SUMMARY In this paper, we propose block matching and learning for color image classification. In our method, training images are partitioned into small blocks. Given a test image, it is also partitioned into small blocks, and mean-blocks corresponding to each test block are calculated with neighbor training blocks. Our method classifies a test image into the class that has the shortest total sum of distances between mean blocks and test ones. We also propose a learning method for reducing memory requirement. Experimental results show that our classification outperforms other classifiers such as support vector machine with bag of keypoints.

key words: color image, block matching, learning vector quantization

1. Introduction

Many researchers have put their efforts into research on object recognition, which can be regarded as a basic problem of artificial intelligence. However, we do not know effective recognition schemes yet for general objects such as buses or phones, since such schemes are now limited to the face and handwritten characters. For example, in the recognition task on the benchmark dataset called TRECVID [1], the state-of-the-art system resulted in poor recognition even if using not only visual information but also sounds and closed captions. In contrast, human beings can distinguish tens of thousands of classes with only visual information, so recognition abilities of machines are inferior to those of human beings.

For achieving object recognition with machines, some significant approaches have recently been proposed. For example, in the recognition task on the benchmark image dataset called Caltech-101 [2], it has been verified by experiment that a combination of Support Vector Machine (SVM) and local feature called bag of keypoints [3], [4] is effective for object recognition [3], [4]. However, it is difficult to classify color images such as landscapes by them because effective local features cannot be extracted from such images. Moreover, learning in SVM will not be finished when the number of training samples and that of classes are both large. This drawback will be a crucial problem because the sizes of training samples and classes in object recognition have a tendency to increase. As another example, Duygulu et al. proposed a classification method that segments images into subregions by using clustering and assigns multi-labels for these regions for classification [5]. However, this approach has already been out of use because it is sensitive to clustering results.

In this paper, we propose block matching and learning for color image classification. We assume here each image has only one class label. In a preprocessing phase, all training images are partitioned into small block images (cf. left side in Fig. 1). Given a test image, it is also partitioned into small blocks. After block partition, k neighbor training blocks corresponding to each test block are selected in each class (cf. middle in Fig. 1). In a classification phase, mean blocks are calculated with selected neighbor blocks in each class, and the total sum of distances between mean blocks and corresponding test ones are measured in individual classes. This total sum of distances can be regarded as the distance between a test image and images reconstructed by mean blocks (cf. right side in Fig. 1). Finally, a test image is classified into the class that has the shortest total sum of distances. SVM and clustering are not used in our classification, so we can overcome the difficulties found in the conventional approaches. However, the number of blocks tends to be large in our method, so we propose a learning rule based on generalized learning vector quantization (GLVQ) [6] for block reduction. The performance of our method is verified with experiments on the WANG color image dataset [7].

2. Proposed Color Image Classification

In block-based image classification, making use of the total sum of the distances between a test block and its nearest training one is the simplest. However, the distance between images suffers sensitivity to geometric transformations (shifts, rotations and others) due to high-dimensionality of vectors. For overcoming this difficulty, it is useful to apply local mean classifier [8] because mean vectors help to reduce influence of geometric transformations. In fact, local mean classifier outperformed the nearest neighbor rule in image classification such as handwritten character recognition [9]. Hence, such generalization capability of local mean classifier also may give high accuracy to our image classification.

Let \( X_i \) (\( i = 1, \ldots, N \)) and \( y_i \) be the \( i \)th arbitrary size training image and its class label \( y_i \in \{ 1, \ldots, C \} \), respectively. Note that \( N \) and \( C \) indicate the number of training images and classes, respectively. When the sizes of images are too diverse, resize images into the same size. According to our experiments, such resizing has little effect on accuracy. In a preprocessing phase, all training images are par-
Let \( N \) be this partitioning. Let \( nq \) be the number of \( n_b \) blocks images called training block images by this partitioning from all the training images. In this paper, the \( i \)th training block is denoted by \( x_i \) (\( i = 1, \ldots, n_b \)) that is represented by a vector of which elements are pixel values of the \( i \)th training block. In the case of color images, the dimensionality of \( x_i \) is \( m \times n \times 3 \). Of course, \( x_i \) obtained from \( X_i \) has the same class label of \( X_i \), i.e., \( y_i \).

When an arbitrary size test image \( Q \) is given, partition it into block images called test blocks of which sizes are \( m \times n \) pixels. Assume \( n_q \) test blocks are obtained from \( Q \) by this partitioning. Let \( q_i \) (\( i = 1, \ldots, n_q \)) be the vector of which elements are pixel values of the \( i \)th test block of \( Q \). Let \( N_i^{(j)} = \{x_i \in \text{class } j \mid k\text{-neighbor training blocks of } q_i \} \) be the set of \( k \) neighbor training blocks of \( q_i \) which are selected from class \( j \) using Euclidean distance \( d(q_i, x_i) = \|q_i - x_i\|^2 \).

After this selection, the mean block of the selected neighbor training blocks in class \( j \) is calculated by

\[
m_{i}^{(j)} = \frac{1}{k} \sum_{x_i \in N_i^{(j)}} x_i. \tag{1}
\]

We calculate \( n_q \) mean blocks \( m_1^{(j)}, \ldots, m_n^{(j)} \) corresponding to each test block \( q_i \) in all classes. After that, the total sum of distances between \( q_i \) and \( m_i^{(j)} \) in class \( j \) is calculated by

\[
D_j = \sum_{i=1}^{n_q} d(q_i, m_i^{(j)}).
\tag{2}
\]

The above \( D_j \) is equivalent to the distance between a test image and an image reconstructed by mean blocks in class \( j \).

In a classification phase, a test image \( Q \) is classified into the class that has the minimum \( D_j \), i.e., the class of \( Q \) (denoted as \( \omega \)) is determined as follows:

\[
\omega = \arg \min_{j=1, \ldots, C} D_j. \tag{3}
\]

3. Block Reduction with Learning

In our method, a large amount of blocks is required, so we have to reduce the number of blocks. For reducing blocks, we can apply clustering such as \( k \)-means to blocks. However, reduction with unsupervised procedures might result in poor recognition performance. Hence, we apply supervised reduction to our classification for achieving block reduction. In this paper, we focus on GLVQ [6] as supervised block reduction. In GLVQ, prototypes called codebooks (reference vectors) are updated by a steepest descent method that minimizes a cost function defined by the distance between a training sample and the nearest codebook [6]. However, we cannot apply GLVQ to our method directly because our
classification rule is defined with the total sum of distances between mean blocks and test ones. Hence, we change the cost function of GLVQ for our classification, i.e., we define the cost function by using the total sum of distances between the set of blocks. Consequently, the update rule of our learning is different from the original one, i.e., all neighbor codebooks are updated into optimal positions for improving accuracy.

Let \( \mathbf{c}_m^{(j)} \) be the \( m \)th vector called codebook block belonging to class \( j \) \((j = 1, \ldots, C)\). In our experiment, \( \mathbf{c}_m^{(j)} \) is initialized randomly by using a \( x_t \in \text{class } j \). Hence, the dimensionality of \( \mathbf{c}_m^{(j)} \) is the same as those of training blocks. Of course, the total number of codebook blocks is smaller than that of training blocks.

When a training image \( \mathbf{X}_i \) is given, codebook blocks are updated based on the total sum of distances between a training block from \( \mathbf{X} \) and its corresponding mean block of \( k \)-neighbor codebook blocks. Let \( \mathbf{x}_i^{(l)} \) \((l = 1, \ldots, n_i)\) be the \( l \)th training block obtained from \( \mathbf{X}_i \), where \( n_i \) is the number of training blocks from \( \mathbf{X}_i \). In practice, when a training image \( \mathbf{X}_i \) is given, \( k \)-neighbor codebook blocks corresponding to each training block \( \mathbf{x}_i^{(l)} \) are selected in individual classes. Let \( N^{(j)}_i = \{ \mathbf{c}_m^{(j)} \mid k \text{-neighbor codebook blocks of } \mathbf{x}_i^{(l)} \} \) be the set of \( k \)-neighbor codebook blocks of \( \mathbf{x}_i^{(l)} \) which are selected from class \( j \) using Euclidean distance. After this selection, the mean block of the selected codebook blocks of class \( j \) corresponding to \( \mathbf{x}_i^{(l)} \) is calculated by

\[
\mathbf{m}_i^{(j)} = \frac{1}{k} \sum_{\mathbf{c}_m^{(j)} \in N^{(j)}_i} \mathbf{c}_m^{(j)}.
\]  

(4)

We calculate \( n_i \) mean blocks \( \mathbf{m}_i^{(1)}, \ldots, \mathbf{m}_i^{(C)} \) corresponding to each training block \( \mathbf{x}_i^{(l)} \) in all classes. After that, the total sum of distances between \( \mathbf{x}_i^{(l)} \) and \( \mathbf{m}_i^{(j)} \) is calculated by

\[
D_j = \sum_{i=1}^{n_i} d(\mathbf{x}_i^{(l)}, \mathbf{m}_i^{(j)}).
\]  

(5)

In a training phase, codebook blocks are updated according to the above distance.

Let \( \mathbf{m}_i^{(1)} \) be the mean block corresponding to \( \mathbf{x}_i^{(l)} \) obtained from the same class of \( \mathbf{X}_i \). In contrast, \( \mathbf{m}_i^{(2)} \) is the mean block corresponding to \( \mathbf{x}_i^{(l)} \) obtained from the nearest different class from \( \mathbf{X}_i \). In this paper, we define the relative distance difference \( \mu(\mathbf{X}_i) \) as follows:

\[
\mu(\mathbf{X}_i) = \frac{\bar{D}_y - \bar{D}_z}{\bar{D}_y + \bar{D}_z}.
\]  

(6)

The above \( \mu(\mathbf{X}_i) \) satisfies \(-1 < \mu(\mathbf{X}_i) < 1\). If \( \mu(\mathbf{X}_i) \) is negative, \( \mathbf{X}_i \) is classified correctly; otherwise, \( \mathbf{X}_i \) is misclassified. As well as [6], we should minimize the following cost function for improving accuracy:

\[
S = \sum_{i=1}^{N} f(\mu(\mathbf{X}_i)),
\]  

(7)

where \( f(\mu) \) is a nonlinear monotonically increasing function. To minimize \( S \), a steepest descent method with a small positive constant \( \alpha (0 < \alpha < 1) \) is adopted to Eq. (7) with respect to \( \mathbf{c}_m^{(j)} \) \((j = 1, \ldots, C)\):

\[
\mathbf{c}_m^{(j)} \leftarrow \mathbf{c}_m^{(j)} - \alpha \frac{\partial S}{\partial \mathbf{c}_m^{(j)}}, \quad (\mathbf{c}_m^{(j)} \in N^{(j)}_i).
\]  

(8)

When the square of Euclidean distance \( d(\mathbf{x}_i^{(l)}, \mathbf{m}_i^{(j)}) = \|\mathbf{x}_i^{(l)} - \mathbf{m}_i^{(j)}\|^2 \) is used, the following update rule is given:

\[
\mathbf{c}_m^{(j)} \leftarrow \mathbf{c}_m^{(j)} + \alpha \frac{\partial f}{\partial \mu} \frac{\bar{D}_y}{\bar{D}_y + \bar{D}_z}(\mathbf{x}_i^{(l)} - \mathbf{m}_i^{(j)}),
\]  

(9)

\[
\mathbf{c}_m^{(z)} \leftarrow \mathbf{c}_m^{(z)} - \alpha \frac{\partial f}{\partial \mu} \frac{\bar{D}_y}{\bar{D}_y + \bar{D}_z}(\mathbf{x}_i^{(l)} - \mathbf{m}_i^{(z)}).
\]  

(10)

In this paper, \( f(\mu, t)(1 - f(\mu, t)) \) is used for \( \partial f/\partial \mu \), where \( t \) and \( f(\mu, t) \) are learning time and a sigmoid function \( 1/(1 - e^{-t^2}) \), respectively.

As shown in the above equations, \( \mathbf{c}_m^{(y)} \) and \( \mathbf{c}_m^{(z)} \) are updated by attractive and repulsive forces from \( \mathbf{x}_i^{(l)} \), respectively. As mentioned in [6], the convergence condition of GLVQ depends on the relation of these forces, i.e., GLVQ converges when attractive forces are larger than repulsive ones. This relation of forces is dependent on the definition of \( \mu \), and we can confirm Eq. (6) satisfies this condition as well as the original GLVQ algorithm. Hence, it can be expected that our learning converges to a good solution.

After training, a test image \( \mathbf{Q} \) is classified by the same classification rule using trained codebook blocks instead of making use of the original training ones.

4. Experimental Results

We tested our method on a small image dataset called the WANG database [7], formed by 10 image classes, African people, beach, buildings, buses, dinosaurs, elephants, flowers, horses, mountains, and food. Each class consists of 100 images of which sizes are 80 × 120 or 120 × 80 pixels. Thus, the total number of images is 1000. In experiments, these images were partitioned into block images of which sizes were 5 × 5, 10 × 10, and 20 × 20 pixels.

4.1 Classification Performance

We investigated classification performance of our method. Recognition rates were estimated by 10 fold cross validation with varying the number of neighbors \( k \) (from \( k = 1 \) to \( k = 30 \)) and block sizes. We compared our method with each classifier: bag-of-keypoints with SVM, color histogram features with \( k \)-nearest neighbor rule (kNN) and SVM. For bag-of-keypoints, SIFT features were extracted from gray-scale images and transformed them into 80 visual words. As color feature vectors, RGB color histograms with a size of \( 4^3 = 64 \) bins were extracted from each image. For SVM, the Gaussian kernel \( K(x, y) = e^{-\gamma \|x-y\|^2} \) was used for nonlinear mapping. Parameter values of each classifier were
optimized by validation. In this experiment, block reduction was not applied to our classification.

The recognition rate of each method, its standard deviation, and optimized parameters using validation are summarized in Table 1. This table indicates that color information is important for classification in the WANG dataset. However, our method outperformed other classifiers that used only color histograms because our method adopted not only color information but also spatio-information to classification. In addition, the recognition rates obtained with $2 \leq k \leq 30$ were higher than those of $k = 1$ in all block sizes. Hence, the use of local mean classifier was effective for improving accuracy.

4.2 Effectiveness of Block Reduction

We evaluated the effectiveness of block reduction described in Sect. 3. In this experiment, we selected 10 images per class randomly for test images. Consequently, the total number of test images was 100. The leftover 900 images were used as training images. This random splitting was performed independently for ten repetitions. The parameter $\alpha = 0.01$ was used for a learning rate, and we selected the number of neighbors $k$ as shown in Table 1. Figure 2 shows the mean training recognition rate with respect to the number of iteration. This result was obtained with 100 codebook blocks of which sizes are $5 \times 5$ pixels per class. In other words, the number of blocks was reduced from 38400 to 100 per class by learning. As shown in Fig. 2, training recognition rate increased almost monotonically as the number of iteration increased. Test recognition rates obtained with various block sizes are shown in Table 2. From this, test recognition rates obtained with learning were as well or better than the cases of the use of all blocks (cf. Tabel 1). Hence, we can reduce the number of blocks by our learning.

5. Conclusion

In this paper, we proposed block matching and learning for color image classification. Our method uses block images instead of extracting local features, so it can classify color images such as landscapes that tend to be misclassified by SVM. Future work will be dedicated to applying our method to a large-scale dataset such as Caltech-101 [2].

Acknowledgement

This research was supported by The Ministry of Education, Culture, Sports, Science and Technology in Japan under a Grant-in-Aid for Division of Young Researchers.

References