Electronics and Communications in Japan, Part 2, Vol. 86, No. 3, 2003
Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. J82-D-II, No. 4, April 1999, pp. 685–693

Knowledge Extraction from Scenery Images and Recognition Using Fuzzy Inference Neural Networks

Hitoshi Iyatomi and Masafumi Hagiwara

Faculty of Science and Technology, Keio University, Yokohama, 223-8522 Japan

SUMMARY

A new method of extracting various types of human knowledge from scenery images in linguistic form and labeling the various regions of unknown images in an effective manner is proposed. In this system, partitioning of the input/output space can be performed in a self-organizing manner. Rule extraction is performed in parallel and automatically by using a knowledge extraction network (KEN). Depending on the type of image to be learned, two-phase processing including a self-organization phase and an LMS learning phase is performed by several KENs. The overall network output is then unified and knowledge extraction and region recognition are performed linguistically. Software that performs a series of such operations has been implemented and it has been confirmed that knowledge extraction and better image recognition can be performed linguistically even if human knowledge is not provided. © 2003 Wiley Periodicals, Inc. Electron Comm Jpn Pt 2, 86(3): 82–90, 2003; Published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/ecjb.10138

Key words: neural network; fuzzy inference; scenery image; knowledge extraction; image recognition.

Contract grant sponsor: Funded in part by the Japan Technical Support Program (Project Number JSP-RFT F 97 I 00103).

1. Introduction

Image understanding is an important topic in today's highly information-oriented society, and much research has been performed in this field. However, most past research has required a priori heuristic knowledge of the image region in question for recognition [1–4]. In addition, the problem of space partitioning as a preprocessing stage must also be solved [5–7]. Although this method gives better recognition results under certain constraints, it lacks flexibility. Thus, when the general recognition problem is encountered, we must solve a difficult problem of defining and representing suitable knowledge of the object.

Based on these considerations, and using the learning features of layered neural networks, various methods for recognizing objects at pixel level have been developed [8–10]. Recognition methods that use the pixel as the unit do not need clear knowledge or definition of regions in the image, which makes automatic processing easy. In Ref. 8, objects were extracted by partitioning the region in a self-organizing manner, and fuzzy measures were used to enhance the antinoise characteristics. However, the shapes of only relatively simple objects were treated in these studies. In Ref. 9, a method of region recognition that combined the use of several neural networks was proposed. However, the images treated were limited to satellite photos. Reference 10 reports better recognition results for scenery images without requiring knowledge of the objects. However, much effort in the adjustment of parameters was involved. In addition, since a multilayer neural network with a back-propagation learning algorithm was used, the allotment of the hidden-layer neurons was not clear. Hence, knowledge extraction from the network was not possible.

Automatic knowledge acquisition by computers is very difficult, but is the subject of much research owing to its wide range of applications. Methods that combine the better learning power of neural networks with strong fuzzy linguistic processing have been reported.

ANFIS, reported by Shing and Jang [11], uses neural networks to extract the shape of the membership function and the fuzzy rules. However, a priori partitioning of the input space is required, and the system characteristics are greatly dependent on preprocessing. In contrast, Wang [12] has proposed the extraction of fuzzy rules by automatic partitioning of the input space. However, since these methods do not consider the output space, the extracted rules are essentially nonoptimal. Nishina and Hagiwara [13] have described the automatic partitioning of the input space and automatic extraction of fuzzy rules, and have proposed the Fuzzy Inference Neural Network (FINN), which has a simple structure.

In this paper, using an improved FINN as the basic element, a new system is proposed in which the extraction of linguistic knowledge from scenery images is performed automatically and various regions of the image can be recognized. The proposed system does not require heuristic knowledge, and automatic recognition of scenery images can be performed at the pixel level. In addition, knowledge extraction from the high commonsense level to a low level is possible.

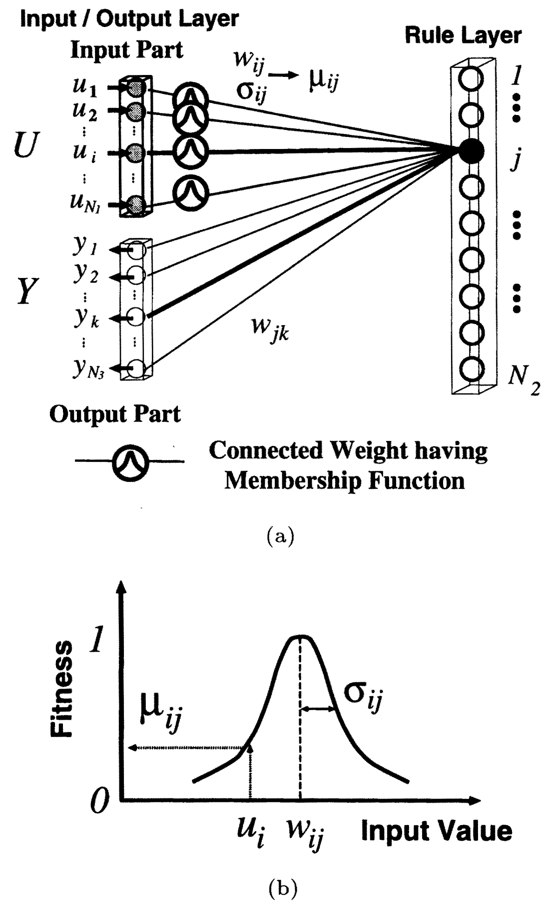


Fig. 1. (a) The structure of FINN. (b) An example of membership function.

2. FINN: The Fuzzy Inference Network

2.1. The structure of the FINN

The proposed system performs input/output space partitioning in a self-organizing manner and the FINN is used for automatic rule extraction, Here we give a simple explanation of the FINN as proposed in Ref. 13.

The structure of the FINN is shown in Fig. 1(a). It consists of two layers, the input/output layer and the rule layer. The input/output layer consists of an input part and output part and there is complete connection between the input part and the rule layer, and between the rule layer and the output part. From the input part, membership functions are attached to the rule layer, and from the rule layer, consequent parameters are attached to the output part. The nodes in the rule layer represent one fuzzy inference rule. However, from the input part, the premise parts of if–then fuzzy inference rules are represented in the rule layer, and the consequent parts are represented from the rule layer to the output part.

2.2. Operation of the FINN

We next explain the operation of the FINN. If we take the number of input variables as N_1, the number of fuzzy rules as N_2, and the number of output variables as N_3, then the input to the FINN can be represented as follows:

$$\boldsymbol{U} = (u_1, u_2 \cdots, u_i \cdots, u_{N_1}) \tag{1}$$

The premise membership function from input part node i to node j of the rule layer is a bell-shaped curve as shown in Fig. 1(b) and is given by

$$\mu_{ij} = \exp\left(-\frac{(u_i - w_{ij})^2}{\sigma_{ij}^2}\right) \qquad j = (1, 2, \cdots, N_2) \tag{2}$$

Here ω_{ij}, σ_{ij} are the central value and variance of the membership function, respectively. In the rule layer, the minimum ambiguity value μ_{ij} entered at each node is taken and the fitness ρ_j is calculated as follows:

$$\rho_j = \min[\mu_{1j}, \mu_{2j} \cdots, \mu_{ij}, \cdots, \mu_{N_1 j}] \qquad (3)$$

Thus, from the consequent parameters ω_{jk} of each rule, we obtain the final inference outputs $\hat{y}_k$ by taking the weighted mean of the fuzzy inference rules with fitness ρ_j. Here $\hat{y}_k$ is a quantity that expresses the degree of membership of the input $\boldsymbol{U}$ in each class k and is given by

$$\hat{y_k} = \frac{\sum_j^{N_2} (w_{jk}\rho_j)}{\sum_j^{N_2} \rho_j} \qquad k = (1, 2, \cdots, N_3) \qquad (4)$$

The fuzzy inference rule obtained from node j of the rule layer can be obtained as follows:

If u_1 is $\tilde{w}_{1j}$, and $\cdots$, u_i is $\tilde{w}_{ij}, \cdots, u_{N_1}$ is $\tilde{w}_{N_1 j}$
then $\hat{y_k}$ is w_{jk}

Here $\tilde{w}_{ij}$ is a neighborhood of w_{ij} that depends on the magnitude of σ_{ij}.

2.3. Learning in the FINN

Learning in the FINN [13] consists of three phases: the self-organizing phase, the rule extraction phase, and the LMS learning phase.

2.3.1. The self-organizing phase

A conceptual diagram of the self-organizing phase is shown in Fig. 2. In the self-organizing phase, the central value w_{ij} of the membership function and the consequent parameter w_{jk} are determined by using the Kohonen algorithm [14].

For unified treatment of the input space and output space, first the N_1-dimensional input vector $\boldsymbol{U}$ given in Eq. (1) is combined with the corresponding expected N_3-dimensional signal vector $\boldsymbol{Y}$:

$$\boldsymbol{Y} = (y_1, y_2, \cdots, y_k, \cdots, y_{N_3}) \qquad (5)$$

in the form of an $(N_1 + N_3)$-dimensional learning vector $\boldsymbol{I}$ as follows:

$$\boldsymbol{I} = (\boldsymbol{U}, \boldsymbol{0}) + (\boldsymbol{0}, \boldsymbol{Y}) \qquad (6)$$

By using the Kohonen self-organizing algorithm for the learning vector, analogous vectors can be placed closer on the map. Moreover, similar fuzzy rules may also be placed closer on the map.

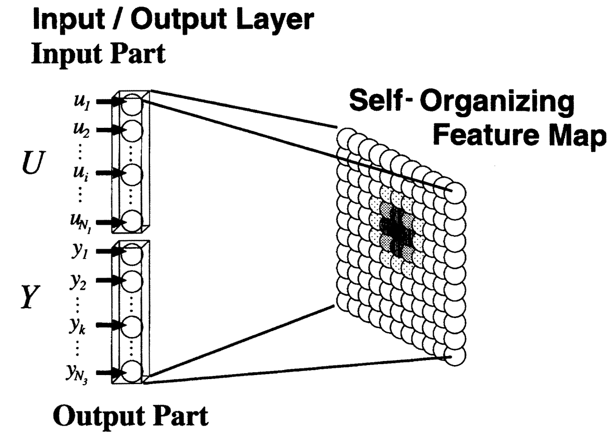


Fig. 2. Self-organizing phase.

For ease of understanding, the connection vectors $\boldsymbol{W}_j^U$ (N_1-dimensional) and $\boldsymbol{W}_j^Y$ (N_3-dimensional) from the input part and the output part to neuron j of the map layer are combined as follows to obtain the weight vector $\boldsymbol{W}_j$ ($N_1 + N_3$ dimensions):

$$\boldsymbol{W_j} = (\boldsymbol{W_j^U}, \boldsymbol{0}) + (\boldsymbol{0}, \boldsymbol{W_j^Y}) \qquad (7)$$

First, the weight factor $\boldsymbol{W}_j$ of neuron j and the Euclidean distance of the input vector $\boldsymbol{I}$ are calculated, and the winning neuron s with the shortest distance is sought:

$$||\boldsymbol{W_s} - \boldsymbol{I}|| = \min_j ||\boldsymbol{W_j} - \boldsymbol{I}|| \qquad (8)$$

Next, the winning neuron s and the weight factor of the neighborhood are updated as follows:

$$\boldsymbol{W_j}(t+1) = \boldsymbol{W_j}(t) + \epsilon_{self}(t) h(j, s, t)(\boldsymbol{I} - \boldsymbol{W_j}(t)) \qquad (9)$$

Here ε_{self} is the learning coefficient in the self-organizing phase and is a decreasing function of time. $h(j, s, t)$ is a neighborhood function given by

$$h(j, s, t) = \exp\left(-\frac{{d_{js}}^2}{\sigma(t)^2}\right) \qquad (10)$$

Here d_{js} is the Euclidean distance on the map of neuron j that updates the winning neuron s and the weight. $\sigma(t)$ is a quantity that decreases exponentially from $\sigma_{initial}$ to σ_{final} as learning progresses. Thus, the weight is updated over a wide range at the start of learning and the range is subsequently narrowed.

Repeating the above procedure, the characteristic relationships between input and output are automatically classified. The weight vectors of the neurons of the input/output layer obtained in this way correspond to the premise and consequent parts of one fuzzy rule.

2.3.2. The rule extraction phase

In the rule extraction phase, the Euclidean distances of any two weighting factors on the map that are less than the threshold value ξ are combined and a new mean weight factor is generated. This operation is repeated until the difference in the Euclidean distances of the weight factors less than the predetermined ξ vanishes. In this way the system can be constructed with fewer rules.

2.3.3. The LMS learning phase

In the self-organization phase, the central value of the membership function and the consequent parameter are roughly determined. The least mean square (LMS) learning rule is used to determine the variance of the membership function, and final adjustment of the central value and the consequent constant is performed at the same time. To guard against overlearning, this process should be delayed to the minimum extent. In LMS learning, all parameters are optimized to minimize the total error function. In practice, we minimize the error function E_p defined by the following equation, which is equivalent to minimizing the total error function E:

$$E_p = \frac{1}{2}\sum_{k}^{N_3}(y_k - \hat{y_k})^2 \tag{11}$$

$$E = \sum_p E_p \tag{12}$$

Here y_k is the teacher signal or the expected signal, and $\hat{y}_k$ is the output of the inference network; p is the number of patterns used in learning. In the proposed system it can be obtained as the product of the number of training images and the number of pixels in each image.

If x is the variable that is to be optimized, then the variation of the variable updated by the LMS algorithm becomes proportional to $-\delta E/\partial x$. Hence, the formula for updating the weights becomes

$$w_{jk}(t+1) = w_{jk}(t) + \epsilon_w(y_k - \hat{y_k})\frac{\rho_j}{\sum_n^{N_2}\rho_n} \tag{13}$$

$$w_{ij}(t+1) = w_{ij}(t) + \epsilon_w \times \sum_k^{N_3}(y_k - \hat{y_k})\left(\frac{w_{jk}\sum_n^{N_2}\rho_n - \sum_n^{N_2}(w_{nk}\rho_n)}{\left(\sum_n^{N_2}\rho_n\right)^2}\right) \times q_{ij}\mu_{ij}\frac{2(u_i - w_{ij})}{\sigma_{ij}{}^2} \tag{14}$$

Similarly, the formula for updating the variance of the membership function can be obtained as follows:

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) + \epsilon_\sigma \times \sum_k^{N_3}(y_k - \hat{y_k})\left(\frac{w_{jk}\sum_n^{N_2}\rho_n - \sum_n^{N_2}(w_{nk}\rho_n)}{\left(\sum_n^{N_2}\rho_n\right)^2}\right) \times q_{ij}\mu_{ij}\frac{2(u_i - w_{ij})^2}{\sigma_{ij}{}^3} \tag{15}$$

$$q_{ij} = \begin{cases} 1, & \rho_j\ (= \min[\mu_{1j}, \cdots, \mu_{N_1 j}]) = \mu_{ij} \\ 0, & \text{else} \end{cases} \tag{16}$$

Here ϵ_w and ϵ_σ are the learning coefficients.

3. The Proposed System

3.1. Outline of the proposed system

The processing in the proposed system is shown in Fig. 3. The subnetworks are knowledge extraction networks (KENs) that are basically extensions of the FINN. Each KEN performs a different kind of learning depending on the type of input image, such as mountains, water, snow, and the like, and each network has a high recognition capability for one particular type of image.

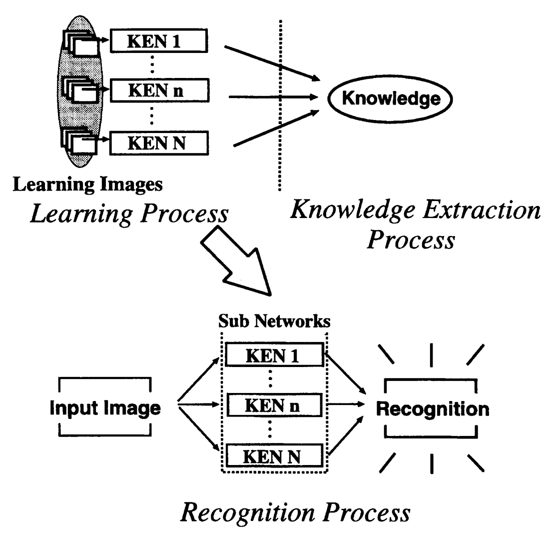


Fig. 3. Processings in the proposed system.

The proposed system consists of the following processes.

(1) The learning process

This process can be further divided into the self-organizing phase and the LMS learning phase.

(2) The knowledge extraction process
(3) The recognition process

The learning process (1) is performed independently at each KEN, but the knowledge extraction process (2) and recognition process (3) are performed collectively at the same time by all of the KENs.

3.2. Structure of the knowledge extraction network (KEN)

The structure of the KEN is shown in Fig. 4. To obtain suitable knowledge, the KEN has at least two rule layers that can be extended in two dimensions. Each node of the two two-dimensional rule layers, as in the FINN of Ref. 13, has complete connectivity with the input/output layer and can perform operations similarly to the one-dimensional case. The method of generation is also not much changed from the one-dimensional case, but due to the extension of the rule layer in two dimensions, effective learning can be performed in the self-organizing phase that uses the Kohonen algorithm.

In the input part of the input/output layer, color information of the VSH system such as the coordinates (x, y), intensity (I), hue (H), and saturation (S) at each pixel of the image can be entered. Attribute information on pixels forming parts of the sky or mountains can be entered in the output part of the I/O layer. This information can be simultaneously entered in the I/O layer, and learning between the two rule layers can be performed.

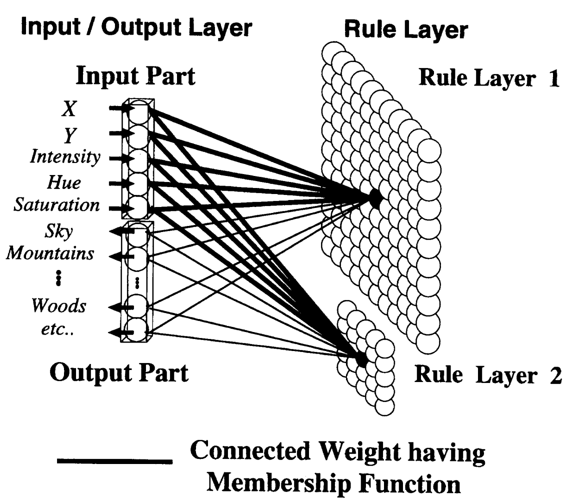


Fig. 4. Knowledge extraction network (KEN).

These two rule layers have an additional significance. Since in the large map (Rule Layer 1), a large number of fuzzy rules can be combined and stored in the membership function, in addition to using it for the generation of large amounts of knowledge, it can also be used for inference in image recognition. On the other hand, the small rule layer (Rule Layer 2) is constructed so that the totality of learning information can be represented by a small number of rules. As a result of learning, the general rules are roughly represented in Rule Layer 2, and the knowledge obtained from this layer is quite general. It is therefore used to judge the reliability of knowledge obtained in Rule Layer 1, to evaluate generality, and to eliminate contradictory knowledge.

3.3. The learning process

The proposed system is constructed from a FINN with the use of several extended KENs. The points in the learning process that are different from FINN as described in Section 2 are now explained.

3.3.1. Elimination of rule extraction processing

In Ref. 13, a single system with one FINN was constructed. However, in the proposed system, several KENs are used, which greatly increases the size of the system. Thus, instead of adjusting the number of rules in each KEN, it is better to extract knowledge or adjust the number of rules collectively, using the output of all of the KENs. It is therefore unnecessary to use a rule extraction phase for the adjustment of rules during learning by individual KENs. We can extract knowledge or adjust the number of rules after completing learning by the KENs.

3.3.2. Updating of the self-organizing phase

In the self-organizing phase of the FINN described in Ref. 13, the Kohonen self-organizing algorithm [14] was used unaltered and the winning neurons were determined by Eq. (8) based on the Euclidean distance. In the proposed system, the importance of the input signal and the training signal is considered, and the distance is calculated by the following equation:

$$\|\boldsymbol{W}_j - \boldsymbol{I}\|^2 = (\delta \boldsymbol{W}_j^U)^2 + \alpha(\delta \boldsymbol{W}_j^Y)^2 \qquad (17)$$

Here $\delta \boldsymbol{W}_j^U$ is the Euclidean distance between the input and the weighting vector of the input part of the I/O layer, and

δW_j^Y is the Euclidean distance between the output (training signal) and the weighting vector of the output part of the I/O layer; α (> 1) is a coefficient.

Increasing α results in sensitivity to different output results, with application of different rules. Thus, if the inputs are similar but different outputs are produced, different rules can be applied.

3.3.3. Updating of the LMS learning phase

In the FINN used in Ref. 13 by Nishina and Hagiwara, the same learning coefficient ε_{LMS} was used for the central value and the variance of the membership function. If ε_{LMS} is increased, the central value of the new membership function obtained in the self-organization phase is significantly altered, and the numerical information to be obtained as knowledge in the rule layer may deviate from the normalized range. As a result, extraction of appropriate knowledge does not occur. Conversely, if the learning coefficient is decreased, learning is greatly slowed.

Thus, in the proposed system, to make effective use of the connection weights of the neurons obtained by self-organization, different values of the learning coefficient are used for the central value and variance of the membership function, and the problem is solved by decreasing the former value.

3.4. The knowledge extraction process

As explained in Section 3.3.1, the KEN of the proposed system has no rule extraction phase. Instead, rule extraction, aggregation, and transformation to linguistic expressions are performed in the knowledge extraction phase after the end of learning in each KEN. For each element (e.g., position and color information) of a fuzzy if–then rule that can be represented by connection weights and a membership function in Rule Layer 1, a suitable linguistic expression is obtained by considering the type of membership function. Next, the numbers of similar rules among the linguistically transformed rules are summed. This result is used as a measure of the generality of the rules, and the group of rules is considered as a knowledge item. The same operation is performed in Rule Layer 2. A practical example is given in Section 4.1.

3.5. The image recognition phase

The process of input image recognition in the proposed system is shown in Fig. 5. The number of KENs is taken as N. Each KEN, as explained in Section 3.1, performs learning separately according to the type of input image. The input image is divided into D horizontal zones and is entered as input into each of N networks. The recognition result is obtained by aggregating the outputs of all networks. First, the three-dimensional color information

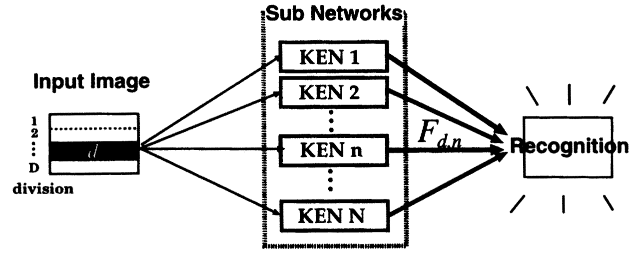


Fig. 5. Recognition process of input image.

(R, G, B) vector C_d in part d (= 1, 2, . . . , D) of the input image is determined from the average color value of that part. The output weight $F_{d,n}$ for part d of network n is determined from the inverse of the Euclidean distance between C_d and the color characteristic vector $L_{d,n}$ generated by d blocks of the image learned by each network n (= 1, 2, . . . , N). Here $L_{d,n}$ is a special quantity that represents the average RGB vector of the pixels in the d (= 1, t, . . . , D) blocks of all learned images after all n networks (n = 1, 2, . . . , N) have performed learning. This $L_{d,n}$ is stored by each network as information separate from the weights:

$$F_{d,n} = \frac{\sum_{d}^{D} \|\boldsymbol{C}_d - \boldsymbol{L}_{d,n}\|}{\|\boldsymbol{C}_d - \boldsymbol{L}_{d,n}\|} \tag{18}$$

Using this $F_{d,n}$, the final recognition result $R(x, y)$ of the input image can be calculated for each pixel as follows:

$$R(x,y) = \arg\max_k I_k \tag{19}$$

Here I_k is

$$I_k = \frac{\sum_{n}^{N} (\hat{y}_{k,n} \times F_{d,n})}{N} \tag{20}$$

and can be calculated for each pixel (x, y). k (= 1, 2, . . . , N_3) is the degree of membership of the pixel in the sky or mountain type, and $\hat{y}_{k,n}$ is the result of inference by network n.

From this $F_{d,n}$, when an untrained image is provided at the input, the recognition results of a network that has learned a similar image are important. This is because a large effect should be obtained from a small amount of image learning due to the combination of parts. The image is divided horizontally, because in scenery images, the variation in the vertical direction is larger than that in the

Table 1. Recognition labels

Sky	Woods/trees
Clouds	Grass/meadows
Distant mountains (objects)	Shadow
Water (ocean, river, other)	Snow
Rock/sand	Other, unknown

horizontal direction. The validity of this assumption is also confirmed by experiments in which the width of the membership function in the x direction after the end of learning is found to be greater than that in the y direction.

4. Experimental Results

The proposed system was implemented in software and its performance was evaluated. The 10 recognition labels used in learning/recognition are shown in Table 1 and the system parameters are shown in Table 2.

4.1. Knowledge acquisition

An example of extracted knowledge is shown in Table 3. The number in parentheses represents the number of neurons of the rule layer having that knowledge. A large value of this number represents common sense. A number marked with an asterisk represents knowledge that can also be obtained by the small rule layer (Rule Layer 2) and knowledge that has a higher commonsense value. From Table 3 it is seen that suitable knowledge has beenobtained.

Table 2. System parameters

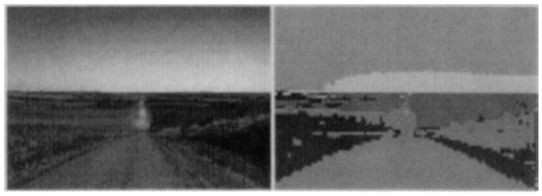
Variable	Value
Number of networks N	4
Number of horizontal image partitions D	4
Total number of learning images	25
Dimensions of input part N_1	5
Number of nodes of Rule Layer 1, N_{2-1}	1024
Number of nodes of Rule Layer 2, N_{2-2}	25
Dimensions of output part N_3	10
Number of image pixels at time of learning	40×28
Number of image pixels at time of recognition	100×70

4.2. Image recognition

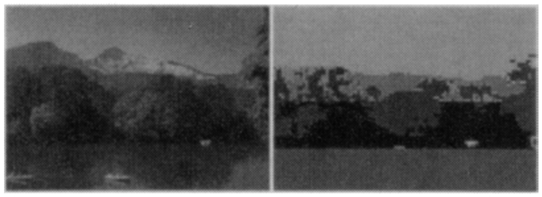
Examples of the input and recognition images are shown in Fig. 6. All images are RGB 24-bit full-color images.

Table 3. Examples of extracted knowledge

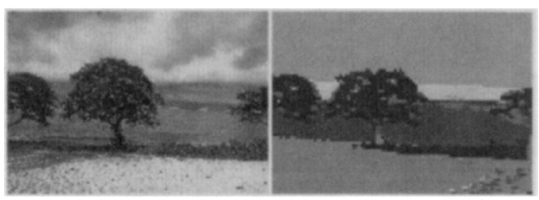
SKY
no relationship with left and right, exists in upper side and has color of blue family. (34*)
no relationship with left and right, exists in upper side and has color of bright blue family. (28)
CLOUD
on the left side, exists in upper side and has color of non-bright blue family. (8)
on the right side, exists in upper side and has color of white family. (3*)
WATER
at left–right center, exists in lower side and has color of non-bright blue family. (25*)
on left, exists in lower side and has color of blue family. (23)
MOUNTAINS or FAR-OFF OBJECTS
almost no relationship with left or right, exists at the center of upper–lower region and has color of blue family. (18*)
on right, exists in upper side and has color of dark blue family. (17)
GRASS, MEADOWS
almost no relationship with left or right, exists at lower side and has color of green family. (10*)
at right end, exists at lower side, dark color that is independent of color. (3)
FOREST, TREES
at left end, exists on upper side and has color of green family. (8*)
at right end, exists at the upper–lower center and has color of dark gray family. (5)
ROCKS, SAND
at right side, exists in lower region and has color of yellow family. (10)
at left end, exists in lower region and has color of dull orange family. (7*)



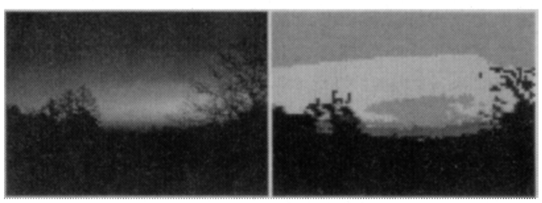
(a)
(**6094/7000**)



(b)
(**5557/7000**)



(c)
(**3284/7000**)



(d)
(**4750/7000**)

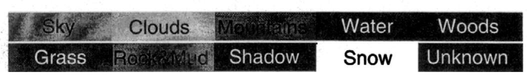


Fig. 6. Examples of input and recognition images.

All regions of all images have been well recognized even if the images were unlearned. Below each image we show the number of pixels corresponding to recognition by humans, along with the total number of pixels (7000). The scenery image contains a large number of difficult regions, even if 10 appropriate label types are used by the human. Thus, the evaluation result is not necessarily absolute, but it is presented for reference.

For example, the recognition result in Fig. 6(c) appears good, but its agreement with recognition by humans is not good. This is because the wide upper half-region may be treated as sky or clouds; it is treated as clouds by humans, but the system has recognized it as sky. In Fig. 6(d), the image is sky with sunset glow and has been well recognized. This is because there is a network that is especially designed for sunset scenes.

For reference, recognition by the system and by a human for an unlearned image are summarized in Table 4. The values in the table are the numbers of pixels corresponding to each feature; they are averages for 20 images. There is a large difference in recognition between sky and clouds, and between trees and grass, but in this region it is also difficult for the human to recognize the difference.

5. Conclusions

A method of knowledge extraction from scenery images and image recognition by means of a fuzzy inference neural network has been proposed and implemented. Experiments have shown that extraction of suitable knowledge from the image and recognition of unknown images can be effectively performed without the necessity of providing human knowledge. Many image parts are difficult for humans to recognize, and thus many problems are still unsolved. This paper proposes a new approach to image understanding and a new method for automatic extraction of linguistic knowledge and for image recognition. In addition to increasing the accuracy of recognition, future investigations will focus on layerwise knowledge and on its

Table 4. The recognition difference between human and the system (average of 20 images)

		Recognition by the system									
		Sky	Cloud	Water	Mountain	Rocks	Woods	Grass	Shadow	Snow	Others
Recognition by human	Sky	**1778**	234	0	13	91	8	8	43	27	12
	Cloud	424	**93**	0	2	13	5	0	0	17	0
	Water	1	10	**406**	20	3	21	0	4	3	0
	Mountain	185	25	1	**161**	12	103	0	4	17	14
	Rocks	31	9	8	10	**290**	102	6	59	20	14
	Woods	59	20	14	101	95	**514**	40	69	4	10
	Grass	5	2	3	7	177	236	**304**	12	10	9
	Shadow	12	3	20	24	108	86	53	**410**	4	0
	Snow	45	14	4	3	3	4	0	0	**108**	0
	Others	14	0	3	1	6	9	0	0	7	**1**

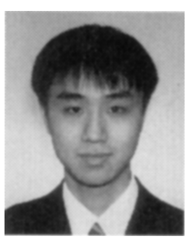
combination to achieve higher degrees of image recognition and understanding.

Acknowledgment. This research was funded in part by the Japan Technical Support Program (Project Number JSP-RFT F 97 I 00103).

REFERENCES

1. Hirata, Shirai, et al. Scene analysis using 3-dimensional information obtained from monochrome images. Trans IEICE 1992;J75-D-II:1839–1847.
2. Bruce AD, Robert TC, John B, Allen RH, Edward MR. The schame system. Int J Computer Vision 1989;2:209–250.
3. Thomas MS, Martin AF. Context-based vision: Recognizing objects using information from both 2-D and 3-D imagery. IEEE Trans Pattern Anal Syst 1991;13:1050–1065.
4. Yamane J, Sakauchi M. A construction of a new image database system which realizes fully automated image keyword extraction. IEICE Trans Inf Syst 1993;E76-D:1216–1223.
5. Sakata et al. Image partitioning by unified processing with the k-means algorithm. Trans IEICE 1997;80-D-II:2995–3003.
6. Iwada M, Nagahashi H. Dynamic region separation in color images by means of a neural network. Trans IEICE 1997;80-D-II;2995–3003.
7. Umatani et al. Partitioning of scenery images by using partial color spaces. Tech Rep IEICE 1995;PRU95-10.
8. Ghosh A, Pal NR, Pal SK. Self-organization for object extraction using a multilayer neural network and fuzziness measures. IEEE Trans Fuzzy Syst 1993;1:54–68.
9. Nogami Y, Jyo Y, Yoshioka M, Omatsu S. Remote data analysis by Kohonen feature map and competitive learning. IEEE Int Conf Syst Man Cybern 1997;1:524–529.
10. Ikeda et al. Scenery image detection by object sketch and automatic index generation. Trans IEICE 1996;79-D-II:1025–1033.
11. Shing J, Jang R. ANFIS: Adaptive-network-based fuzzy inference system. IEEE Trans SMC 1993;23:665–685.
12. Wang L-X. Training of fuzzy logic systems using nearest neighborhood clustering. Proc 2nd IEEE Int Conf on Fuzzy Systems 1993;1:93–100.
13. Nishina T, Hagiwara M. Fuzzy inference neural network. Neurocomputing 1997;14:223–239.
14. Kohonen T. The self-organizing map. Proc IEEE 1990;78:1464–1480.
15. Iyatomi H, Hagiwara M. Knowledge extraction from scenery images and the recognition using fuzzy inference neural networks. IEEE Int Conf on Systems Man and Cybernetics, 1998, San Diego, p 4486–4491.

AUTHORS (from left to right)



Hitoshi Iyatomi received his B.E. degree in electrical engineering from Keio University in 1998 and is currently an M.E. candidate. His research interests include neural networks, image recognition, and knowledge extraction.

Masafumi Hagiwara (member) received his B.E. and D.Eng. degrees in electrical engineering from Keio University in 1982 and 1987. He then joined the faculty of Keio University as a research associate and is now an associate professor there. His research interests include demodulation and synchronization using digital signal processing, and neural networks. He was the recipient of the IEICE Niwa Memorial Award in 1985 and its Shinohara Memorial Award in 1986. He received a best paper award from the IEEE Consumer Electronics Society in 1989. He is a member of IEEE, INNS, the Information Processing Society, and IEE Japan.