in two images: template and source.

$$cor = \frac{\sum_{i=0}^{N-1}(x_i-\bar{x})\cdot(y_i-\bar{y})}{\sqrt{\sum_{i=0}^{N-1}(x_i-\bar{x})^2\cdot\sum_{i=0}^{N-1}(y_i-\bar{y})^2}} \quad \text{..............................(4)}$$

x is the template gray level image
$\bar{x}$ is the average grey level in the template
image y is the source image section
$\bar{y}$ is the average grey level in the source
image N is the number of pixels in the
selected image
N= template image size = (columns * rows)

The value *cor* is between –1 and +1, with larger values representing a stronger relationship between the two images: templates and source.

c. ***Similarity Measures***

The other type of metric used for comparing two templates ***or*** two feature vectors is the similarity measures. In this type of measures, ***the bigger the value of measure the***

greater the similarity between the two objects. The most common form of the similarity measure is the ***vector inner product (or the cross – correlation based)***. Using our

definition for the two vector A and B, we can define the vector inner product by the following equation:

$$\sum_{i=1}^{n} aibi = (a_1b_1 + a_2b_2 + \text{........} + a_nb_n) \quad \text{................................} (5)$$

4. Template Matching Filters:

Search template matching algorithms can be described in terms of a matching filter, where instances of a template *f(x, y)* are to be detected in a search area *s(x, y)*. The filter is designed to give maximal response where a region of the search area matches the template. The filter response, *z(x, y)*, is given by:

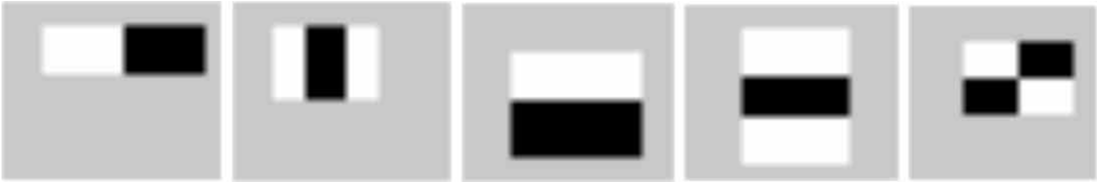
$$z(x, y) = s(x, y) * h(x, y) \quad \text{..} \quad (6)$$

where *h (x, y)* is the matching filter

* denotes the convolution operation.

A basic method of template matching uses a convolution mask (template), tailored to a specific feature of the search image, which we want to detect. This technique can be ***easily*** performed on ***Grey images***. The convolution output will be highest at places where the image structure matches the mask structure, where the image values get multiplied by large mask values (value of 1).

This method is sometimes referred to as Linear Spatial Filtering and the template is called a filter mask.



Five Haar-like features used in Viola-Jones Face Detection Algorithm.

Note the background of a template is painted gray to highlight the pattern's support. Only those pixels marked in black or white are used when the corresponding feature is calculated.

5. Improving the Accuracy of the Matching:

Improvements can be made to the matching method by using more than one template; these other templates can have different scales and rotations.

a. Hybridizing

It is also possible to improve the accuracy of the matching method by ***hybridizing*** the feature-based and template-based approaches. Naturally, this requires that the search and template images have features that are apparent enough to support feature matching.

b. Normalization

Due to the fact that ML calculations use Euclidean distance between information components, the distance ratio between a large component and a small component would produce undesirable results. Thereafter, all provisions must be scaled at the same level. Calculations based on Min-Max or Z-scores should be able to scale information to a specific range (e.g. 0-1).

6. Template Matching Problems:

Problems with template matching are:

a. Intolerance to deviations: this technique ***requires*** a separate template for each scale and orientation.

b. Template matching become thus ***too expensive***, especially for large templates; large number of templates required.

c. Sensitive to noise and occlusion.

d. Cannot specifically support similarity-difference Judgments.

7. **Template Matching Applications:**

   1. Template matching with various average face pyramid levels.
   2. 3D reconstruction.
   3. Motion detection.
   4. Object recognition.
   5. Panorama reconstruction.
   6. Medical image processing.
   7. Geo-statistical simulation which could provide a fast algorithm.

Lecture 10

Classification & ID3

:Introduction

The Pattern Recognition task of assigning an object to a class is said to be a *classification task.* Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts. The model is derived based on the analysis of a set of training data (i.e., data objects for which the class labels are known). The model is used to predict the class label of objects for which the class label is unknown.

Whereas classification predicts categorical (discrete, unordered) labels, regression models are continuous-valued functions. That is, regression is used to predict missing or unavailable numerical data values rather than (discrete) class labels. The term prediction refers to both numeric prediction and class label prediction. Regression analysis is a statistical methodology that is most often used for numeric prediction. Regression also encompasses the identification of distribution trends based on the available data.

Classification is supervised (using class information to design a classifier), while clustering is unsupervised (allocating to groups without class information).

The quality of a feature vector is related to its ability to discriminate examples from different classes. Examples from the same class should have similar feature values, while examples from different classes having different feature values.

Classification process is divided into two main steps. The first is the **training step** where the classification **model is built**. The second is the **classification itself**, in which the trained **model is applied** to assign unknown data object to one out of a given set of class label. (as shown in Fig.1).

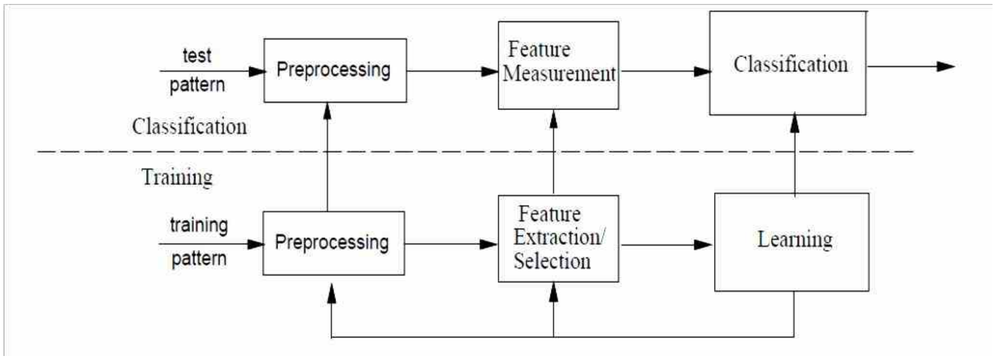


Figure 1:- model is utilized for classification

Note

1. Sometimes classification algorithm act as recognition algorithm, e.g. fruit classification, while most of the times it does not, e.g. vehicle classification.
2. When the classifier does not act as recognizer, the classification algorithm is used as a dimensionality reduction process preceding the recognition step.

In this lecture we will focused on Statistical Classifiers:

1. Parametric Classification

Parametric methods estimate the class by building the parameter estimation probability distribution based on the distribution of the input data. In non-parametric methods, a function (e.g. kernel) is introduced to estimate the class.

Bayes Decision Theory

We assume that the *a* ***priori probabilities*** P(c1), P(c2) are known. This is a very reasonable assumption, because even if they are not known, they can easily be estimated from the available training feature vectors. Indeed, if N is the total number of available training patterns, and N1,N2 of them belong to c1 and c2, respectively, then P(c1) ≈ N1/N and P(c2) ≈ N2/N.

The other statistical quantities assumed to be known are the class-conditional probability density functions ***pdf*** (|), *i*=1, 2, describing the distribution of the feature vectors in each of the classes. If these are not known, they can also be estimated from the available training data. The pdf (|) is sometimes referred to as the *Probability function of x given class ci.*

That is, the feature vectors can take any value in the *l*-dimensional feature space. In the case that feature vectors can take only discrete values, density functions (|)become probabilities and will be denoted by (|).

$$(\quad |) = \frac{(|)()}{()}$$

The ***Bayes classification rule*** can now be stated as

If $P(c1/x) > P(c2/x)$ x is classified to $c1$

If $P(c1/x) < P(c2/x)$ x is classified to $c2$

The case of equality is detrimental and the pattern can be assigned to either of the two classes.

$p(\boldsymbol{x})$ is not taken into account, because it is the same for all classes and it does not affect the decision.

<u>Naïve Bayesian Classification</u> **: a Probabilistic Model**

The Bayesian Classification represents a supervised learning method as well as a statistical method for classification. It is based on the Bayesian theorem; Parameter estimation for Naive Bayesian models uses the method of maximum likelihood.

<u>Theory:</u>

Derivation:

D : Set of tuples

- ✓ Each Tuple is an „n“ dimensional attribute vector
- ✓ X : (x1,x2,x3,…. xn)

Let there be „m“ Classes : C1,C2,C3…Cm

Naïve Bayes classifier predicts X belongs to Class Ci iff

- ✓ P (Ci/X) > P(Cj/X) for 1<= j <= m , j <> i

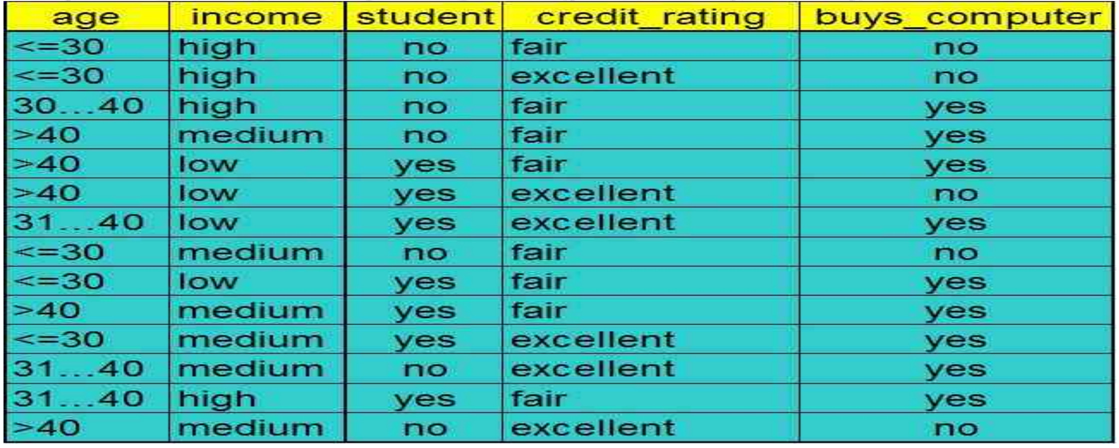
Maximum Posteriori Hypothesis

- ✓ P(Ci/X) = P(X/Ci) P(Ci) / P(X)
- ✓ Maximize P(X/Ci) P(Ci) as P(X) is constant

With many attributes, it is computationally expensive to evaluate P(X/Ci). Naïve Assumption of “class conditional independence”

$$P(X\,/.Ci) = \prod_{k=1}^{n} P(x_k\,/\,Ci)$$

$$P(X/Ci) = P(x1/Ci) * P(x2/Ci) * \ldots * P(xn/\ Ci)$$



age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
30…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Class:
C1:buys_computer=„yes‟
C2:buys_computer=„no‟
Data sample:
X = (age<=30, Income=medium, Student=yes, Credit_rating=Fair)

Sol:

P(buys_computer="yes")=9/14
P(buys_computer="no")=5/14

➢ Compute P(X/C_i) for each class

P(age="<30" / buys_computer="yes") = 2/9=0.222
P(age="<30" / buys_computer="no") = 3/5 =0.6
P(income="medium" / buys_computer="yes")= 4/9 =0.444
P(income="medium" / buys_computer="no") = 2/5 = 0.4
P(student="yes" / buys_computer="yes)= 6/9 =0.667
P(student="yes" / buys_computer="no")= 1/5=0.2
P(credit_rating="fair" / buys_computer="yes")=6/9=0.667
P(credit_rating="fair" / buys_computer="no")=2/5=0.4

➢ X=(age<=30 ,income =medium, student=yes,credit_rating=fair)

P(X/C_i) : P(X/buys_computer="yes")= 0.222 × 0.444 × 0.667 × 0.0.667 =0.044
P(X/buys_computer="no")= 0.6 × 0.4 × 0.2 × 0.4 =0.019

P(X/C_i) × P(C_i) : P(X/ buys_computer="yes") × P(buys_computer="yes")=0.028
P(X/ buys_computer="no") × P(buys_computer="no")=0.007

Since, P(X/C_1)×P(C_1) > P(X/C_2)×P(C_2)
Then , P(C_1/X) > P(C_2/X) from Bayes theorm

X belongs to class "buys_computer=yes"

Advantage:

1. It is particularly suited when the dimensionality of the inputs is high (huge data).
2. Required short computation time.
3. Requires a small amount of training data to estimate the parameters.
4. Easily updated when new training data is received.
5. In spite of its over-simplified assumptions, it often performs better in many complex real world situations.

Disadvantage: requires a very large number of records to obtain good results.

2. Nonparametric Classification : Model Free

In Naïve Bayesian Classifier, we have assumed that the probability density functions are known. However, this is not the most common case. In many problems, the underlying *pdf* has to be estimated from the available data. There are various ways to approach the problem. Sometimes we may know the type of the *pdf* (e.g., Gaussian), but we do not know certain **parameters**, such as the **mean** values or the variances. In contrast, in other cases we may not have information about the type of the *pdf* but we may know certain statistical parameters, such as the mean value and the variance. Depending on the available information, different approaches can be adopted.

Although all of these methods are model-free, their tuning to the particular distributions of the feature vectors is still based on statistical considerations. They are either based on the idea of estimating the *pdf* of the pattern distributions or simply address the problem of choosing the best discriminating thresholds as a compromise in conflicting cost requirements.

Usually the model-free techniques are only used in low dimensional (small data) situations.

Non-Metric Methods

Most practical pattern recognition methods address problems of the sort, where feature vectors are real-valued and there exists some notion of metric. But suppose a classification problem involves *nominal* data | for instance descriptions that are discrete and without any natural notion of similarity or even ordering.

Tree Classifiers

It requires a very large design set for proper training, probably much larger than what we have available. Also, the feature subset that is the most discriminating set for some classes can perform rather poorly for other classes. As an attempt to overcome these difficulties, a ***"divide and conquer"*** principle through the use of multistage classification has been proposed, the so-called ***decision tree*** classifiers, also known as ***hierarchical classifiers,*** where an unknown pattern is classified into a class using decision functions in successive stages.

Decision trees

A decision tree (DT) is a flow chart-like tree structure that represents the knowledge for classification, where each internal node (no leaf node) denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (or *terminal node*) holds a class label and the top most node in a tree is the **root node**. Decision trees are used for classification where each path is a set of decisions leading to one class. In pattern classification, the variable to be used for splitting at each node is to be determined. A path from the root to a leaf node represents a set of rules which leads to a classification (as shown in Fig.2)

By using decision tree, It is natural and intuitive to classify a pattern through a sequence of questions, in which the next question asked depends on the answer to the current question. This approach is particularly useful for non-metric data. Such a sequence of questions is displayed in a directed *decision tree* or simply *tree*, where by convention the first or *root node* is displayed at the top, connected by successive (directional) *links* or *branches* to other nodes.

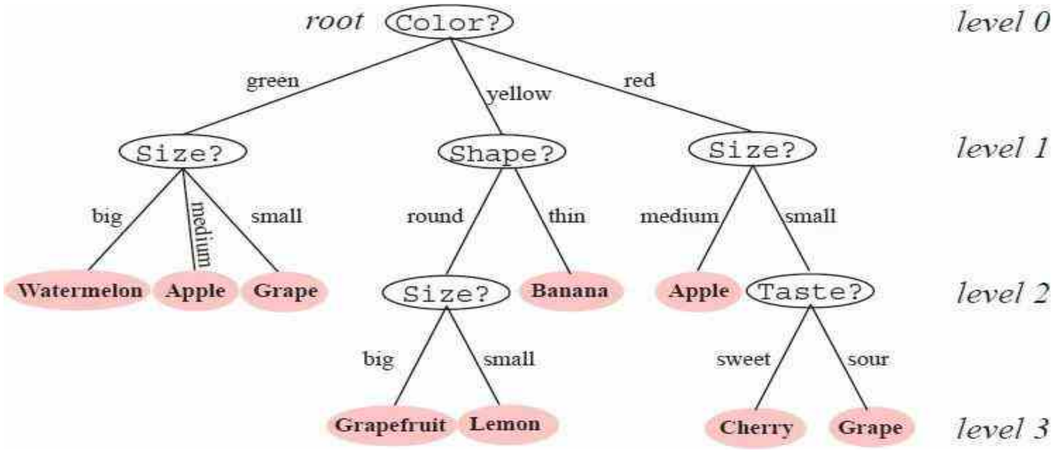


Figure (2): A simple Decision Tree

A decision tree model consists of two parts: creating the tree (building the rules "Model" in training phase) and applying the tree (testing phase) to the data.

The classification of a particular pattern begins at the root node, which asks for the value of a particular property of the pattern. Based on the answer, we follow the appropriate link to a descendent node.

In the decision trees, the links must be mutually distinct and exclusive. It means that one and only one link will be followed. We continue this way until we reach a leaf node, which has no further question. Each leaf note bears a category label, and the test pattern is assigned the category of the leaf note reached.

<u>The Requirements of DT Algorithms</u>

There are several requirements that must be met before applying decision tree algorithms:

1. Decision tree algorithms represent supervised learning, and as such require preclassified target variables.
2. The training dataset should be rich and varied.
3. The target attribute classes must be discrete. That is, one cannot apply decision tree analysis to a continuous target variable.

Stopping Criteria

The splitting phase continues until a stopping criterion is triggered. The following conditions are common stopping rules:

1. All values in the training set belong to the same class.
2. The maximum tree depth has been reached.
3. The best splitting criteria is not greater than a certain threshold.

Characteristic of Decision Tree Classification:

- At each stage of the tree classifier a simpler problem with a smaller number of features can be solved. This has an additional benefit: in practical multi-class problems it may be possible that by using a multistage approach these conditions are approximately met, affording optimal classifiers at each stage.
- In the decision tree, it is a straightforward matter to render the information in such a tree as logical expressions.
- Another benefit of decision trees is that, by employing a sequence of simple queries, they lead to **rapid** classification.
- Finally, decision trees provide a natural way to incorporate prior knowledge from human experts.

Advantages

1. The rules are simple and easy to understand.
2. Able to handle both categorical and numerical data.
3. It produces the accurate result.
4. It takes the less memory.
5. It robustness to noise.

Disadvantage

1. It has long training time.
2. Decision trees can have significantly more complex representation for some concepts due to replication problem.
3. It has a problem of over fitting.

Classification Performance

Accuracy is one important measure that is used to evaluate the performance of classification . To calculate the classification accuracy, it is applied on test set, then count the number of correct predictions and divide it by the total number of predictions, multiply it by 100 to convert it into a percentage. But the accuracy is not enough to evaluate the performance in some datasets, especially when most objects are assigned to specific class. For this, we need another measures to evaluate a classification performance for each class in dataset in addition to accuracy that evaluates the overall correctness of the classifier .

$$Accuracy = \left[\frac{No.of\ correctly\ classified\ samples}{Total\ no.of\ samples}\right] * 100\%$$

Precision and Recall are very efficient to evaluate the classification model for each class when the accuracy is high. Precision is a fraction of correct predictions for specific class from the total number of *predictions* (Error + Correct) for the same class. Recall (also known as sensitivity) is a fraction of correct predictions for the specific class from the total number of *actual* objects that belong to the same class .

ID3 Classifier algorithm

In decision tree learning, ID3 (Iterative Dichotomiser 3) is an algorithm used to generate a decision tree from a dataset. ID3 is typically used in the machine learning and natural language processing domains. The ID3 algorithm selects the best feature at each step while building a Decision tree. *What are the steps in ID3 algorithm are as follows:*

1.Calculate entropy for dataset.
2.For each attribute/feature.
  2.1. Calculate entropy for all its categorical values.
  2.2. Calculate information gain for the feature.
3.Find the feature with maximum information gain.
4.Repeat it until we get the desired tree.

How does it work?

Information Gain calculates the reduction in the entropy and measures how well a given feature separates or classifies the target classes. The feature with the highest Information Gain is selected as the best one.

- The ID3 follows the Occam‟s razorprinciple.
- Attempts to create the smallest possible decision tree.

Advantage of ID3

• Understandable prediction rules are created from the training data.
• Builds the fastest tree.
• Builds a short tree.
• Only need to test enough attributes until all data is classified.
• Finding leaf nodes enables test data to be pruned, reducing number of tests.

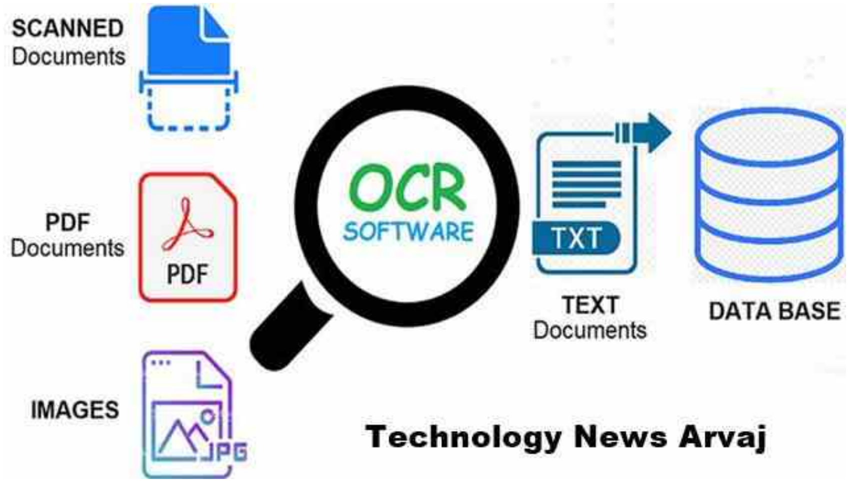
Disadvantage of ID3

• Data may be over-fitted or over classified, if a small sample is tested.
• Only one attribute at a time is tested for making a decision.
• Classifying continuous data may be computationally expensive, as many trees must be generated to see where to break the continuum.

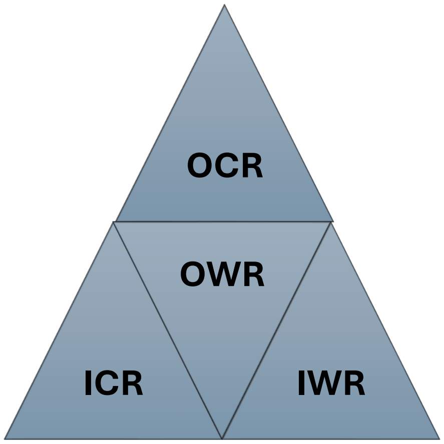
Lecture 11:

Optical Character Recognition (OCR)

- Definition: An **OCR** (**Optical Character Recognition**) **system** is a computerized scanning **system** enabling you to scan text documents into an electronic computer file which you can then edit using a word processor on your computer. **Optical Character Recognition** is the machine recognition of printed text characters.

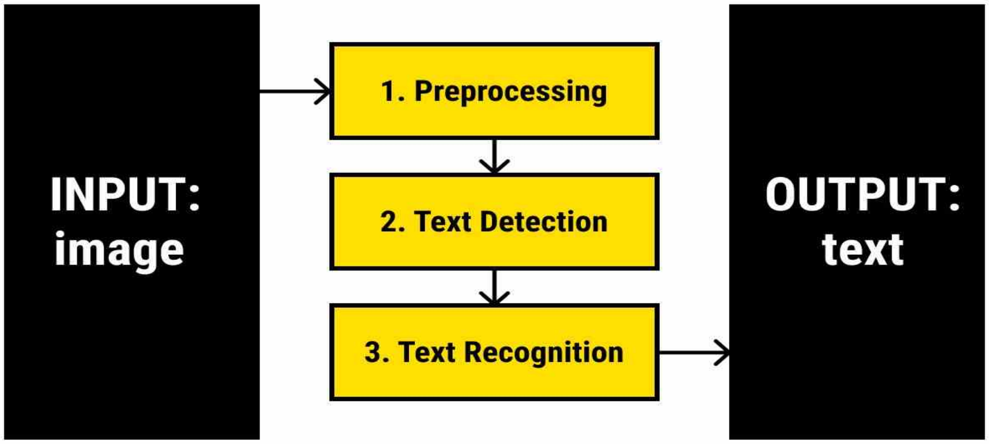


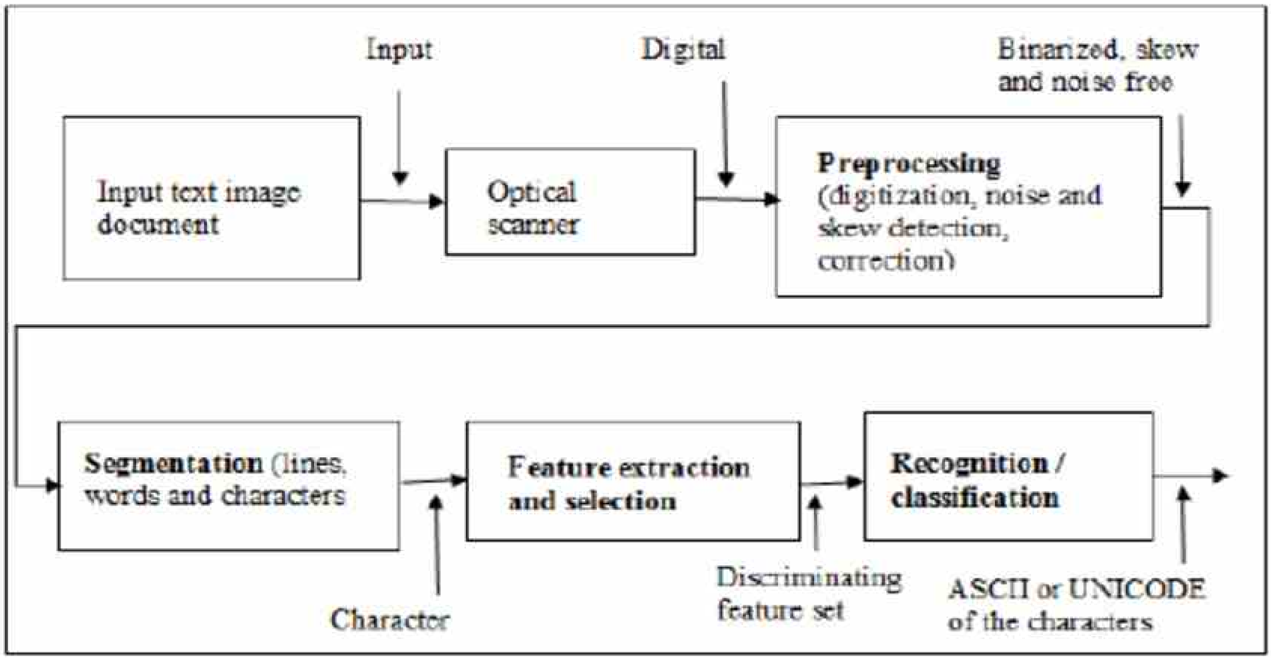
Types:-



- Optical character recognition (OCR) – targets typewritten text, one character at a time.
- Optical word recognition – targets typewritten text, one word at a time.
- Intelligent Character Recognition (ICR): Handwritten print script text one character at a time.
- Intelligent Word Recognition (IWR): Handwritten print script text one word at a time.

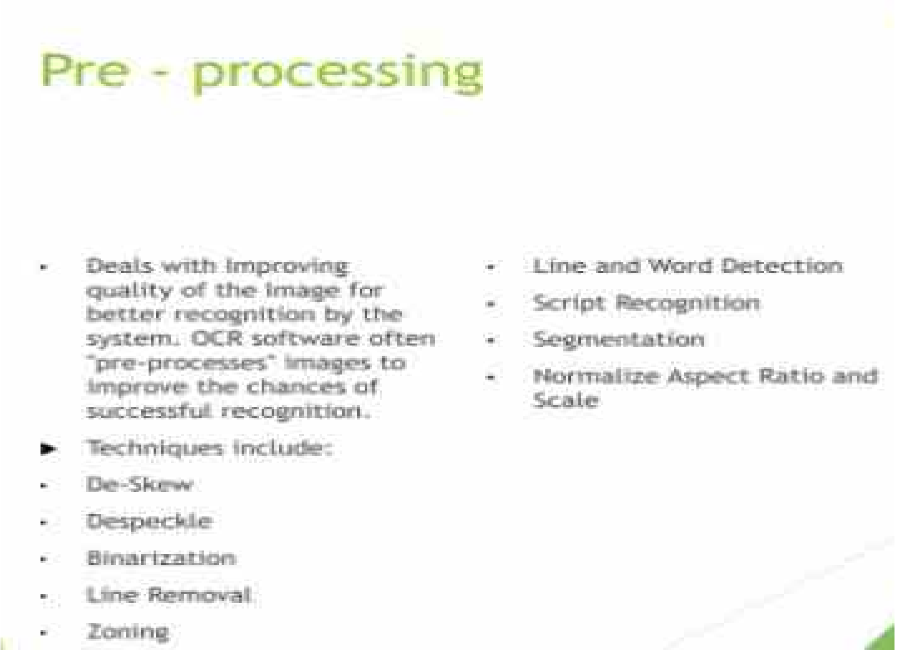
Process:-




Input
Digital
Binarized, skew and noise free
Input text image document
Optical scanner
Preprocessing (digitization, noise and skew detection, correction)
Segmentation (lines, words and characters
Feature extraction and selection
Recognition / classification
Character
Discriminating feature set
ASCII or UNICODE of the characters

- **Pre-processing**

- properly alignment when scanned, it may need to be tilted a few degrees clockwise or counterclockwise in order to make lines of text perfectly horizontal or vertical.
- Remove spots
- Convert an image in binary image (Colors)
- Effectiveness for quality
- Line removal – Cleans up boxes and lines
- Layout: Columns, paragraphs, captions, tables etc.
- Line and word detection: Establishes baseline for word and character shapes, separates words if necessary.
- Script recognition



Pre - processing

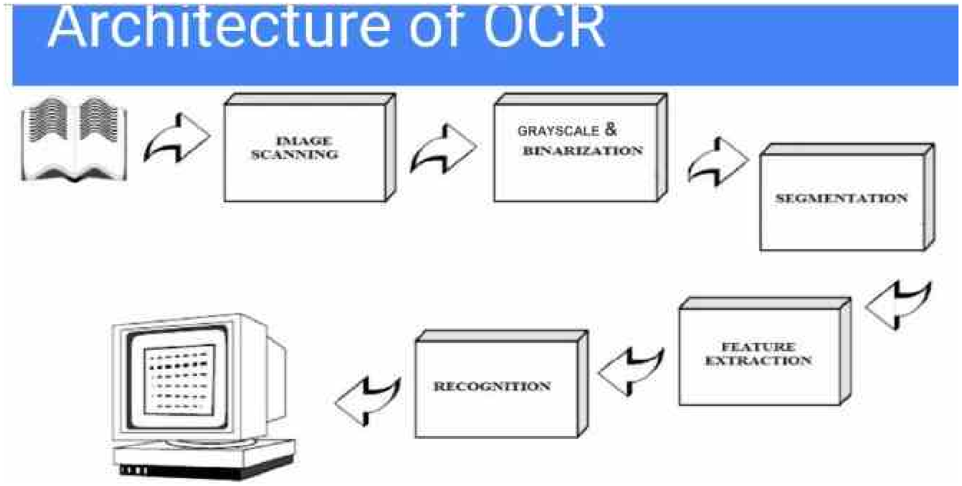
- Deals with Improving quality of the Image for better recognition by the system. OCR software often "pre-processes" images to improve the chances of successful recognition.
- Techniques include:
- De-Skew
- Despeckle
- Binarization
- Line Removal
- Zoning
- Line and Word Detection
- Script Recognition
- Segmentation
- Normalize Aspect Ratio and Scale

- Post-processing

- Out put accuracy:
- Correct errors: For example, "Washington, D.C." is generally far more common in English than "Washington DOC".

- Evaluation

- Character Accuracy
- Page Quality
- Confusions: Generating the letter “c” when the correct character is an “e”.
- Word Accuracy
- Input formats
- Output formats
- Installation and basic features
- Implementation



Architecture of OCR

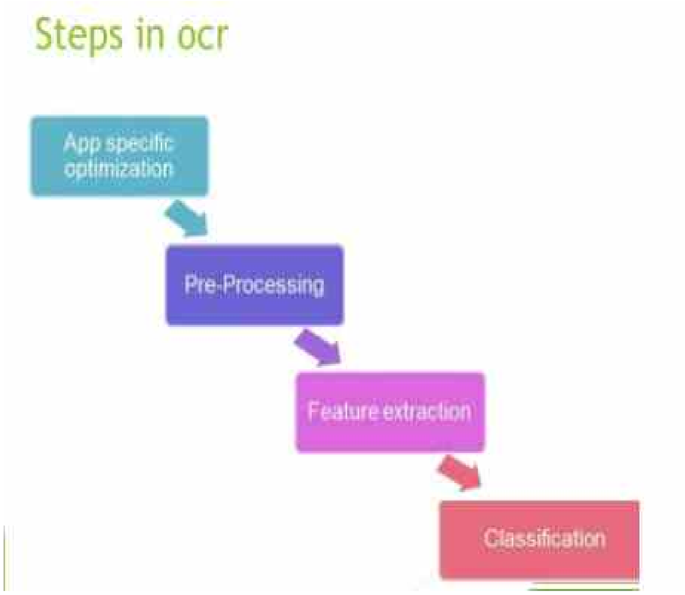
IMAGE SCANNING

GRAYSCALE & BINARIZATION

SEGMENTATION

FEATURE EXTRACTION

RECOGNITION



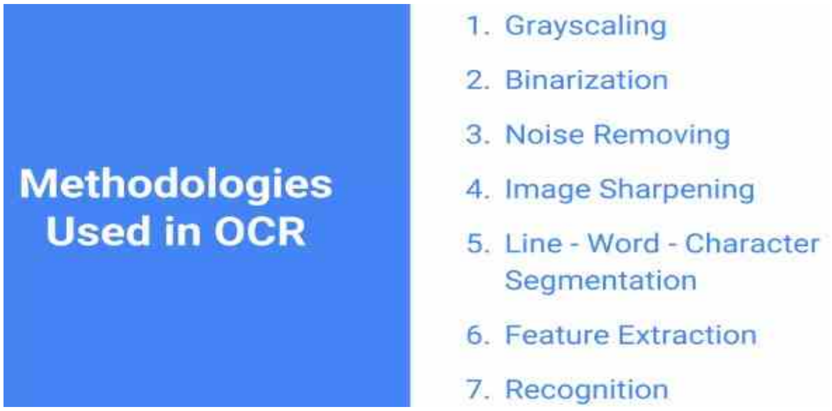
Steps in ocr

App specific optimization

Pre-Processing

Feature extraction

Classification



Methodologies Used in OCR

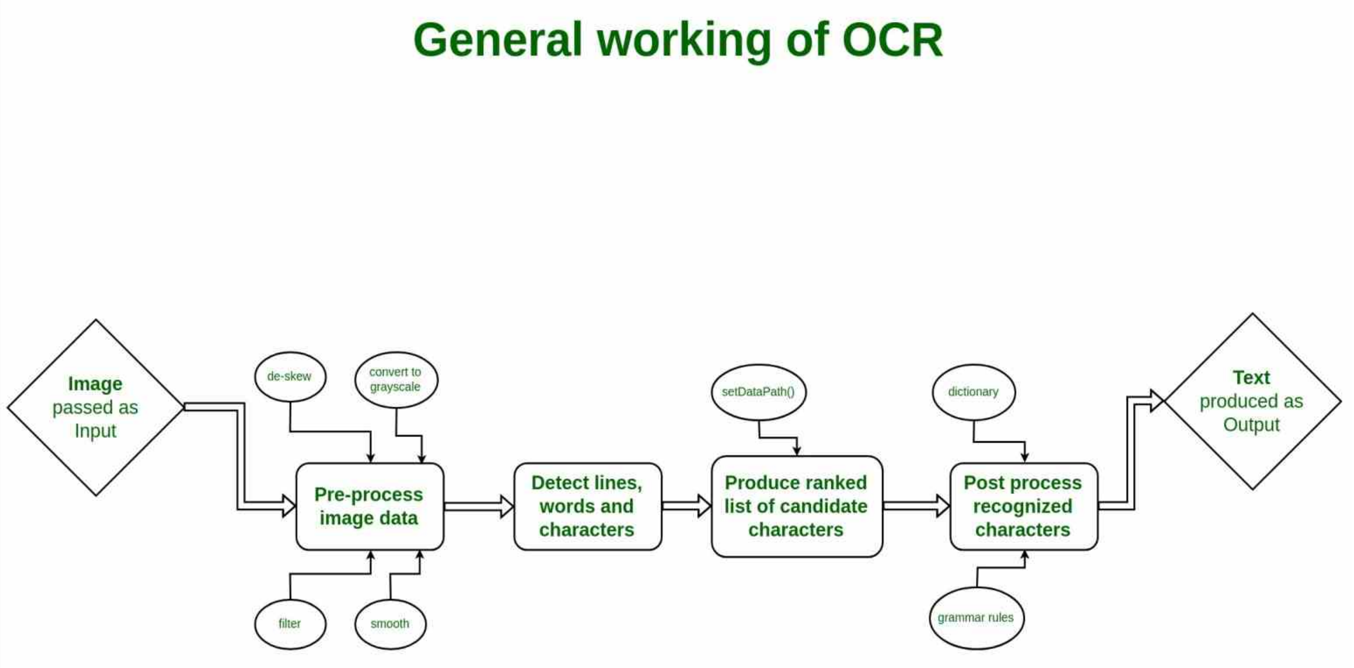
1. Grayscaling
2. Binarization
3. Noise Removing
4. Image Sharpening
5. Line - Word - Character Segmentation
6. Feature Extraction
7. Recognition

Advantages

- Higher Productivity
- Cost Reduction
- High Accuracy
- Increased Storage Space
- Superior Data Security
- Text-searchable Documents
- Improves quality
- Makes Documents Editable

Disadvantages

- No **OCR** scanning system is infallible
- Time-consuming
- Proofreading
- Lack of accuracy
- Additional works
- Limitation of documents: Face problems in old documents.


General working of OCR
Image passed as Input
de-skew
convert to grayscale
filter
smooth
Pre-process image data
Detect lines, words and characters
setDataPath()
Produce ranked list of candidate characters
dictionary
grammar rules
Post process recognized characters
Text produced as Output