9

Content-based Image Classification via Visual Learning

Hiroki Nomiya[1] and Kuniaki Uehara[2]
[1]*Graduate School of Science and Technology, Kobe University*
[2]*Graduate School of Engineering, Kobe University*
Japan

1. Introduction

As the information processing ability of computers improves, real-world images are increasingly used in various data mining applications. Thus, flexible and accurate image recognition methods are strongly needed. However, real-world images generally contain a wide variety of objects which have complex features (e.g., shapes and textures). Therefore, the accurate recognition of real-world objects is difficult because of three main problems. Firstly, although an image is generally given as a set of pixels, pixels alone are insufficient for the description and recognition of complex objects. Thus, we must construct more discriminative features from pixels. Secondly, finding useful features to describe complex objects is problematic because appropriate features are dependent on the objects to be recognized. Thirdly, real-world images often contain considerable amounts of noise, which can make accurate recognition quite difficult. Because of these problems, the recognition performance of current recognition systems is far from adequate compared with human visual ability.

In order to solve these problems and facilitate the acquisition of a level of recognition ability comparable to that of human visual systems, one effective method consists of introducing learning schemes into image understanding frameworks. Based on this idea, *visual learning* has been proposed (Krawiec & Bhanu, 2003). Visual learning is a learning framework which can autonomously acquire the knowledge needed to recognize images using machine learning frameworks. In visual learning, given images are statistically or logically analyzed and recognition models are constructed in order to recognize unknown images correctly for given recognition tasks. Visual learning attempts to emulate the ability of human beings to acquire excellent visual recognition ability through observing various objects and identifying several features by which to discriminate them.

The key to the development of an efficient visual learning model resides in features and learning models. Image data contain various types of informative features such as color, texture, contour, edge, spatial frequency, and so on. However, these features are not explicitly specified in input image data. Therefore, *feature construction* is needed. Feature construction is the process of constructing higher-level features by integrating multiple lower-level (primitive) features. Appropriate feature construction will greatly contribute to recognition performance. In addition, since useful features depend on the given image data,

Source: Data Mining and Knowledge Discovery in Real Life Applications, Book edited by: Julio Ponce and Adem Karahoca,
ISBN 978-3-902613-53-0, pp. 438, February 2009, I-Tech, Vienna, Austria

feature selection is also needed to determine appropriate features according to the given image data. Thus, we propose both feature construction and feature selection methods to develop a better visual learning framework.
In order to utilize features efficiently, it is necessary to develop an appropriate visual learning model. In most existing visual learning models, a single learner based on a single learning algorithm is trained using input image data. However, this learning model is inefficient compared with human visual models. Human visual systems can be divided into multiple modules, accomplishing excellent visual ability through the cooperation of these modules (Marr 1982). In other words, introducing modularity into visual learning framework leads to improved recognition performance. To introduce modularity, we adopt an *ensemble approach*, a kind of learning approach in which multiple learners called base learners (they correspond to modules) are simultaneously trained and their learning results integrated into a single hypothesis. Based on this ensemble approach, we develop a novel visual learning model in which multiple base learners can be trained through cooperation with each other. Through the introduction of cooperation among multiple learners, recognition accuracy can be considerably improved. The learning strategy of the proposed visual learning model enables more flexible and accurate recognition applicable to a wide variety of data mining tasks using various types of visual data. We verify the flexibility and recognition performance of our method through an object recognition experiment using real-world image data, and a facial expression recognition experiment using video data.

2. Visual learning

At the beginning of computer vision and image understanding research, the recognition process was human-intensive. That is, a large part of domain knowledge was defined and given by hand. As the amount and complexity of image data increase, however, conventional image recognition methods have difficulty in recognizing real-world objects for several reasons. First, it is quite difficult to provide sufficient domain-specific knowledge manually due to the complexity of large-scale real-world recognition problems. Next, although the recognition performance of conventional methods is sufficient for limited domains, such as facial recognition and hand-written character recognition, these methods have great difficulty in effecting a flexible recognition that can discriminate a wide variety of real-world objects. Finally, the recognition performance of these methods tends to be affected by noisy images which contain cluttered backgrounds, bad lighting conditions, occluded or deformed objects, and so on. These problems must be solved in order to develop a recognition framework comparable to human visual systems.
The first problem can be solved by the framework of visual learning. Visual learning autonomously acquires domain-specific knowledge in order to recognize images. This knowledge is derived by machine learning frameworks which statistically or logically analyze input image data and construct recognition models to recognize unknown images correctly for the given recognition task. Most machine learning frameworks can be easily applied to a wide variety of recognition problems and can provide domain-specific knowledge.
Although machine learning frameworks automatically derive domain-specific knowledge, however, they often have difficulty in acquiring the knowledge because of the second problem caused by the data structure of image data. That is, an image has various features which are useful for the discrimination of objects in the image: for example, color

histograms and spatial frequency are widely used to describe images. However, these features are not explicitly specified in input data because an input image is usually given only as a set of pixels. Generally, since an image consists of a large number of pixels, the input images contain a large amount of irrelevant or redundant data. To solve this issue, feature construction is required to construct more informative features from the given image data. For instance, color histograms can be constructed by analyzing the distribution of the intensity values of the given pixels. Since the learning efficiency and performance depend on the features input into the machine learning algorithm, the feature construction method has a great influence on recognition performance. In other words, constructing appropriate features leads to an accurate recognition which is able to solve the second problem. For real-world image recognition, a crucial problem stems from the large variety of objects to be recognized. The problem is that appropriate features are generally dependent on the given object. Therefore, it is essential for flexible recognition to develop an efficient feature selection method to select appropriate features according to the given object.

As for the third problem, noisy images, which contain occlusion, deformation, or bad lighting conditions, often worsen the learning performance. To reduce the influence of these noisy images, some kind of image preprocessing such as image filtering is frequently used (Krawiec & Bhanu, 2003). However, the elimination of any type of noise using image preprocessing is extremely difficult. Thus, we attempt to deal with noisy images by developing a noise-robust learning model based on the ensemble approach. In the learning model, interaction among multiple base learners provides robustness to noise by detecting and eliminating noisy images. We show an example of the learning model in Fig. 1.

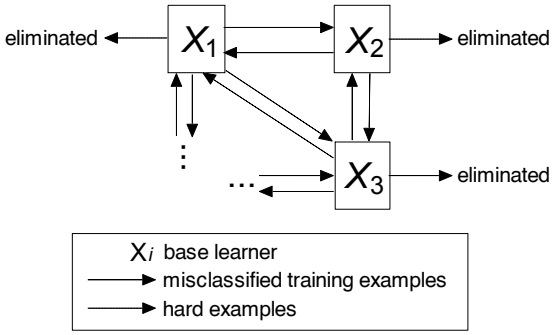


Fig. 1. The learning model with the collaboration of multiple base learners.

In this learning model, an arbitrary number of base learners are collaboratively trained. Specifically, each base learner X_i (i = 1, 2, … , n, where n is the number of base learners) sends its misclassified training examples to the other base learners $\{X_1,\cdots X_{i-1},X_{i+1},\cdots,X_n\}$. The examples misclassified by X_i are eliminated from its training set. Similarly, the other base learners send their misclassified training examples to X_i and eliminate the examples from their own training sets. X_i is then trained using the examples sent from the other base learners.

The examples that are misclassified by all of the base learners are regarded as *hard examples* and are eliminated from the training sets of all base learners. Hard examples mean the examples which are very difficult to classify correctly; thus, noisy images correspond to

hard examples. Hard examples should be eliminated because they cause overfitting. Overfitting is the phenomenon in which base learners become too specialized through the recognition of hard examples, so that they often fail to recognize non-hard examples, which are the majority of given examples. Since resolving the problem of overfitting is sometimes crucial for ensemble learning, this learning model considerably reduces the influence of noisy images and improves recognition accuracy.

3. Collaborative visual learning

Since the key point of the learning model mentioned in the previous section is the collaboration of multiple base learners, we call this model *collaborative ensemble learning*. Its learning framework is based on a boosting algorithm (Freund & Schapire, 1997). Each example (i.e., image) is assigned a weight that measures the difficulty of correctly classifying the example, and each base learner is iteratively trained using these weights. The iteration step is called a round. The weights of all examples are updated at the end of each round to assess their classification difficulty. A higher weight means that the example is more difficult to classify correctly. In the machine learning domain, it has been proven that training a base learner using examples with high weights improves the classification performance of the base learner (Dietterich, 2000). At the end of each round, a base learner generates a hypothesis to classify unseen examples. Through the learning process, multiple hypotheses are generated and are ultimately integrated into a final hypothesis. The integration is performed by, for example, voting by multiple hypotheses. The final hypothesis corresponds to the prediction of an ensemble classifier and generally has much better classification performance than a hypothesis by a single base learner.

3.1 A weighting algorithm to detect hard examples

To describe the collaborative ensemble learning model, we first formulate an object recognition task as a classification problem. The training set S is represented as $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, where m is the number of training examples. x_i and $y_i \in \{1, \ldots, C\}$ correspond to an image and a class label respectively. C is the number of classes—that is, the recognition problem is to distinguish C kinds of objects.

Our learning algorithm is based on AdaBoost, which has the crucial problem of susceptibility to hard examples. This problem stems from the fact that AdaBoost gives excessively high weights to hard examples, so that overfitting tends to occur. In order to prevent overfitting, we improve the weighting algorithm that determines weights so that the weights of hard examples are appropriately controlled. We propose a weighting algorithm based on the following two points: (1) To prevent overfitting, when an example is regarded as a hard example by a base learner X, that example should not be used as a training example for X; and (2) To train all base learners collaboratively, when X misclassifies a training example, its weight for other base learners should be increased so that the example is used to train the other base learners. We thus define the weight distribution for our weighting algorithm. The weight distribution represents the importance of each example. The examples which have high weight distribution values are useful to improve the classification performance of base learners. The weight distribution $D_{i,t}^{l}$ of the *i*-th example x_i for the *l*-th classifier X^l at the *t*-th round is calculated as follows:

$$D_{i,t}^{l} = \frac{\delta_{i,t}^{l} d_{i,t}^{l}}{\sum_{i=1}^{m} \delta_{i,t}^{l} d_{i,t}^{l}}, \quad (1)$$

where $\delta_{i,t}^{l}$ is 0 if x_i is regarded as a hard example by X^l (i.e., the weight of x_i becomes higher than the threshold[1]); otherwise $\delta_{i,t}^{l}$ is 1, and

$$d_{i,t}^{l} = \frac{1}{n-1} \sum_{j \neq l}^{n} \left(\frac{w_{i,t}^{j}}{\sum_{i=1}^{m} w_{i,t}^{j}} \right), \quad (2)$$

where n is the number of base learners and $w_{i,t}^{j}$ is the weight of x_i (calculated in the same way as a weight used in AdaBoost) for X^l at the t-th round. When an example x_i is regarded by X^l as a hard example, $\delta_{i,t}^{l} = 0$. Thus, from equation (1), $D_{i,t}^{l}$ is 0 and x_i is not used in any subsequent rounds. In this way, hard examples are removed from the training set. Equation (2) represents the collaboration of base learners. $d_{i,t}^{l}$ is determined based on the weights of other base learners. Since the weight $w_{i,t}^{j}$ of a misclassified example increases, the values of both $d_{i,t}^{l}$ and $D_{i,t}^{l}$ increase when most of the other base learners misclassify the example. Consequently, the example is learned by X^l.

Here, we show an example of the weighting process. We consider a case in which the training set consists of six examples $\{x_1, x_2, x_3, x_4, x_5, x_6\}$ whose class labels are $\{c_1, c_1, c_2, c_2, c_3, c_3\}$ respectively, and three base learners X^1, X^2 and X^3 are trained simultaneously. Assuming that each base learner classifies each example at the first round as shown in Table 1 (a) (the class labels shown in boldface represent the correct classification), the weight distribution of each example for the second round is calculated according to equation (1) as shown in Table 1(b).

For example, x_4 was correctly classified by X^1 and X^2 while misclassified by X^3. On the other hand, x_2 was correctly classified only by X^3. Thus, x_4 should be learned by X^1 and X^2 while x_2 should be learned by X^3. From Table 1 (b), the weight distribution of x_4 for X^1 and X^2 is much higher than for X^3. The weight distribution of x_2 for X^3 is higher than for X^1 and X^2. Since each base learner learns the examples which have higher weight distributions, both x_2 and x_4 (as well as the other examples) are learned by appropriate learners in the next round.

	x_1	x_2	x_3	x_4	x_5	x_6
X^1	**c_1**	c_2	c_3	**c_2**	**c_3**	**c_3**
X^2	c_2	c_3	**c_2**	**c_2**	c_1	**c_3**
X^3	**c_1**	**c_1**	**c_2**	c_3	**c_3**	c_2

(a) Predicted class labels

	x_1	x_2	x_3	x_4	x_5	x_6
X^1	.20	.13	.08	.20	.20	.20
X^2	.09	.15	.22	.22	.09	.22
X^3	.21	.22	.21	.08	.21	.08

(b) Weight distributions

Table 1. Predicted class labels and weight distributions.

[1] The threshold is determined by searching for the optimal value using the beam search, based on the NadaBoost algorithm (Nakamura et al., 2004), which is more robust to hard examples than AdaBoost.

3.2 Construction of an ensemble classifier using collaboration of base learners

Assuming that the number of rounds is T, the *l*-th base learner trained in the *t*-th round is represented as X_t^l (t = 1, … , T). At the end of the *T*-th round, we integrate the base learners $\{X_t^l\}_{t=1}^T$ into an ensemble classifier X^l and determine the prediction of X^l by integrating the predictions of all base learners $\{X_t^l\}$ using a weighted voting method. Specifically, the prediction of the ensemble classifier X^l is determined as the class label which is predicted by the majority of base learners. The prediction $X^l(x)$ of the ensemble classifier X^l for an example x is defined as follows:

$$X^l(x) = \arg\max_{c\in\{1,\ldots,C\}} \sum_{t=1}^{T} \alpha_t^l [X_t^l(x) = c], \tag{3}$$

where $[X_t^l(x) = c]$ is 1 if $X_t^l(x) = c$ and otherwise is 0. $\alpha_t^l = \log\{(1-\varepsilon_t^l)/\varepsilon_t^l\}$, where ε_t^l is the classification error of X^l. Thus, a higher value for α_t^l means a lower classification error.

In the learning process, each base learner is specialized to classify non-hard examples precisely. Thus, we must determine whether a given example is a hard example or a non-hard example. To distinguish hard examples from non-hard examples, we define a criterion and call it *class separability*. Class separability is defined so that it is proportional to the classification performance of a base learner for each class. When X_t^l can correctly classify the examples whose class labels are c, the class separability for class c is high because X_t^l can distinguish these examples from the other examples. On the other hand, if X_t^l misclassifies these examples, the class separability for class c is low. Since hard examples are frequently misclassified, the class separability of a hard example will be much lower than that of a non-hard example. Here, we consider the case in which X_t^l predicts the class label of an unknown example x as class c. If the class separability of X_t^l for class c is high, x will be a non-hard example. That is, the possibility that the prediction is correct will be high. We define the class separability $s_t^l(c)$ for class c as follows:

$$s_t^l(c) = \begin{cases} s_{t+}^l(c) & \text{if } X_t'^l(x) = c, \\ s_{t-}^l(c) & \text{otherwise,} \end{cases} \tag{4}$$

where

$$X_t'^l(x) = \arg\max_{c\in\{1,\ldots,C\}} \sum_{\tau=1}^{t} \alpha_\tau^l [X_\tau^l(x) = c],$$

$$s_{t+}^l(c) = \frac{n_{t,c,c}^l}{\sum_{i=1}^{C} n_{t,c,i}^l},\ s_{t-}^l(c) = \frac{\sum_{i\neq c}^{C}\sum_{j\neq c}^{C} n_{t,i,j}^l}{\sum_{i\neq c}^{C}\sum_{j=1}^{C} n_{t,i,j}^l}.$$

$n_{t,i,j}^l$ denotes the number of examples whose class label is i and which were classified into class j. $s_{t+}^l(c)$ is high when the examples whose class labels are c are correctly classified

into class c. $s^l_{t-}(c)$ is high when the examples whose class labels are $c'(c' \neq c)$ are classified into class c'. If the prediction $X'^l_t(x)$ of the *l*-th classifier is c, then $s^l_{t+}(c)$ is used as the class separability. Otherwise, $s^l_{t-}(c)$ is used as the class separability. For example, in the case shown in Table 1, $s^1_{1+}(c_1) = 1/2$ because X^1 correctly classifies x_1 and misclassifies x_2. $s^1_{1-}(c_1) = 4/4 = 1$ because X^1 correctly classifies x_3, x_4, x_5 and x_6. Similarly, the class separabilities of the other base learners for the class c_1 are calculated as follows: $s^2_{1+}(c_1) = 0, s^2_{1-}(c_1) = 3/4, s^3_{1+}(c_1) = 1,$ and $s^3_{1-}(c_1) = 1$. These values of class separability for c_1 indicate that X^3 will give the most accurate prediction in the classification of the examples whose class labels are c_1.

When classifying an unseen example, we first obtain the predictions of all classifiers $\{X^l\}^n_{l=1}$, where n is the number of features. We next calculate the class separability of each classifier and select the classifier with the highest class separability as the most reliable classifier. The prediction $F(x)$ of the ensemble classifier for an example x is then determined by the following equation:

$$F(x) = X^{l^*} \text{ such that } l^* = \arg\max_l \sum_{\tau=1}^{t} s^l_\tau(X'^l_\tau(x)). \tag{5}$$

Finally, we show the algorithm list of the collaborative ensemble method as follows:

1. Initialize: t = 1, $D^l_{i,1} = 1/m$ and $\delta^l_{i,1}$ = 1 for all i and l, *T_list* = {}. *T_list* is the list to retain up to b weight thresholds, where b is beam width.
2. For each base learner, construct the training set S^l_t by sampling from the original training set S according to the weight distribution $D^l_{i,t}$.
3. Train each base learner using S^l_t and obtain X^l_t.
4. If $t = T$ and *T_list* is empty, then make the final prediction F and finish the learning process; otherwise, go to step 5.
5. For all l, classify all training examples using X^l_t, then decrease the weights of correctly classified examples and increase the weights of misclassified examples.
6. Obtain the possible thresholds W^l_t.
7. Calculate the accuracy of each threshold by estimating the classification accuracy using the threshold.
8. Add the threshold to *T_list*.
9. If the number of thresholds in *T_list* is more than b, remove the thresholds which have lower accuracy.
10. Select the most accurate threshold from *T_list* to detect hard examples, then remove the threshold from *T_list*.
11. Set $\delta^l_{i,t+1}$ to 0 if x_i is regarded as a hard example by X^l_t, otherwise to 1.
12. Calculate the weight distribution $D^l_{i,t+1}$ for each l and i.

13. $t \leftarrow t+1$ and go to step 2.

In the above algorithm, we efficiently search for the optimal (or suboptimal) threshold based on beam search. The process corresponds to steps 5 to 10 above. After all base learners are trained, each weight is updated in step 5. According to the weights, the possible thresholds W_t^l are determined in step 6. We next evaluate the classification accuracy for each threshold in step 7. In step 8, the thresholds are added to a list *T_list*, which is used by beam search to restrict the search space. *T_list* retains at most b thresholds, where b corresponds to the beam width. If the number of possible thresholds is higher than b, the thresholds which have a lower accuracy are removed from *T_list* in step 9 and are not used for the search. In step 10, a threshold W_t^{l*} which has the highest accuracy is selected and removed from *T_list*. Hard examples are detected using W_t^{l*}. This process is repeated while *T_list* is not empty. As a result, the optimal (or suboptimal) threshold can be found.

3.3 Experiment

We carried out several object recognition experiments to verify the performance of our method using the images in the ETH-80 Image Set database (Leibe & Schiele, 2003). This data set contains 8 different objects: apples, cars, cows, cups, dogs, horses, pears, and tomatoes. We used 20% of the examples as training examples and the remainder as test examples. The number of rounds was experimentally set to 100. We constructed five types of base learners using five types of features as given in Fig. 2.

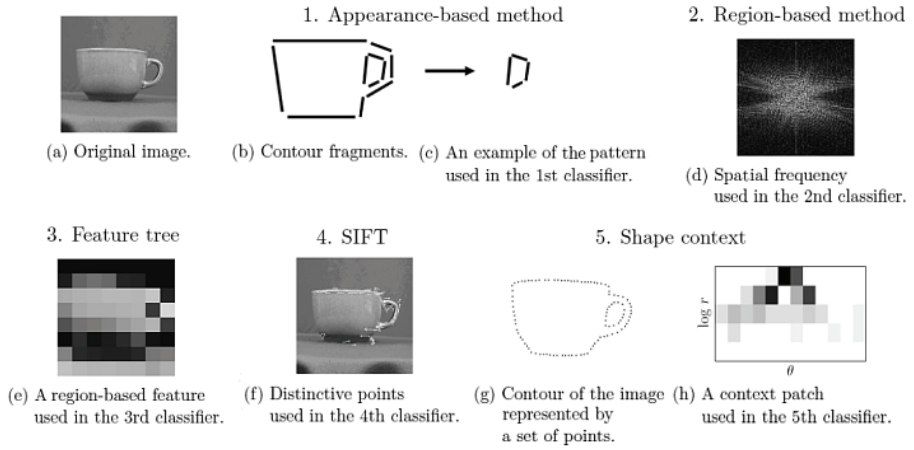


Fig. 2. The features used in this experiment.

The first base learner is an appearance-based recognition method which utilizes contour fragments (Nomiya & Uehara, 2007), as shown in (b). The set of contour fragments is grouped into meaningful structures called patterns as depicted in (c). The second base learner is based on the distributions of pixel intensity values (Nomiya & Uehara, 2007). The distributions are represented by a Generic Fourier Descriptor (GFD). A GFD is obtained by calculating the spatial frequency of an image as described in (d). The third base learner is a feature tree (Nomiya & Uehara, 2005). This method generates region-based features by image filtering as shown in (e). It then combines several features into several decision trees

called feature trees. The fourth base learner is based on Scale Invariant Feature Transform (SIFT) (Lowe, 2004). SIFT generates deformation-invariant descriptors by finding distinctive points in objects (indicated by white arrows in (f)). These points represent the characteristics of the object based on image gradients. The fifth base learner uses the shape context method (Belongie et al., 2001). In this method, the contour of an object is described by a set of points as illustrated in (g). Using this set of points, log-polar histograms of the distance and angle between two arbitrary points, called shape contexts, are calculated for all points. An example of shape context is given in (h). This method discriminates the object by matching its shape contexts with the shape contexts in the training set.

In this experiment, we evaluate the proposed method from the following three viewpoints: firstly, in order to verify the effectiveness of the learning model of the proposed method, we evaluate the recognition performance of the collaborative ensemble learning model. Secondly, we assess the usefulness of the proposed method as an object recognition method by comparing it with several existent object recognition methods. Finally, we use noisy image data to verify the robustness of the proposed method to noise.

3.3.1 Evaluation of collaborative ensemble learning model

To verify the effectiveness of our collaborative ensemble learning model, we construct four types of ensemble classifiers, L_2, L_3, L_4 and L_5, by integrating two, three, four and five base learners respectively. L_i consists of the first, second, … , and i-th base learners. We then compare their performance. The result of the experiment is provided in Fig. 3.

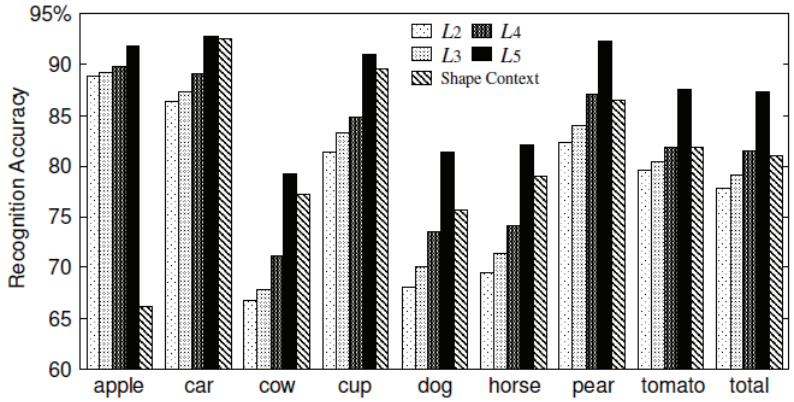


Fig. 3. The recognition accuracy for each ensemble classifier and shape context method.

The recognition accuracy is proportional to the number of integrated base learners. In particular, the accuracy improves significantly for animals with complex shapes and textures. This result implies that diverse features are required to discriminate correctly between complex objects and that our method can effectively utilize various features. L_5 achieved much higher accuracy than the other ensemble classifiers. Although this improvement is due to the high classification performance of the shape context learner (i.e., the fifth base learner), our method fully outperforms the shape context method. Since the shape context method depends on the shapes of objects, it sometimes fails to distinguish between objects which have similar shapes, such as apples and tomatoes. Thus, our method

selects an appropriate base learner other than the shape context learner for the classification of apples and tomatoes, leading to a higher recognition accuracy for our method.

3.3.2 Comparison with other object recognition methods

We compare our recognition performance with those of the following six object recognition methods. The first is the shape context (Belongie et al., 2001). The second is the multidimensional receptive histogram (Schiele & Crowley, 2000), which describes the shapes of objects using statistical representations. The third is color indexing (Swain & Ballard, 1991), which discriminates an object using RGB histograms calculated from all the pixels in the object. The fourth is based on local invariant features (Grauman & Darrell, 2005) that are generated by a gradient-based descriptor and are robust to the deformation of images. The fifth is the learning-based recognition method (Marée et al., 2005), in which an object is described by randomly extracted multi-scale subwindows in the image and classified by an ensemble of decision trees. The sixth is the boosting-based recognition method (Tu, 2005), in which a probabilistic boosting-tree framework is introduced to construct discriminative models. The recognition accuracy is shown in Table 2.

Swain & Ballard (1991)	64.85	Grauman & Darrell (2005)	81
Marée et al. (2005)	74.51	Belongie (2001)	86.40[2]
Tu (2005)	76	**Proposed method**	**87.27**
Schiele & Crowley (2000)	79.79		

Table 2. The recognition accuracy for each object recognition methods (in %).

Our recognition accuracy is higher than those of all other recognition methods. We utilize multiple features for recognition and thus can discriminate a wider variety of objects than single-feature recognition methods. In addition, this result indicates that our learning strategy for selecting optimal base learners is effective. Since the criterion for the determination of optimal base learners is determined by observing the collaborative learning process, this result indicates the effectiveness of our collaborative learning framework.

3.3.3 Robustness over hard examples

In order to verify the robustness to hard examples of our method, we carry out an experiment using the Caltech image data set, which contains many hard examples. We use the images of six kinds of objects from the data set: airplanes, cars, Dalmatians, faces, leopards and motorbikes. We use 20% of the examples as training examples and the remainder as test examples. We construct two types of ensemble classifiers and compare these ensemble classifiers with our method. The first ensemble classifier, X_1, does not eliminate any hard examples and never increases the weights of hard examples even if they are misclassified. The second ensemble classifier, X_2, also does not eliminate hard examples,

[2] This recognition accuracy is reported by (Leibe & Schiele, 2003). This accuracy has been achieved, however, using over 98% of the examples in the training set while we use only 20%. In addition, its recognition accuracy is proportional to the number of the training examples as shown in (Belongie et al., 2001). The recognition accuracy using 20% of examples in the training set is 81.06%.

but it does increase the weights of all misclassified examples even if they are hard examples. First, we show the recognition accuracy for each class and total recognition accuracy in Fig. 4.

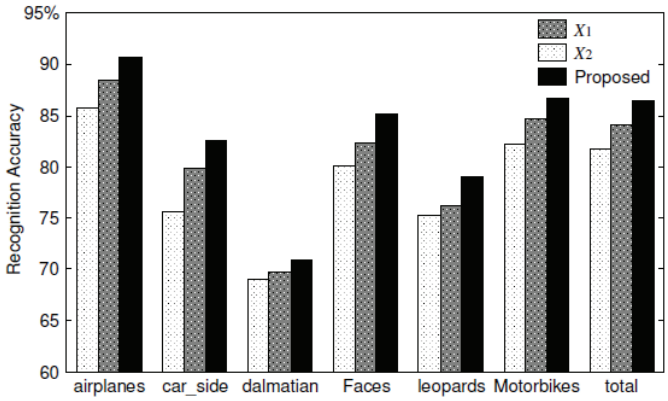


Fig. 4. The recognition accuracy for each class.

The recognition accuracy of X_2 is the worst because it does not take hard examples into consideration. X_1 outperforms X_2 by giving smaller weights to hard examples and thus reducing their influence to some extent. Given that our recognition accuracy is the best for all objects, it seems that detecting and eliminating hard examples from training examples leads to more efficient learning.

Through these experiments, we confirm the effectiveness of our collaborative ensemble learning framework. However, it is difficult to determine the appropriate combination of base learners because finding the optimal combination is an open problem. Although our method enables an ensemble classifier to be less sensitive to the combination of classifiers due to the determination of the optimal base learner for a given example, the accuracy of the proposed method can improved if we can determine the optimal combination of base learners. Thus, we should find an efficient method to determine the optimal combination.

4. Application to facial expression classification problem using multistream time-series data

4.1 Visual learning in facial expression recognition

In this section, we mention the application of our visual learning framework to problems in facial expression classification which is a challenging domain in computer vision understanding. In facial expression recognition, distinguishing slight differences in face images is required. However, it is quite difficult to accomplish this because the number of characteristic points on a face is small. Moreover, the movement of each point is subtle, while multiple points are mutually correlated in most facial expressions. For accurate recognition, a variety of recognition methods have been proposed using various features which can be extracted or constructed from input image data. We divide the features typically used in facial expression recognition into three levels: low-level features, medium-level features and high-level features.

Low-level features are based on the intensity of each pixel in an image. Since an image is generally given as a set of pixels and a pixel is described by its intensity value, low-level

features can be obtained by directly processing the given image data using some kind of statistical analysis and dimensionality reduction techniques, such as principal component analysis and linear discriminant analysis (Donato et al., 1999). Since low-level features can be directly and easily generated from the given image data, they can be utilized for a variety of recognition problems (e.g., object recognition and handwritten character recognition). However, low-level features seem to be rather insufficient for accurately distinguishing slight difference among various types of facial expressions.

In facial expression recognition, it is effective to observe the movement of particular parts in a face because facial expressions can be described as the combination of the movement of facial muscles. According to this idea, several features which are able to describe partial movement in a face have been proposed (Bourel et al., 2002). These features, which we call medium-level features, can more precisely describe characteristic parts of a face. By selecting appropriate parts which are relevant to the facial expression and analyzing their movement, more accurate recognition can be performed. However, since useful parts are highly dependent on the expression, finding appropriate parts is a difficult task.

Features more complex than medium-level features are defined by modeling whole face (Essa & Pentland, 1997). We call these features high-level features. The models are able to analyze multiple parts in a face simultaneously and can be more effective than medium-level features. High-level features are typically constructed by taking multiple images from various directions and generating 3D models from input 2D images. This can be a crucial problem, however, because it is troublesome and time-consuming to construct facial models through such processes.

Taking the tradeoff between the better quality of higher-level features and the lower cost of constructing lower-level features into account, we propose a facial expression recognition method based on medium-level features. Although the representational ability of a single medium-level feature is lower than that of a single high-level feature, more discriminative features can be constructed by combining multiple medium-level features. As was mentioned in Section 3, ensemble learning enables the utilization of multiple features. Thus, we can deal with multiple medium-level features by introducing our visual learning framework.

Generally, only a few medium-level features around salient parts on a face (e.g., eyes, eyebrows and mouth) are used for recognition when analyzing various facial parts, and finding the appropriate combination of facial parts is essential to distinguishing a wide variety of facial expressions. In order to solve this issue and find an appropriate combination depending on facial expressions, we utilize a motion capture system which can precisely observe the movement of diverse facial parts (Osaki et al., 2000).

The facial expression data obtained from the motion capture system are represented as multistream time-series data. Multistream time-series data consist of multiple time-series sequences which are mutually correlated. Since multistream time-series data generally contain a large amount of information which includes redundant data, it is necessary to select useful streams from the given streams in order to achieve accurate recognition. Thus, we propose an efficient method of assessing the usefulness of each stream and finding appropriate streams based on an effective criterion to measure the similarity among multiple streams. To verify the effectiveness of the proposed method, we perform several facial expression recognition experiments.

4.2 Feature construction method

In order to generate facial expression data, we utilize a motion capture system to capture the movement of several points on a face. Specifically, the facial expression data are captured by 35 markers on the subject's face as depicted in Fig.5 and described by 35 streams which represent the movement of each marker.

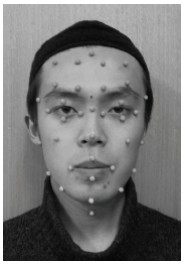


Fig. 5. Markers used by motion capture system.

The location of each marker is determined according to the definition of Facial Action Coding System (FACS) (Ekman & Friesen, 1978), which is designed for measuring and describing facial behavior. FACS was developed by analyzing and determining the relationship between the contraction of each facial muscle and the appearance of the face. In FACS, specific measurement units called Action Units (AUs) are defined to describe facial expressions. AUs represent the muscular activity that leads to the changes in facial expression. Although numerous AUs are specified, the following 17 AUs are considered to be sufficient to describe basic facial expressions.

No.	Name	No.	Name	No.	Name
1	Inner Brow Raiser	9	Nose Wrinkler	17	Chin Raiser
2	Outer Brow Raiser	10	Upper Lip Raiser	20	Lip Stretcher
4	Brow Lowerer	12	Lip Corner Puller	23	Lip Tightener
5	Upper Lid Raiser	14	Dimpler	25	Lips Part
6	Cheek Raiser	15	Lip Corner Depressor	26	Jaw Drop
7	Lid Tightener	16	Lower Lip Depressor		

Table 3. Main Action Units (AUs).

In Table 3, for example, AU 1 corresponds to the raising of the inner corner of the eyebrow, while AU 4 corresponds to the puckering up of the outer corner of the eyebrow. Combining these AUs allows for various types of facial expressions to be described. We show the combinations of AUs for several basic facial expressions (Surprise, Anger, Happiness and Sadness) in Table 4.

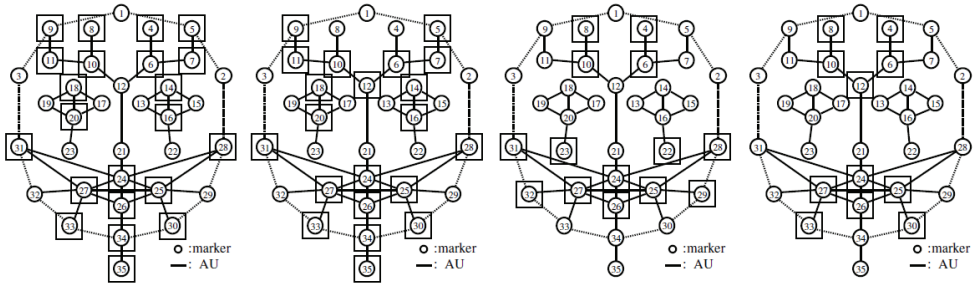
Expression	AU numbers (intensity)
Surprise	1(100), 2(40), 5(100), 10(70), 12(40), 16(100), 26(100)
Anger	2(70), 4(100), 7(60), 9(100), 10(100), 12(40), 15(50), 26(60)
Happiness	1(60), 6(60), 10(100), 12(50), 14(60), 20(40)
Sadness	1(100), 4(100), 15(50), 23(100)

Table 4. Combination of AUs for each expression.

In Table 4, the numerical values in parentheses are intensity values of AUs. A higher intensity value means stronger activity of an AU, and the maximum value is 100. For

example, in Sadness, AUs 1, 4 and 23 are strongly activated and AU 15 is weakly activated because brows and lips show characteristic changes in the expression of sadness. Based on the combination of AUs, corresponding combinations of streams (i.e., markers shown in Fig. 5) are determined for each expression as illustrated in Fig. 6. In Fig. 6, the markers encircled by squares denote the corresponding streams.

The input data given by the motion capture system simply represents the movement of each marker. It is a kind of primitive feature and is thus inadequate for recognition. To construct higher-level features, we estimate the stress from each marker. This is because each facial expression is described as the movement of particular facial muscles which can be represented by the stress for each marker. Due to the difficulty of directly measuring the stress, we instead estimate the stress using finite element method (FEM). FEM is widely used for estimating the deformation of an object caused by the given stress. Since our goal is to obtain the stress given to each marker, we consider the inverse problem of FEM. That is, we estimate the stress from the deformation of facial muscles. In order to describe the stress estimation process, we first show the settings of this problem in Fig. 7.



1.Head	7.LBrowOut	13.LEyeIn	19.REyeOut	25.LLipCorner	31.RUpperLine
2.LHead	8.LFronsIn	14.LEyeTop	20.REyeBottom	26.LipBottom	32.RMiddleLine
3.RHead	9.LFronsOut	15.LEyeOut	21.Nose	27.RLipCorner	33.RLowerLine
4.LFronsIn	10.RBrowIn	16.LEyeBottom	22.LCheek	28.LUpperLine	34.Chin
5.LFronsOut	11.RBrowOut	17.REyeIn	23.RCheek	29.LMiddleLine	35.Neck
6.LBrowIn	12.BetweenBrows	18.REyeTop	24.LipTop	30.LLowerLine	

Fig. 6. Corresponding markers (streams) for each expression.

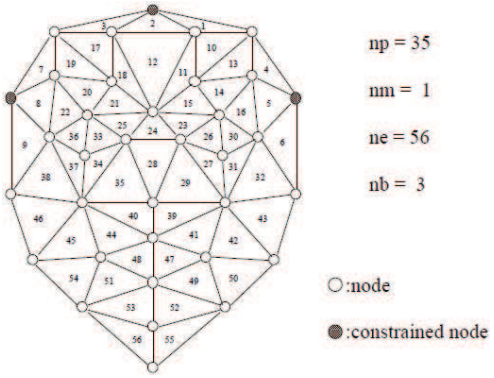


Fig. 7. The problem settings of inverse FEM for facial expression recognition.

Fig. 7 represents the settings of the inverse problem of FEM for facial expression recognition. A face is divided into several regions by the markers on the face. The regions and markers are called elements and nodes, respectively. Since we use 35 markers, the number of nodes np is 35. A node is represented by a circle in Fig. 7. Each element is defined as a triangular region whose vertices consist of three nodes. Thus, the number of elements ne is 56. We fix the number of materials nm at 1 because a face is uniformly covered with skin and set the number of constraints (i.e., the number of fixed points) nb at 3 because the three markers (Head, LHead and RHead) represented by • are fixed.

Under this setting, we observe the deformation of each element by measuring the movement of each marker and then estimate the stress given to each node. The process of solving the inverse problem of FEM proceeds as follows:

1. Calculating the element rigidity matrix
   Using the above settings, calculate the element rigidity matrix $[EK]$ as follows:

$$[EK] = tS[B]^T[D][B]$$

   where $[B]^T$ denotes the transpose of $[B]$ and

$$S = \frac{1}{2}\begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{vmatrix}, \quad [D] = \frac{E}{(1+\nu)(1-\nu)}\begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}$$

$$[B] = \frac{1}{2S}\begin{bmatrix} y_j - y_k & 0 & y_k - y_i & 0 & y_i - y_j & 0 \\ 0 & x_k - x_j & 0 & x_i - x_k & 0 & x_j - x_i \\ x_k - x_j & y_j - y_k & x_i - x_k & y_k - y_i & x_j - x_i & y_i - y_j \end{bmatrix},$$

   x_n and y_n $(n = i, j, k)$ are the x-coordinate and y-coordinate of the three nodes i, j and k which form an element. E, ν and t denote Young's modulus, Poisson's ratio, and board thickness, respectively. We experimentally set the value of E and ν to 0.14[MPa] and 0.45, respectively.
2. Constructing the whole rigidity matrix
   Construct the whole rigidity matrix $[TK]$ using the element rigidity matrix $[EK]$ so that each element of $[TK]$ corresponds to $[EK]$ calculated for each element in step 1.
3. Estimating the stress for each node
   Calculate the node force vectors $\{F\}$ according to the following equation:

$$\{F\} = [TK]\{d\}$$

   where $\{d\}$ denotes the displacement of the nodes output by the motion capture system. The node force vectors $\{F\}$ represent the stress given to each node. Thus, we utilize them as the higher-level features.

4.3 Feature selection method

Through the above feature construction process, the higher-level features are constructed for each node (i.e., marker). However, using all of the features is inefficient because the large amount of information which they contain often includes redundant data. Since such redundancy makes the computational complexity excessively large and does not contribute to the improvement of the recognition accuracy, it is necessary to select useful features. To perform efficient feature selection, because the constructed features are represented as multistream time-series data, we propose an effective method to evaluate the usefulness of streams in multistream time-series data.

We propose an effective criterion to assess the usefulness of each stream based on a novel similarity measure called Angular Metrics for Shape Similarity (AMSS) (Nakamura et al., 2007). To measure the similarity between time-series data, AMSS first divides a time-series sequence into several subsequences which are represented by a set of vectors. It then calculates the similarity based on the angles between two subsequences. Using angles for calculating similarity, AMSS can be robust to the difference in spatial locations of two time-series sequences compared with conventional similarity measures such as Dynamic Time Warping (Berndt & Clifford, 1996).

In order to evaluate the usefulness of a stream based on AMSS, we consider a *C*-class multistream time-series data classification problem. We assume that the input data consist of several examples. An example x contains a set of streams, which are described by vector sequences, and is represented as $x = \{x^1, \cdots, x^p\} = \{(\vec{x}_{11}, \cdots, \vec{x}_{1q_1}), \cdots (\vec{x}_{p1}, \cdots, \vec{x}_{pq_p})\}$, where p is the number of streams, $\vec{x}_{rs}$ is the *s*-th vector in the *r*-th stream, and q_k is the length of the *k*-th stream. Each example has the class label y ($y \in \{1, \cdots, C\}$). We represent a multistream time-series data set as $\{(x_1, y_1), \cdots, (x_m, y_m)\}$ where m is the number of examples.

To define a criterion to evaluate the usefulness of a stream, we assume that examples with the same class label have similar streams while examples with different class labels have dissimilar streams. The streams which satisfy these assumptions are useful to classifying examples accurately because if these assumptions are satisfied, the examples can be appropriately separated into each class. Thus, we measure the similarity between arbitrary combinations of two streams and, by determining whether the streams satisfy these assumptions, find the optimal streams.

According to the first assumption, we first define the similarity $SS(n)$ among the examples which have the same class label for each stream ($n = 1, \cdots, p$) as follows:

$$SS(n) = \sum_{k=1}^{m-1} \sum_{l=k+1}^{m} A_s(x_k^n, x_l^n) w(x_k^n) w(x_l^n), \tag{12}$$

where

$$A_s(x_k^n, x_l^n) = \begin{cases} AMSS(x_k^n, x_l^n) & \text{if } y_k = y_l \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

$$w(x_j^n) = \sum_{i=1}^{q_n} \| \vec{x}_{jni} \|, \tag{14}$$

$\vec{x}_{jni}$ corresponds to the *i*-th vector in the *n*-th stream $\vec{x}_{ni}$ of the *j*-th example x_j. $AMSS(x_k^n, x_l^n)$ denotes the similarity between x_k^n and x_l^n as calculated by AMSS. Due to space limitations, we do not show the details of AMSS here. The detailed algorithm of AMSS is described in (Nakamura et al., 2007). In equation (12), the coefficients $w(x_k^n)$ and $w(x_l^n)$ are used as weights. In order to utilize informative streams, we introduce these weights. We show an example in Fig. 8.

In Fig. 8, x_i^1 is similar but not identical to x_j^1. On the other hand, x_i^2 and x_j^2 are identical. Thus, the similarity between x_i^2 and x_j^2 is higher than that between x_i^1 and x_j^1. From the viewpoint of information theory, however, the entropy of x_i^1 and x_j^1 is much higher than those of x_i^2 and x_j^2. Thus, x_i^1 and x_j^1 are more informative and represent the characteristics of streams. As a result, comparing x_i^1 with x_j^1 is more effective than comparing x_i^2 and x_j^2. Since the weights in equation (12) reflect the information contained in each stream, introducing the weights enables the utilization of informative streams.

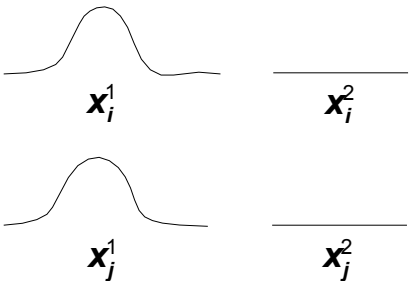


Fig. 8. Informative streams (left) and uninformative streams (right).

Next, with respect to the second assumption, we define the similarity $DS(n)$ for each stream among the examples which have different class labels as follows:

$$DS(n) = \sum_{k=1}^{m-1} \sum_{l=k+1}^{m} A_d(x_k^n, x_l^n), \tag{15}$$

where

$$A_d(x_k^n, x_l^n) = \begin{cases} AMSS(x_k^n, x_l^n) & \text{if } y_k \neq y_l \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

In the calculation of $DS(n)$, we do not use weights. Without weighting, uninformative streams as shown in Fig. 8 make the value of $DS(n)$ higher, and a higher $DS(n)$ means that the stream is less useful. Consequently, the streams are regarded as useless streams. Based on $SS(n)$ and $DS(n)$, we define the usefulness of the *n*-th stream $U(n)$ by the following equation:

$$U(n) = \frac{SS(n)}{DS(n)}. \tag{17}$$

Using equation (17), we can estimate the usefulness of each stream. However, multistream time-series data generally contain a number of streams, with the number of possible combinations of the streams exponentially increasing as the number of streams increases. Thus, selecting useful streams is a difficult task. In order to determine the optimal number of streams, we propose an effective method based on the idea of class separability mentioned in Section 3.2. For the stream selection, the class separability is defined as follows:

$$s_+^k(c) = \frac{e_{c,c}^k}{\sum_{i=1}^{C} e_{c,i}^k},\ s_-^k(c) = \frac{\sum_{i\neq c}^{C}\sum_{j\neq c}^{C} e_{i,j}^n}{\sum_{i\neq c}^{C}\sum_{j=1}^{C} e_{i,j}^k}, \tag{18}$$

where $e_{i,j}^k$ denotes the number of examples whose class labels are i and which are classified into the class j using the k most useful streams (i.e. streams that have the k highest values of usefulness). Based on the class separability, we determine the optimal number of streams k^* so that the following evaluation function is maximized:

$$k^* = \arg\max_k \left\{ \sum_{c=1}^{C} s_+^k(c) s_-^k(c) \right\}. \tag{19}$$

That is, k^* streams should be selected where k^* is the number of streams which maximizes the products of $s_+^k(c)$ and $s_-^k(c)$ for each class. When classifying an unseen example x, the predicted class label y is given by

$$y = \arg\max_c \left\{ \frac{\sum_{i=1}^{m}\sum_{n=1}^{k^*} f(x^n, x_i^n, c)}{m_c} \right\}, \tag{20}$$

where

$$f(x^n, x_i^n, c) = \begin{cases} AMSS(x^n, x_i^n) & \text{if } y_i = c \\ 0 & \text{otherwise,} \end{cases} \tag{21}$$

and m_c is the number of examples whose class labels are c.

4.4 Experiment

For the evaluation of the proposed method, we perform two types of experiments. In the first experiment, we verify the usefulness of our feature construction and feature selection methods by comparing the recognition performance using the streams selected by the proposed method with the performance using the streams defined to be useful by FACS. We then apply our collaborative ensemble learning framework to a facial expression recognition problem and assess the effectiveness through the second experiment.

The motion data used in this experiment are obtained using the optical motion capture system HiRES (4 cameras) by Motion Analysis company. The motion data has a sampling frequency of 60Hz and a length of 5 seconds. Thus, a total of 300 frames are used per markers. We divide these frames into 30 groups, so that each group contains 10 frames, then generate time-series data for each stream whose length is 30 by averaging the frames in each group. Since we use the time-series data of horizontal and vertical movement of the markers, each stream consists of 2-dimensional time-series data. The motion data contains four types of expressions: *Surprise, Anger, Happiness,* and *Sadness*. Thus, the classification task is to distinguish these four expressions (i.e., a 4-class classification problem).

4.4.1 Comparison with FACS

We carry out several facial expression recognition experiments using facial expression data from five subjects. For each subject, we obtain 24 examples (6 examples for each expression), for a total of 120 examples. We perform 5-fold cross-validation using 96 examples as training examples and 24 examples as test examples. In addition, we perform person-independent and person-dependent experiments. A person-independent experiment is an experiment in which the training set consists of examples from four subjects and the test set consists of examples from the remaining one subject. In a person-dependent experiment, the training set and test set include examples from the same subject (but not identical examples).

For the collaborative ensemble learning, we construct four base learners which are specialized to recognize *Surprise, Anger, Happiness* and *Sadness*. These base learners were generated using the following equation instead of equation (16).

$$A_s(x_k^n, x_l^n) = \begin{cases} AMSS(x_k^n, x_l^n) & \text{if } y_k = y_l = c \\ 0 & \text{otherwise,} \end{cases} \tag{22}$$

where c is the class label (that is, *Surprise, Anger, Happiness,* or *Sadness*). To classify an unseen example, we perform weighted voting using these four base learners and using their class separability as weights. For the comparison with FACS, we construct four base learners based on the stream defined by FACS and integrate them using weighted voting. We show the result of experiment in Table 5.

Method	Avg. # of streams	Surprise	Anger	Happiness	Sadness	Total
Person-independent						
Proposed	24.3	100	63.3	86.7	73.3	80.8
FACS	17.0	100	43.3	66.7	60.0	67.5
Person-dependent						
Proposed	27.4	100	83.3	90.0	76.7	87.5
FACS	17.0	100	60.0	73.3	80.0	78.3

Table 5. The recognition accuracy of the proposed method and FACS (in %).

Our method outperforms FACS in all expressions except for *Surprise* and total recognition accuracy. Since *Surprise* is most discriminative expression because of the intensive movement of facial muscles, both methods perfectly classified the example of *Surprise*. On the other hand, because distinguishing *Anger* from the other expressions is relatively difficult, the recognition accuracy for *Anger* is generally lowest. From this result, we verify the effectiveness of our feature construction and selection method.

The number of selected streams of our method is quite larger than that of FACS. For example, in the person-independent experiment, the average numbers of selected streams for *Surprise, Anger, Happiness* and *Sadness* are 26.4, 19.0, 27.4, and 24.2, respectively. Thus, the overall average number of selected streams is 24.3. We show an example of selected streams for the person-independent experiment in Fig. 9.

As for *Anger* and *Sadness*, most streams are regarded as useful streams while the number of streams defined by FACS is relatively small. This is because *Anger* and *Sadness* are more difficult to classify correctly than the other two expressions. In fact, the recognition accuracy for *Anger* and *Sadness* is relatively low. The number of selected streams for *Surprise* and *Happiness* is smaller than for the other expressions, but larger than those of FACS. This result implies that most AUs can contribute to the discrimination of facial expressions.

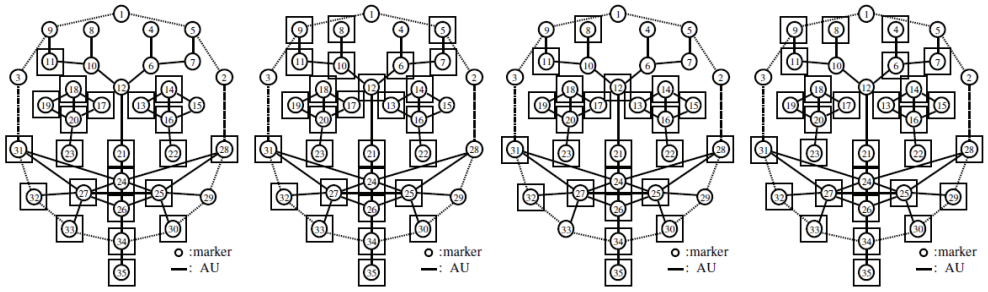


Fig. 9. Example of selected streams for each expression.

Although the recognition performance of our method is higher than that of FACS, the number of streams used is also higher than that of FACS. Since this may be unfair, we perform a recognition experiment using the same number of streams as FACS. We show the result in Table 6.

	Surprise	Anger	Happiness	Sadness	Total
Person-dependent	100	56.7	90.0	80.0	81.7
Person-independent	100	83.3	83.3	66.7	83.3

Table 6. The recognition accuracy of the proposed method using the same number of streams as FACS (in %).

Compared with the recognition accuracy of FACS in Table 5, our method fully achieves high recognition performance. Thus, we confirm the advantage of employing the streams selected by our method over the streams defined by FACS. In the person-dependent experiment, the recognition accuracy is higher than the accuracy shown in Table 5. Generally, person-independent problems are more difficult than person-dependent problems because the distributions of training set and test set can be considerably different. Therefore, we should introduce a method of more accurately estimating the distribution of the test set. However, there is no significant difference between the recognition accuracy of the person-independent cases in Table 5 and Table 6 under the *t*-test with a 5% significance level. This result indicates that our method is able to find the sub-optimal combination of streams which is comparable to the optimal combination with respect to recognition performance. Therefore, our feature selection method seems to be fully effective.

4.4.2 Verification of the effectiveness of collaborative ensemble learning

We introduce our collaborative ensemble learning framework, as shown in Section 3. We use the aforementioned four base learners and train them according to the collaborative ensemble learning algorithm shown in Section 3.2. The prediction of the resulting ensemble classifier $f(x)$ for a given example x is determined based on equation (5) as follows:

$$f(x) = Y^{l^*} \text{ such that } l^* = \arg\max_{l=\{1,2,3,4\}} \sum_{\tau=1}^{t} s_\tau^l (Y_\tau^l(x)) \ , \tag{23}$$

where $Y_\tau^1, Y_\tau^2, Y_\tau^3$ and Y_τ^4 correspond to the hypotheses generated at the τ -th round by the base learners for *Surprise, Anger, Happiness* and *Sadness,* respectively. $s_\tau^l (l = 1,2,3,4)$ represents the class separability for each base learner and is calculated by equation (4). We perform a person-independent experiment with 100 rounds of learning. The result of our experiment is shown in Table 7.

Collaborative ensemble learning						Weighted voting
1 round	5 rounds	10 rounds	20 rounds	50 rounds	100 rounds	
81.5%	81.5%	82.5%	83.0%	85.0%	85.2%	80.8%

Table 7. Recognition accuracy by collaborative ensemble learning and weighted voting.

The recognition accuracy is proportional to the number of rounds. Although the recognition accuracy of the collaborative ensemble learning method is slightly higher than that of the weighted voting method in early rounds, the difference is significant after approximately the 50th round. This result shows that the interaction of multiple base learners is effective in dealing not only with image data but also with multistream time-series data. From this result, we confirm the flexibility of our learning model.

These experimental results show that the streams selected by the proposed method lead to better recognition results than the streams defined by FACS. The results reflect the effectiveness of our method as a data-mining framework in facial expression recognition. However, there is room for improvement in the determination of the optimal combinations of streams in the person-independent case. Thus, we should try to estimate the distribution of each stream more accurately and improve the performance of our stream selection method. In addition, we verify the applicability of our collaborative ensemble learning framework to facial expression recognition problems. In the experiment, the difference between the recognition accuracy at the 50th round and that at the 100th round is insignificant although the computational complexity differs considerably. However, the number of rounds is determined experimentally. In order to improve the learning efficiency, a method of automatically determining the optimal number of rounds is needed.

5. Visual learning revisited

The performance of a visual learning model is closely related to (1) the learning model and (2) features. We refer to several visual learning methods shown in Table 8 from these two viewpoints.

At an early stage of the study of visual learning, a successful object detection method was proposed based on AdaBoost (Viola & Jones, 2001). In this method, a cascade classifier,

which is a kind of ensemble classifier, is constructed. An example of a cascade classifier for facial recognition is shown in Fig. 10.

Reference	Learning model	Features
Viola and Jones, 2001	Cascade classifier + AdaBoost	Primitive feature + feature selection
Marée et al., 2005	Decision tree + ensemble learning	Primitive feature
Krawiec and Bhanu, 2003	Evolutionary computation + closed-loop learning	Feature construction + feature selection
Proposed method	Collaborative learning + modular approach	Feature construction + feature selection

Table 8. Visual learning models.

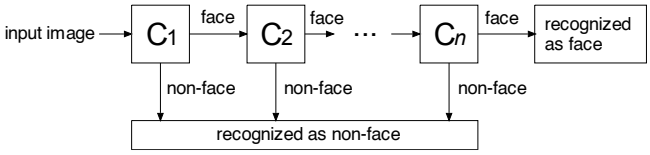


Fig. 10. An example of cascade classifier.

In Fig. 10, C_i (i = 1, 2, … , n, where n is the number of base classifiers) represents the base classifier. For a facial recognition task, if all the base classifiers classify the given image as a face image, then the image is recognized as a face image. Otherwise, the given image is recognized as a non-face image. This recognition process is efficient because, when a given image is classified by a certain classifier as a non-face image, subsequent classifiers are not used. In addition, the cascade classifier is able to select a small number of useful features from a large number of extracted features and can thus quickly and accurately detect objects. However, the features used in this method are primitive features because they are obtained only from pixel intensity values. Moreover, all the base learners use the same features, called Harr-like features. Since recognition performance is greatly dependent on the representational ability and diversity of the features (i.e., various types of high-level features are required), this method seems to be insufficient to describe and recognize complex objects. In addition, the learning model of this method is rather simple because it only optimizes the parameters used in AdaBoost rather than constructing or selecting features. Therefore, this method can be regarded as the most primitive visual learning model.

To utilize multiple features effectively, the ensemble approach has already been introduced into visual learning. For example, in (Marée et al., 2005), an ensemble classifier is constructed using decision tree classifiers as base learners. The learning model of this method is shown in Fig. 11. In Fig. 11, T_i (i = 1, 2, …, n, where n is the number of base learners) represents the base learner (i.e., decision tree classifier). The learning result of each base learner is integrated into an ensemble classifier using weighted voting by all base learners. However, in this ensemble approach, the base learners are separately constructed

with only their learning results integrated. Thus, there is no interaction among the base learners. In addition, all visual learners are based on the same primitive features directly obtained from the color information of each pixel. Since the performance of the ensemble approach tends to be proportional to the diversity of features, higher-level features are needed from the viewpoints of the flexibility and accuracy of recognition.

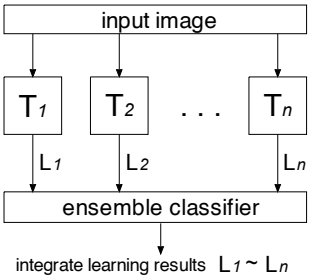


Fig. 11. Ensemble visual learning model.

As a principal learning structure of human visual systems, Krawiec et al. introduced a closed-loop learning scheme based on evolutionary computation (Krawiec & Bhanu, 2003). In this method, high-level features are constructed from the given primitive features (the intensity values of pixels) by combining several image processing operations, such as image filtering and the application of mathematical or logical computation for some pixels. In order to construct appropriate high-level features, the optimal combination of image processing operations is sought through the learning loop. In the learning loop, the combination of image processing operations is determined and its effectiveness is evaluated using evolutionary computation. The learning framework is shown in Fig. 12.

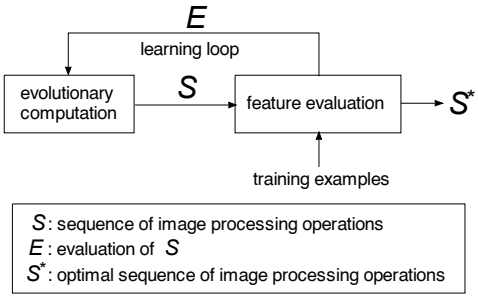


Fig. 12. Evolutionary (closed-loop) visual learning model.

The process of constructing and evaluating features is iteratively performed during the learning process. The evaluation of the constructed features is fed back to the evolutionary computation algorithm and better features then searched. Finally, the best feature S^* is output at the end of learning process. This feature construction strategy represents a sophisticated learning framework that is consistent with human visual learning process. However, the effectiveness of the feature construction method is dependent on the

predefined image processing operations, and the determination of appropriate image processing operations is an open problem. The proposed visual learning approach has the properties of modularity and closed-loop learning, both essential properties in human visual systems; thus, they make the proposed method more efficient than conventional visual learning methods. However, our method still has the following two main problems.

The first problem is related to features. In the facial expression recognition, we propose a feature construction method based on stress estimation. Additionally, we propose a feature selection method based on the evaluation of the usefulness of each stream. We verify the effectiveness of our method through the comparison of its recognition performance with that of FACS. However, the experimental result shows that our feature construction and selection methods cannot always find the optimal combination of streams. This implies that our method is rather simple because we construct higher-level features for each stream separately. A facial expression is represented by the complex movements of several points on a face. This means that multiple streams are mutually correlated. Therefore, we should improve the feature construction process so that the higher-level features are constructed by integrating multiple streams which are mutually correlated. More generally, we should further analyze human visual systems and attempt to model them in order to develop satisfactory feature construction frameworks for various visual recognition problems. The second problem with our method resides in the representation of knowledge obtained through the learning process. Our method can provide the knowledge for the recognition of visual data as the useful features. This knowledge can be used for data mining, but in order to utilize the learning results of our method fully in some data mining domains, the knowledge should be systematized by analyzing and organizing it in the learning process.

5. References

Belongie, S.; Malik, J. & Puzicha, J. (2001). Matching Shapes, *Proceedings of the 8th IEEE International Conference on Computer Vision,* pp. 454-463

Berndt, D. J & Clifford, J. (1996). Finding Patterns in Time Series: A Dynamic Programming Approach, *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, Menlo Park, CA, pp. 229-248

Bourel, B.; Chibelushi, C. C. & Low, A. A. (2002). Robust Facial Expression Recognition Using a State-Based Model of Spatially-Localised Facial Dynamics, *Proceedings of the 5th International Conference on Automatic Face and Gesture Recognition*, pp. 106-111

Das, G.; Gunopulos, D & Mannila, H. (1997). Finding Similar Time Series, *Proceedings of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery*, pp. 88-100

Dietterich, T. G. (2000). Ensemble Methods in Machine Learning, *Proceedings of the 1st International Workshop on Multiple Classifier Systems,* pp. 1-15

Donato, G.; Bartlett, M. S.; Hager, J.C.; Ekman, P. & Sejnowski, T. J. (1999). Classifying Facial Actions, *Pattern Analysis and Machine Intelligence*, Vol. 21, No. 10, pp. 974-989

Ekman, P & Friesen, W. (1978). Facial Action Coding System: A Technique for the Measurement of Facial Movement, *Consulting Psychologists Press*, Palo Alto, CA

Essa I. A. & Pentland, A. P. (1997). Coding, Analysis, Interpretation, and Recognition of Facial Expressions, Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, pp. 757-763

Freund, Y. & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences*, Vol. 55, No.1, pp. 119-139

Grauman, K. & Darrell, T. (2005). Efficient Image Matching with Distributions of Local Invariant Features, *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* Vol.2, pp. 627-634

Leibe, B. & Schiele, B. (2003). Analyzing Appearance and Contour Based Methods for Object Categorization, *Proceedings of International Conference on Computer Vision and Pattern Recognition,* pp. 409-415

Lowe, D. (2004). Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110

Krawiec, K. & Bhanu, B. (2003). Visual Learning by Evolutionary Feature Synthesis, *Proceedings of the 20th International Conference on Machine Learning,* pp. 376-383

Marée, R.; Geurts, P.; Piater, J. & Wehenkel L. (2005). Random Subwindows for Robust Image Classification, *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* Vol.1, pp. 34-40

Marr, D. (1982). *Vision,* W. H. Freeman and Company

Nakamura, M.; Nomiya, H & Uehara, K. (2004). Improvement of Boosting Algorithm by Modifying the Weighting Rule, *Annals of Mathematics and Artificial Intelligence*, Vol.41, pp. 95-109

Nakamura, T.; Taki, K. & Uehara, K. (2007). Time Series Classification by Angular Metrics for Shape Similarity, *Proceedings of Workshop and Challenge on Time Series Classification (CTSC '07/KDD 2007)*, pp. 42-49

Nomiya, H. & Uehara, K. (2005). Feature Construction and Feature Integration in Visual Learning, *Proceedings of ECML2005 Workshop on Sub-symbolic Paradigm for Learning in Structured Domains,* pp. 86-95

Nomiya, H. & Uehara, K. (2007). Multistrategical Image Classification for Image Data Mining, *Proceedings of International Workshop on Multimedia Data Mining*, pp.22-30

Osaki, R.; Shimada, M. & Uehara, K. (2000). A Motion Recognition Method by Using Primitive Motions, *Proceedings of the 5th IFIP 2.6 Working Conference on Visual Database Systems*, pp. 117-128

Schiele, B. & Crowley, J. L. (2000). Recognition without Correspondence using Multidimensional Receptive Field Histograms, *International Journal of Computer Vision,* Vol. 36, No. 1, pp. 31-50

Swain, M. J. & Ballard, D. H. (1991). Color indexing, *International Journal of Computer Vision,* Vol.7, No.1, pp. 11-32

Tu, Z. (2005). Probabilistic Boosting-Tree: Learning Discriminative Models for Classification, Recognition, and Clustering, *Proceedings of the 10th IEEE International Conference on Computer Vision,* Vol.2, pp. 1589-1596

Turk, M. A. & Pentland, A. P. (1991). Face Recognition Using Eigenfaces, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition,* pp.586-591

Viola, P. & Jones M. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition,* pp.511-518

Data Mining and Knowledge Discovery in Real Life Applications

Edited by Julio Ponce and Adem Karahoca

ISBN 978-3-902613-53-0
Hard cover, 436 pages
Publisher I-Tech Education and Publishing
Published online 01, January, 2009
Published in print edition January, 2009

This book presents four different ways of theoretical and practical advances and applications of data mining in different promising areas like Industrialist, Biological, and Social. Twenty six chapters cover different special topics with proposed novel ideas. Each chapter gives an overview of the subjects and some of the chapters have cases with offered data mining solutions. We hope that this book will be a useful aid in showing a right way for the students, researchers and practitioners in their studies.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Hiroki Nomiya and Kuniaki Uehara (2009). Content-based Image Classification via Visual Learning, Data Mining and Knowledge Discovery in Real Life Applications, Julio Ponce and Adem Karahoca (Ed.), ISBN: 978-3-902613-53-0, InTech, Available from: http://www.intechopen.com/books/data_mining_and_knowledge_discovery_in_real_life_applications/content-based_image_classification_via_visual_learning

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821