RESEARCH ARTICLE OPEN ACCESS

A Review: Classification in Pattern Recognition

Nikam V.R.

Department of Computer Science

Dr S.C. Gulhane college of Arts, Commerce & Science

ABSTRACT

There is a vast difference in recognition of pattern and object. It has lot of application in object detection, Geological fault detection, approximation, and medical diagnosis are of great practical significance those are encompasses by pattern and object recognition. It has been well understood that these problem can perform well effortlessly by human brain. However, their solution using computer has, in many case proved to be immensely difficult. In order to have the best opportunity of developing effective solutions, it is important to consider the various existing approaches and methods for pattern, diagram and object recognition with the machine. In this chapter we are discussing the general introduction of pattern, diagram and object recognition and various techniques or methods those are used for this purpose starting from conventional statistical inference to modern approaches of soft computing integrated non-linear dynamics. Probabilistic approach integrated with neural network gives the better understanding of pattern and object detection. The efficiency of Ant colonization algorithm with non-linear mathematics gives better integration result then probabilistic approach integrated with neural network algorithm.

Keyword:-

1.1 Pattern recognition

Pattern recognition is a branch of machine learning that focuses on the recognition of patterns and regularities in data, although it is in some cases considered to be nearly synonymous with machine learning [1]. Pattern recognition systems are in many cases trained from labeled "training" data (supervised learning), but when no labeled data are available other algorithms can be used to discover previously unknown patterns (unsupervised learning).

Pattern recognition encompasses a wide range of information processing problem of great practical significance, form speech recognition and the classification of hand written characters, to fault detection in machinery and medical diagnosis. It has been well understood that these problem can perform well and effortlessly by human brain. However, their solution using computer has, in many case proved to be immensely difficult. In order to have the best opportunity of developing effective solutions, it is important to consider the various existing approaches and methods for pattern recognition by the machine. In this chapter we are discussing the different approaches of pattern recognition starting from conventional statistical inference to modem approaches of soft computing.

The terms pattern recognition, machine learning, data mining and knowledge discovery in databases (KDD) are hard to separate, as they largely overlap in their scope. Machine learning is the common term for supervised learning methods and originates from artificial intelligence, whereas KDD and data mining have a larger focus on unsupervised methods and stronger connection to business use. Pattern recognition has its origins in engineering, and the term is popular in the context of computer vision: a leading computer vision conference is named Conference on Computer Vision and Pattern Recognition. In pattern recognition, there may be a higher interest to formalize, explain and visualize the pattern, while machine learning traditionally focuses on maximizing the recognition rates. Yet, these domains have evolved substantially from their roots in artificial intelligence, engineering and statistics, and they've become increasingly similar by integrating developments and ideas from each other.

In machine learning, pattern recognition is the assignment of a label to a given input value. In statistics, discriminant analysis was introduced for this same purpose in 1936. An example of pattern recognition is classification, which attempts to assign each input value to one of a given set of classes (for example, determine whether a given email is "spam" or "non-spam"). However, pattern recognition is a more general problem that encompasses other types of output as well. Other examples are regression, which assigns a real-valued output to each input; sequence labeling, which assigns a class to each member of a sequence of values (for example, part of speech tagging, which assigns a part of speech to each word in an input sentence); and parsing, which assigns a parse tree to an input sentence, describing the syntactic structure of the sentence.

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. This is opposed to pattern matching algorithms, which look for exact matches in the input with pre-existing patterns. A common example of a pattern-matching algorithm is regular expression matching, which looks for patterns of a given sort in textual data and is included in the search capabilities of many text editors and word processors. In contrast to pattern recognition, pattern matching is generally not considered a type of machine learning, although pattern-matching algorithms (especially with fairly general, carefully tailored patterns) can sometimes succeed in providing similar-quality output of the sort provided by pattern-recognition algorithms.

Pattern recognition is generally categorized according to the type of learning procedure used to generate the output value. Supervised learning assumes that a set of *training data* (the training set) has been provided, consisting of a set of instances that have been properly labeled by hand with the correct output. A learning procedure then generates a *model* that attempts to meet two sometimes conflicting objectives: Perform as well as possible on the training data, and generalize as well as possible to new data (usually, this means being as simple as possible, for some technical definition of "simple", in accordance with Occam's Razor, discussed below). Unsupervised learning, on the other hand, assumes training data that has not been hand-labeled, and attempts to find inherent patterns in the data that can then be used to determine the correct output value for new data instances. A combination of the two that has recently been explored is semi-supervised learning, which uses a combination of labeled and unlabeled data (typically a small set of labeled data combined with a large amount of unlabeled data). Note that in cases of unsupervised learning, there may be no training data at all to speak of; in other words, the data to be labeled *is* the training data.

Note that sometimes different terms are used to describe the corresponding supervised and unsupervised learning procedures for the same type of output. For example, the unsupervised equivalent of classification is normally known as clustering, based on the common perception of the task as involving no training data to speak of, and of grouping the input data into *clusters* based on some inherent similarity measure (e.g. the distance between instances, considered as vectors in a multi-dimensional vector space), rather than assigning each input instance into one of a set of pre-defined classes. Note also that in some fields, the terminology is different: For example, in community ecology, the term "classification" is used to refer to what is commonly known as "clustering".

The piece of input data for which an output value is generated is formally termed an *instance*. The instance is formally described by a vector of *features*, which together constitute a description of all known characteristics of the instance. (These feature vectors can be seen as defining points in an appropriate multidimensional space, and methods for manipulating vectors in vector spaces can be correspondingly applied to them, such as computing the dot product or the angle between two vectors.) Typically, features are either categorical (also known as nominal, i.e., consisting of one of a set of unordered items, such as a gender of "male" or "female", or a blood type of "A", "B", "AB" or "O"), ordinal (consisting of one of a set of ordered items, e.g., "large", "medium" or "small"), integer-valued (e.g., a count of the number of occurrences of a particular word in an email) or real-valued (e.g., a measurement of blood pressure). Often, categorical and ordinal data are grouped together; likewise for integer-valued and real-valued data. Furthermore, many algorithms work only in terms of categorical data and require that real-valued or integer-valued data be *discretized* into groups (e.g., less than 5, between 5 and 10, or greater than 10).

Recognition is a basic property of all human beings; when a person sees an object, he or she first gathers all information about the object and compares its properties and behaviors with the existing knowledge stored in the mind. If we find a proper match, we recognize it. The concept of recognition is simple in the real world environment, but in the world of computer science, recognizing any object is an amazing feat. The functionality of the human brain is amazing; it is not comparable with any machines or software. The act of recognition can be divided into two broad categories:

1. Concrete item recognition, it involves the ecognition of spatial samples such as fingerprints, weather maps, pictures and physical objects and the recognition of temporal samples such as, waveforms and signatures.

2. Abstract item recognition, it involves the recognition of a solution to a problem, an old conversation or argument.

Pattern recognition, as a subject, spans a number of scientific disciplines, uniting them in search for a solution to the common problem of recognizing the pattern of a given class and assigning the name of identified class. But what is a pattern? Watanabe [2] defines a pattern as opposite of a chaos; it is an entity, vaguely defined, that could be given a name. Pattern could be a fingerprint image, a handwritten cursive word, a human face, or a speech signal.

For example, some pattern recognition system applications include the following. In weather prediction, input data are in the form of weather maps, and the output is a forecast. The symptoms serve as input data in medical diagnosis while disease identity serves as the output. The predicted market ups and downs are the desired output for stock market prediction when input data are the financial news and charts.

Pattern recognition is the categorization of input data into identifiable classes through the extraction of significant attributes of the data from irrelevant background detail. A pattern class is a category determined by some common attributes. Therefore, a pattern is the description of a category member representing a pattern class. A pattern class is a family of patterns that shares some common properties. Pattern recognition by machine involves techniques for assigning patterns to their classes automatically with as little human interventions is possible. Pattern recognition aims to classify data (patterns) based on either a priori knowledge or on statistical information extracted from the patterns. The patterns to be classified are usually groups of measurements or observations, defining points in an appropriate multidimensional space [4].

A complete pattern recognition system consists of a sensor that gathers the observations to be classified or described; a feature extraction mechanism that computes numeric or symbolic information from the observations; and a classification or description scheme that does the actual job of classifying or describing observations, relying on the extracted features.

Pattern recognition problems may be logically divided into two broad categories. First involves the study of pattern recognition capabilities of human beings and other living organism, while the second involves the development of theory and techniques for the design of device capable of performing a given recognition task for a specific applications.

The first area deals with the subjects of psychology, physiology and biology, but the second area deals with the computer, information science and artificial intelligence. The pattern recognition is concerned primarily with description and analysis or classification of measurements taken from physical or mental process. The area of pattern recognition deals with the following:

i. Character Recognition: Optical or hand written character recognition,

ii. Visual Image Recognition,

iii. Voice Data Recognition,

iv. Speech Analysis,

v. Man and Machine Diagnostics,

vi. Person Identification and Industrial Inspection.

During the last few years the researchers have proposed many mathematical approaches to solve the pattern recognition problems. The available methods of pattern recognition may be categorized into three basic principles:

1. Statistical Methods; consisting the sub disciplines like discriminant analysis, feature extraction, error estimation, cluster analysis

2. Structural Methods consisting grammatical inference and parsing

3. Artificial Intelligence based Method

1.2 Image Transform and Matrix

Two dimensional unitary transforms play an important role in image processing. The term image transform refers to a class of unitary matrices used for representation of images.

In analogy with I-D signals that can be represented by an orthogonal series of basis functions , we can similarly represent an image in terms of a discrete set of basis arrays called "**basis images**". These are generated by unitary matrices.

Alternatively an $(N \times N)$ image can be represented as $(N^2 \times 1)$ vector. An image transform provides a set of coordinates or basis vectors for the vector space.

1.1.1. I-D-Transforms:

For a one dimensional sequence $\{\ u(n), \quad n = 0,1......N-1\ \}$ representing a vector $\vec{u}$ of size N, a unitary transform is : $\vec{v} = \underline{\boldsymbol{A}}\ \vec{u}$

$$\Rightarrow \qquad v(k) = \sum_{n=0}^{N-1} a(k,n)u(n), \quad \text{for}$$

$0 \le K \le N-1$ (2.1)

where $\underline{\boldsymbol{A}}^{-1} = \underline{\boldsymbol{A}}^{*\boldsymbol{T}\ *T}$ (unitary)

This implies , $\vec{u} = \underline{\boldsymbol{A}}^{*\boldsymbol{T}}\ \vec{v}$

or, $u(n) = \sum_{k=0}^{N-1} v(k)a^*\ (k,n)$, for

$0 \le n \le N-1$ (2.2)

Equation (2.2) can be viewed as a series representation of sequence *u(n)* . The columns of

$\underline{\boldsymbol{A}}^{*\boldsymbol{T}}$ i.e the vectors $\vec{a}_k^{\,*} \overset{\Delta}{=} \left\{ a^*(k,n), \quad 0 \le n \le N-1 \right\}^T$ are called the "basis vectors" of $\underline{\boldsymbol{A}}$.

The series coefficients *v(k)* give a representation of original sequence u(n) and are useful in compression , filtering , feature extraction and other analysis.

1.2.2Two dimensional Orthogonal and Unitary transforms:

As applied to image processing, a general orthogonal series expansion for an $N \times N$ image is a pair of transformations of the form :

$$v(k,l) = \sum\sum_{m,n=0}^{N-1} u(m,n)a_{k,l}(m,n) \quad , \quad 0 \le k,l \le N-1 \qquad (2.3)$$

$$v(k,l) = \sum\sum_{m,n=0}^{N-1} u(m,n)a_{k,l}(m,n) \quad , \quad 0 \le k,l \le N-1 \qquad (2.3)$$

$$u(m,n) = \sum\sum_{k,l=0}^{N-1} v(k,l)a^*_{k,l}(m,n) \quad , \quad 0 \le m,n \le N-1 \qquad (2.4)$$

where $\{a_{k,l}(m,n)\}$ is called an " image transform."

It is a set of complete orthogonal discrete basis functions satisfying the properties:-

1) Orthonormality: $\sum\sum_{m,n=0}^{N-1} a_{k,l}(m,n)a^*_{k',l'}(m,n)$

$= \delta(k-k', l-l')$

2) Completeness : $\sum\sum_{k,l=0}^{N-1} a_{k,l}(m,n)a^k_{k,l}(m',n')$

$= \delta(m-m', n-n')$

The elements $v(k,l)$ are transform coefficients and $\underline{\boldsymbol{V}} \triangleq \{v(k,l)\}$ is the transformed image.

The orthonomality property assures that any truncated series expansion of the form

$$U_{P,Q}(m,n) \triangleq \sum_{k=0}^{P-1}\ \sum_{l=0}^{Q-1} v(k,l)a^*_{k,l}(m,n) \quad , \quad \text{for} \quad P \le N, \qquad Q \le N$$

will minimize the sum of squares error

$$\sigma_e^2 = \sum\sum_{m,n=0}^{N-1} \left[u(m,n) - U_{P,Q}(m,n)\right]^2$$

where coefficients $v(k,l)$ are given by (3).

The completeness property assures that this error will be zero for $P = Q = N$

1.2.3. Separable Unitary Transforms:

The number of multiplications and additions required to compute transform coefficients $v(k,l)$ in equation(3) is $O(N^4)$. This is too large for practical size images. This is the form as expressed in the figure 2.1

If the transform is restricted to be separable,

i.e $a_{k,l}(m,n) = a_k(m)b_l(n)$

$\underline{\underline{\Delta}}\ a(k,m)b(l,n)$

where $\{a_k(m), k = 0(1)n-1\}$,

and $\{b_l(n), l = 0(1)N-1\}$ are 1D complete orthogonal sets of basis vectors.

On imposition of completeness and orthonormality properties we can show that $\underline{\boldsymbol{A}}\ \underline{\underline{\Delta}}\ \{a(k,m)\}$,

and $\underline{\boldsymbol{B}}\ \underline{\underline{\Delta}}\ \{b(l,n)\}$ are unitary matrices.

i.e

$$\underline{A}\underline{A}^{*T} = \underline{I} = \underline{A}^T \underline{A}^* \text{ and } \underline{B}\underline{B}^{*T} = \underline{I} = \underline{B}^T \underline{B}^*$$

Often one chooses $\underline{\boldsymbol{B}}$ same as $\underline{\boldsymbol{A}}$

$\therefore$ $v(k,l) = \sum\sum_{m,n=0}^{N-1} a(k,m)u(m,n)a(l,n)$

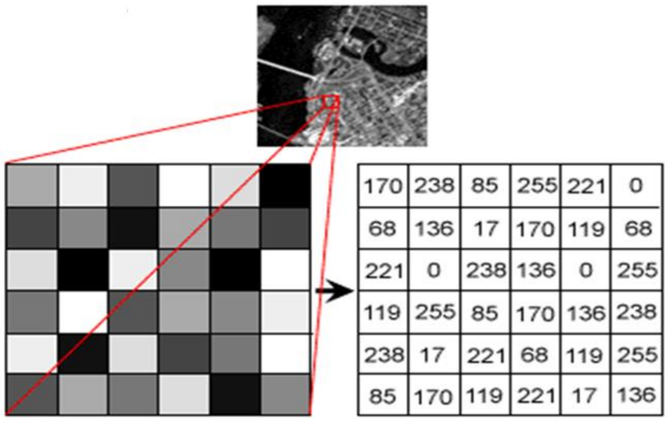
$$\leftrightarrow \underline{V} = \underline{A}\,\boldsymbol{U}\,\underline{A}^T \qquad (2.5)$$

And $u(m,n) = \sum\sum_{k,l=0}^{N-1} a^*(k,m)v(k,l)a^*(l,n)$

$$\leftrightarrow \underline{U} = A^{*T}\ \boldsymbol{V}\ A^* \qquad (2.6)$$

Eqn (2.5) can be written as $\underline{V}^T = \underline{A}(\underline{AU})^T$

$\Longrightarrow$ Eqn (2.5) can be performed by first transforming each column of $\underline{\boldsymbol{U}}$ and then transforming each row of the result to obtain rows of $\underline{\boldsymbol{V}}$.



170	238	85	255	221	0
68	136	17	170	119	68
221	0	238	136	0	255
119	255	85	170	136	238
238	17	221	68	119	255
85	170	119	221	17	136

Figure 2.1:- Showing the basic structure of matrix for any image.

1.2.3.1. Basis Images : Let $\vec{a}_k^*$ denote k^{th} column of $\underline{A}^{*T}$.

Let us define the matrices $\underline{A}_{k,l}^* = \vec{a}_k^* a_l^{*T}$ and matrix inner product of two $N \times N$ matrices $\underline{F}$ and $\underline{G}$ as

$$\langle \underline{F}, \underline{G} \rangle = \sum\sum_{m,n=0}^{N-1} f(m,n)g^*(m,n)$$

`.2.3.1. **Basis Images :** Let $\vec{a}_k^*$ denote k^{th} column of $\underline{A}^{*T}$.

Let us define the matrices $\underline{A}_{k,l}^* = \vec{a}_k^* a_l^{*T}$ and matrix inner product of two $N \times N$ matrices $\underline{F}$ and $\underline{G}$ as

$$\langle \underline{F}, \underline{G} \rangle = \sum\sum_{m,n=0}^{N-1} f(m,n)g^*(m,n)$$

Then equ (2.6) and (2.5) give a series representation.

$$\underline{\boldsymbol{U}} = \sum\sum_{k,l=0}^{N-1} v(k,l)\,\underline{A}_{k,l}^*$$

and $v(k,l) = \langle \underline{u}, \underline{A}_{k,l}^* \rangle$

Any image $\underline{U}$ can be expressed as linear combination of N^2 matrices. $\underline{A}_{k,l}^*$ called "basis images".

Therefore any $N \times N$ image can be expanded in a series using a complete set of N^2 basis images.

1.1.3.2. Dimensionality of Image transforms

The $2N^3$ computations for $\underline{V}$ can also be reduced by restricting the choice of $\underline{A}$ to <u>fast transforms</u>. This implies that $\underline{A}$ has a structure that allows factorization of the type,

$$\underline{A} = \underline{A}_{(1)}\underline{A}_{(2)}\cdots\cdots\underline{A}_{(p)}$$

where $\underline{A}_{(i)}$, $i = 1,\ p(p << N)$ are matrices with just a few non zero entries say r where $r << N$ Therefore a multiplication of the type : $\vec{y} = \underline{A}\vec{x}$ is accomplished in rpN operations. For several transforms like Fourier, Sine, Cosine, Hadamard etc, $p = log_2 N$, and operations reduce to the order of $N\,log_2 N$ or $N^2\,log_2 N$ (for $N \times N$ images). Depending on the transform, an operation is defined as 1 multiplication + 1 addition. Or, 1 addition or subtraction as in Hadamard Transform.

2.2.3.3. Kronecker products:

If $\underline{A}$ and $\underline{B}$ are $M_1 \times M_2$ and $N_1 \times N_2$ matrices we define Kronecker product as:

$$\underline{A} \otimes \underline{B} \underset{=}{\Delta} \left\{a(m,n)\underline{B}\right\}$$

Consider the transform, $\underline{V} = \underline{A}\ \underline{U}\ A^T$

or, $$v(k,l) = \sum \sum_{m,n=0}^{N-1} a(k,m)u(m,n)a(l,n)$$

(2.7)

If $\vec{v}_k$ and $\vec{u}_k$ denote k^{th} and m^{th} row vectors of $\underline{V}$ and $\underline{U}$ then (1) becomes ,

$$\vec{v}_k^T = \sum_m a(k,m)\left[\overrightarrow{AU_m^T}\right]$$

$= \sum_m [\underline{A} \otimes \underline{A}]_{k,m}\ \vec{u}_m^T$ where $[\ ]_{k,m}$ is the $(k,m)^{th}$ block of $[\underline{A} \otimes \underline{A}]$

If $\underline{U}$ and $\underline{V}$ are row ordered into vectors $\vec{v}$ and $\vec{u}$ respectively, then $\underline{V} = \underline{A}\underline{U}A^T \Longrightarrow \vec{v} = (\underline{A} \otimes \underline{A})\vec{u}$

The number of operations required for implementing equation(2.7) reduces from $O(N^4)$ to $O(2N^3)$.

1.3 Neural Network Based Methods

The pattern recognition approaches discussed so far are based on direct computation through machines. Direct computations are based on statistical analysis techniques. The neural approach applies biological concepts to machines for pattern recognition. The outcome of this effort is invention of artificial neural networks. Neural networks can be viewed as massively parallel computing systems consisting of an extremely large number of simple processors with many interconnections. Neural network models attempt to use some organizational principles (such as learning, generalization, adaptivity, fault tolerance, distributed representation, and computation) in a network of weighted directed graphs in which the nodes are artificial neurons and directed edges (with weights) are connections between neuron outputs and neuron inputs.

The main characteristics of neural networks are that they have the ability to learn complex nonlinear input-output relationships, use sequential training procedures, and adapt themselves to the data. The most commonly used family of neural networks for pattern classification tasks [3] is the feed-forward network, which includes multilayer perceptron and Radial-Basis Function (RBF) networks. These networks are organized into layers and have unidirectional connections between the layers. Another popular network is the Self-Organizing Map (SOM), or Kohonen-Network [4], which is mainly used for data clustering and feature mapping. The learning process involves updating network architecture and connection weights so that a network can efficiently perform a specific classification/clustering task. The increasing popularity of neural network models to solve pattern recognition problems has been primarily due to their seemingly low dependence on domain-specific knowledge (relative to model-based and rule-based approaches) and due to the availability of efficient learning algorithms. Neural networks provide a new suite of nonlinear algorithms for feature extraction (using hidden layers) and classification (e.g., multilayer perceptron's).

In addition, existing feature extraction and classification algorithms can also be mapped on neural network architectures for efficient (hardware) implementation. In spite of the seemingly different underlying principles, most of the well known neural network models are implicitly equivalent or similar to classical statistical pattern recognition methods. Ripley [5] and Anderson et al. [6] also discuss the relationship between neural networks and statistical pattern recognition. Most neural networks conceal the statistics from the user. Despite these similarities, neural networks do offer several advantages such as, unified approaches for feature extraction & classification and flexible procedures for finding good, moderately nonlinear solutions. It consists of massive simple processing units with a high degree of interconnection between each unit. The processing units work cooperatively with each other and achieve massive parallel distributed processing. The design and function of neural networks simulate some functionality of biological brains and neurons systems. The advantages of neural networks are their adaptive-learning, self-organization and fault-tolerance capabilities. For these outstanding capabilities, neural networks are used for pattern recognition applications. The goal in pattern recognition is to use a set of example solutions to some problem to infer an underlying regularity which can subsequently be used to solve new instances of the problem. Examples include hand-written digit recognition, medical image screening and fingerprint identification. In the case of feed-forward networks, the set of example solutions (called a training set), comprises sets of input values together with corresponding sets of desired output values. The training set is used to determine an error function in terms of the discrepancy between the predictions of the network, for given inputs, and the desired values of the outputs given by the training set. A common example of an error function would be the squared difference between desired and actual output, summed over all outputs and summed over all patterns in the training set. The learning process then involves adjusting the values of the parameters to minimize the value of the error function. This kind of feedback would be used to reconstruct the input patterns and make them free from error; thus increasing the performance of the neural networks. These kinds of networks are called as auto associative neural networks. As the name implies, they use back-propagation algorithms. However, effective learning algorithms were only known for the case of networks in which at most one of the layers comprised adaptive interconnections. Such networks were known variously as perceptrons and Adalines [7], and were seriously limited in their capabilities [8]. Research into artificial neural network was stimulated during the 1980s by the development of new algorithms capable of training networks with more than one layer of adaptive parameters [9].

1.4 Fuzzy Based Methods

Pattern recognition tasks ideally suit fuzzy theory, as patterns are very often inexact, ambiguous, or corrupted. In the Handwritten Characters Recognition case example, the patterns to be recognized are not well defined and are ambiguous in many cases. Fuzzy rules for handwritten character recognition are given

In Yamakawa [10]. Pattern recognition and classification are usually considered as very similar tasks. Classes can be described by fuzzy classification rules. Fuzzy rules can be used for classification purposes when the objects to be classified are noisy, corrupted, blurry, etc. Fuzzy rules can cope with those ambiguities. Contradictory fuzzy rules can be accommodated in one system, the tradeoff being achieved through the inference mechanism. This characteristic of fuzzy systems is very important for their applications in solving classification problems. Classification problems, when data examples labeled with class labels are available, have been successfully solved by classic statistical methods, especially when the data set is unambiguous and dense. If the data set is sparse in the problem state space, the problem is rather difficult. The confidence factors represent the percentage of instances of a given class which fall in a particular "patch" of the problem space:

Rule 1: IF A1 is M and A2 is S, THEN Class1 (CF = 1)

Rule 2: IF A1 is L and A2 is S, THEN Class1 (CF = 0.45)

Rule 3: IF A1 is L and A2 is S, THEN Class2 (CF = 0.55)

Rule 4: IF A1 is L and A2 is M, THEN Class2 (CF = 0.75)

Rule 5: IF A1 is L and A2 is M, THEN Class1 (CF = 0.25)

The above set of rules is ambiguous and contradictory. Rules 2 and 3 have the same antecedents, but different consequences, which is also true for rules 4 and 5. This situation is impossible for a symbolic production system to cope with. A fuzzy production system can infer a proper classification, as for every input data vector all rules may fire to some degree and all of them contribute to the final solution to different degrees. The points at the border between the two classes should be correctly classified by the fuzzy rules as they take support not only from one of the contradictory rules but from a rule that has lateral fuzzy labels with it.

when the data examples are not labeled with the class or pattern labels, that is, when the classes they belong to are not known, then different clustering techniques can be applied to find the groups, the clusters, in which data examples are grouped. The clusters show the typical patterns, the imilarity, and the ambiguity in the data set. In addition to the exact clustering, fuzzy clustering can be applied too.

Fuzzy clustering is a procedure of clustering data into possibly overlapping clusters, such that each of the data examples may belong to each of the clusters to a certain degree. The procedure aims at finding the cluster centers V_i (i = 1, 2, . . .,c) and the cluster membership functions μ_i which define to what degree each of the n examples belong to the ith cluster. The number of clusters c is either defined a priori (supervised type of clustering) or chosen by the clustering procedure (unsupervised type of clustering). The result of a clustering procedure can be represented as a fuzzy relation $\mu_{i,k}$, such that:

$$\sum_{i=1}^{c} \mu_{1,k} = 1$$

for each k = 1, 2, . . .,n; (the total membership of an instance k to all the clusters equals 1)

$$\sum_{i=1}^{c} \mu_{1,k} > 0$$

for each i = 1, 2, . . .,c (there are no empty clusters) .

A widely used algorithm for fuzzy clustering is the C-means algorithm suggested by J. Bezdek [11, 12]. Fuzzy clustering is an important data analysis technique. It helps to understand better the ambiguity in data. It can be used to direct the way other techniques for information processing are used afterward. For example, the structure of a neural network to be used for learning from a data set can be defined to a great extent after knowing the optimal number of fuzzy clusters.

1.5 Components of Pattern Recognition

Pattern recognition technique extracts a random pattern of human trait into a compact digital signature, which can serve as a biological identifier. The biometric systems use pattern recognition techniques to classify the users and identify them separately.

The components of pattern recognition are as follows –

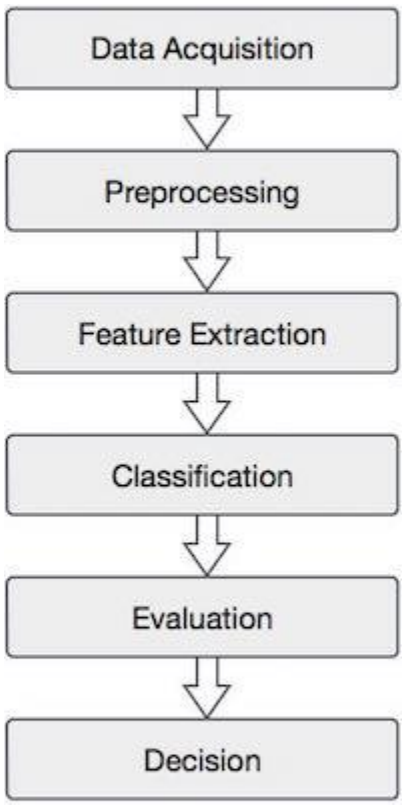


Fig 2.2 Components of Pattern Recognition

.6. The ACO Algorithm for Image Feature Selection

Given a feature set of size n, the feature selection problem is to find a minimal feature subset of size 5 *(s< n)* while maintaining a fairly high classification accuracy in representing the original features. Most of the ACO based algorithms for feature selecting uses a complete graph, on which the ants try to construct a path with part of the nodes. Since the a solution of feature selection is a subset of those selected features, there is not any ordering among the components of the solution. It is not necessary to use a path in a complete graph to represent such solution. In the ACO on such complete graph, ant on one node (feature) selects an edge connecting another node (feature) based on the pheromone and heuristic information assigned on this edge between the two nodes (features). But in the feature selection problem, one feature to be selected is independent with the last feature added to the partial solution. Therefore, it unnecessary to use a complete graph with $O(n^2)$ edges in the ACO algorithm.

To efficiently apply an ACO algorithm for feature selection, we must redefine the way that the representation graph is used. We proposed ant optimization algorithm on a discrete search space represented by a digraph with only *O(n)* arcs as shown in Figure 2.3, where the nodes represent features, and the arcs connecting two adjacent nodes indicating the choice of the next feature.

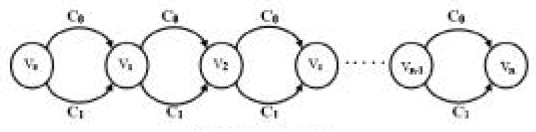


Fig.2.3 The Digraph

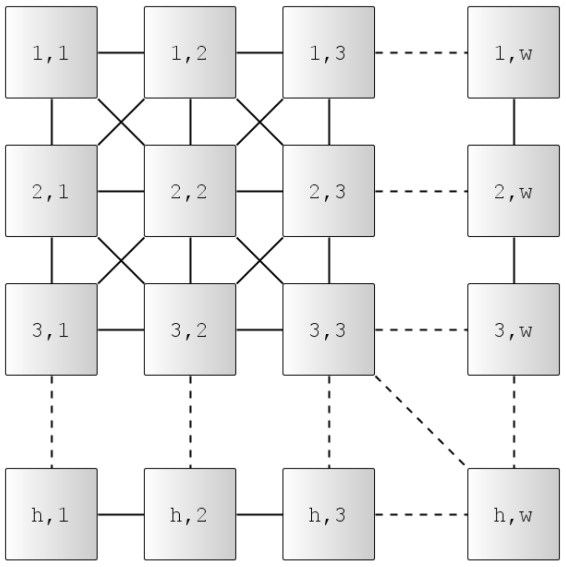
Denote the n features as $f1f_2$, ... f_n, the /'th node v, is used to represent feature f An additional node v_0 is placed at the beginning of the graph where each ant starts it search. As shown in Figure 2.3, the ants travel on the digraph from v_0 to v_b and then to v_2, and so on. The ant terminates its tour and outputs this feature subset as it reaches the last node v_n . When an ant completes the search from v_0 to v_n, the arcs on its trace form a solution.

1.6.1 Ant Colony Optimization (ACO)

Ant colony optimization (ACO) is an evolution simulation algorithm proposed by M. Dorigo et al [13]. Ants communicate with each other using pheromones. They leave pheromone trails on the ground in order to mark a path between their colony and a food source for other members in the colony to follow. The more ants following the same path, the higher its pheromone concentration becomes. Over time pheromone trails evaporate. The longer it takes for an ant to walk back and forth, the more time the pheromone has to evaporate. Hence, pheromone density remains higher at shorter and more favorable paths where pheromone is deposited at a much higher rate. This behavior helps ants successfully establish, and follow, the better paths. ACO is inspired by this foraging behavior. There exists several ant colony optimization algorithms, whereas the original algorithm, known as Ant System (AS), originates from the early nineties. Since then, a number of other algorithms, among the more successful variants MAX-MIN Ant System (MMAS) and Ant Colony System (ACS), were introduced. Several ACO-based approaches have been proposed to the edge detection problem. In this project, a new ACO-based approach is applied to image edge detection. The approach makes use of improvements introduced in ACS, with the addition of a new direction control feature. The reason for introducing direction control is to make the ants better suited for detecting edges belonging to long and narrow openings, hence, edges belonging to cracks.

1.6.2 ACO in Image Edge Detection

Image edge detection deals with extracting edges in an image by identifying pixels where the intensity variation is high. Basically edge detector are usually subjected for evaluation of observer [14][15]. There are many well-known edge detection algorithms. Potvin[16], laporte[17] and Canny[18], to
mention a few. Although originating from the early days of computer vision, some are still considered state-of-the-art edge detectors. Ant Colony Optimization introduces a different approach to image edge detection. In ACO,



artificial ants «walk on» the image depositing pheromone where the intensity variation is high. A how two dimensional image can be represented as a two-dimensional graph with the image pixels as its nodes (Figure 2.4). A pixel is connected to all adjacent pixels in an 8-connectivity neighborhood.

Fig.**2.4**: A graph representation of a how two-dimensional image.

ACO[1] is an iterative probabilistic algorithm where ants are guided, by pheromone information, towards optimal paths in a graph. At each iteration, a number of artificial ants are considered. Each ant in Algorithm, The proposed ACO-based approach to image edge detection.

Fig.**2.4**: A graph representation of a how two-dimensional image.

1. Initialization Phase

Initialize a gray scale intensity value matrix, a pheromone matrix, a normalized intensity variation matrix (the heuristics) and a set of eight polar angles.

2. Construction Phase

```
for construction step (round) n = 1 : N
    position all ants.
    for movement step l = 1 : L
        for ant k = 1 : K
            Select, and move ant to, next pixel.
            Immediate local update of the pixel's pheromone (pheromone decay).
        end
    end
    Offline update of all visited pixels pheromone (pheromone evaporation).
end
```

3**. Decision Phase**

The solution is made based on the values in the final pheromone matrix. incrementally builds a (complete) solution by moving from pixel to pixel in the graph. An ant can only move to adjacent pixels with the constraint of not visiting any pixel more than once within the same iteration. The movement of the ants is affected by local variations in pixel intensity values. Ants select the following pixel to visit through a stochastic mechanism influenced by pheromone and heuristic information. The heuristic information is only dependent on the specific problem. Additionally, ants deposit a certain amount of pheromone on the traversed pixels. The actual amount deposited depends on the quality of the pixel. Subsequent ants make use of this pheromone information as a guide steering them towards the more promising regions in the graph. The goal is to construct a final pheromone matrix that reflects the edge information in the image. Each element in the pheromone matrix corresponds to a pixel in the image and indicates whether the pixel is on an edge or not. The proposed ACO-based approach to image edge detection can be thought of as a three-phased process as described in Algorithm 1: The first phase is an initialization phase. The second, which is the construction phase, covers all ant movement. During this phase the pheromone information is continuously updated as the ants «walk on» the image. In the last phase, the decision phase, the solution is made from the values of the elements in the final pheromone matrix.

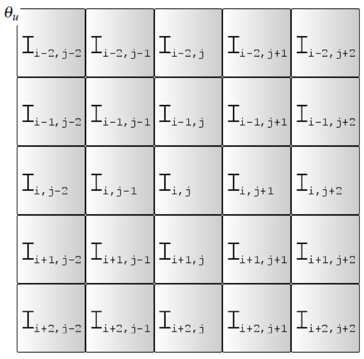
1.6.3 Initialization Phase

First, an intensity value matrix is made. Every element in the intensity value matrix has a value based on the gray scale intensity level related to the corresponding image pixel. Secondly, a pheromone matrix is constructed. Every element in the pheromone matrix is assigned a small nonzero value t_{init} , stating an initial pheromone level. The heuristic information, one of the main aspects in the following construction phase, may be calculated already in the initialization phase. This is due to the fact that the heuristic only depends on the intensity values of the pixels in the image. In other words, the heuristic information is a (normalized) intensity variation matrix which is fixed for every construction step:

$$\eta_{i,j} = \frac{V_c(I_{i,j})}{V_{max}} \qquad (2.10)$$

• Ii; j is the intensity value of the pixel (i; j).
• Vc(Ii; j) reflects the intensity variation between the pixel (i; j) and a local group of surrounding
pixels c, called a clique (Figure 2.5).

• Vmax represents the maximum intensity variation in the whole image and serves as a normalization factor. The value of hi; j is large for pixels located in regions of the image containing sharp intensity variations. ence, pixels representing edges in the image. Figure



$I_{i-2,j-2}$	$I_{i-2,j-1}$	$I_{i-2,j}$	$I_{i-2,j+1}$	$I_{i-2,j+2}$
$I_{i-1,j-2}$	$I_{i-1,j-1}$	$I_{i-1,j}$	$I_{i-1,j+1}$	$I_{i-1,j+2}$
$I_{i,j-2}$	$I_{i,j-1}$	$I_{i,j}$	$I_{i,j+1}$	$I_{i,j+2}$
$I_{i+1,j-2}$	$I_{i+1,j-1}$	$I_{i+1,j}$	$I_{i+1,j+1}$	$I_{i+1,j+2}$
$I_{i+2,j-2}$	$I_{i+2,j-1}$	$I_{i+2,j}$	$I_{i+2,j+1}$	$I_{i+2,j+2}$

Polar Angle = Value (in radians)		
$\theta_6 = \frac{5\pi}{4}$	$\theta_7 = \frac{3\pi}{2}$	$\theta_8 = \frac{7\pi}{4}$
$\theta_5 = \pi$	$(i\,,\,j)$	$\theta_1 = 0$
$\theta_4 = \frac{3\pi}{4}$	$\theta_3 = \frac{\pi}{2}$	$\theta_2 = \frac{\pi}{4}$

Figure 2.5: The clique at pixel (i,j).

Calculating the intensity variation at pixel (i; j) is done as follows:

$$V_c(I_{i,j}) = \begin{array}{l} |I_{i-2,j-2} - I_{i+2,j+2}| + |I_{i+2,j-2} - I_{i-2,j+2}| + |I_{i-1,j-1} - I_{i+1,j+1}| + |I_{i+1,j-1} - I_{i-1,j+1}| + \\ |I_{i-2,j-1} - I_{i+2,j+1}| + |I_{i+2,j-1} - I_{i-2,j+1}| + |I_{i-1,j-2} - I_{i+1,j+2}| + |I_{i+1,j-2} - I_{i-1,j+2}| + \\ |I_{i-2,j} - I_{i+2,j}| + |I_{i,j-2} - I_{i,j+2}| + |I_{i-1,j} - I_{i+1,j}| + |I_{i,j-1} - I_{i,j+1}| \end{array}$$

2.11

Large differences in intensity values between pixels located 180 degrees of each other, relative to pixel (i; j), gives a large variation in intensity. In other words, pixels located at edges have large intensity variations, hence, large values for Vc(Ii; j). Note however, that the borders of an image (the two outermost pixels) does not satisfy a complete clique. For instance, the pixel located at the top left corner (1;1) does not have any west- nor north-laying neighbor pixels. Hence, no (normalized) intensity variation is calculated on the image borders. Instead are all border pixels initialized with a very small nonzero value. A small value is chosen for the simple
reason of making the image borders less attractive to ants. A set of eight polar angles is defined (in radians). Each angle qu is relative to the movement, in a horizontal right direction, of an ant located at pixel (i; j) and its adjacent neighboring pixels. Hence, there are only eight possible values for qu (Table 2.1).

1.6.4 Construction Phase

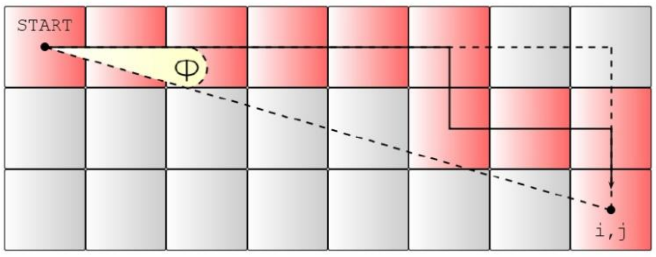
Ant Movement

Ants execute a predefined number of rounds (construction steps). In every round n each ant k is consecutively moved a fixed number of movement steps l. An ant moves from its current pixel (i; j) to one of its, not previously visited, neighboring pixels (x;y). Which neighbor pixel to go to is determined by a transition probability matrix:

$$p^{(n)}_{(i,j),(x,y)} = \frac{\left[\left(\tau^{(n-1)}_{x,y}\right)^{\alpha} (\eta_{x,y})^{\beta}\right] + [r(\cos(\theta_u - \varphi_{i,j}) + 1)]}{\Sigma_{(x,y)\in\Omega_{(i,j)}} \left[\left(\tau^{(n-1)}_{x,y}\right)^{\alpha} (\eta_{x,y})^{\beta}\right] + [r(\cos(\theta_u - \varphi_{i,j}) + 1)]}$$

2.12

- $\tau^{(n-1)}_{x,y}$ is the pheromone information at the neighbor pixel (x,y).
- α determines the influence of the pheromone information.
- $\eta_{x,y}$ is the (normalized) intensity variation at the neighbor pixel (x,y).
- β determines the influence of the intensity variation.
- r determines the influence of the direction control.
- θ_u is the polar angle given the ants current pixel (i,j) and its neighbor pixel (x,y).
- $\varphi_{i,j}$ is the angle given the ants current pixel (i,j) and its starting pixel (i_{start}, j_{start}) (Figure 4).
- $\Omega_{(i,j)}$ represents the set of neighborhood pixels for the ant currently located at pixel (i,j).
- The denominator represents a normalization of the transition probability $p_{(i,j)}$.



The angle $\varphi_{i,j}$ given the ants current pixel (i,j) and its starting pixel (i_{start}, j_{start}).

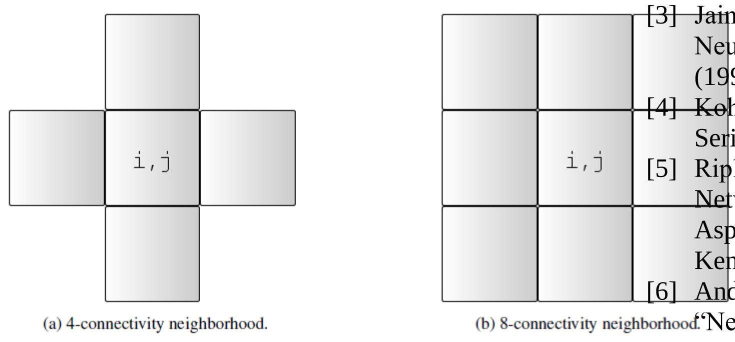
Figure 2.6

Calculation of the angle $\varphi_{i,j}$ is done as follows:

$$\varphi_{i,j} = \begin{cases} \arctan\left(\frac{i-i_{start}}{j-j_{start}}\right) & if\ j > j_{start} \\ \arctan\left(\frac{i-i_{start}}{j-j_{start}}\right) - \Pi & if\ j < j_{start}\ and\ i < i_{start} \\ \arctan\left(\frac{i-i_{start}}{j-j_{start}}\right) + \Pi & if\ j < j_{start}\ and\ i \geq i_{start} \\ -\frac{\Pi}{2} & if\ j = j_{start}\ and\ i < i_{start} \\ \frac{\Pi}{2} & if\ j = j_{start}\ and\ i > i_{start} \end{cases}$$

2.13

The value of $j_{i;j}$ solely depends on the ants current pixel location (i; j) with respect to its starting pixel (i_{start} ; j_{start}). It is not (directly) affected by any of the intermediate movement steps. Note however, that j is neither a part of the original ACO, but introduced as a part of the new direction control feature. During ant movement, there are two issues which are considered crucial. The first is related to the already discussed (normalized) intensity variation (1), more specifically Ƞx;y in (3). The second is about defining the permissible range of their movement, namely 'Ω(i; j) in (3). Ants may only move to adjacent (neighboring) pixels (Figure 2.7). In other words, it cannot move to a pixel which is not directly connected with the pixel where it is currently located. Various neighborhoods have been proposed in the literature, whereas the one adopted in this report is the 8-connectivity version (Figure 2.7b).


i,j
i,j
(a) 4-connectivity neighborhood.
(b) 8-connectivity neighborhood.

Adjacent pixels using various neighborhoods.

Fig.2.7

This method of feature detection is not ideal. The output of this unamplified filter at a corner is the autocorrelation of the feature rather than the ideal 2-D discrete Dirac delta function. If multiple copies of the pattern with different contrasts are present in the input, it will be difficult or impossible to segment the desired features by thresholding the convolution alone.

CONCLUSION

We have studied various 2 dimensional object and pattern recognition techniques in various format. We have analyzed the various result in terms of matrix format, and tried to get out the set and function relation out of it. This relation are then pushed into some of the soft computing techniques like ACO, ANN, GA and fuzzy logic, and the study on its different parameters. In this chapter we are also discussing on conventional statistical inference to modern approaches of soft computing integrated non-linear dynamics. Probabilistic approach integrated with neural network gives the better understanding of pattern and object detection. We have derived the image inversion form for the direct input to our soft computing techniques for further processing.

REFERENCES

[1] Bishop, Christopher M. (2006). Pattern Recognition and Machine Learning (PDF). Springer. p. vii. Pattern recognition has its origins in engineering, whereas machine learning grew out of computer science. However, these activities can be viewed as two facets of the same field, and together they have undergone substantial development over the past ten years.

[2] S. Watanabe, "Pattern Recognition: Human and Mechanical", New York: Wiley, 1985

[3] Jain, A.K., Mao, J. and Mohiuddin, K.M., "Artificial Neural Networks: A Tutorial.", Computer, pp. 31-44, (1996).

[4] Kohonen, T., "Self-Organizing Maps." , Springer Series in Information Sciences, vol. 30, (1995).

[5] Ripley, B., "Statistical Aspects of Neural Networks.", Networks on Chaos: Statistical and Probabilistic Aspects. U. Bornndorff-Nielsen, J. Jensen, and W. Kendal, eds., Chapman and Hall, (1993).

[6] Anderson, J., Pellionisz, A. and Rosenfeld, E., "Neurocomputing 2: Directions for Research.", Cambridge Mass.: MIT Press, (1990).

[7] Widrow,B. and Lehr,M.A.,"30 years of adaptive neural networks: perceptron, madeline, and backpropagation.", Proceedings of the IEEE 78 (9), pp. 1415-1442. (1990).

[8] Minsky, M. L. and Papert, S.A., "Perceptrons.", Cambridge, MA: MIT Press. Expanded Edition,(1990).

[9] Rumelhart, D.E., Hinton, G.E. and Williams,R.J., "Learning internal representations by error propagation.",Parallel Distributed Processing: Explorations in the Microstructure of Cognition Cambridge, MA: MIT Press. Reprinted in Anderson and Rosenfeld, vol. 1, pp. 318-362. (1988).

[10] Yamakawa, T.,"Pattern recognition hardware system employing a fuzzy neuron.", Proceedings of the International Conference on Fuzzy Logic and Neural Networks, Iizuka, Japan, pp 943-948,(1990).

[11] Bezdek, J., "Analysis of Fuzzy Information.", Boca Raton, Fla, CRC Press,(1987).

[12] Bezdek, J. and Pal, S., "Fuzzy Models for Pattern Recognition.", New York, IEEE Press,(1992).

[13] Impedovo, S., "Fundamentals in Handwriting Recognition." NATO-Advanced Study Institute, vol. 124, Springer-Verlag (1994)

[14] Mori, S., Suen, C.Y. and Yamamoto, K., "Historical review of OCR research and development," Proceeding of the IEEE, vol.80 (7), pp. 1029-1058 (1992).

[15] Dorigo.M,Birattari.M,Stützle.T,Ant Colony Optimization:Artificial Ants as a Computational Intelligence Technique,IEEE Computational Intelligence Magazine,(11);28-29.(2006)

[16] M. Fisher. Vehicle routing. Handbooks of Operations Research and Management Science, Chapter 1, 8:1-31,1995

[17] Sergios Theodoridis, Konstantinos Kourtoumbas. Pattern Recognition Second Edition page 582

[18] M.Gendreau, G. Laprte, and J-Y. Potvin. Metaheuristics for the vehicle routing problem. Management Science, 40:1276-1290,1994.