SOM-Based Vector Recognition with Pre-Grouping Functionality

Yuto KUROSAKI†, Masayoshi OHTA†, Nonmembers, Hidetaka ITO††, and Hiroomi HIKAWA†, Members

SUMMARY This paper discusses the effect of pre-grouping on vector classification based on the self-organizing map (SOM). The SOM is an unsupervised learning neural network, and is used to form clusters of vectors using its topology preserving nature. The use of SOMs for practical applications, however, may pose difficulties in achieving high recognition accuracy. For example, in image recognition, the accuracy is degraded due to the variation of lighting conditions. This paper considers the effect of pre-grouping of feature vectors on such types of applications. The proposed pre-grouping functionality is also based on the SOM and introduced into a new parallel configuration of the previously proposed SOM-Hebb classifiers. The overall system is implemented and applied to position identification from images obtained in indoor and outdoor settings. The system first performs the grouping of images according to the rough representation of the brightness profile of images, and then assigns each SOM-Hebb classifier in the parallel configuration to one of the groups. Recognition parameters of each classifier are tuned for the vectors belonging to its group. Comparison between the recognition systems with and without the grouping shows that the grouping can improve recognition accuracy.

key words: SOM, vector recognition, position identification, parallel classifiers

1. Introduction

Various pattern recognition systems have been developed that are of practical use, e.g., the personal identification and man-machine interaction. Pattern recognition can be defined as the categorization of input data into identifiable classes, which is a mapping process of the input vectors to a finite set of clusters each of which is associated to a class. The input image is converted to a feature vector and its class is determined by a vector classifier that searches the closest prototype of the cluster.

The self-organizing map (SOM) [1] is a form of unsupervised learning neural networks. The SOM is well known for its capability of nonlinear mapping from a given high-dimensional input space to a lower-dimensional map of neurons. One of the important features of the SOM is vector quantization embedded in the mapping process, with which resembling vectors are mapped to a single neuron. Another interesting nature of the SOM is its topology preserving nature, i.e., vectors that are neighbors in the input space will also be represented close to each other on the map. Due to these features, the SOM has been successfully used for a wide range of applications, such as information visualization and data analysis. Also, the SOM was used for various clustering and pattern recognition applications [2]–[5].

While the reliability and interpretability of the obtained mapping are foremost important for SOM-based pattern recognition, they greatly depend on careful selection of parameters such as the learning rate and the neighborhood function that are not intuitive to users [6]. Therefore, it is necessary to interpret the obtained mapping appropriately so that the final classification is made accurately. One of the effective options is to use another SOM for that task, and this kind of structure is often referred to as the hierarchical SOM [3]. Another approach is the hybrid model with the self-organizing map and a supervised neural network [4]. In a hand-sign recognition system [5], a hybrid network called the SOM-Hebb classifier was employed consisting of a SOM network and a Hebbian learning network. This classifier was implemented in a form of FPGA-based hardware. In this classifier, the Hebbian-learning network was a single-layer feed-forward network and it was used to relate the neurons in the SOM to classes.

To extend the success of the above SOM-Hebb classifier to broader applications, while retaining the advantage of the classifier such as simplicity and design reuse, which are desirable for hardware implementation, this paper introduces a functionality for pre-grouping of feature vectors. In general, image recognition is highly affected by lighting conditions. For accurate recognition, the lighting condition during recognition should be the same as that when training images were taken because the prototypes are formed by using the training images. For example, images in the same position taken in the morning and daytime have different brightness, and the weather condition should also be taken into account.

This paper thus proposes a new type of image recognition system that employs grouping and multiple SOM-Hebb classifiers. In this system, the input images are sorted into groups having similar brightness profiles, and the images in each group are classified by the SOM-Hebb classifier that has been tuned for the particular group. The SOM-Hebb classifiers are tuned to different lighting conditions to cope with the variation of the lighting conditions. For the selection of the appropriate SOM-Hebb classifier, another SOM is employed, which performs the pre-grouping of images in terms of their brightness. This choice of another SOM for the intended pre-grouping is based on the overall design simplicity as well as the preference for a method of unsu-
In terms of the use of multiple classifiers, the random forest [7] employs a similar approach, and is utilized in a wide range of applications, such as image classification and object recognitions. However, the random forest is a collective learning algorithm that develops multiple decision trees, and its result is determined by voting of the trees. On the other hand, the proposed system is based on a SOM-based pre-grouping functionality and the groupwise optimization of SOM-Hebb classifiers that form a parallel configuration.

Expanding and elaborating our preliminary report [8], this paper focuses on detailed evaluation of the SOM-based pre-grouping and the groupwise classifier optimization. The classification performance of the proposed system is examined by applying it to tasks of position identification, which identifies the position of a camera from its input image. The presented examples include outdoor as well as indoor position identification. The remainder of this paper is organized as follows: Section 2 describes the details of the proposed classifier system by applying it to the position identification application. Then the performance of the system is examined by experiments. The experimental results are given in Sect. 3, followed by conclusions in Sect. 4.

2. Grouping-Based Vector Recognition with an Application to Camera Position Identification

2.1 System Summary

The overall flow of the proposed group-based image recognition is shown in Fig. 1. This recognition system consists of an unsupervised SOM for grouping (hereafter referred to as SOM-G) and $N_G$ supervised SOM-Hebb classifiers each of which is tuned to match a specific group of input images. The grouping is performed in an unsupervised way because we here aim to perform rough clustering that does not directly correspond to the final classes. First, a low-dimensional feature vector of the input image of $P \times Q$ pixels is extracted and fed to the SOM-G. According to the mapping done by the SOM-G, the input image is forwarded to one of the groupwise classifiers, and its recognition result is used as the final recognition result. The basic configuration of each classifier is the same as used in the hand sign recognition system that was proposed in [5], and the task of the classifier consists of binarization, high-dimensional feature vector extraction, and the SOM-Hebb classification. The SOM used in each groupwise classifier is hereafter referred to as SOM-C.

The proposed group-based recognition is applied to a position identification system that identifies the camera position by finding a similar landscape with a position data. This similarity search is carried out by one of the individual vector classifiers. For example, images taken at a certain place in a cloudy day have low brightness, while images taken in a sunny day are very bright. This variation of the lighting conditions significantly affects the image recognition. Since the position identification is carried out by comparing input images with pre-memorized images called prototypes, it is desirable that the brightness of input images and that of prototype images are close enough to each other. In the proposed system, this requirement is handled by a grouping preprocess using the SOM-G.

Before proceeding into details, note that we employ several SOMs (one SOM-G and $N_G$ SOM-Cs constituting the SOM-Hebb classifiers) with basically the same structure as shown in Fig. 2 and working principles but with different sizes and vector dimensions. In a general form, each $i$-th competitive-layer neuron of a SOM (SOM-G or SOM-C) is associated with a $D$-dimensional weight vector

$$\vec{m}_i = (\mu_{i0}, \mu_{i1}, \ldots, \mu_{iD-1})$$

that serves as a cluster prototype when a $D$-dimensional input feature vector

$$\vec{x} = (\xi_0, \xi_1, \ldots, \xi_{D-1})$$

is presented to the input layer.
The input image is given in 24-bit RGB color with pixel values \( R(x, y), G(x, y), \) and \( B(x, y) \), where \( x = 0, \ldots, P - 1 \) and \( y = 0, \ldots, Q - 1 \) denote the pixel coordinates. These RGB pixel values will be processed in two distinctly different ways for extracting two types of input feature vectors for the SOM-G and the SOM-Cs, respectively. In what follows, we refer to the feature vector for the SOM-G as \( \vec{x}_G \) and that for the SOM-C as \( \vec{x}_C \), respectively.

### 2.2 The Feature Vector \( \vec{x}_G \) for Grouping

To extract the feature vector \( \vec{x}_G \) for grouping, the RGB pixel values are first converted to 256-level grayscale values \( Y(x, y) \) using the standard NTSC conversion formula

\[
Y(x, y) = 0.299 \cdot R(x, y) + 0.587 \cdot G(x, y) + 0.114 \cdot B(x, y).
\]  

(3)

An example of the obtained grayscale image is shown in Fig. 3.

At the same time, the grayscale image is partitioned into a small number (\( K \)) of sub-images as shown in Fig. 4 for \( K = 4 \). Then, within each sub-image, the grayscale values are averaged over the pixels as \( A(s)(s = 0, \ldots, K - 1) \) to obtain a rough representation of the distribution of brightness. While the values of \( A(s) \) could be used to construct the feature vector \( \vec{x}_G \), those values are further converted using the discrete Fourier transform (DFT) into frequency-domain data to increase robustness against camera positioning errors. Thus we adopt \( K \) as the dimension of the feature vector \( \vec{x}_G \), which is then constructed from the DFT amplitude spectrum (with frequency index \( n \)) as vector elements

\[
\vec{e}_{GN} = F_S(n) = \frac{\sqrt{R_S^2(n) + I_S^2(n)}}{K}
\]  

(4)

where,

\[
R_S(n) = \sum_{s=0}^{K-1} A(s) \cdot \cos\left(\frac{2\pi sn}{K}\right),
\]

(5)

\[
I_S(n) = \sum_{s=0}^{K-1} A(s) \cdot \sin\left(\frac{2\pi sn}{K}\right).
\]

(6)

### 2.3 Grouping by the Self-Organizing Map

The feature vectors constructed from (4) are grouped using the vector quantization ability of the SOM-G with \( N_G \) neurons. Operation of the SOM is divided into two phases, i.e., the learning and recall phases. In the initial learning phase, the map is trained with a set of input vectors. Once the learning phase is finished, the weights of the neurons are kept fixed and then used in the recall phase.

The learning phase starts with an appropriate initialization, in which small random numbers are assigned to the weight vectors. Subsequently, the input vectors \( \vec{x} \) (for the SOM-G, \( \vec{x} = \vec{x}_G \)) in the training set (training vectors) are fed to the map in multiple iterations. For each training vector, the distances between the training vector and all the weight vectors are calculated, and a winner neuron with index

\[
c = \arg \min_i ||\vec{x} - \vec{m}_i||
\]

(7)

is determined such that it has the smallest distance to the training vector. Here, \( || \cdot || \) denotes the Euclidean distance.

After the winner neuron is determined, the weight vectors of the winner neuron and its neighborhood neurons are updated according to

\[
\vec{m}_i(t + 1) = \vec{m}_i(t) + h_{ci} \cdot (\vec{x} - \vec{m}_i(t))
\]

(8)

so that those weight vectors become closer to the training vector as the iteration index \( t \) increases. Here, \( h_{ci} \) is the neighborhood function defined as

\[
h_{ci} = \alpha(t) \exp\left(-\frac{||\vec{p}_c - \vec{p}_i||^2}{2\sigma^2(t)}\right)
\]

(9)

where, \( \vec{p}_c \in \mathbb{R}^2 \) and \( \vec{p}_i \in \mathbb{R}^2 \) are the position vectors of the winner neuron \( c \) and neuron \( i \), respectively. Thus the magnitude of the update decreases as the distance to the winner neuron increases. The neighborhood function also includes two other factors: the learning coefficient \( \alpha(t) \) (0 < \( \alpha(t) < 1 \)) and the neighborhood radius \( \sigma(t) \), both of which are some monotonically decreasing functions of the iteration index.

The learning of the SOM is carried out by repeatedly giving the training vectors, and the feature map of the training vectors is gradually formed. In the resulting feature map, the weight vector of each winner neuron is placed in the center of the corresponding cluster of training vectors. Therefore those weight vectors can be used as prototype vectors.
Using this nature of the SOM, the vector quantization is carried out in the recall phase. During the recall phase, only the winner neuron search is conducted. Clusters of the feature vectors are formed on the basis of the rough representation of the distribution of the brightness of images. For vectors belonging to one of the clusters, a neuron whose weight vector is placed in the cluster becomes the winner. Therefore, by grouping the images according to the winner neuron, images in the same group are supposed to have similar rough landscape pattern and similar brightness. Consequently, the number of groups is equal to the number of neurons ($N_G$) of the SOM-G.

2.4 Recognition Flow within Each Groupwise Classifier

This section discusses the flow of recognition carried out for each group. The winner neuron index generated by the SOM-G is used to route the input image to one of the $N_G$ groupwise classifiers. The selected classifier then searches the class of the landscape position to which the input image belongs. The flow of the position recognition is further detailed in Fig. 5, which is basically the same as used in the hand-sign recognition system [5]. Each groupwise classifier is comprised of a preprocessor (for binarization and feature-vector construction) and a SOM-Hebb classifier (comprised of a SOM-C and a Hebbian learning network).

At this stage, the input image in the RGB color format is preprocessed to generate the feature vector $\vec{x}_C^e$. The pre-processing consists of binary quantization, and horizontal- and vertical-projection histogram calculations that are followed by two DFTs. The DFTs calculate the magnitude spectrum of the histogram data. The feature vector $\vec{x}_C^e$ is extracted from the magnitude spectrum, and is fed to the SOM-Hebb classifier that finally identifies the landscape class.

2.4.1 Binarization

In the first sub-module, the RGB pixel values are converted to the binary values $I(x, y)$ using the formula

$$ I(x, y) = \begin{cases} 255 & \text{if } k_0 - 10 < \frac{G(x, y)}{R(x, y)} \cdot 100 < k_0 + 10, \\ 0 & \text{if } k_1 - 10 < \frac{B(x, y)}{R(x, y)} \cdot 100 < k_1 + 10, \\ 0 & \text{otherwise} \end{cases} $$

(10)

where, $k_0$ and $k_1$ are the threshold parameters that specifies the characteristic ranges of the relative values of green and blue against red, respectively. An example of the binarization is shown in Fig. 6. During the learning process, the best values for $k_0$ and $k_1$ are determined for each group to maximize the recognition accuracy of the groupwise classification.

2.4.2 Horizontal and Vertical Projection Histogram

The horizontal and vertical projection histograms of $I(x, y)$ are calculated in the next sub-module. The projection is defined here as an operation that maps a binary image into a one-dimensional array called a histogram. The histogram value is the sum of pixel values along a particular direction. Horizontal projection histogram $P_H(y)$ and vertical projection histogram $P_V(x)$ are defined by

$$ P_H(y) = \sum_{x=0}^{P-1} I(x, y), $$

(11)

and

$$ P_V(x) = \sum_{y=0}^{Q-1} I(x, y). $$

(12)

Figure 7 shows examples of the horizontal and vertical projection histogram.
2.4.3 Discrete Fourier Transforms

At the final stage of preprocessing, two sets of DFT operations are performed on the projection histograms $P_H(y)$ and $P_V(x)$ to compute the magnitude spectra (with frequency index $k$)

$$F_H(k) = \frac{\sqrt{A_H^2(k) + B_H^2(k)}}{Q}, \quad (13)$$
$$F_V(k) = \frac{\sqrt{A_V^2(k) + B_V^2(k)}}{P}, \quad (14)$$

where,

$$A_H(k) = \sum_{j=0}^{Q-1} P_H(y) \cdot \cos\left(\frac{2\pi y k}{Q}\right), \quad (15)$$
$$B_H(k) = \sum_{j=0}^{Q-1} P_H(y) \cdot \sin\left(\frac{2\pi y k}{Q}\right), \quad (16)$$
$$A_V(k) = \sum_{x=0}^{L-1} P_V(x) \cdot \cos\left(\frac{2\pi x k}{P}\right), \quad (17)$$
$$B_V(k) = \sum_{x=0}^{L-1} P_V(x) \cdot \sin\left(\frac{2\pi x k}{P}\right). \quad (18)$$

Figure 8 shows the spectra $F_H(k)$ and $F_V(k)$ computed from the histograms shown in Fig. 7. As shown in this example, most of the feature information of images concentrates in lower frequency components. Thus, we adopt a relatively small (in comparison with the resolution) number $L$ as the dimension of the feature vector $\vec{x}_C$, which is then constructed from the DFT spectra as vector elements

$$\xi_{Ci} = \begin{cases} F_H(i) & (0 \leq i < L/2) \\ F_V(i - L/2) & (L/2 \leq i < L). \end{cases} \quad (19)$$

2.4.4 SOM-Hebb Classifier

The feature vectors $\vec{x}_C$ constructed from (19) are finally classified by the SOM-Hebb classifier to identify the positions where the input images were taken. As shown in Fig. 9, the SOM-Hebb classifier takes the form of a hybrid network consisting of the SOM-C with $N_C$ neurons and a single layer feedforward network with $H$ output nodes corresponding to $H$ classes. The feedforward network is a supervised network that is trained using the Hebbian learning algorithm [9]; hence it is called the Hebb network in this paper.

As previously explained in Sect. 2.3, the SOM performs vector quantization. For each input feature vector $\vec{x}_C$, one winner neuron is determined, and the input vector is mapped to the winner neuron that represents a cluster. Therefore, from the winner neuron, the class of the input vector can be identified. Considering that a single class may consist of multiple clusters, the clusters belonging to the same class must be associated in some ways. In the present system, this association is performed by the Hebb network.

During the learning phase, the training feature vectors $\vec{x}^*_C$ and the corresponding teaching signals $\tau_0, \tau_1, \ldots, \tau_{H-1}$ indicating the classes are fed to the SOM-Hebb network.

A training vector makes one of the neurons of the SOM-C the winner neuron and one of the winner information signal $w_i$ becomes ‘1’. Then the winner signal activated by the input is connected to the corresponding output node that is indicated by the teaching signal if strong synchronization is found between two signals. The output node may be connected to multiple $w_i$ signals because the class may consist of multiple vector clusters.

If there are too many neurons against the number of classes in the SOM-C, ineffective neurons that have no connections tend to be formed. Since the presence of such ineffective neurons with no connection leads to the failure of classification, those ineffective neurons are searched and culled after the training. This culling is performed by assigning dummy large values to the weight vector elements of the ineffective neurons so that they do not become winners.

3. Experiments

3.1 Indoor Position Identification

3.1.1 Input Images

Figure 10 shows the floor plan of the room where the indoor position identification experiments were performed. The input images for the experiments were captured in $P \times Q = \ldots$.
320 × 240 pixels by the camera onboard the mobile robot shown in Fig. 11. The robot traces the oval lane laid on the floor of the room and transmits the captured images to a remote computer that executes the proposed recognition task. Under various lighting conditions due to different time and weather, images were captured as the robot passed through each of ten locations numbered as 0 to 9 in Fig. 10. Hence the number of classes is \( H = 10 \).

### 3.1.2 Setup of the Experiments

The system parameters were set as follows:

- Number of groups: \( N_G = 1, 4 \)
- The dimension of the feature vector \( \mathbf{x}_G \): \( K = 4 \)
- Total number of the neurons in the parallel configuration of SOM-Cs: \( N_C \times N_C = 16, 36, 48 \)
- The dimension of the feature vector \( \mathbf{x}_C \): \( L = 12 \)
- Binarization parameters: \( k_0 \) and \( k_1 \) were searched so that the classification accuracy was maximized.

For the feature vector \( \mathbf{x}_G \), the dimension \( K \) was decided by preliminary experiments, results of which is summarized in Table 1. 550 images were taken at each position, and 350 of them (3500 in total) were used as the learning data. The remaining 200 images (2000 in total) were used to examine the classification performance of the system. To evaluate the effect of the grouping, systems with different numbers of groups were compared in terms of the classification accuracy. In each comparison pair, the total numbers of neurons in the parallel configuration of SOM-Cs were set to be identical.

### 3.1.3 Training

Using the 3500 training images, the training of the system was first carried out. The training procedure is given below:

#### Table 1 Optimization of the dimension \( K \) of the feature vector \( \mathbf{x}_G \).

<table>
<thead>
<tr>
<th>( N_G \times N_C )</th>
<th>( N_C )</th>
<th>( K )</th>
<th>Recognition rate [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 ( \times ) 2</td>
<td>4 ( \times ) 4</td>
<td>2</td>
<td>90.95</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>92.79</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>92.17</td>
</tr>
</tbody>
</table>

#### Table 2 Optimization of the binarization parameters \( k_0 \) and \( k_1 \), without grouping (\( N_G = 1, N_C = 4 \times 4 = 16 \)).

<table>
<thead>
<tr>
<th>( k_0 )</th>
<th>10</th>
<th>115</th>
<th>120</th>
<th>125</th>
<th>130</th>
<th>135</th>
<th>140</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k_0 )</td>
<td>10</td>
<td>115</td>
<td>120</td>
<td>125</td>
<td>130</td>
<td>135</td>
<td>140</td>
</tr>
<tr>
<td>( k_1 )</td>
<td>10</td>
<td>115</td>
<td>120</td>
<td>125</td>
<td>130</td>
<td>135</td>
<td>140</td>
</tr>
</tbody>
</table>

#### Table 3 Optimization of the binarization parameters \( k_0 \) and \( k_1 \), with grouping (\( N_G = 4, N_C = 2 \times 2 = 4 \)).

<table>
<thead>
<tr>
<th>( k_0 )</th>
<th>10</th>
<th>115</th>
<th>120</th>
<th>125</th>
<th>130</th>
<th>135</th>
<th>140</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k_0 )</td>
<td>10</td>
<td>115</td>
<td>120</td>
<td>125</td>
<td>130</td>
<td>135</td>
<td>140</td>
</tr>
<tr>
<td>( k_1 )</td>
<td>10</td>
<td>115</td>
<td>120</td>
<td>125</td>
<td>130</td>
<td>135</td>
<td>140</td>
</tr>
</tbody>
</table>

#### Table 4 Optimization of the binarization parameters \( k_0 \) and \( k_1 \), with grouping (\( N_G = 4, N_C = 2 \times 2 = 4 \)).

<table>
<thead>
<tr>
<th>( k_0 )</th>
<th>10</th>
<th>115</th>
<th>120</th>
<th>125</th>
<th>130</th>
<th>135</th>
<th>140</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k_0 )</td>
<td>10</td>
<td>115</td>
<td>120</td>
<td>125</td>
<td>130</td>
<td>135</td>
<td>140</td>
</tr>
<tr>
<td>( k_1 )</td>
<td>10</td>
<td>115</td>
<td>120</td>
<td>125</td>
<td>130</td>
<td>135</td>
<td>140</td>
</tr>
</tbody>
</table>

### 3.2 Floor plan for the indoor position identification experiments.

![Fig. 10](image1.png)

### 3.3.1 Mobile robot.

![Fig. 11](image2.png)
Table 4  Summary of the indoor recognition results.

<table>
<thead>
<tr>
<th>$N_G \times N_C$</th>
<th>$N_G$</th>
<th>$N_C$</th>
<th>$k_0$</th>
<th>$k_1$</th>
<th>$M_G$</th>
<th>$M_R$</th>
<th>Individual group recognition rate [%] $(100 \times M_R/M_G)$</th>
<th>Recognition rate [%] $(100\sum M_R/\sum M_G)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>1</td>
<td>4 x 4</td>
<td>60</td>
<td>115</td>
<td>2000</td>
<td>1709</td>
<td>85.45</td>
<td>85.45</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>94.95</td>
</tr>
<tr>
<td>36</td>
<td>1</td>
<td>6 x 6</td>
<td>60</td>
<td>110</td>
<td>2000</td>
<td>1926</td>
<td>96.30</td>
<td>96.30</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>99.20</td>
</tr>
<tr>
<td>48</td>
<td>1</td>
<td>6 x 8</td>
<td>60</td>
<td>110</td>
<td>2000</td>
<td>1939</td>
<td>96.95</td>
<td>96.95</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>99.30</td>
</tr>
</tbody>
</table>

(1) The SOM-G for the grouping was trained.

(2) Using the SOM-G, the training images were grouped into $N_G$ groups.

(3) Each SOM-Hebb classifier was trained using the above grouped training images.

(4) The recognition rate (for the training images) of each classifier was evaluated.

(5) For various values of $k_0$ and $k_1$, (2) ~ (4) were repeated to find the best pair of $k_0$ and $k_1$.

Table 2 shows the recognition rates of the system for $N_G = 1$ and $N_C = 4 \times 4 = 16$, i.e., without grouping. In this table, the recognition rates are shown for the pairs of various values of $k_0$ and $k_1$. Numbers in bold indicate the best pair of $k_0$ and $k_1$, and the corresponding recognition rate. This result shows that $k_0 = 60$ and $k_1 = 115$ provided the best performance. Likewise, the recognition rates of the system with grouping for $N_G = 4$ and $N_C = 2 \times 2 = 4$ are shown in Table 3. Note that total number of the neurons in the parallel configuration of SOM-Cs is set to the same as that in the system without grouping. Using the same procedure, the best pairs of $k_0$ and $k_1$ for the systems with 36 and 48 ($= N_G \times N_C$) neurons were obtained, and using the best parameters all classifiers were configured.

3.1.4 Evaluation of the Recognition Rate

Experimental results of the systems with different number of groups are summarized in Table 4 that shows the recognition rates for systems with $N_G = 1$ (without grouping) and $N_G = 4$ (with grouping). The table summarizes the recognition rates for the individual groups as well as the overall recognition rate. $M_G$ is the number of test data that were assigned to each group, and $M_R$ is the number of test data that were correctly recognized.

The results in the table show that the system with grouping provides better recognition accuracy compared to that of the system without the grouping. Comparison between the two systems with the same total number of neurons shows that the grouping method provided the better performance. In the case of 16 neurons, the recognition accuracy was improved from 85.45% without grouping to 94.95% with grouping. Likewise, for 36 neurons, the recognition rate was improved from 96.30% without grouping to...
Table 5  Summary of the outdoor recognition results.

<table>
<thead>
<tr>
<th>$N_G \times N_C$</th>
<th>$N_G$</th>
<th>$N_C$</th>
<th>$k_0$</th>
<th>$k_1$</th>
<th>$M_G$</th>
<th>$M_R$</th>
<th>Individual group recognition rate [%] ($100 \times M_R/M_G$)</th>
<th>Recognition rate [%] ($100 \sum M_G/\sum M_C$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>144</td>
<td>1</td>
<td>12 × 12</td>
<td>100</td>
<td>100</td>
<td>3000</td>
<td>2306</td>
<td>79.87</td>
<td>97.97</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>6 × 6</td>
<td>95</td>
<td>100</td>
<td>539</td>
<td>533</td>
<td>98.89</td>
<td>90.40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6 × 6</td>
<td>100</td>
<td>105</td>
<td>895</td>
<td>741</td>
<td>82.79</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6 × 6</td>
<td>100</td>
<td>105</td>
<td>729</td>
<td>603</td>
<td>82.72</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>14 × 14</td>
<td>105</td>
<td>105</td>
<td>3000</td>
<td>2817</td>
<td>93.90</td>
<td>93.90</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>7 × 7</td>
<td>100</td>
<td>110</td>
<td>539</td>
<td>538</td>
<td>99.81</td>
<td>95.60</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7 × 7</td>
<td>95</td>
<td>95</td>
<td>895</td>
<td>812</td>
<td>90.73</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>7 × 7</td>
<td>100</td>
<td>105</td>
<td>729</td>
<td>682</td>
<td>93.55</td>
<td></td>
</tr>
</tbody>
</table>

99.20% with grouping. For 48 neurons, the recognition rate was improved from 96.95% to 99.30%. With grouping, the binarization parameters $k_0, k_1$ and the weight vectors of the SOM-C can be tuned for individual groups having similar rough landscape pattern and similar brightness. This makes the classification task within each group significantly easier, and all of the groupwise recognition rates shown in Table 4 exceeds the recognition rate attained by a single large SOM-C.

3.2 Outdoor Position Identification

To further support the effectiveness of the proposed system, we extend our performance evaluation to the case of outdoor position identification. This case is more difficult than the previous one due to severe variation in lighting and weather conditions, and also due to moving objects such as people and cars.

Figure 12 shows the 20 positions (classes) that were to be classified in the following experiments. A typical variation of images that belong to a single class is illustrated in Fig. 13. The prepared data set was comprised of 300 images for each class, totaling in 6000 images, half of which were used for learning and the other half for recognition. Each image was a 360 degree panoramic image of $740 \times 238$ pixels, so that the images were shift invariant under the present feature vector construction. The dimensions of the feature vectors $x_G$ and $x_C$ were set to $K = 8$ and $L = 30$, respectively. The total number of the neurons in the parallel configuration of SOM-Cs was set to $N_G \times N_C = 144, 196$.

Recognition results are summarized in Table 5. In the case of 144 neurons, the recognition rate was significantly improved from 79.87% without grouping to 90.40% with grouping. For 196 neurons, the recognition rate was also improved from 93.90% to 95.60%. Thus the grouping is effective also for outdoor position identification. In addition, the grouping allows early rise in the recognition rate for relatively small numbers of total neurons, which is highly suited for small hardware and/or more complex tasks.

4. Conclusion

This paper has discussed the effect of pre-grouping of feature vectors on SOM-based vector recognition, with an application to image classification. The proposed pre-grouping functionality, also based on the SOM, was introduced into a new parallel configuration of the SOM-Hebb classifiers, retaining their advantages such as simplicity and easy design reuse. The overall system was applied to outdoor as well as indoor position identification from images. The system first performed the grouping of images according to the rough representation of the brightness profile of images, and then assigned each SOM-Hebb classifier in the parallel configuration to one of the groups. Recognition parameters of each classifier were then tuned for the vectors belonging to its group. Comparison between the recognition systems with and without the grouping showed that the grouping can improve the recognition accuracy.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP16K00344.

References


Yuto Kurosaki received the B.E. degree in electrical and electronic engineering from Kansai University, Japan, in 2016.

Masayoshi Ohta received the B.E. and M.E. degrees in electrical and electronic engineering from Kansai University, Japan, in 2015 and 2017, respectively.

Hidetaka Ito received the B.E., M.E., and Ph.D. degrees in electrical engineering from Kyoto University, Japan, in 1988, 1991, and 1996, respectively, and the MSc degree in satellite engineering from the University of Surrey, UK, in 1991. Since 1994, he has been with Kansai University, where he is currently an associate professor. His research interests include nonlinear dynamics and neural networks.

Hiroomi Hikawa received the B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University, Japan, in 1984, 1986, and 1989, respectively. He then served as a post-doctoral researcher at University of South Florida. From 1992 to 2008, he worked at the Computer Science and Intelligent Systems Department of Oita University. In 2008, he joined Department of Electrical and Electronic Engineering of Kansai University where he is a Professor. His research interests include architecture for signal processing and neural networks.