A Neural Network Architecture for Generalized Category Perception

Brian B Miller, *Member IEEE*, 275 Ruth Avenue, Mansfield, OH 44907
Frank L. Merat, *Member IEEE*, Associate Professor, Dept. of Electrical Engineering and Applied Physics, Case Western Reserve University, Cleveland, OH 44106

I. INTRODUCTION

The recognition of objects given a complete or partial set of features is inherent in human intelligence. The fields of pattern recognition and artificial intelligence, among others, have addressed this topic with a variety of models which lack consistency and generality. Thus, it is the goal of this paper to set forth a generalized model for object recognition (classification).

System models utilizing neural networks have been suggested for category perception. There are many publications dedicated to the fundamentals of neural networks and parallel distributed processing [1], [2], [6], [7], [8], [10], [11], [12]. The most popular of the neural based models is the multi-layer perceptron [12], capable of recognizing non-linear class boundaries. However, such a model when applied to pattern recognition has manifested poor generalization capabilities for inputs significantly separated in feature space from any of the training patterns [9]. All of these models lack three very important properties; simplicity, generality for use in a variety of applications, and training ease. The concept of a probability based recognition system has been proposed by several researchers [1], [2], [12]. Given an object(s) for recognition, the presence or lack of features in the input provides the basis for a probability weighted decision on the classification. Following this reasoning, the proposed system is based on the principles of probability. We refer to this architecture as the GCPM (Generalized Category Perception Model.)

II. FORMULATION

The function of the GCPM is to establish the presence of an object in an input feature set. For an object to be present in the input it must be supported by a set of features found in the input. If they are not present in the input, there is a low probability that the feature set was a result of that object. The term "feature" refers to a characteristic of an object that is measurable or the confidence in the presence of a non-measurable feature. Measurable features are inherently quantifiable (e.g., geometric measurements). Non-measurable features are a confidence value in the interval [0,1] which represent the probability that a feature is present in the input (e.g., topology, structure, geometric primitives, ...).

There is an association between the value of a measurable feature, or the presence of a non-measurable feature, and the probability of the presence of an object. Given an object, it is expected that all measurable properties will be (approximately) constant between appearances of the object. This suggests the existence of a probability density function between measurable feature values and the presence of object classes. Likewise, the presence of non-measurable features should be consistent. Combining both sets of features, an object can then be identified by the confidence in non-measurable features and the evaluation of measurable features This concept is the basis for the GCPM.

A. *Discrete Probabilities for Measurable Features*

Assume that n is the maximum number of features that may appear in the input, and there are at most p possible object classes. Let the input be the feature set

$$\mathbf{x} = \{x_1, x_2, \cdots, x_n\}, \tag{1}$$

where all members of x are continuous valued if the feature is measurable and in the interval [0,1] if non-measurable Additionally, let the outputs corresponding to the feature input be described by

$$\mathbf{o} = \{o_1, o_2, \cdots, o_p\}, \tag{2}$$

where all elements of o are in the interval [0,1]. These outputs o_i, can be physically interpreted as the relative probability that the input feature set, x, was due to presence of object class i.

Similar to [13], each object class has an associated set of features. Letting x_j be the set of features for object class j they define the entire set of features under consideration as

$$\mathbf{F} = \cup_{j=1}^{p} X_j . \tag{3}$$

F is the union of the feature vectors for every possible object class. The measurable features must be transformed into a probability in the interval [0,1] for each object class in order to make a probability weighted decision on the feature vector classification. One method for determining these probability density functions, in the supervised learning mode, is presented below.

Using a sample of objects (training patterns) from each class where the features in F are known, a discrete valued probability density function may be determined. Consider that a measurable feature from F, is under consideration. At discrete values of the feature the training set shows how many of each object class appear in a region of support (*ROS*) around the value. Thus, given the *ROS*, the probability of each object class can be directly determined.

For each measurable feature in F let *range* and *ROS* be defined as:

$$range = \max\_value - \min\_value \tag{4}$$
$$ROS = \left[y - \kappa_1 * \beta * range, y + \kappa_2 * \beta * range \right] \tag{5}$$

where β , κ_1, and κ_2 are determined by

$$\beta \propto \frac{1}{\text{Training Points}}, \qquad \kappa_1 + \kappa_2 = 1 \tag{6}$$

0-7803-1901-X/94 $4.00 ©1994 IEEE

and y is the discrete value of the feature under consideration. Proper selection of these parameters allows application specific knowledge to be incorporated into the perception process. Once *ROS* is obtained for each measurable feature in **F**, the probabilities of membership at these points can be computed. The discrete points obtained in this process can be the basis for estimating a continuous approximation to the probability density function for each class using an artificial neural network (ANN).

B. Probability Density Function Approximation

It has been proven that an ANN with one hidden layer, and an input and output layer is sufficient to approximate any measurable function and thus can be used as a universal function approximator [4], [5]. A schematic representation of the three-layered ANN architecture is shown in Figure 1. The input to each network is a measurable feature value, and the output is the probability for each object class.

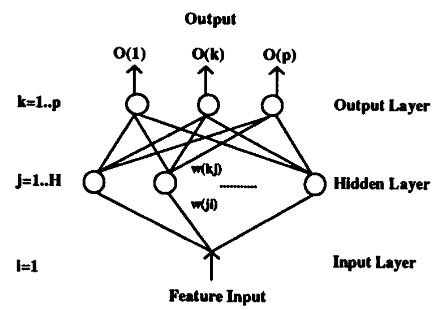

Fig. 1. Architecture of a three-layered ANN

III. NEURAL NETWORK MODEL

The complete ANN model for solving this problem is shown in Figure 2. For each measurable feature in the feature set, layers 1 through 3 represent an instance of the three-layered network described in Figure 1. Figure 2 (a) provides a schematic representation of this architecture. The dotted lines of layer 4, in Figure 2 (a), not associated with a node in layer 3 are for the effects of the non-measurable features on the final class probabilities. These inputs are described in Figure 2 (b). For each non-measurable feature, there is a uni-directional link between layers 1 and 3, without layer 2 (the hidden layer of Figure 1). Each neuron is labeled with an S or L indicating the node has sigmoidal or linear activation.

The output of layer 3, in Figure 2 (a), represents the probability that the measurable feature value was associated with each object class. Note that for each feature, there are p output nodes in layer 3 (one per object class). Layer 3 inputs, in Figure 2 (b), are uni-directional connections between the input confidence value and all object classes. The weights between layers 3 and 4, in both Figures 2 (a) and (b), have the important

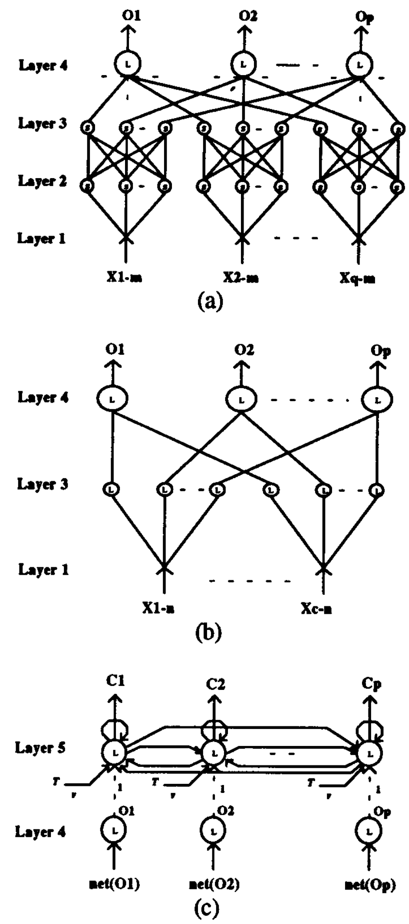

Fig. 2. Complete neural network model (a) Measurabl Feature Model (b) Non-measurable Feature Model (c Structure of the output layer

role of combining the evidence from all features in the featur vector to obtain the final class probabilities, **o**.

The output layer, shown in Figure 2 (c), has as input the output probabilities **o**, and a dynamic classificatio threshold, *T*. The output layer contains lateral inhibitor connections [1] which create competition between the objec classes. Depending on the selection of v, the system wil operate in a single object or multi-object recognition mode The outputs, *c1* to *cp*, represent the final classifications for eacl object class. In both the single and multiple object modes, an output which is greater than zero corresponds to the objec being extant in the input feature set.

B. Dynamic Behavior of the System

The system is completely feedforward in its operation until th excitation of the output layer. A feature vector is input into th system and transformed through the weights of layers 1 througl 4 to relative object class probabilities. These probabilities ar used as an initial bias to layer 5. The net input to node a i layer 5 is

$$\mathrm{net}(a)_t = \begin{cases} o(a) & t=0 \\ c(a)_{t-1} - \sum_{b=1}^{p} \varepsilon_{ab} c(b)_{t-1} & t>0 \end{cases} \quad (7)$$

where ε_{ab} are the link weights for the lateral connections in layer 5 from node b to node a, and c is the final classification output of Figure 2 (c). The output, $c(a)$, for node a is a non-linear function of the net input and the dynamic classification threshold, T, and can be written as

$$c(a)_t = \frac{g[\mathrm{net}(a)_t]}{\sum_{b=1}^{p} g[\mathrm{net}(b)_t]} \quad (8)$$

where v is the link weight of the threshold, and

$$g(\eta) = \begin{cases} \eta & \text{for } \eta > vT_t \\ 0 & \text{for } \eta \le vT_t \end{cases} \quad (9)$$

T_t is given by

$$T_t = \frac{1}{\sum_{b=1}^{p} f[c(b)_{t-1}] + \rho} \quad (10)$$

with

$$f(\alpha) = \begin{cases} 1 & \alpha > 0 \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

Note in equation 8 that the output $c(a)$ is normalized by all other outputs which results in the sum of all the outputs being equal to 1 at each time instant. The function g has the role of eliminating all output classes which are not above the dynamic threshold, T. The threshold, at any time, is slightly less (due to ρ) than 1 divided by the number of output classes still in the competition. Proper selection of parameters v,ε, and ρ will allow for operation in single object and multiple object recognition modes.

IV. SYSTEM TRAINING

There are 4 sets of weights to determine; layers 1 to 3 of Figure 2 (a), layers 1 to 3 of Figure 2 (b), layers 3 to 4 for both measurable and non-measurable features, and the weights of the output layer. Each are discussed in the following subsections.

A. Learning of the Weights for Measurable Features

The weights in layers 1 to 3 of Figure 2 (a) are determined during a supervised training process utilizing the back-propagation of error algorithm [1] [2]. Training patterns are obtained by using the methodology described in Section II. Given all the weights and thresholds for each ANN it is possible to obtain the probabilities of class membership given any measurable feature value in $\boldsymbol{F}$.

B. Selection of the Weights for the Non-Measurable Features

The weights in layers 1 to 3 of Figure 2 (b) are chosen to associate a non-measurable feature input with every object class. In other words, each of the weights is set to unity. Not every feature is a member of the feature set for every object class. Thus a filtering process must be performed to only excite an output node if the input feature is associated with the class. This will be discussed in detail in the next subsection.

C. Selecting Weights for the Combination of Evidences

The link weights from layers 3 to 4 for the GCPM are next to be determined. They have the important role of combining all the evidence from each of the features in the input feature set, $\boldsymbol{F}$. Four parameters, $v_1 - v_4$, determine the importance of a feature i in recognizing object class m. v_1 is a uniqueness measure, v_2 creates weight distribution, v_3 is a user defined consistency measure, and v_4 associates features to object classes. Thus,

$$v_1(o_m, f_i) = 1/N_i \quad (12)$$

$$v_2(o_m, f_i) = e^{-\# X_m} + 1.0 \quad (13)$$

$$v_3(o_m, f_i) = C(o_m, f_i) \quad (14)$$

$$v_4(o_m, f_i) = \begin{cases} 1 & \text{if } f_i \in X_m \\ 0 & \text{if } f_i \notin X_m \end{cases} \quad (15)$$

where N_i is the number of object classes to which feature i belongs, $\# X_m$ is the number of features in the feature set for object m, and C is user-defined. v_4 may be viewed as having the same functionality as a switch. If the feature is associated with an object class, the switch is closed allowing the effect of the feature to be transmitted to layer 4. If the feature is not associated with an object, the switch is opened and the influence of the feature is removed.

The final importance of any feature i to object o_m is written as

$$v(o_m, f_i) = \prod_{j=1}^{4} v_j(o_m, f_i) \quad (16)$$

If a perfect feature set for an object is input to the system, the output probability for that class should be fully activated. This requires the weights for a feature i going from output m of layer 3 to output m of layer 4 be chosen as

$$w_{mi} = \frac{v(o_m, f_i)}{\sum_{j=1}^{n} v(o_m, f_j)}. \quad (17)$$

To avoid full activation of two object class outputs when one has a feature vector which is a subset of another feature vector the actual selection of the weights are modified as

$$w_{mi} = \frac{v(o_m, f_i)}{\psi + \sum_{j=1}^{n} v(o_m, f_j)} \tag{18}$$

where ψ is a constant. Additionally, layer 4 nodes have a linear activation implying that the output value is simply equal to the net input.

D. Selection of the Weights in the Output Layer

The final weights to be determined are those in the dynamic output layer. Layer 4 provides the final relative probabilities that the input feature set was created by each of the output classes. The role of layer 5 is to take those probabilities and finalize the classification process. As described, recognition can be accomplished in the single or multiple object recognition mode. With the proper selection of v, ε_{ab}, and ρ, described in equations 7 through 11, both modes of operation can be accomplished. In the single mode, it is known *a priori* that only one class may appear in the input at one time. Thus, the role of the output layer should be to simply select the maximum output of layer 4, and suppress the other outputs. This can be achieved by choosing the parameters to realize the MAXNET configuration [1]. In some instances, the input feature set may have been composed by the superposition of many objects. Combining the effects from both modes of operation, ε_{ab}, the inhibitory weight, is chosen as

$$\varepsilon_{ab} = \begin{cases} \varepsilon & \text{if } X_a \in X_b\text{: for } a \neq b \\ (1-v)\varepsilon & \text{if } a \neq b \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

and the classification threshold weight, v, is

$$v = \begin{cases} 1 & \text{multiple object recognition mode} \\ 0 & \text{single object recognition mode} \end{cases} \tag{20}$$

The value of ρ in equation 10 is chosen to avoid the recognition of objects which don't exist in the input. A logical approach in classification is to eliminate many object classes early in the competition. This requires a threshold which is (relatively) the largest when many object classes are in the competition, and decreases as object classes are eliminated. Thus ρ is written as

$$\rho = e^{\sum_{b=1}^{p}(\zeta - f[c(b)_t])} \tag{21}$$

ζ and ψ (equation 18) are usually optimized simultaneously during system training. This concept is presented in Section V. When a stable condition is reached only the object classes present in the input will have final output classifications greater than zero. In the multi-object mode, like the single object mode, a stable condition is reached as the time step approaches infinity.

V. EXPERIMENTAL RESULTS

The GCPM has been tested with two different experimen utilizing the process presented. The first experiment tests th validity of the non-measurable model and the second th measurable model. In the first stage of both experiment training patterns were presented to the network and appropriat learning and selection of weights was performed. At th completion of the learning stage, the system was tested fc recognition capabilities. This section describes each of th experiments and discusses the significance of the results.

The first experiment consisted of six non-measurabl features with four possible output classes. The 6-bit featur vectors are shown in Table I. The complete neural networ model is similar to Figures 2 (b) and (c) with 6 inputs, 24 node in layer 3 (4 per feature), 4 nodes in layer 4 (one per outpı class), and 4 nodes in layer 5. The binary vectors wer randomly chosen with the exception of the fourth object clas which was purposely chosen as a subset of object 2. The value represent the confidence levels in the presence of nor measurable features (e.g., binary pixel values). The values (*v3* and ε were chosen as 1.0 and 0.1, respectively. Parameter ψ and ζ were optimized using the Guided Evolutionar Simulated Annealing (GESA) algorithm [14] which is a hybri of simulated annealing optimization [15]. The values of ψ and ζ which yielded the best results were 0.52 and 0.88.

During the recognition stage of the experiment, th feature confidence values were contaminated with uniforml distributed noise which varied from 0.0 to each of the nois levels between 0.0 to 0.5. Table II contains the results of th experiment for the recognition of the four objects in the singl mode of operation. The data was collected over a sample (1000 runs of the algorithm at each of 6 noise levels. Note th for all the objects 100% recognition was obtained even at nois levels of 0.4 except for 0.2% misclassification of object 2 at noise level of 0.4. During this experiment robust recognitic was possible even at noise levels of 0.5 and below for operatic in the single object mode.

Table III contains the portion of the results for tł recognition of the objects in the multi-object mode when onl single objects were presented at the input. Three classes (results are indicated in Table III; "True", "False", and "Extra True is defined as recognizing only the objects which wei contained in the input feature vectors. False implies that sorr other object(s) were identified other than those contained in tł input. Finally, extra means that the correct objects wei recognized along with another object(s). Nearly 100' recognition of the objects up to noise levels of 0.3 is possib without the identification of extra objects when object 4 excluded. The recognition of extra objects along with tł correct object becomes increasingly prevalent at noise leve beyond 0.4. This is because the lack of confidence in tł presence (or absence) of features creates enough support for tl extra objects to be identified.

Table IV shows the results using the GCPM whe superposed objects of Table I are input in the multi-obje mode. The results show excellent recognition capabilities fı superposed objects up to noise levels of 0.3. The percent extra and false objects identified also increases over inpu which were created by the contamination of single object feature sets. This is because multiple objects superimposed upon each other create feature sets which closely resemble feature sets from other object classes. This data suggests that multi-object recognition for superposed objects is robust up to noise levels of 0.3. Situations where the combinations of objects results in a feature set which contains other objects as subsets should be avoided to eliminate excessive recognition of extra or false classes. In reality, this will be a rare occurrence in implementing the GCPM.

TABLE I
FEATURE VECTORS FOR EXPERIMENT 1

Object	$f1$	$f2$	$f3$	$f4$	$f5$	$f6$
1	1	0	0	1	1	0
2	1	1	0	0	1	1
3	0	1	1	0	1	0
4	1	1	0	0	1	0

TABLE II
SINGLE MODE RECOGNITION

Noise	Percent Correct Recognition			
	Object 1	Object 2	Object 3	Object 4
0.0	100	100	100	100
0.1	100	100	100	100
0.2	100	100	100	100
0.3	100	100	100	100
0.4	100	99.8	100	100
0.5	100	95.4	100	93.0

TABLE III
MULTI-OBJECT MODE - SINGLE OBJECTS

Object	Noise Level					
1	0.0	0.1	0.2	0.3	0.4	0.5
True	100.0	100.0	100.0	100.0	92.8	85.7
False	0.0	0.0	0.0	0.0	0.0	0.0
Extra	0.0	0.0	0.0	0.0	7.2	14.3
2	0.0	0.1	0.2	0.3	0.4	0.5
True	100.0	100.0	100.0	100.0	99.6	94.5
False	0.0	0.0	0.0	0.0	0.0	0.0
Extra	0.0	0.0	0.0	0.0	0.4	5.5
3	0.0	0.1	0.2	0.3	0.4	0.5
True	100.0	100.0	100.0	99.9	93.4	81.6
False	0.0	0.0	0.0	0.0	0.0	0.0
Extra	0.0	0.0	0.0	0.1	6.6	18.4
4	0.0	0.1	0.2	0.3	0.4	0.5
True	100.0	100.0	100.0	76.8	49.4	26.5
False	0.0	0.0	0.0	0.0	0.0	6.4
Extra	0.0	0.0	0.0	23.2	50.6	67.1

TABLE IV
MULTI-OBJECT MODE - SUPERPOSED OBJECTS

Objects	Noise Level					
1&2	0.0	0.1	0.2	0.3	0.4	0.5
True	100.0	100.0	99.8	93.2	82.7	70.5
False	0.0	0.0	0.0	0.0	0.1	0.7
Extra	0.0	0.0	0.0	0.0	0.0	0.0
1&3	0.0	0.1	0.2	0.3	0.4	0.5
True	100.0	100.0	100.0	88.9	74.4	57.1
False	0.0	0.0	0.0	0.6	5.0	17.4
Extra	0.0	0.0	0.0	3.5	7.1	5.5
1&4	0.0	0.1	0.2	0.3	0.4	0.5
True	100.0	100.0	99.9	93.3	84.2	62.2
False	0.0	0.0	0.0	0.0	0.0	5.7
Extra	0.0	0.0	0.0	0.0	0.0	0.5
2&3	0.0	0.1	0.2	0.3	0.4	0.5
True	100.0	100.0	99.7	93.6	80.7	73.6
False	0.0	0.0	0.0	0.0	0.3	1.2
Extra	0.0	0.0	0.0	0.0	0.0	0.0
3&4	0.0	0.1	0.2	0.3	0.4	0.5
True	100.0	100.0	99.7	94.7	83.1	58.3
False	0.0	0.0	0.0	0.0	0.0	6.8
Extra	0.0	0.0	0.0	0.0	0.0	0.9

In the second experiment, the task was to solve th real problem of determining the number of spheres (circles associated with any region in an image. A version of the imag utilized in the test is provided in Figure 3. The problem i based

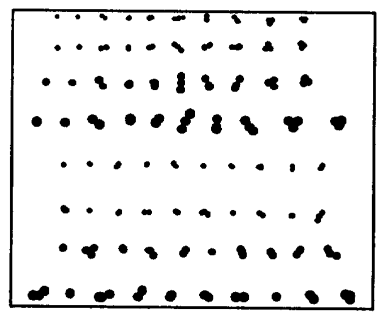

Fig. 3. Overlapping spheres utilized for experiment 2

on the decomposition of overlapping particles in the study o fluid motion using seed particles. The image contains fou different sized spheres at various levels of overlap an orientation. Overlapping can only occur with spheres of th same size and no more than three spheres can be present in an one region. Thus, there are 3 possible output classes. The to four rows were used for training, and the bottom four fo verification. The features used for recognition are measurabl features. Specifically, the circumference divided by the radiu of the sphere, and the major axis of the bounding ellipse [3 divided by the radius is contained in the feature vector for al three classes. The complete system model is similar to Figure 2 (a) and (c) with 2 inputs (1 per feature), 24 nodes in layer

(12 hidden nodes per measurable feature), 6 nodes in layer 3 (3 per feature), and 3 nodes in layers 4 and 5 (1 per output class).

During the learning stage of the experiment, discrete values of the features in the feature vector were extracted and used for learning the continuous probability density functions as described in Section II. The parameter β was chosen to be 0.15 because a fairly small amount of training points were utilized. If a larger amount of patterns had been used, this parameter could have been decreased. Additionally, a self-imposed goal was to ensure misclassification always occurred to the class with the least number of spheres per region. For example, if a two sphere region were to be misclassified, it should be recognized as a single sphere region and not a triple sphere region. By selecting parameters κ_1 and κ_2 to be 0.7 and 0.3, respectively, such selective misclassification can be obtained. This demonstrates one of the most powerful aspects of the GCPM, control over the classification of bounding regions with measurable features. $v3$ (consistency measure) was chosen to be 1.5 for the normalized circumference, and 1.0 for the normalized major axis feature. The circumference of a region is an exact measurement, while the major axis is an approximation to the bounding ellipse. Thus, the circumference can be considered to be a more consistent measure than the major axis, and is therefore given additional weighting in the final classification. Once again, the utilization of this parameter provides the user with powerful control of the recognition process.

The recognition task involved extracting the feature vectors from the verification set, inputting them into the neural network system, and determining the final region classification. Because only one class may be present per region, single mode operation was used during recognition ($v = 0$). The results from this experiment are shown in Table V. The percent correct classification was expected to decrease as the number of spheres per region increased. This is because a triple sphere region may look like a double or single sphere region if there is a great deal of overlap. Likewise, a double sphere region could appear to be a single sphere region if the spheres were 100% overlapped. A greater percentage of the double and triple sphere regions could have been correctly identified if selective misclassification were not utilized ($\kappa_1 = \kappa_2 = 0.5$).

TABLE V
RESULTS FROM EXPERIMENT 2

Spheres Per Region	Recognition Results
1	100.0
2	97.5
3	80.0

VI. DISCUSSION AND CONCLUSIONS

A generalized model for learning object classes has been presented and the associated neural network model discussed. In developing the paradigm, learning rules for determining the weights for both measurable and non-measurable features are derived. The GCPM has many strong points which allow for robust perception in the presence of noise. First, it provides an effective way of dealing with both measurable and non-measurable features. A straight forward approach in going from feature values to the probabilities of class memberships is inherent in the system. Second, the user is given control over selective misclassification using parameters κ_1 and κ_2, which can bias the probability density functions. Additionally, the user's experience can be utilized in the appropriate selection of the consistency measure of a feature, v_3. Third, the capability to operate the system in both single and multiple object recognition modes is provided. Both modes of operation exhibit robust classification, even with significant amounts of noise on the input feature vectors. This ability was demonstrated in experiments 1 and 3. By accounting for each of these situations, the GCPM can truly be categorized as general.

References

[1] Y. H. Pao, *Adaptive Pattern Recognition and Neural Networks.* Addison-Wesley, Reading, Mass, 1989.

[2] D.E. Rumehalt, G. Hinton, and G. Williams, "Learning internal representation by error propagation," *Parallel Distributed Processing*, vol. 1. MIT Press, Cambridge, Mass, 1986.

[3] Robert M. Haralick and Linda G. Shapiro, *Computer and Robot Vision.* vol. 1. Addison-Wesley, Reading Mass, 1992.

[4] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, No. 2, pp. 303-314, 1989.

[5] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359-366, 1989.

[6] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE Acout., Speech, Signal Processing*, vol. 4, pp. 4-22, 1987.

[7] S. E. Fahlmann and G. E. Hinton, "Connectionist architecture for artificial intelligence," *IEEE Computer*, vol. 20, pp. 100-109, 1987.

[8] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, pp. 386-408, 1958.

[9] G. G. Lendaris and I. A. Harb, "Improved generalization in ANNs via use of conceptual graphs: a character recognition task as an example case," in *Proc. IJCNN-90*, San Diego, CA, 1990, pp. I-551 - I-556.

[10] T. Kohonen, *Self-Organization and Associative Memory.* Berlin, Germany: Springer-Verlag, 1988.

[11] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," in *Proc. Nat. Academy Sciences*, 1982, pp. 2554-2558.

[12] J. J. Hopfield and D. W. Tank, "Neural computation of decision in optimization problems," *Biol. Cybern.*, vol. 52, pp. 141-152, 1985.

[13] J. Basak, C. A. Murphy, S. Chaudhury, D. D. Majumder, "Connectionist model for perception: theory and implementation," *IEEE Transactions on Neural Networks*, vol. 4, pp. 257-269, 1993.

[14] P. Yip, "The role of regional guidance in optimization: The guided evolutionary simulated annealing approach," *PhD Dissertation*, Case Western Reserve University, 1993.

[15] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671-680, 1983.