Comparison of Advanced Techniques of Image Classification

M.Seetha, * I.V.MuraliKrishna, Member, IEEE ** B.L. Deekshatulu, Life Fellow, IEEE

Associate Professor, CSE Dept., CBIT, HYDERABAD-500075, E-mail: smaddala2000@yahoo.com

*Director, R & D Division, JNTU, HYDERABAD-500072, E-mail: ivm@ieee.org

** Visiting Professor, HCU, HYDERABAD

Abstract: The conventional statistical approaches for image classification use only the gray values. Different advanced techniques in image classification like Artificial Neural Networks (ANN), Support Vector Machines (SVM), Fuzzy measures and Genetic Algorithms (GA) are developed for image classification. The use of textural features in ANN helps to resolve misclassification. SVM was found competitive with the best available machine learning algorithms in classifying high-dimensional data sets. Fuzzy measures show the detection of textures by analyzing the image by stochastic properties. The genetic algorithm searches a space of image processing operations that produce suitable feature planes, and a conventional classifier uses those feature planes to output a final classification. A comparative study of some of these techniques for image classification is made to identify relative merits.

Keywords: Image classification, neural networks, support vector machines, fuzzy measures, genetic algorithms.

I.INTRODUCTION

Digital image processing is a collection of techniques for the manipulation of digital images by computers. Classification generally comprises four steps: l. Pre-processing. e.g. atmospheric correction, noise suppression, and finding the band ratio, principal component analysis, etc, 2. Training: Selection of the particular feature which best describes the pattern, 3. Decision: Choice of suitable method for comparing the image patterns with the target patterns and 4: Assessing the accuracy of the classification.

II.TECHNIQUES OF IMAGE CLASSIFICATION

Image classification is an important task for many aspects of global change studies and environmental applications. This paper emphasizes on the analysis and usage of different advanced classification techniques like Artificial Neural Networks, Support Vector Machines, Fuzzy measures and Genetic algorithms for digital image classification. Finally the study depicts the comparative analysis of different classification techniques with respect to several parameters.

A. Artificial Neural Network (ANN)

ANN is a parallel distributed processor [1] that has a natural tendency for storing experiential knowledge. Image classification using neural networks is done by texture feature extraction and then applying the back propagation algorithm.

1)Architecture of Neural Network and Texture Feature Extraction Algorithm:

In the design four textural features namely the angular second moment, contrast, correlation and variance are considered. In order to capture the spatial dependence of gray-level values, which contribute to the perception of texture, a two dimensional dependence, and texture analysis matrix is considered. Figure 1 shows the architecture of NN with combined gray value and textural features. In this, four layers consisting of three inputs, seven first layer hidden nodes, eleven second layer hidden nodes and five output nodes are considered.

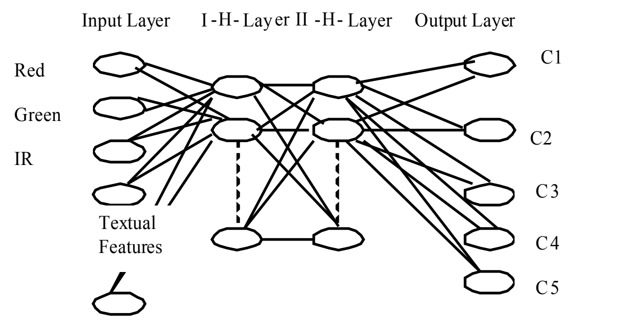

Figure:1 : Architecture of NN with combined gray value and textural features

The gray-tone co occurrence matrix is used for extracting textural features. It is a two dimensional matrix of joint probabilities Pd, r (i, j) between pairs of pixels, separated by a distance, d in a given direction r [2].

2)Training the Combined Network using BKP

The following assumes the sigmoid function f (x)

$$f(x) = \frac{1}{1 + e^{-x}} \qquad \text{--- (1)}$$

The back propagation algorithm is implemented using following steps:

1. Initialize weights to small random values.
2. Feed input vectors X0, X1………., X6 through the network and compute the weighting sum coming into the unit and then apply the sigmoid function. Also, set all desired outputs d0, d1..., d5 typically to zero except for that corresponding to the class the input is from.
3. Calculate error term for each output unit as

$$\delta_j = y_j(1 - y_j)(d_j - y_j) \qquad \text{--- (2)}$$

where d*j* is the desired output of node j; and y_j is the actual output.

4. Calculate the error term of each of the hidden units as

$$\delta_j = x_j(1 - x_j)\sum_k \delta_k w_{jk} \qquad \text{--- (3)}$$

Where k is over all nodes in the layers above node j; and j is an internal hidden node.

5. Add the weight deltas to each of

$$W_y(t+1) = W_y(t) + \eta\, \delta_j x_i \qquad \text{--- (4)}$$

All the steps excepting 1 are repeated till the error is within reasonable limits and then the adjusted weights are stored for reference to the recognition algorithm.

B. Support Vector Machines

The support vector machine (SVM) is superior of all machine learning algorithms. SVM employs optimization algorithms to locate the optimal boundaries between classes. The applicability of SVM for image classification is explored in this study.

1) Development of SVM

The support vector machine (SVM) is a machine learning algorithm based on statistical learning theory. There are a number of publications detailing the mathematical formulation and algorithm development of the SVM [8, 9, and 10]. The inductive principle behind SVM is structural risk minimization (SRM). The risk of a learning machine (*R*) is bounded by the sum of the empirical risk estimated from training samples (*R*emp) and a confidence interval (ψ): $R \leq R\text{emp} + \psi$ [8]. The strategy of SRM is to keep the empirical risk (*R*emp) fixed and to minimize the confidence interval (ψ), or to maximize the margin between a separating hyper plane and closest data points (Figure 2). A separating hyperplane refers to a plane in a multidimensional space that separates the data samples of two classes. The optimal separating hyperplane is the separating hyperplane that maximizes the margin from closest data points to the plane. Currently, one SVM classifier is able to separate only two classes. Integration strategies are needed to extend this method to classifying multiple classes.

2) The optimal separating hyperplane

Let the training data of two separable classes with k samples be represented by(x_1, y_1), (x_k, y_k) where $x \in R^n$ is an n-dimensional space, and $y \in \{+1, 1\}$ is class label. Suppose the two classes can be separated by two hyperplanes parallel to the optimal hyperplane (Figure 2(*a*)):

$$w.x_i + b \geq 1 \qquad \text{--- (5)}$$

$$w.x_i + b \leq -1 \qquad \text{--- (6)}$$

where **w**= $(w_1 \ldots w_n)$ is a vector of n elements. Inequalities (5) and (6) can be combined into a single inequality:

$$y_i[w'x_i + b] \geq 1 \qquad \text{--- (7)}$$

As shown in Figure 2, the optimal separating hyperplane is the one that separates the data with maximum margin. This hyperplane can be found by minimizing the norm of w, or the following function:

$$F(w) = \tfrac{1}{2}(w'w) \qquad \text{--- (8)}$$

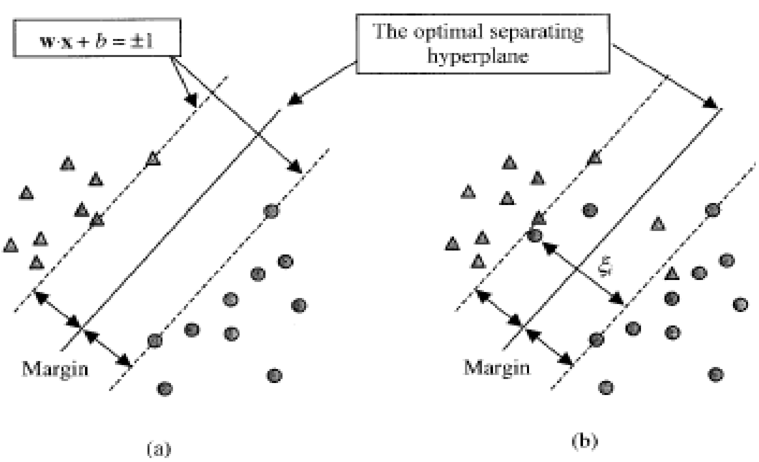

Figure 2: The optimal separating hyperplane between (*a*) separable samples and (*b*) non-separable data samples.

The saddle points of the following Lagrange gives solutions to the above optimization problem:

$$L(w,b,\alpha) = \tfrac{1}{2}(w'w) - \sum_{i=1}^{k} \alpha_i y_i[w'x_i + b] - 1\} \qquad \text{--- (9)}$$

where $\alpha_i \geq 0$ are Lagrange multipliers [11]. The solution to this optimization problem requires that the gradient of L (w, b, α) with respect to w and b vanishes, giving the following conditions:

$$w = \sum_{i=1}^{k} y_i \alpha_i x_i \qquad \text{--- (10)}$$

$$\sum_{i=1}^{k} \alpha_i y_i = 0 \qquad \text{--- (11)}$$

By substituting (10) and (11) into (9), the optimization problem becomes: maximize

$$L(\alpha) = \sum_{i=1}^{k} \alpha_i - \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k} \alpha_i \alpha_j y_i y_j (x_i ' x_j) \quad \text{--- (12)}$$

under constraints, $\alpha_i \geq 0$ i=1,.. , k.

Given an optimal solution $\alpha^0 = (\alpha^0_{1},....,\alpha^0_k)$ to (12), the solution w^0 to (9) is a linear combination of training samples:

$$w^0 = \sum_{i=1}^{k} y_i \alpha_i^0 x_i \quad \text{--- (13)}$$

According to the Kuhn–Tucker theory [11] only points that satisfy the equalities in (5) and (6) can have non-zero coefficients α_i^0 .These points lie on the two parallel hyperplanes and are called support vectors (Figure 2). Let $\mathbf{x}^0(1)$ be a support vector of one class and $\mathbf{x}^0(\ 1)$ of the other, then the constant b^0 is calculated as follows:

$$b^0 = \tfrac{1}{2}[w^0 ' x^0(1) + w^0 ' x^0(-1)] \quad \text{--- (14)}$$

The decision rule that separates the two classes can be written as:

$$f(x) = sign\left(\sum_{\sup portvector} y_i \alpha_i^0 (x_i ' x) - b^0\right) \quad \text{--- (15)}$$

3) Implementation of Support vector machines

To generalize the above method to non-linear decision functions, the support vector machine implements the following method: it maps the input vector x into a high-dimensional feature space H and constructs the optimal separating hyperplane in that space. Suppose the data are mapped into a high-dimensional space H through mapping function Φ:

$$\Phi : R^n \to H \quad \text{--- (16)}$$

A vector x in the feature space can be represented as $\Phi(\mathbf{x})$ in the high-dimensional space H. Since the only way in which the data appear in the training problem (8) are in the form of dot products of two vectors, the training algorithm in the high dimensional space H would only depend on data in this space through a dot product, i.e. on functions of the form $\Phi(\mathbf{x}_i)' \Phi(\mathbf{x}_j)$. Now, if there is a kernel function K such that

$$K(x_i, x_j) = \Phi(x_i)' \Phi(x_j) \quad \text{--- (17)}$$

then K is used in the training program without knowing the explicit form of Φ. Thus if a kernel function K can be found, a classifier can be used in the high-dimensional space without knowing the explicit form of the mapping function for training.. The optimization problem (12) is rewritten as:

$$L(\alpha) = \sum_{i=1}^{k} \alpha_i - \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad \text{--- (18)}$$

and the decision rule expressed in equation (11) becomes:

$$f(x) = sign\left(\sum_{\sup port\ vector} y_i \alpha_i^0 K(x_i, x) - b^0\right) \quad \text{--- (19)}$$

By the systematic development of SVM, the kernel function plays a major role in locating complex decision boundaries between classes. By having input data into dimensional space, the kernel function converts non-linear boundaries in the original data space into linear ones in the high dimensional space. Hence the selection of kernel function and appropriate values for corresponding kernel parameters, affect the performance of the SVM.

C. Fuzzy Measures

There are several image processing algorithms for the detection of information in the form of sound defined features such as edges, skeletons, and contours etc as in [13, 14, and 15]. In Fuzzy measures, different stochastic relationships are identified to describe properties of an image. If the fuzzy property is more related to a region, then a fuzzy measure is used. Fuzzy function is used if a stochastic property is to be described by a particular distribution of gray values. The fusion of these two stochastic properties is

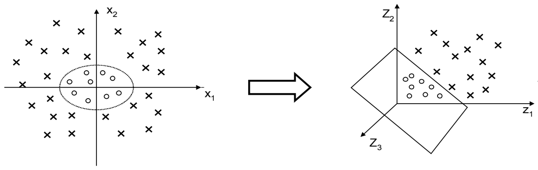

Fig. 3 mapping nonlinear data to a higher dimensional feature space

represented as a fuzzy measure and fuzzy function defines on an area which is achieved by a fuzzy integral. The result of fuzzy integral is a new fuzzy measure.

3) Stochastic Information by Fuzzy Measures

The mathematical basis for including the importance of a stochastic property is the fuzzy measure. By the fuzzy measure the properties described by different kinds of relationships are mapped into the closed interval [0, 1]. The fuzzy measure, as defined in [19], has a term with the combination of all elementary fuzzy measures multiplied by a factor λ. The factor λ has an effect similar to a weight factor for the interaction between the properties. If λ= 0 then g can be used as a probability measure.

The coupling of the elementary fuzzy measures (densities) $g_1(x_1)$ over the elementary region x_1 with another elementary fuzzy measure $g(x_2)$ over the other elementary region x_2 is defined by:

$$g_\lambda(x_1 \cup x_2) = g_\lambda(x_1) + g_\lambda(x_2) + \lambda g_\lambda(x_1) g_\lambda(x_2) \qquad \text{--- (20)}$$

where $\lambda = (1+\lambda g(x_1))\,(1+\lambda\, g(x_2) + \lambda\, g_\lambda(x_1)\, g_\lambda(x_2)) - 1$ is a coupling constant used as a substitution for the loss of additivity. For a set of elements A= {x_i} the relationship above is used recursively to give:

$$g(A) = \sum_{i=1}^{n} g(x_i) + \lambda \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} g(x_i)g(x_j) + .. + \lambda^{n-1} g(x_1)..g(x_n) \qquad \text{--- (21)}$$

This is written as product

$$g(X) = \frac{1}{\lambda}\left[\prod_{x_j \in X} (1+\lambda g(x_i) - 1 \right] \quad \text{where} \quad \lambda \neq 0 \qquad \text{--- (22)}$$

The coupling parameter λ is obtained by solving the equation

$$1+\lambda = \left[\prod_{x_j \in X} (1+\lambda g(x_i)) \right] \qquad \text{--- (23)}$$

The mathematical concept for calculating the measure for the coupled elementary properties for small areas is shown here.

4) Stochastic Information by Fuzzy Functions

The fuzzy functions are mostly values over single pixel points. The values of the neighbor pixel are of stochastic nature and normally not directly correlated with this value. These fuzzy functions are described normally by a characterization over a threshold. Outside of such a characteristic threshold the values depend very weakly on the real value. Inside the interval the values generate fuzzy properties for the adapted condition. These fuzzy functions are also normalized and mapped on an interval given by a boundary.

5) Fusion of Fuzzy Properties by Fuzzy Integrals

The values over the possible region of textures represented by a fuzzy measure $g(x_{k,l})$ are connected with the values $h(x_{i,j})$ at the pixels representing the strength of the properties for a texture. The value $h(x_{i,j})$ is described by a fuzzy function where the values are normalized to 1. The functional relationship between the fuzzy measure and the fuzzy function is represented by the fuzzy integral [19], because it is well adapted to the problem of detection of textures. The fuzzy measure [19] is combined with the fuzzy function in the form

$$f_A h_\alpha(x) \oplus dg = \sup_{\alpha \in [0,1]} \{\min[\alpha, g(A \cap H_\alpha)]\},$$

$$H_\alpha = \{x | h_\alpha \geq \alpha\} \qquad \text{--- (24)}$$

Here h_α is the cut of h at the constant α. For h_α (x), the values at the pixel-points are used, representing the property of a texture. α is the threshold where the assumption is fulfilled, that the property is used in the minimal condition. The region A is given as the image region where surely a specific texture is expected. It may be also the whole image for the pixel region and the whole possible range for a fuzzy function. The important property of a fuzzy measure is that its value is mapped on the closed interval [0, 1]. This is given by the calculated value of the fuzzy integral. This gives the possibility to use the result of a fuzzy integral as a new fuzzy measure g_2

$$g_{2_{i,j}}(A) = \int_A h_\alpha(x) \oplus dg_1 \qquad \text{--- (25)}$$

This newly produced fuzzy measure is linked with the region obtained by another stochastic property for the texture so that f_2 obtained

$$f_2 = \int_A h'_{\alpha'}(x) \oplus dg_2 \qquad \text{--- (26)}$$

In such away a set of fuzzy functions $\{f1, f_2, fn\}$ is obtained by fuzzy measures $\{g_1, g2, g3\}$. The summation of all combinations of fuzzy measures with fuzzy functions makes sure, that all possible properties in all combinations, which should be considered, are used. In order to achieve this different approaches have to be applied for the elimination of the elementary stochastic properties within an image.

D. Genetic Algorithms

The techniques of image classification ranging from maximum likelihood to neural networks depend on feature vectors formed by the intensity values in each spectral channel for each pixel. But the spectral information alone is not sufficient to exactly identify a pixel. The features of its neighborhood, like texture, or the average value of nearby pixels are necessary to get good spectral information. Hence to choose these features automatically a new evolutionary hybrid genetic algorithm is used.

6)The Genetic Programming System

The genetic programming system based on a linear chromosome [22] manipulates image processing programs that take the raw pixel data planes and transform them into a set of feature planes. This set of feature planes have a multi-spectral image derived from the original image through a certain sequence of image processing operations. The system then applies a conventional supervised classification algorithm to the feature planes to produce a final output image plane. The pixel in the output image plane specifies whether that feature is there or not. Figure 3 illustrates this hybrid scheme. In this structure finally raw data planes are transformed into a set of feature planes by an image processing program that is evolved by genetic algorithm.

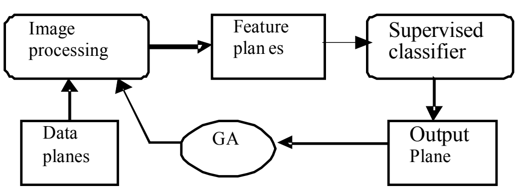

Fig ure 3: The Structure of Genetic programming System

In the system a fixed-length linear chromosome and standard one-point crossover are used. According to [23], these two are significantly better than more flexible crossover schemes. A single chromosome is made up of a string of genes, each one of which corresponds to a particular image processing operation. Each gene has one or more inputs, and one or more outputs. An input can be taken from any one data plane in the original image (there are as many data planes as there are spectral channels), or from any one ‘scratch plane’. Scratch planes are temporary holding places where a single image plane can be held. The gene performs some image processing operation on its inputs and produces one or more planes of output data. Each of these planes is written to a different scratch plane. The whole chromosome is evaluated by starting with the gene at the left end, and sequentially stepping through the genes in order, one-by-one. It is a requirement when chromosomes are created that no gene is allowed to read from a scratch plane that has not been written to by at least one gene to the left of it. The feature planes which are passed on to the backend classifier are specified by the user as a subset of the scratch and data plane.

Each gene corresponds to a different image processing operation, but the details of that operation can be influenced by gene parameters In general, genes produce output that is roughly on the order of the same scale as their input. Thus by using genetic algorithm a robust classifier is developed.

III. COMPARATIVE ANALYSIS OF VARIOUS MACHINE LEARNING ALGORITHMS

The advanced image classification techniques using Artificial Neural Networks, Support Vector Machines, Fuzzy measures and Genetic Algorithms are analyzed and compared with respect to several parameters. The benefits and limitations of these classification techniques are as in Table II. Artificial neural networks have the advantages mainly of more tolerance to noise inputs and representation of boolean function apart from others. But too many attributes may result in over fitting. In support vector machines over fitting is unlikely to occur. The computational complexity and the complexity of decision rule are reduced in SVM. Fuzzy measures have the benefit of identification of various stochastic relationships to describe the properties of the image. But priori knowledge is very important to get good results. Genetic algorithms are primarily used in optimization and always have a good solution. But the computation of scoring function is non trivial.

TABLE II
BENEFITS AND LIMITATIONS OF VARIOUS MACHINE LEARNING ALGORITHMS

Machine Learning Algorithm	Benefits	Assumptions and / or Limitations
Neural Network	•can be used for classification or regression •able to represent Boolean functions (AND, OR, NOT) •tolerant of noisy inputs •instances can be classified by more than one output	•difficult to understand structure of algorithm •too many attributes can result in over fitting •optimal network structure can only be determined by experimentation
Support Vector Machine	•models nonlinear class boundaries •over fitting is unlikely to occur •computational complexity reduced to quadratic optimization problem •easy to control complexity of decision rule and frequency of error	•training is slow compared to Bayes and Decision trees •difficult to determine optimal parameters when training data is not linearly separable •difficult to understand structure of algorithm
Fuzzy logic	• different stochastic relationships can be identified to describe properties	•Priori knowledge is very important to get good results •precise solutions are not obtained if the direction of decision is not clear
Genetic Algorithm	•can be used in feature classification and feature selection •primarily used in optimization always finds a “good” solution (not always the best solution) • can handle large, complex, non differentiable and multimodal spaces • Efficient search method for a complex problem space • good at refining irrelevant and noisy features selected for classification	•computation or development of scoring function is nontrivial •not the most efficient method to find some optima, rather than global •complications involved in the representation of training/output data

TABLE III

COMPARATIVE ANALYSIS OF DIFFERENT IMAGE CLASSIFICATION TECHNIQUES WITH RESPECT TO VARIOUS PARAMETERS

Parameter	Artificial Neural Networks	Support Vector Machines	Fuzzy logic	Genetic Algorithms
Type of approach	Non-parametric	Non-parametric with binary classifier	Stochastic	Large time series data
Non-linear decision boundaries	Efficient when the data have only few input variables.	Efficient when the data have more input variables	Depends on priori knowledge for decision boundaries.	Depends on the direction of decision
Training speed	Network structure, momentum rate, learning rate, converging criteria	Training data size, kernel parameter, class separability	Iterative application of the fuzzy integral	Refining irrelevant and noise genes
Accuracy	Depends on number of input classes.	Depends on selection of optimal hyper plane	Selection of cutting threshold	Selection of genes
General performance	Network structure	Kernel parameter	Fused fuzzy integral	Feature selection

The comparative analysis of different image classification algorithms with respect to several parameters is as in Table III. The artificial neural networks and support vector machines follows non-parametric approach whereas fuzzy measures use stochastic properties for image classification. The selection of non-linear boundary is efficient when the data have only few input variables in ANN and vice versa in SVM. In fuzzy logic it depends on priori knowledge whereas in genetic algorithms it depends on the direction of decision. The training speed in the neural networks depends on network structure, momentum rate, learning rate and converging criteria. In SVM it depends on training data size and class separability. The Fuzzy measures incorporate the training speed depending on the isolation of the relevant information by iterative application of the fuzzy integral. The training speed could be improved by refining irrelevant and noisy genes in genetic algorithms. Along with these parameters, the accuracy and general performance are also compared (Table III).

IV. CONCLUSIONS

This paper attempts to study and compare artificial neural networks and other methods of machine learning algorithms for image classification. The study concluded that the neural network approach of classification improves the accuracy and the finer information from the individual class is obtained by using textures. This study emphasizes that kernel type and kernel parameter affect the shape of the decision boundaries as located by the SVM and thus influence the performance of the SVM. It is found that the optimum results are obtained if the stochastic information is fused by the fuzzy integral. The combined genetic algorithm plus conventional classifier system achieves higher performance than either the conventional classifier or the Genetic algorithm alone. The neural network topology described in this study is determined manually. The study can be extended further to apply the genetic algorithm for neural network structure optimization.

REFERENCES

[1] S Haykin, *Neural Network - a Comprehensive Foundation; a Computational Approach to Learning and Machine Intelligence*, Macmillan, NY, 1994.

[2] J. M. Zurada, *Introduction to Artificial Neural Networks System,* Jaico Publishing House.
[3] Freeman, "Artificial Neural Network Algorithm" *Applications and Programming*, Comp. and Neural Systems Series, 1990.
[4] A. Kulkarni, *.Artificial Neural Network for Image Understanding, Van* Nostrand Reinhold, New York, 1994.
[5] J. Anderson, *An Introduction to Neural Network*.
[6] T Jackson*, Neural Computing,* An Introduction.
[7] Haralick and Shanmugan, "Textural Features for Image Classification", IEEE Transactions on Systems, Man and Cybernatics, vol. SMC 3, no 6, pp. 610, November 1973.
[8] V.N.Vapnik, *T he Nature of Statistical L earning T heory,* (New York: Springer-Verlag), 1995.
[9] V. N Vapnik, *Statistical L earning T heory* (New York: Wiley), 1998.
[10] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition", *Data Mining and Knowledge Discovery*, 2, 1998, pp. 121–167.
[11] R. K. Sundaram, *A First Course in Optimization Theory,* (New York: Cambridge University Press), 1996.
[12] B. Campbell, "Spatial correlation effects upon accuracy of supervised classification of land cover", *Photogrammetric Engineering and Remote Sensing*, 47, pp. 355–363, 1981.
[13] P. J .Bickel, C. A. J. Klaassen, Y. Ritov and J. A. Wellner," Efficient and Adaptive Estimation for Semiparametric Models", The Hopkins Univ. Press, Baltimore and London. 1993,
[14] S. P Banks, 1990**,** "Signal Processing, Image Processing and Pattern Recognition", Prentice Hall, New York.
[15] J. Teubner, *Digital Image Processing*, Prentice Hall, New York, 1993.
[16] H. Hetzheim, "Using Martingale Representation to Adapt Models for Non-Linear Filtering", Proceedings of ICSP 93, International Academic Publ., Beijing , pp 32-35, 1993.
[17] H. Hetzheim, "Anwendung der Fuzzy-Logic in der Bildverarbeitung," Vision Jahrbuch' 93, Sprechsaal Pub., Coburg, pp. 47-53, 1993.
[18] M. H. Hetzheim, "Detection of stochastic structures in Images by Fuzzy and Choquet Integrals", ACCV'95, Second Asian Conference on Computer Vision ,III , pp 116-120,1993.
[19] Sugeno, "Theory of fuzzy integrals and its application", Doctoral Thesis, Tokyo Institute of Technology, (1974).
[20] U. H.G. KreBel, "Pairwise Classification and Support Vector Machines," in B. Scholkopf, C. J. C. Burges, and A. J. Smola, Editors, "*Advances in Kernel Methods: Support Vector Learning",* pp 255-268, The MIT Press, Cambridge, MA, 1999.
[21] J. Richards and X. Jia, Remote Sensing Digital Image Analysis, Springer, New York, 3rd ed., 1999.
[22] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone, Genetic Programming: An Introduction, Morgan Kaufmann, San Francisco, CA, 1998.
[23] F. Francone, M. Conrads, W. Banzhaf, and P. Nordin, "Homologous crossover in genetic programming," in Proc. of Genetic and Evolutionary Computation Conference (GECCO'99), vol. 2, pp. 1021–1026, Morgan Kaufmann, San Francisco, CA, 1999.