# Learning Fuzzy Rules for Pattern Classification and High-Level Computer Vision

*Chung-Hoon Rhee

Abstract

In many decision making systems, rule-based approaches are used to solve complex problems in the areas of pattern analysis and computer vision. In this paper, we present methods for generating fuzzy IF-THEN rules automatically from training data for pattern classification and high-level computer vision. The rules are generated by constructing minimal approximate fuzzy aggregation networks and then training the networks using gradient descent methods. The training data that represent features are treated as linguistic variables that appear in the antecedent clauses of the rules. Methods to generate the corresponding linguistic labels (values) and their membership functions are presented. In addition, an inference procedure is employed to deduce conclusions from information presented to our rule-base. Two experimental results involving synthetic and real data are given.

## Ⅰ. Introduction

For many high-level computer vision systems, rule-based approaches have been used to design systems that try to perform complex tasks such as image understanding and scene interpretation [1]-[4]. In these systems, common-sense knowledge about the world is represented in terms of rules, and the rules are then used by an inference mechanism to arrive at a meaningful interpretation of the contents of the image. Furthermore, determination of properties and attributes of image regions and spatial relationships among image regions is critical for high-level vision processes involved in tasks such as autonomous navigation, medical image analysis, and scene interpretation. In domains such as the blocks world [5] and the world of generalized cylinders [6], the properties (features) of the objects in the image can be precisely defined. Therefore, existing techniques for scene description and interpretation perform quite well. However, when the attributes of objects and spatial relationships between objects are not well-defined (as in the case of outdoor scenes), traditional techniques may not be adequate. For example, in a rule-based system to interpret outdoor scenes, a typical rule may be

IF a **region** is **RATHER GREEN** AND **HIGHLY TEXTURED** AND
IF the **region** is **SOMEWHAT BELOW** a **Sky region**
THEN it is **Trees**.

The above rule may be translated as:

IF the **greenness** of a **region** is **RATHER HIGH** AND
the **texturedness** is **HIGH** AND
the **belowness** of it in relation to **Sky regions** is **SOMEWHAT HIGH**
THEN it is **Trees**.

The terms such as "RATHER HIGH," "HIGH," and "SOMEWHAT HIGH" are known as linguistic labels. Linguistic labels, as well as attributes such as "green" and "textured" defy precise definitions, and they are best modeled by fuzzy sets (fuzzy linguistic variables [7]). Similarly, spatial relationships among regions such as "left-of," "above," and "below" are difficult to model using the all-or-nothing traditional techniques [8]. Therefore, we believe that a fuzzy approach to high-level vision will yield more realistic results.

In the existing rule based systems for high-level vision, the rules are usually enumerated by experts. However, this procedure is rather tedious if the number of rules is large. Moreover, in a fuzzy approach, one needs to specify not only the rules but also the membership functions associated with the various linguistic labels. The membership functions for the labels need to relate to feature data if the system is to perform reasonably well. Therefore, the designing and modeling of membership functions is a crucial aspect in rule generation. There have been many studies in the control area that relate to modeling of membership functions and rule generation [9]-[12]. These methods have been shown to be very effective. However, very little work has been done in applying these ideas to

*Department of Electronic Engineering Hanyang University
Manuscript Received: May 31, 1997.

the area of rule-based computer vision. We believe that this is a promising area of research and therefore, this is the focus of this paper.

In this paper, we present methods to generate fuzzy IF-THEN rules as well as the membership functions of linguistic labels associated with the rules automatically from training data. The proposed methods are particularly suitable for pattern classification and high-level vision. In Section 2, we describe the fuzzy aggregation operator (e.g., generalized mean), and fuzzy aggregation networks which we propose to use in our approach. In Section 3, we present a method for generation of fuzzy rules from training data that involves features with the various steps involved. In Section 4, we present a method for inference that can be employed with our rule-base to deduce conclusions from input information. Section 5 presents experimental results involving our rule generation and inference method for computer vision applications. These examples involve synthetic data and real data from images. Finally, Section 6 gives the summary and conclusions.

## Ⅱ. Fuzzy Aggregation Operators and Networks

In [13][14], the properties of several union and intersection connectives, the generalized mean, and the $\gamma$-model have been investigated. The behavior of these connectives when they are used in fuzzy aggregation networks has also been investigated. In particular, the generalized mean operator [15] (given below) has several attractive properties.

$$g_p(x_1, \ldots, x_n; w_1, \ldots, w_n) = \left( \sum_{i=1}^{n} w_i x_i^p \right)^{1/p},$$

$$\text{where } \sum_{i=1}^{n} w_i = 1. \tag{1}$$

For example, the mean value always increases with an increase in $p$ [15]. Thus, by varying the value of $p$ between $-\infty$ and $+\infty$ we can obtain various types of aggregation for values between min and max. For example, $g_{-\infty}$ is the min operator, $g_0$ is the geometric mean, and $g_{+\infty}$ is the max operator. Therefore, in the extreme cases, this operator can be used as union or intersection. Also, the $w_i$'s can be thought of as the relative importance factors for the different criteria due to the constraint.

In [13][14][16][17], the authors discuss methods for managing the uncertainty inherent in properties (features) while decision making by means of hierarchical fuzzy aggregation networks. In these hierarchical networks, each node aggregates the degree of satisfaction of a particular criterion. The inputs to each node are the degrees of satisfaction of each of the sub-criteria, and the output is the aggregated degree of satisfaction of the criterion. Such hierarchical networks are known as fuzzy-connective-based aggregation networks, or fuzzy aggregation networks for short.

It has been shown that optimization procedures based on gradient descent can be used to determine the proper type of aggregation connective and parameters at each node, given only an approximate structure of the network and given a set of training data that represent the inputs at the bottom-most level and the desired outputs at the top-most level [13][14]. Also, it has been shown that such networks are capable of detecting certain types of redundant features in a decision-making problem. If the attributes, properties, and relationships used in antecedent clauses of rules in high-level vision systems are interpreted as criteria, then one can model rule-based region labeling also as a hierarchical network. In this paper, we use this idea and the redundancy detection capability of fuzzy aggregation networks to automatically generate fuzzy rules from training data.

## Ⅲ. Methods for Learning Fuzzy Rules Involving Features

Properties (features) of classes can provide the necessary entities for proper rule generation. Some features can be used to discriminate among classes more effectively than others depending upon the type of classes involved. However, when features are not well defined, the classification process can become very challenging. In light of the ideas mentioned above, we develop a method for generating fuzzy rules that deals with features that may not be well defined (as in the case of natural scenes). The rules are automatically generated from training data. The proposed method for generating fuzzy IF-THEN type rules consists of the following four stages: Ⅰ) estimation of class membership functions, Ⅱ) elimination of redundant criteria (features), Ⅲ) estimation of the membership functions of linguistic labels that best describe the non-redundant features, and finally Ⅳ) construction of an approximate network structure that can be used for generating the rules that best describe the training data.

### 3.1 Stage Ⅰ: Estimation of Class or Criterion Membership Functions

Class membership functions can provide a way of de-

termining the degree of satisfaction of classes or criteria (features) used in the antecedent clauses of the rules of type IF-THEN [16][17]. They can also be used to compute the degrees of satisfaction of criteria in the bottom most layer of the aggregation networks. We now present a method for estimating these membership functions.

In this method, a normalized histogram from the training data is treated as a possibility distribution [14][18] and the membership in each class for a particular feature value is then directly calculated using these possibility functions. One advantage of this approach is that the membership values are independent of the membership values in the other classes. Therefore, addition of new classes to the problem can be handled easily. However, one problem with this method is that when the number of available training samples is small, we need to use interpolation or smoothing techniques to obtain a reasonable membership function. We now describe one method to do this [16].

Let $x_1, \ldots, x_K$ denote $K$ features, for example position and texturedness. We first normalize the training feature values so that they fall in the domain [0, 1]. Let there be one such domain for each of the $K$ features. Each of these domains is then fuzzily partitioned into $Q$ levels represented by $Q$ corresponding fuzzy sets. Let $\boldsymbol{x}_u^j = (x_{1u}^j, \ldots, x_{Ku}^j)$ denote the $u^{th}$ feature vector from class $j$, where $j = 1, \ldots, M$. We now define the class $j$ membership function $\tilde{\mu}_k^j$ over the domain of feature $x_k$ to be computed as:

$$\tilde{\mu}_k^j(i) = \frac{1}{N^j} \sum_{u=1}^{N^j} f_{ki}^j(x_{ku}^j) \quad \text{for } i = 1, \ldots, Q, \tag{2}$$

where $f_{ki}^j(\,)$ is the membership function of level $i$ defined over the domain of $x_k$ for class $j$, and $N^j$ is the number of class $j$ samples. Equation (2) gives memberships of features at the $Q$ (fuzzy) levels and can be converted to normalized memberships in $Q$ crisp intervals as follows:

$$m_k^j(x_k) = \frac{\tilde{\mu}_k^j(i)}{\max\{\tilde{\mu}_k^j(1), \ldots, \tilde{\mu}_k^j(Q)\}} \quad \text{for } \beta_i \le x_k < \beta_{i+1}, \tag{3}$$

where

$$\beta_0 = 0,\ \beta_i = \frac{1}{Q-1}\left(i - \frac{1}{2}\right) \text{ for } 1 \le i \le Q-1, \text{ and } \beta_Q = 1.$$

In (3), $m_k^j(x_k)$ denotes the class $j$ membership function for feature $x_k$. The shape and support of the membership functions used to describe the $Q$ levels (i.e., $f_{ki}^j(\,)$) controls the type and extent of the interpolation. One example is shown in Figure 1.

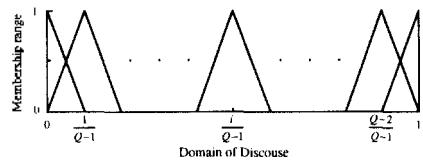



Figure 1. An example of $Q$ fuzzy sets representing $Q$ levels of a feature.

### 3.2 Stage II : Elimination of Redundant Criteria(Features)

In this stage, a method to remove redundant criteria (features) from the antecedent clauses of the rules is proposed [16]. This is achieved by training a three-layer aggregation network of the type shown in Figure 2. The input layer to the network consists of $K$ features (attributes or properties) that represent a feature vector $x$. Each feature is then aggregated to the middle layer. The class membership functions computed using the method described in Stage I are used as activation functions in the nodes of the middle layer. For training purposes, the desired output for a feature vector coming from a particular class is 1 at the node representing the class, and it is 0 at all the other nodes. The generalized mean compensative fuzzy aggregation operator is used at the output nodes in the top layer. The network is then trained to learn the aggregation connective (generalized mean) parameter values that best describe the input/output relationships. The learning is implemented using a modified gradient descent approach which is explained in detail in [13][14]. As the weights approach zero, redundant features are detected.

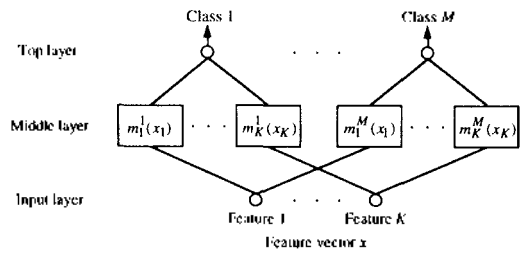



Figure 2. Aggregation network used for detecting redundant features.

### 3.3 Stage III : Determination of the Membership Functions of the Linguistic Labels

In order to insure a reliable set of rules for a particular application, the linguistic labels must be modeled so that they adequately and compactly describe each criterion (feature) involved. Therefore, after eliminating redundant

features (Section 3.2), the next step in rule generation is to generate the membership functions for the various linguistic labels (such as LOW, MEDIUM, and HIGH) that each non-redundant feature can take. These membership functions may be estimated from the membership functions $m_k^j(x_k)$ given in (3). We now explain the procedure to estimate membership functions of linguistic labels.

A suitable parametrized function is chosen to model the membership function of a linguistic label. Let $h_k(x_k, p)$ denote the chosen parametrized function where $p = (p_1, \ldots, p_n)$ is the parameter vector. Then each of the functions $m_k^j(x_k)$ is approximated by a set $H_k^j = \{h_k^j(x_k, p_i) \mid i = 1, \ldots, L_k^j\}$ of such parametrized functions, where $L_k^j$ is the number of parametrized functions required for a reasonable approximation of $m_k^j(x_k)$. We then form the union $H_k$ of all such sets $H_k^j$ for $j = 1, \ldots, M$. If any two functions $h_k(x_k, p_l)$ and $h_k(x_k, p_m)$ (henceforth denoted by $h_{kl}$ and $h_{km}$ for short) in $H_k$ have very similar parameter values, then one of them is removed. (Two parametrized functions $h_{kl}$ and $h_{km}$ obviously have distinct parameter values if they both come from the same set $H_k^j$. If they come from sets corresponding to different $j$'s (i.e., classes), it is still unlikely that they have similar parameter values. Otherwise, it usually means that feature $x_k$ is not good because the functions $m_k^j(x_k)$ for two distinct classes overlap too much.) The resulting $H_k$ consisting of a set of distinct parametrized functions can be interpreted as the set of membership functions of the linguistic labels defined over the domain of discourse of feature $x_k$. If we denote the cardinality of $H_k$ by $L_k$, we will have $L_k$ linguistic labels to describe the feature $x_k$. Thus, determining the membership functions of the linguistic labels reduces to the problem of estimating the parameters $p_i$ and the number of parametrized functions required for a reasonable approximation of class membership functions $m_k^j(x_k)$.

To illustrate the above procedure, we shall consider Gaussian type functions as a suitable parametrized function to model the membership functions of the linguistic labels, and describe a method for estimating the parameters and the number of parametric functions required for a reasonable approximation of each of the class membership function $m_k^j(x_k)$ [16][17]. This is shown as follows.

Consider a Gaussian function given by

$$G(x, c, \sigma) = a \exp\left[i - \frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2\right], \tag{4}$$

where $a$ is the height, $c$ is the mean value, and $\sigma$ is the standard deviation. Hence, the parameter vector $p = (a, c, \sigma)$ for a Gaussian function. If a membership function $m_k^j(x_k)$ consists of multiple peaks, we can model it by a sum of $L_k^j$ Gaussians as follows.

$$m_k^j(x_k) \approx \tilde{G}_k^j(x_k) = \sum_{i=1}^{L_k^j} a_{ki}^j G_{ki}^j(x_k), \tag{5}$$

where

$$G_{ki}^j(x_k) = \exp\left[-\frac{1}{2}\left(\frac{x_k - c_{ki}^j}{\sigma_{ki}^j}\right)^2\right]$$

is the $i^{th}$ parametrized function for feature $x_k$ in class $j$, and $a_{ki}^j$, $c_{ki}^j$, and $\sigma_{ki}^j$ denote the height, the mean value, and the standard deviation of the Gaussian, respectively. Hence, $p_{ki}^j = (a_{ki}^j, c_{ki}^j, \sigma_{ki}^j)$.

In order to approximate the class $j$ membership function for feature $x_k$ as a sum of parametrized functions, we could minimize the following objective function

$$J_k^j(p_k^j) = \frac{1}{2} \sum_{i=1}^{L_k^j} [h_k^j(x_k, p_{ki}^j) - m_k^j(x_k)]^2, \tag{6}$$

where $h_k^j(x_k, p_{ki}^j)$ is the $i^{th}$ parametrized function (i.e., gaussian in this case) chosen to model the membership function $m_k^j(x_k)$ for feature $x_k$ in class $j$. We can now use a gradient descent method to estimate the parameter vector $p_{ki}^j$ by the following update rule

$$p_{ki}^{j(\text{new})} = p_{ki}^{j(\text{old})} - \rho \frac{\partial J_k^j}{\partial p_{ki}^j}, \tag{7}$$

where $\rho$ is a positive learning constant. The parameter vector is iteratively updated until there is little or no change in the parameter values. This occurs when the partial derivatives of $J_k^j$ with respect to each component of $p_{ki}^j$ are approximately equal to zero (i.e., when the chosen approximation of parametrized functions $h_k^j(x_k, p_{ki}^j)$ closely matches the membership function $m_k^j(x_k)$.

For the case of Gaussian functions, the partial derivatives of $J_k^j$ with respect to each component of $p_{ki}^j$ are

$$\frac{\partial J_k^j}{\partial a_{ki}^j} = \frac{\partial J_k^j}{\partial \tilde{G}_k^j(x_k)} \frac{\partial \tilde{G}_k^j(x_k)}{\partial a_{ki}^j} = (\tilde{G}_k^j(x_k) - m_k^j(x_k)) G_{ki}^j(x_k), \tag{8}$$

$$\frac{\partial J_k^j}{\partial c_{ki}^j} = (\tilde{G}_k^j(x_k) - m_k^j(x_k)) \frac{a_{ki}^j G_{ki}^j(x_k)(x_k - c_{ki}^j)}{(\sigma_{ki}^j)^2}, \tag{9}$$

$$\text{and} \quad \frac{\partial J_k^j}{\partial \sigma_{ki}^j} = (\tilde{G}_k^j(x_k) - m_k^j(x_k)) \frac{a_{ki}^j G_{ki}^j(x_k)(x_k - c_{ki}^j)}{(\sigma_{ki}^j)^3}. \tag{10}$$

It is well known that when using gradient descent methods, the choice of the initial values for the parameters is critical, due to the local minima problem. Therefore, for the case of Gaussian functions, we can use the following heuristic approach [17] consisting of 4 steps to obtain the initial parameters.

*Step 1*: Generate the membership functions $m_k^j(x_k)$ as explained in Section 3.1.

*Step 2*: Using a least squares approximation, fit a polynomial $p(x)$ of the lowest possible degree (i.e., to avoid overfitting) such that the fit to each $m_k^j(x_k)$ has a reasonably small error.

*Step 3*: Calculate the extrema (maxima and minima) values for $p(x)$ in Step 2 and determine the number of Gaussians by the number of positive valued maxima, ignoring the ones that have small peaks.

*Step 4*: Initialize the heights of the Gaussians by the maxima values (peak values), the mean values as the locations of these peaks, and the standard deviation as the shortest value among the distances between the mean of each Gaussian and the nearest minima or roots of $p(x)$.

### 3.4 Stage IV: Approximate Network Structure for Rule Generation

The final stage is to obtain the compact set of rules which may contain multiple antecedent clauses joined together either conjunctively or disjunctively. To achieve this, we use a three-layer fuzzy aggregation network. We initially start with an approximate structure for the aggregation network which is then trained to detect redundant connections, if any. As in Section 3.2, the target (desired) values for the training data are chosen to be 1 for the class from which the training data are extracted, and 0 for the remaining classes. When the training is complete and all the redundant connections are eliminated, the resulting network is interpreted as a set of decision rules. The nodes in the middle and top layers can represent either conjunctive or disjunctive nodes depending on the final values of the parameters of the aggregation function. For example, when using the generalized mean as the aggregation connective, then a value much less (greater) than 1.0 for $p$ indicates a conjunction (disjunction) (e.g., $g_{-\infty}$ is the min operator and $g_{+\infty}$ is the max operator). Also, the weights $w_i$ determine the relative importances of the antecedent clauses in a rule. We now present the method for constructing the initial approximate structure for the three-layer network.

Figure 3 shows the approximate network structure for implementing this method [17]. From the figure, the input layer consists of $K^*$ input nodes ($K^* \leq K$), where each input node represents a non-redundant feature for at least one class. The bottom layer consists of $K^*$ groups of nodes, where each group corresponds to the linguistic labels describing a non-redundant feature. The $i^{th}$ node in the $k^{th}$ group uses $h_{ki}$ (the membership function of the $i^{th}$ linguistic label for feature $k$ obtained from Stage III) as the activation function. The middle layer consists of $K^*$ groups of $M$ nodes each, where $M$ is the number of classes. The $i^{th}$ node in group $k$ in the bottom layer is connected to the $j^{th}$ node in the corresponding group in the middle layer if feature $k$ is considered non-redundant for class $j$ and the support of $h_{ki}$ has a non-empty intersection with the support of $m_k^j(x_k)$ (the class $j$ membership function). An illustration is shown in Figure 4. From the figure, the support of $m_k^j(x_k)$ intersects with the support of the linguistic labels $h_{k2}$ and $h_{k3}$. Hence, these linguistic labels would get connected to the $j^{th}$ node in the $k^{th}$ group in the middle layer. The rationale behind this connection is that if feature $k$ is redundant for class $j$ or the support of the membership function of a linguistic label has no intersection with the support of the class membership function, then it cannot appear in the antecedent clause of a rule that describes the class. (Some parametrized membership functions such as Gaussians do not vanish anywhere in the domain. In that case, we use a small $\alpha$-cut of the membership function, rather than the support.) This connection process is repeated for all the groups in the bottom layer. Similarly, the $j^{th}$ node of every group in the middle layer that has a connection from the bottom layer is connected to the $j^{th}$ node of the top layer for $j = 1, \ldots, M$. All middle and top-layer nodes use the generalized mean operator as the activation function. The non-redundant

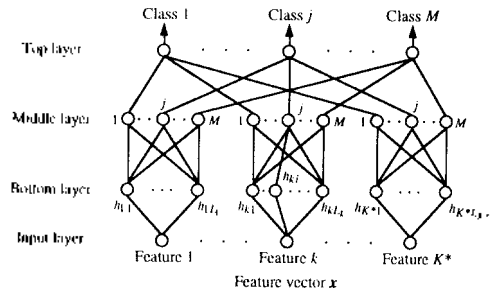



Figure 3. An approximate network structure for generating rules.

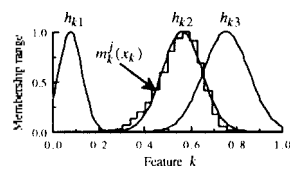



Figure 4. An example illustrating how connections are made between bottom and middle layers of the approximate network structure for generating rules.

features are fed to the corresponding groups of nodes in the bottom layer as inputs. This completes the construction of the initial approximate aggregation network. When the training is complete and all the redundant connections (if any) are eliminated, the resulting network is interpreted as a set of fuzzy rules. In Section 5, we show several examples of rule generation using such fuzzy aggregation networks.

## IV. Method for Inference

To demonstrate the effectiveness of rules that are generated in any rule-based system, it is necessary to deduce conclusions from information that is presented to the system. Hence, a method for inferencing needs to be developed. In this section, we describe a method for inferencing that uses one crisp test feature vector at a time. The method utilizes the rule generation network from the previous section. We now present the method for inference in more detail.

The output vectors in our case are binary vectors filled with zeros in all locations but one, and these correspond to target vectors used while training the rule generation networks. Since the rules are inferred from the rule generation network, ideally the rule generation network will produce the desired binary output when a test input feature vector exactly matches an antecedent clause of one of the rules. If there is no exact match, the output will be a weighted combination of the binary output vectors, where the weights are ideally in proportion to the degree to which the input matches the corresponding antecedent clauses. Thus, the rule generation network (with the aggregation parameters fixed at values obtained at the end of training) can act as a rule-matching network. The output of this network can be defuzzified using the maximum-membership defuzzification scheme. This is shown in Figure 5.

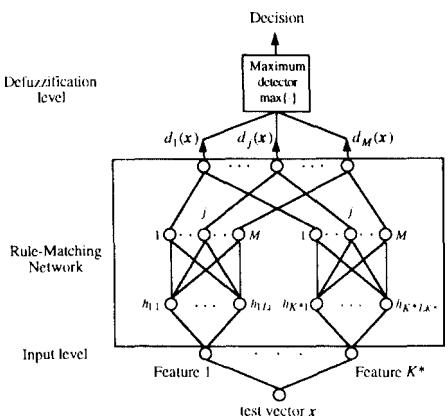



Figure 5. Inference network.

## V. Experimental Results

Example 1) Iris data problem

For our first example, we show some results using the classical "iris data" problem. This problem consists of 3 classes and 4 features (such as petal length), with 50 samples in each class. Each of the four features was first mapped into the interval [0, 1]. Features 1 and 2 are not very good because there is a lot of overlap in the feature values between the classes, as can be seen in Figure 6(a). However, features 3 and 4 are quite good for characterization of the classes, as can be seen in Figure 6(b). We used 20% of the data from each class (i.e., 10 samples per class) for testing and the remaining 80% for training for rules. This was repeated 5 times using different data sets for testing (i.e., every fifth sample in each class starting with sample number 1~5 was used) and the remainder for training. Results for 1 of the 5 trials (i.e., using the test set that starts with sample number 5 in each class) are presented in detail.

Figure 7 shows the class membership functions for the four features using 80% of the data. The domain of the features was quantized into 32 levels and the resulting histograms were obtained using a triangular window function with a support of 7 units. Features 1 and 2 were eliminated using the redundancy detection method dis-

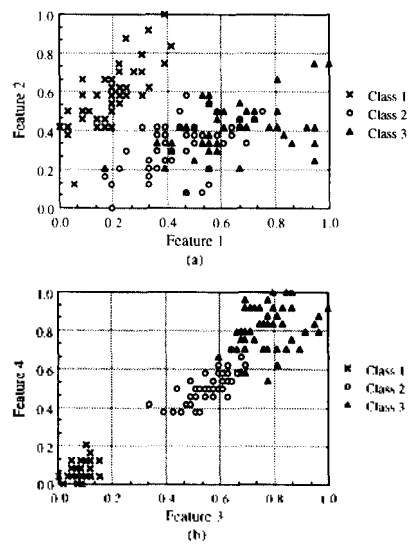



Figure 6. Scatter plot of (a) features 1 and 2, and (b) features 3 and 4 of the "iris data."

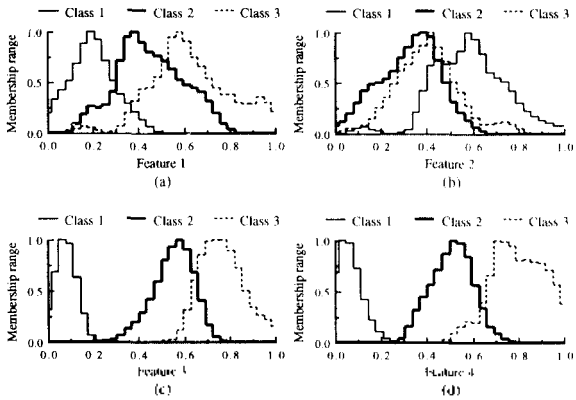



Figure 7. Estimated class membership functions for the "iris data" for (a) feature 1, (b) feature 2, (c) feature 3, and (d) feature 4 using 80% of the data.

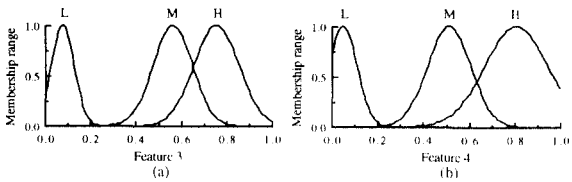



Figure 8. Gaussian fitted linguistic labels of the "iris data" resulting from Figure 7 for the two non-redundant features, (a) feature 3 and (b) feature 4.

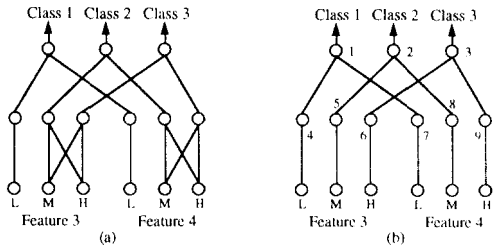



Figure 9. (a) Approximate network structure for generating rules for the "iris data." (b) Reduced network after training.

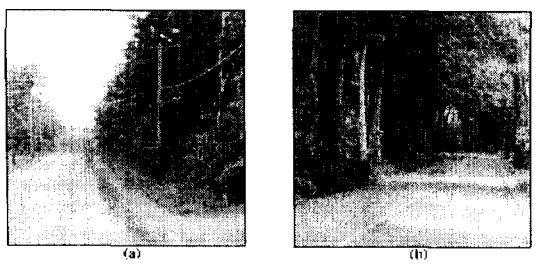



Figure 10. Images of natural scenes used for training: (a) Scene 1 and (b) Scene 2.

Table 1. Values for the weights and parameter $p$ for the reduced network in Figure 9(b).

| node | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| weights | 0.56 | 0.41 | 0.17 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| | 0.44 | 0.59 | 0.83 | | | | | | |
| $p$ | 7.37 | -0.14 | 6.81 | 1.00 | 1.02 | 1.03 | 1.00 | 0.92 | 1.09 |

Table 2. Confusion matrix for 5 trials of the "iris data."

| | | Recognized class | | |
|---|---|---|---|---|
| | | 1 | 2 | 3 |
| True class | 1 | 50 | 0 | 0 |
| | 2 | 0 | 47 | 3 |
| | 3 | 0 | 3 | 47 |

cussed in Section 3.2. Figure 8 shows the membership functions of the resulting linguistic labels generated by fitting Gaussians to the non-redundant class membership functions (i.e., features 3 and 4). Figure 9, and Table 1 show the final results of training. The rules obtained from the final network in Figure 6.19(b) are listed below.

$R_1$: IF **Feature 3** is **LOW** OR **Feature 4** is **LOW**
THEN the class is **Class 1**.

$R_2$: IF **Feature 3** is **MEDIUM** AND **Feature 4** is **MEDIUM**
THEN the class is **Class 2**.

$R_3$: IF **Feature 3** is **HIGH** OR **Feature 4** is **HIGH**
THEN the class is **Class 3**.

It can be seen that this set of rules is a good description of the data set shown in Figure 6(b). The rules generated for each of the remaining 4 trials gave similar results. When the inference procedure described in Section 4 was used to test the remaining 20% of the data, the number of correct classifications varied slightly for each trial. The average rate of correct classification for the 5 trials was 96%. Table 2 shows the confusion matrix for the 5 trials.

In comparison, when a neural network (backpropagation algorithm with 8 hidden units fully connected) was used, the correct classification rate was 95.3%. This indicates that our method (using a smaller network structure) performs as well as backpropagation.

Example 2) Natural scene problem

As our next example, we present the results of a more realistic experiment involving natural scenes. Figures 10 (a) and (b) show two $200 \times 200$ images of natural scenes used in training for rules. The images consist of three regions (classes) "road," "sky," and "vegetation." We used five features: two color features (intensity and excess green), two texture features (homogeneity and entropy), and position (row number). The color features (based on

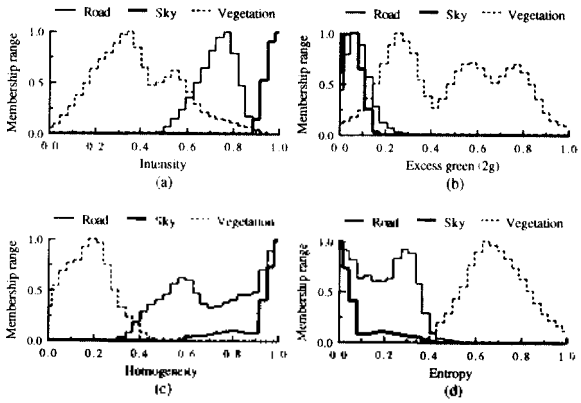



Figure 11. Membership functions of the outdoor scenes for the features (a) intensity (b) excess green, (c) homogeneity, and (d) entropy.

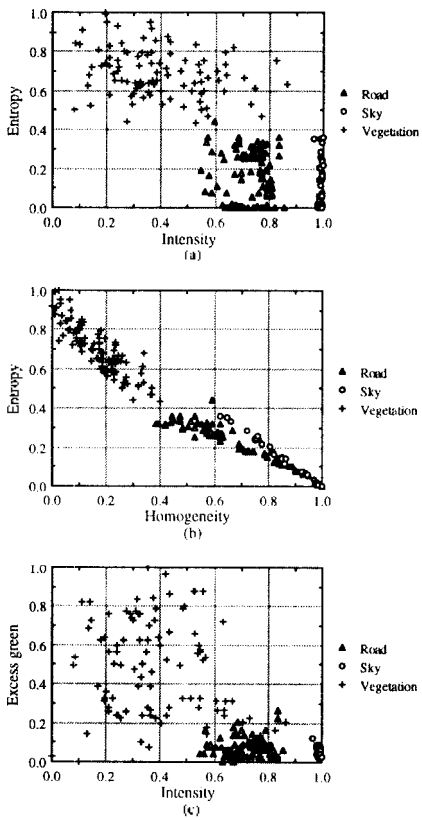



Figure 12. Scatter plot of (a) features intensity and entropy, (b) features homogeneity and entropy, and (c) features intensity and excess green representing the training data of the outdoor scenes in Figure 10.

Ohta's color space [19]) were extracted from the red, green, and blue components of the training scenes after applying a median filter of size $3\times3$ to remove noise points. The intensity feature values were obtained by averaging the three color components (i.e., $(r+g+b)/3$) and the excess green feature values by $2g-r-b$. The two

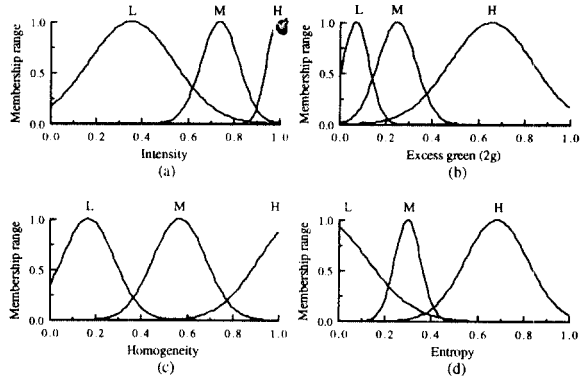



Figure 13. Gaussian fitted linguistic labels resulting from the histograms of the natural scene data for the features (a) intensity, (b) excess green, (c) homogeneity, and (d) entropy.

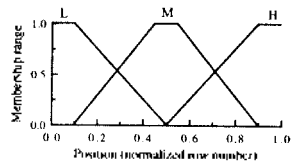



Figure 14. Heuristically predetermined linguistic labels representing the position feature.

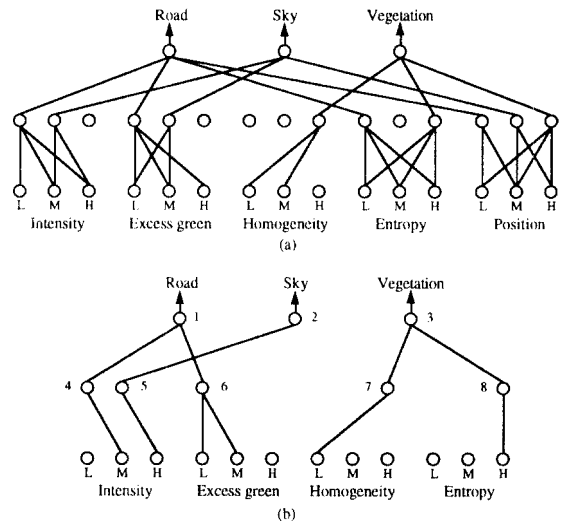



Figure 15. (a) Approximate network structure for generating rules for the outdoor scene. (b) Reduced network after training.

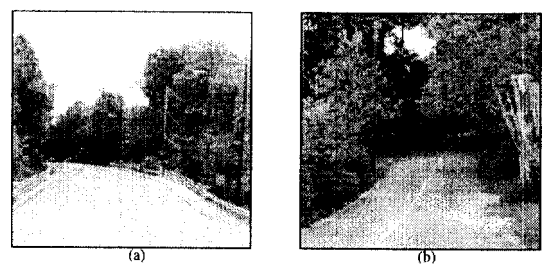



Figure 16. Images of natural scenes used for testing: (a) Scene 3 and (b) Scene 4.

texture features were obtained from the excess green color feature. Appendix A shows the two color and texture feature images for the two training scenes. A total of 100 samples from the class regions (sampled uniformly) were used to represent each class (i.e., 50 samples from each scene representing road and vegetation, and 100 samples from Scene 1 representing sky). Next, the redundancy detection method described in Section 3.2 was used to eliminate redundant features. Figure 11 shows the estimated class membership functions of the two color and texture features using the histogram method after mapping each feature into the interval [0, 1]. As before, the domain of the features was quantized into 32 levels and the resulting histograms were obtained using a triangular window function with a support of 7 units. The redundancy detection method eliminated the homogeneity feature for class "road," the homogeneity and entropy features for class "sky," and the intensity and excess green features for class "vegetation." Figure 12 shows scatter plots involving combinations of the non-redundant features.

Figure 13 shows the estimated membership functions for the linguistic labels that describe the four features in Figure 12. The linguistic labels for the position feature were heuristically predetermined as shown in Figure 14. Figure 15 and Table 3 show the final results of training. After training, the position feature was further eliminated for all classes. The rules obtained from the final network in Figure 15(b) are listed below.

$R_{road}$: IF **Intensity** is **MEDIUM** AND **Excess green** is **(LOW OR MEDIUM)**
THEN the class is **Road**.

$R_{sky}$ : IF **Intensity** is **HIGH**
THEN the class is **Sky**.

$R_{veg}$ : IF **Homogeneity** is **LOW** OR **Entropy** is **HIGH**
THEN the class is **Vegetation**.

These rules are similar to what an expert might elaborate.

We now present some inference results involving the two training images and two test images (see Figure 16). Figures 17 shows results of the inference method applied to one of the training scenes, namely Scene 1. Parts (a), (b), and (c) of Figures 17 shows the membership values (i.e., gray levels toward 0 (255) are considered low (high) membership) of "road", "vegetation", and "sky" respectively, resulting from the outputs of the rule-matching network as discussed in Section 4. Part (d) shows the resulting labeled images after the defuzzification stage. Similarly, Figure 18 shows the resulting labeled images for the

Table 3. Values for the weights and parameter $p$ for the reduced network in Figure 15(b).

| node | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| weights | 0.0004 | 1.0000 | 0.6501 | 1.0000 | 1.0000 | 0.9425 | 1.0000 | 1.0000 |
| | 0.9996 | | 0.3499 | | | 0.0575 | | |
| $p$ | -0.6948 | 0.9218 | 7.3604 | 0.8280 | 1.0044 | 5.5103 | 1.0374 | 1.0257 |

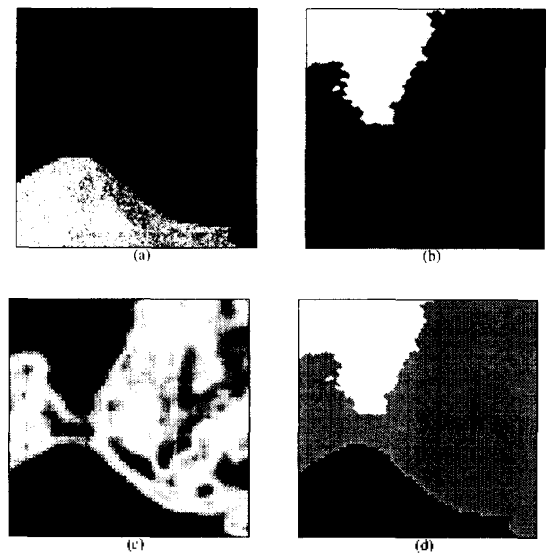



Figure 17. Membership values of (a) road, (b) sky, and (c) vegetation resulting from the outputs of the rule-matching network, and (d) the resulting labeled image after the defuzzification stage for Scene 1.

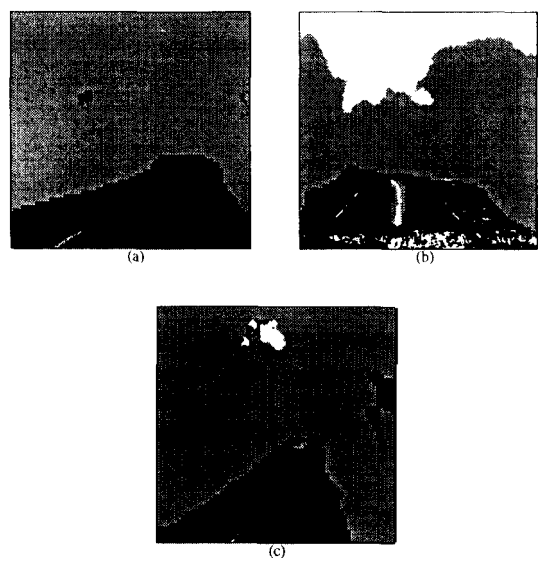



Figure 18. The resulting labeled images after the defuzzification stage for (a) Scene 2, (b) Scene 3, and (c) Scene 4.

other training image and the two test images.

## VI. Summary and Conclusions

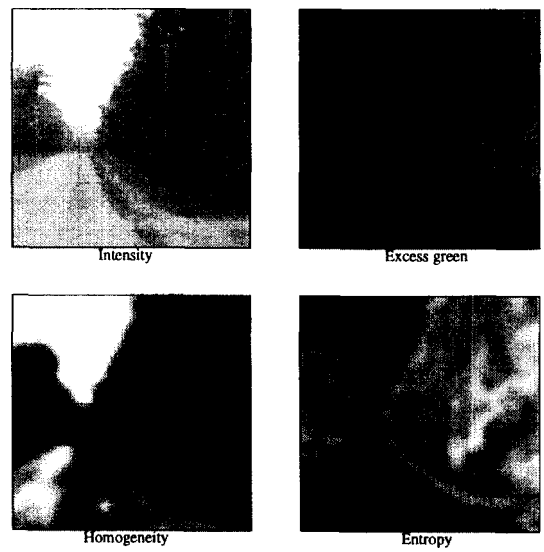
In this paper, we discussed various issues involved in the development of generating fuzzy rules automatically from training data for high-level vision. We suggested methods to generate linguistic labels and their membership functions to describe the features. These methods were used to develop methods to generate a compact set of rules by constructing minimal approximate network structures using ideas from fuzzy aggregation networks. Although we use a gradient decent procedure to train the rule generation network, convergence to a local minimum is unlikely due to the fact that we start with an approximate network which is reasonably close to the final solution. Results from inference show that our proposed methods for rule generation is effective.

## References

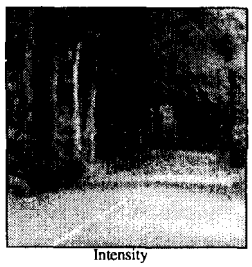

1. T. Binford, "Survey of model-based image analysis systems," *Int. J. Robotics Research*, vol. 1, no. 1, pp. 18-64, 1982.
2. C. Chu and J. Aggarwal, "Image interpretation using multiple sensing modalities," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 14, no. 8, pp. 570-585, Aug. 1992.
3. D. McKeown, W. Harvey, and J. McDermott, "Rule-based interpretation of aerial imagery," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 7, no. 5, pp. 570-585, Sept. 1985.
4. M. Nagao, T. Matsuyama, and H. Mori, "Structural analysis of complex aerial photographs," *Proceedings of the 6th Int. Joint Conf. Artificial Intell., IJCAI*, Tokyo, Japan, pp. 610-616, Aug. 1979.
5. P. Winston, *Artificial Intelligence*. Reading, MA: Addison-Wesley, 1984.
6. R. Brooks, "Symbolic reasoning among 3-D models and 2-D images," *Artificial Intelligence*, vol. 17, pp. 285-349, 1981.
7. L. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning," *Information Sciences*, Part 1, vol. 8, pp. 199-249, 1975, Part 2, vol. 8, pp. 301-357, 1975, Part 3, vol. 9, pp. 43-80, 1975.
8. J. Freeman, "The modeling of spatial relations," *Computer Graphics and Image Processing*, vol. 4, pp. 156-171, 1975.
9. S. Horikawa, T. Furuhashi, and Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm," *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 801-806, Sept. 1992.
10. C. Lin and C. S. G. Lee, "Neural-network-based fuzzy logic control and decision system," *IEEE Trans. Comput.*, vol. 40, no. 12, Dec. 1991.
11. M. Sugeno and G. Kang, "Structure identification of fuzzy model," *Fuzzy Sets Syst.*, vol. 28, no. 1, pp. 15-33, Oct. 1988.
12. L. Wang and J. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst. Man Cybern.* vol. 22, no. 6, pp. 1414-1427, Nov./Dec. 1992.
13. R. Krishnapuram and J. Lee, "Fuzzy-connective-based hierarchical aggregation networks for decision making," *Fuzzy Sets Syst.*, vol. 46, no. 1, pp. 11-27, Feb. 1992.
14. R. Krishnapuram and J. Lee, "Fuzzy-set-based hierarchical networks for information fusion in computer vision," *The Journal of Neural Networks*, vol. 5, no. 2, pp. 335-350, March 1992.
15. H. Dyckhoff and W. Pedrycz, "Generalized means as a model of compensation connectives," *Fuzzy Sets Syst.*, vol. 14, no. 2, pp. 143-154, Nov. 1984.
16. R. Krishnapuram and F. C.-H. Rhee, "Compact fuzzy rule base generation methods for computer vision," *Proceedings of the 2nd IEEE Conf. Fuzzy Systems*, San Francisco, CA, vol. II, pp. 809-814, March 1993.
17. F. C.-H. Rhee and R. Krishnapuram, "Fuzzy rule generation methods for high-level computer vision," *Fuzzy Sets Syst.*, vol. 60, pp. 245-258, Dec. 1993.
18. J. Keller, D. Subhangkasen, K. Unklesbay, and N. Unklesbay, "Approximate reasoning for recognition in color images of beef steaks," *International Journal of General Systems*, vol. 16, no. 4, pp. 331-342, 1990.
19. Y. Ohta, *Knowledge-Based Interpretation of Outdoor Natural Color Scenes*. Boston, MA: Pitman Advanced Publishing Inc., 1985.

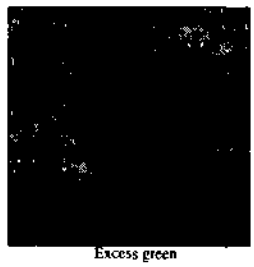

## APPENDIX A
## FEATURE IMAGES OF THE TRAINING SCENES IN SECTION V.

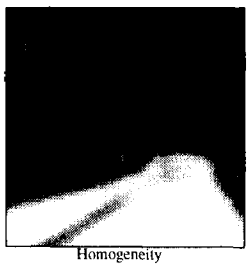
### A.1 Scene 1

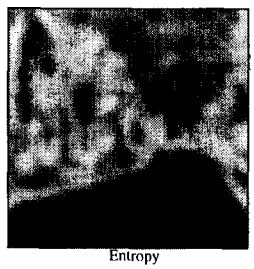


A.2 Scene 2


Intensity


Excess green


Homogeneity


Entropy

▲Chung-Hoon Rhee

Chung-Hoon Rhee was born March 3, 1963, in Seoul, Korea. He received the B.S. degree in electrical engineering from the University of Southern California (USC), Los Angeles, in 1985. He attended the University of Missouri, Columbia, from 1985 to 1987 and from 1990 to 1993, where he received the M.S. and Ph.D. degrees in electrical engineering respectively.

From 1990 to 1993, he was a research assistant in the Department of Electrical and Computer Engineering, University of Missouri, Columbia, where his research involved applications of computer vision, pattern recognition, neural networks, and fuzzy set theory. From 1994 to 1995, he was a Senior Member of the Engineering Staff in the High-speed Network Access Section at the Electronics and Telecommunications Research Institute (ETRI), Taejon, Korea, where his work involved applications of group communications. He has been a faculty member in the Department of Electronic Engineering at Hanyang University, Ansan, Korea, since August 1995. His current interests include applications of computer vision, pattern recognition, and computational intelligence.

Dr. Rhee was a co-chair for the Image Processing and Pattern Recognition area for the 1997 International Conference on Neural Networks held in Houston, TX, June 8-12. He is a member of the IEEE, the Korea Institute of Telematics and Electronics, the Korea Fuzzy Logic and Intelligent Systems Society, and the Acoustical Society of Korea.