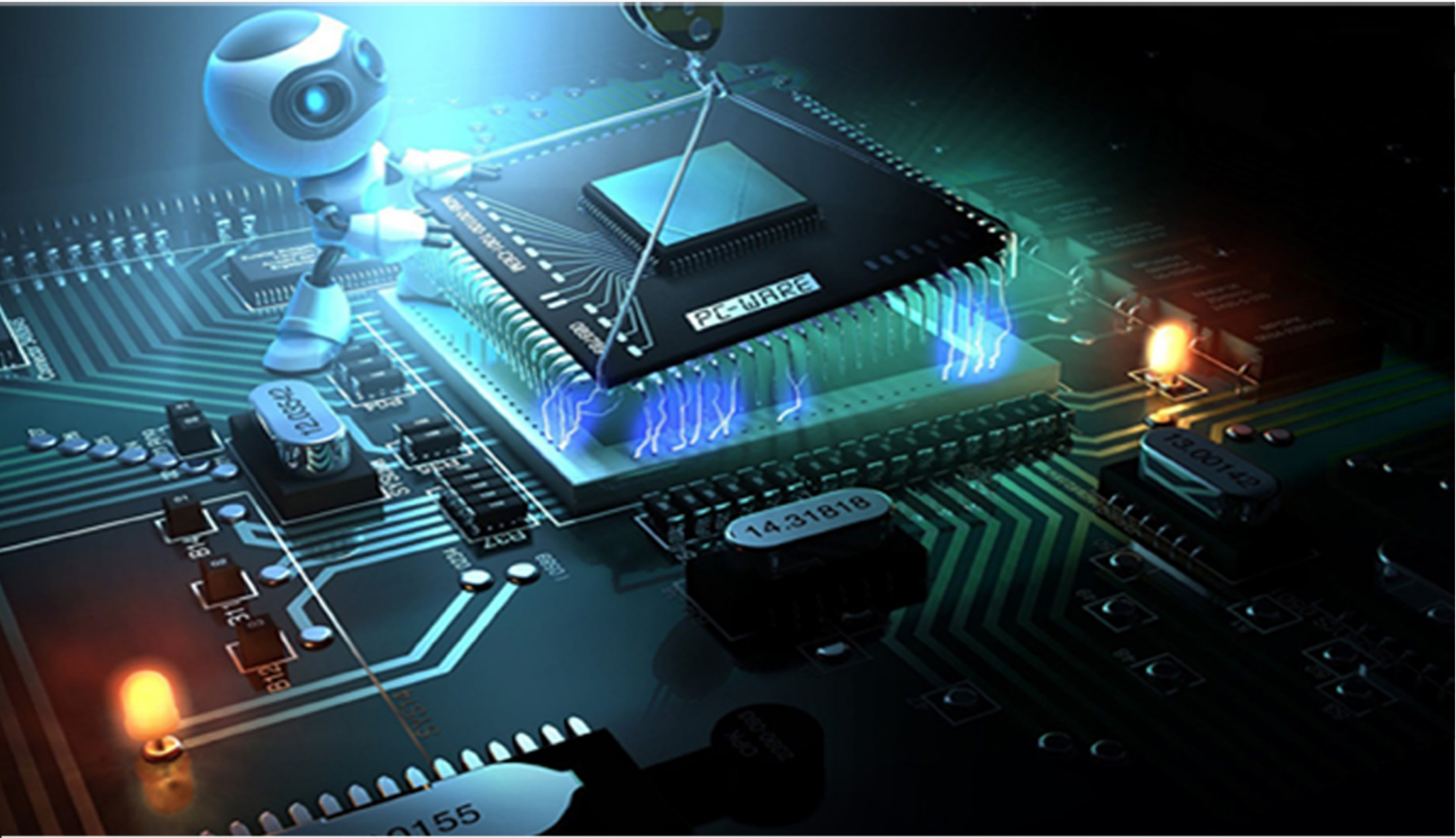
ISSN: 2054-7390

Volume 2. Issue 1 - February 2014

Transactions on Machine Learning and Artificial Intelligence



Society for Science and Education
UNITED KINGDOM

TABLE OF CONTENTS

EDITORIAL ADVISORY BOARD

DISCLAIMER

All the contributions are published in good faith and intentions to promote and encourage research activities around the globe. The contributions are property of their respective authors/owners and the journal is not responsible for any content that hurts someone's views or feelings etc.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TMLAI TRANSACTIONS ON
MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

VOLUME 2, ISSUE 1
ISSN: 2054 - 7390

Neural Networks Trained by Randomized Algorithms

Qin Qin[1]**, Qing-Guo Wang**[2]**, Shuzhi Sam Ge**[3] **and Chao Yu**[4]

*Department of Electrical and Computer Engineering, National University of Singapore,
Singapore, 117576.*

g0800434@nus.edu.sg[1]*, elewqg@nus.edu.sg*[2]*, samge@nus.edu.sg*[3]*, yuchao@nus.edu.sg*[4]

ABSTRACT

In this paper, a new model framework is proposed where a group of neural networks are trained with randomized algorithms. By incorporating randomization of random forests into a training algorithm of a neural network, the repeated running of such a revised training algorithm yields multiple independent neural networks. This group of multiple models jointly may outperform individual models. Simulation studies are conducted on various examples including practical ones such as stock markets and show that the proposed model group overall performs better than single neural network and a random forest. When there is significant noise in the data set, the performance of the former drops relatively less than the latter. In particular, the former produces much lower deviation of the performance and higher mean performance compared with the latter. Therefore, the proposed method has strong ability to classify the noisy data and perform robustly.

Keywords: Neural networks, Random forests, Classification, Complex systems, modeling.

1. INTRODUCTION

Neural network [1, 2] is well known in the area of machine learning and used in a wide range of applications for modeling complex systems. It was used as a basic tool to model the activities of the human brain, and learning mechanisms were designed to acquire knowledge. The process of learning for a neural network consists of adjusting the weights of its nodes to minimize a chosen cost functional [3, 4]. Since the nature of the neural network of biology is nonlinear, the neural network is a technique that can be used to handle nonlinear problems. In applications, the neural network is often utilized to recognize the patterns hidden in the input data and output data by exploring the complicated relationships between them. In [5], Sneak Circuit data were modeled with a neural network on back propagation algorithm with the prediction accuracy rate of up to 60%, which is high in the field of Sneak Circuit prediction. The neural network was combined with Bayesian algorithm in [6] to predict the skewed data. A

DOI: 10.14738/tmlai.21.39
Publication Date:10th February 2014
URL: http://dx.doi.org/10.14738/tmlai.21.39

novel neural network was proposed in [7] with multiple prior knowledge, where its structure consists of three layers with hybrid feed forward modes and it is trained by sequential quadratic programming with experimental results in the industrial processes. In [8], the optimal selection was made from sigmoid, radial basis and polynomial functions. An ensemble algorithm was presented in [9] for neural networks for the prediction of wind power, which shows that the prediction accuracy is enhanced compared with the one produced by single neural network. The k-mean algorithm was incorporated into neural networks to form an ensemble framework in dealing with the classification problem on digital mammograms [10], which shows improved accuracy rate compared with existing framework. The random neural network proposed by Gelenbe [11] has an interconnected network of neurons which exchanges spiking signals and involves complicated feedback loops.

A decision tree [12, 13] is a very different model from a neural network. Its model structure mimics a human decision making process, but not a human brain. Its learning process is also different and carried out in stages by separating a data set into branch-like segments with splitting rules. In a tree, the original node contains the entire data. The terminal nodes are called the leaves, and used for predictions. A random forest [14, 15] is a group of multiple decision trees. These trees in a forest are obtained independently through randomized splitting rules. Each tree grows with a random vector which is sampled independently from a distribution. The sampling for all the trees in the random forest is performed under the same distribution. When the size of the random forest increases, the generalization error of the random forest can converge to a limit, which makes the random forest predictions robust to noise. It is shown [16] that a random forest is able to prevent over-fitting. Random forests have found applications in stock markets [17], where both technical indicators and financial news are used in the model and a higher return rate is obtained than the buy-and-hold strategy. In [18], the direction of the Indian stock market index was predicted using random forest, neural network and support vector machine. Random forests also have many applications in the field of classification problem. Yi et al. [19] use the random forest to classify the stellar spectral data and the experiment results show the random forest can have better classification efficiency and produce lower root mean square errors compared with neural network. In [20], random forest is used to classify the high-dimensional data that reflects patient response to drugs, a comparison experiment is conducted, and the results show that random forest has a stronger power in classifying such high-dimension data. It is shown [21] that better accuracy rates can be obtained by random forest in locating the desired region in segmenting an image.

It is noted that when a neural network model is employed as a single model, it may lack robustness, while a random forest usually adopts the majority rule from its various decision trees, which may enhance robustness but have poor predictions from individual trees, causing performance limitations for the entire forest. It would be desirable to combine the strengths of both techniques, that is, a group of models with the individual models similar to a neural network with good prediction expectation and its group similar to a random forest with prediction robustness, or reduction of variance of prediction errors.

In the view of the above observations, this paper aims to develop a group of neural networks by incorporating randomization during their learning process so as to outperform a neural network and a random forest. Our idea of randomization was motivated by the essence of random forests. However, when attempting to apply the idea of random forests to neural networks, one realizes that two models' structures and their learning techniques are extremely different: the former is actually locally trained in the sense that the training process works on a smaller subset of the data at each stage of learning using a subset of inputs (actually a small subset of the entire inputs), whereas the latter is always globally trained with the full data set at each stage of learning using the complete set of inputs. The extension of random forests to neural networks is not reported in the literature, to our best knowledge, and deserves a careful technical development, which is presented in this paper. It turns out that our new technique can enhance performance compared with either of random forest and neural network when the data is difficult to learn. This should be valuable for modeling of noisy data. For ease of exposition, we consider classification problems in this paper, since the regression problem can be treated similarly with no technical differences or difficulties. It should be also pointed out that the neural network group devised in this paper is different from the standard neural network ensembles in the literature, where either some averaging is used to produce an ensemble from individual neural networks or individual neural networks act on different subsets of the data. Our model works like a random forest, where trees are replaced by neural networks.

The rest of this chapter is organized as follows. The neural network and the random forest are reviewed briefly in Sections 2 and 3, respectively. The proposed method is developed in Section 4. An illustrative example, popular examples and practical examples are presented in Sections 5, 6 and 7, respectively. The paper is concluded in Section 8.

2. NEURAL NETWORKS

A simple neural network is shown in Figure 1. The neural network modeling deals with the following:

1) The function that computes the output from the input based on the weights.

2) The training process that updates the weights between the neurons of different layers.

The function $f(x)$ of a neuron network can be a combination of other functions $h_i(x)$, which can also be the combinations of others functions. The nonlinear weighted sum is a widely adopted form that is utilized in neural network, and it can be described by:

$$f(x) = Q(\sum_i w_i h(x)), \tag{1}$$

where Q is the function that is predefined, for example, the hyperbolic tangent.

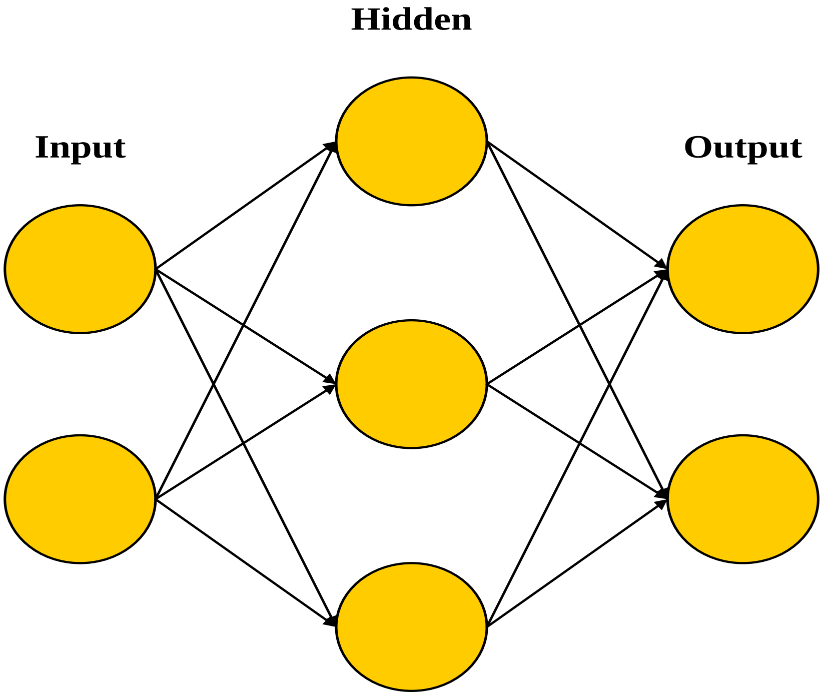


Figure 1 Structure of a simple neural network.

For training, one defines a cost function $J: F \to R$, and then find the optimal f^* such that $J(f^*) \le J(f)$, $\forall f \in F$. For example, we have the data of (x, y), where x represents the input and y the output. We define the cost function:

$$J = E[(f(x) - y)^2]. \tag{2}$$

The task is to find optimal f^* that minimizes the cost function:

$$E[(f^*(x) - y)^2] \le E[(f(x) - y)^2], \quad \forall f \in F. \tag{3}$$

3. RANDOM FORESTS

Random forest is an ensemble of many decision trees. The decision tree is a branch-like graph for the process of decision making. A simple decision tree is depicted in Figure 2. The decision tree splits the data set into subsets according to the splitting criterions. This splitting process is repeated on each subset recursively.

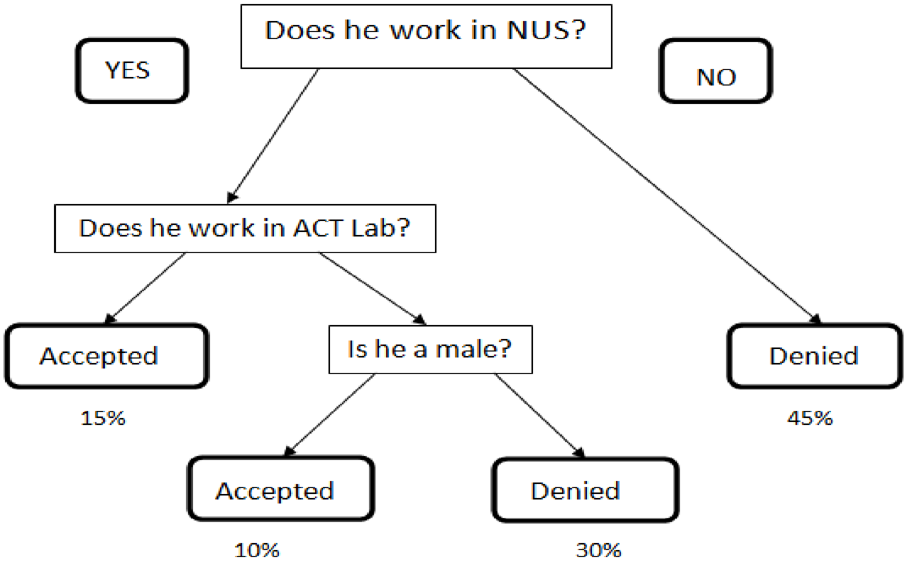


Figure 2 Structure of a simple decision tree.

For a tree, the output is obtained from the input along the way from the root node to the corresponding leaf node.

Let the number of input variables be Q. A random forest is formed by growing each decision tree as follows:

1) Randomly select q variables from Q variables and conduct the splitting at each node;

2) Let the tree grows fully with no pruning;

until the number of trees reaches the desired N. With a random forest, for a test data point, one obtains from each decision tree, its classification label which is called “vote”, and decide the final classification label with the highest votes.

4. THE PROPOSED MODEL

We view a neural network like a decision tree and try to design some randomized algorithm to find a set of neural networks independently. Note that a neural network is trained at each stage of iterative learning with the full data set (this discussion has nothing to do with cross-validation or bootstrap et al, which uses a subset of the data) based on the full set of inputs, and its structure (the layers and neurons) is fixed before and during the training. This is very different from a tree for which its branches and nodes are produced gradually during the training. Due to the above essential difference, it does not look feasible to mimic construction of a random forest by training a neural network with different combinations of inputs, and/or from different subsets of data at different stages. Further, if one chooses different structures (layers and neurons) for different neural networks, there seems no good rational to make such a choice (even for single neural network). As a result, we suppose that N neural networks to be trained have the same structure. Then, the question is how to generate multiple neural networks independently, givens the same structure for all N models and the same set of date and inputs at each training stage? Note that when the structure is fixed, a particular neural network is determined by its weights. Different weights give different neural networks. This led us to devising the changes to the training algorithms so as to randomize certain aspects of a chosen training algorithm to find *N* neural networks independently. Then, each neural network acts like a decision tree and the resulting neural network group looks like a random forest. It is expected that a single neural network may perform better than a single decision tree. Hopefully, with the same degree of randomization and independence of the individual models as the trees, the resulting neural network group performs better than a random forest.

In the view of the above observations, we look at some training algorithm to introduce randomization. The back propagation algorithm of neural networks is widely used and thus taken for our study and illustration. The other algorithms can be chosen as well and the similar development as below can be done easily. Suppose that the training set is given as $\{(x_1, y_1), (x_2, y_2), ...(x_n, y_n)\}$, where x_i , $i = 1, 2, ..., n,$ are the input vectors, and y_i , $i = 1, 2, ..., n,$ are the corresponding outputs. Let the predicted outputs from a neural network is $\hat{y}_i$, $i = 1, 2, ..., n$. The backpropagation algorithm seeks to minimize the following error function:

$$E = \frac{1}{2}\sum_{i=1}^{n}\left\|\hat{y}_i - y_i\right\|^2 . \tag{4}$$

One chooses the initial weights, w_i^0 , $i = 1, 2, ..., m$, and update the weights as follows:

$$w_i^{k+1} = w_i^k - \alpha\frac{\partial E}{\partial w_i}, \ i = 1, 2, ..., m, \tag{5}$$

where α is a learning constant. This updating carries on recursively till $\nabla E = 0$ holds approximately.

Note that the above back propagation algorithm updates all the weights together at each iteration. We now introduce its randomization: randomly choose *p* weights from the total *m* weights to update with (5) while the remaining weights are kept unchanged, at each iteration, and carry on till $\nabla E = 0$ holds approximately. As a result, a neural network has been obtained. Next, choose a new set of initial weights randomly and repeat the above iterations to find the second neural network, and so on till N neural networks are all obtained.

The proposed method is summarized as follow:

1) Set up *N* identical neural networks architectures.

2) For each neural network: set the initial weights randomly; and adopt the back propagation algorithm with the following change: at each iteration, choose randomly *p* weights only to implement the updates while keeping the remaining (*m-p*) weights unchanged.

3) Assign the output to the label which has the most votes from *N* trained neural networks.

Breinman [15] recommends that for a random forest, the number of trees in a forest should be a large number, and the number of randomly selected features at each splitting should be equal to the root square of the number of all features. In this paper, we follow Breinman's above recommendations with some adaptation. In particular, suggest that the number of neural networks in the proposed framework be 50, and the number of randomly selected weights be the root square of the number of all weights by comparing the input number in random forest to the number of weights in our framework, that is, $N = 50$ and $p = \sqrt{m}$.

5. ILLUSTRATIVE EXAMPLE

From this section onwards, we conduct comparative simulation studies of three modeling methods: a single neural network (SNN), a random forest (RF) and the proposed group of multiple neural networks with randomized algorithms (MNN). Throughout these sections, we set $N = 50$ and $p = \sqrt{m}$. This implies that a RF consists of 50 decision trees and an MNN group consists of 50 individual neural networks. We make ten random runs for each of RF, SNN and MNN, from which, the best and worst performance as well as the average performance are evaluated on the test data, respectively, for comparison.

For a start, this section presents an illustrative example. We randomly select 4000 data points in the square formed by the four points: [-1 -1], [-1 1], [1 1] and [1 -1]. Choose this simple function:

$$y = x^2 - 0.25, \qquad (6)$$

to divide the square into two areas. It happens that 2297 data points are above the function curve and we assign their labels as 1, while 1703 data points are below the function curve and are assigned with the label of 0. All the 4000 data points are used as the training data for the experiment. The test data are generated by additional randomly-drawn 400 data points in the same square. Once again, the resulting 211 data points above the function curve are labeled as 1, while others as 0. This is of course a separable case. Later we introduce noise in the data to make the data worse and worse, and thus classification more and more difficult.

Test results are presented in Table 1, where P1 means predicted 1, P0 predicted 0, T1 is true 1, and T0 true 0. To find the accuracy rate from the table, take the average case of SNN for consideration. Since the number of data points with true 1 and predicted 1 is 210 while the number of data points with true 0 and predicted 0 is 186.1, the total number of correct predictions is (210+186.1), which is divided by the total number of test points, 400, to give the classification accuracy rate of 99.03%. In the same manner, the average accuracy rates for RF and MNN are obtained as 98.83% and 98.65%, respectively. The best accuracy rates for them are 99.75%, 99.25% and 99.00%, respectively, whereas the worst accuracy rates are 98.00%, 98.75% and 98.00%, respectively. It follows that the performance of MNN is not better than that of RF and SNN when the data set is easy to classify.

Table 1 Classification for illustrative data (0% noise level).

	SNN		RF		MNN		
	P1	P0	P1	P0	P1	P0	
Average	210	1	210.1	0.9	209.3	1.7	T1
	2.9	186.1	3.8	185.2	3.7	185.3	T0
Best	211	0	211	0	210	1	T1
	1	188	3	186	3	186	T0
Worst	208	3	210	1	209	2	T1
	5	184	4	185	6	183	T0

In order to check the performance on the noisy data, we randomly select 10% of data points from in both training and testing data sets of the first group with label of 1 (above the function curve) and change their labels to 0, and also select 10% of data points in both training and testing data sets from the second group (below the function curve) with label of 0 and change their labels to 1. Then, we re-run the simulation on the changed data. The results are shown in Table 2. The average accuracy rates for SNN, RF and MNN become 99.00%, 95.86% and 98.70%, respectively. The best accuracy rates for them are 99.75%, 96.50% and 98.75%, respectively, whereas the worst accuracy rates are 98.00%, 95.25% and 98.50%, respectively. Thus, the performance of MNN has improved a lot relatively to other two methods and is better than RF now when the data set is added with 10% noise.

Table 2 Classification for illustrative data (10% noise level)

	SNN		RF		MNN		
	P1	P0	P1	P0	P1	P0	
Average	210.1	0.9	201.9	9.1	209	2	T1
	3.1	185.9	7.4	181.6	3.2	185.8	T0
Best	210	1	204	7	209	2	T1
	0	189	7	182	3	186	T0
Worst	206	5	201	10	209	2	T1
	3	186	9	180	4	185	T0

To see the performance on the data set with higher noise levels, we conduct simulation for the noise levels of 20%, 30%, 40% and 50%, respectively. The classification accuracies at all the noise levels are summarized in Table 3. It is seen from Table 3 that

when the data set has no noise or small noise (10% noise level), three methods perform similarly and all achieve the high accuracy rates.

when the data set has significant noises (20%, 30%, 40% and 50% noise levels), SNN and MNN perform much better than RF. Note that MNN produces a much stable performance (much lower performance deviation between the best and worst). In particular, the worst performance of MNN is much better than SNN, and the average performance of MNN is better than SNN in most cases as well.

The results indicate that the proposed method has more power than SNN and RF in classifying the noisy data set.

Table 3 Classification accuracy relative to noise level (illustrative data).

Noise level	Method	Average	Best	Worst
No noise	SNN	99.03%	99.75%	98.00%.
	RF	98.83%	99.25%	98.75%
	MNN	98.65%	99.00%	98.00%
10% noise	SNN	99.00%	99.75%	98.00%
	RF	95.86%	96.50%	95.25%
	MNN	98.70%	98.75%	98.50%
20% noise	SNN	98.30%	99.50%	97.00%
	RF	89.78%	90.75%	88.50%
	MNN	98.25%	98.50%	98.00%
30% noise	SNN	96.45%	98.25%	92.50%
	RF	82.05%	84.25%	80.50%
	MNN	96.95%	97.50%	96.25%
40% noise	SNN	90.78%	93.00%	85.75%
	RF	66.48%	68.50%	63.75%
	MNN	92.73%	93.50%	92.00%
50% noise	SNN	54.30%	66.25%	47.75%
	RF	50.08%	53.00%	47.25%
	MNN	61.03%	64.75%	56.50%

6. POPULAR EXAMPLES

6.1 Fisher's Iris data

This data is included in the Matlab software, where its description is available from Matlab helps and cited as follows: "Fisher's iris data consists of measurements on the sepal length, sepal width, petal length, and petal width of 150 iris specimens. There are 50 specimens from each of three species." In this data set, three species are "setosa", "versicolor" and "virginica". Each species consist of 50 specimens. We take 120 data points (40 data points of "setosa", 40 data points of "versicolor" and 40 data points of "virginica" to maintain class ratios same as those in the raw data) as our training set and the remaining 30 data points (10 data points of "setosa", 10 data points of "versicolor" and 10 data points of "virginica") as the testing set.

The simulation results are shown in Table 4, where P2 means predicted "setosa", P1 means predicted "versicolor", P0 means predicted "virginica"; T2 means true "setosa", T1 means true "versicolor", T0 means true "virginica". From Table 4, we can find the average accuracy rates for SNN, RF and MNN as 95.00%, 91.67% and 96.00%, respectively. The best accuracy rates for these three methods are 100.00%, 93.33% and 96.67%, respectively. The worst accuracy rates for the three methods are 90.00%, 90.00% and 93.33%, respectively. It follows that averagely MNN produces the best results when the data set has no noise. Although the best accuracy rate of SNN is 100%, MNN gives the highest accuracy rate on worst performance. RF gives poor accuracy rates.

Table 4 Classification for fisher's iris data (0% noise level)

	SNN			RF			MNN			
	P2	P1	P0	P2	P1	P0	P2	P1	P0	
Average	10	0	0	10	0	0	10	0	0	T2
	0	10	0	0	10	0	0	10	0	T1
	0	1.5	8.5	0	2.5	7.5	0	1.2	8.8	T0
Best	10	0	0	10	0	0	10	0	0	T2
	0	10	0	0	10	0	0	10	0	T1
	0	0	10	0	2	8	0	1	9	T0
Worst	10	0	0	10	0	0	10	0	0	T2
	0	10	0	0	10	0	0	10	0	T1
	0	3	7	0	3	7	0	2	8	T0

In order to check the performance on noisy data, we randomly select 30% data points that is labeled as "setosa" and change their labels to "versicolor", then we randomly select 30% data points that is labeled as "versicolor" and change their labels to "virginica", finally we randomly select 30% data points that is labeled as "virginica" and change their labels to "setosa". The resulting simulation is exhibited in Table 5. From Table 5, we can determine the average accuracy rates for SNN, RF and MNN as 76.67%, 60.00% and 81.00%, respectively. The best accuracy rates are 86.67%, 73.33% and 83.33%, respectively. The worst accuracy rates are 56.67%, 56.67% and 76.67%, respectively. From the accuracy results, we find that averagely MNN yields the best results when the data set is added with 30% noise. RF still gives poor accuracy rates which decrease obviously. Although the best accuracy rate of SNN is 86.67%, MNN gives a much higher accuracy rate on worst performance than SNN whose accuracy rate is only 56.67%.

Table 5 Classification for fisher's iris data (30% noise level).

	SNN			RF			MNN			
	P2	P1	P0	P2	P1	P0	P2	P1	P0	
Average	7.8	2.2	0	5.6	4.4	0	8	2	0	T2
	1.8	7.6	0.6	0	8.1	1.9	1.8	8.2	0	T1
	0.9	1.5	7.6	2.5	3.2	4.3	0	1.9	8.1	T0
Best	8	2	0	7	3	0	9	1	0	T2
	0	10	0	2	7	1	2	8	0	T1
	0	2	8	0	2	8	0	2	8	T0
Worst	3	7	0	5	5	0	8	2	0	T2
	3	6	1	0	7	3	3	7	0	T1
	0	2	8	0	5	5	0	2	8	T0

In summary, we put the accuracy rates of two cases together in Table 6 for easy comparison. From Table 6, we can see that MNN averagely produces the best results both with and without added noise. RF gives the poorest accuracy rates. The performance of RF decreases apparently when the data set is added with noise.

In the case of 30% noise added, although SNN can achieve 86.67% classification accuracy rate in its best results, its worst accuracy rate is very low (56.67%). On the contrary, MNN achieves 83.33% classification accuracy rate in its best results, while its worst accuracy rate remains high (76.67%). The results demonstrate that MNN has more power in classifying noisy data than SNN and RF.

Table 6 Classification accuracy relative to noise level (fisher's iris data)

Noise level	Method	Average	Best	Worst
No noise	SNN	95.00%	100.00%	90.00%
	RF	91.67%	93.33%	90.00%
	MNN	96.00%	96.67%	93.33%
30% noise	SNN	76.67%	86.67%	56.67%
	RF	60.00%	73.33%	56.67%
	MNN	81.00%	83.33%	76.67%

6.2 Crabs classification data

This data is included in the Matlab software, where its description is available from Matlab helps and cited as follows: "In this demo we attempt to build a classifier that can identify the sex of a crab from its physical measurements. Six physical characteristics of a crab are considered: species, frontal lip, rear width, length, width and depth. The problem on hand is to identify the sex of a crab given the observed values for each of these 6 physical characteristics." In this data set, we randomly select 160 data points as the training set and take the remaining 40 data points as the testing set.

The simulation results are shown in Table 7, where P1 means predicted "Male", P0 means predicted "Female"; T1 means true "Male", T0 means true "Female". From Table 7, we obtain the average accuracy rates for SNN, RF and MNN as 88.00%, 80.50% and 87.75%, respectively. The best accuracy rates are 95.00%, 85.00% and 90.00%, respectively. The worst accuracy rates are 77.50%, 77.50% and 85.00%, respectively. Thus, RF gives the poorest accuracy rates. SNN and MNN are similar in terms of average performance. The former is better with best performance while the latter is better with worst performance when these two methods are compared with each other.

Table 7 Classification for crabs classification data (0% noise level).

	SNN		RF		MNN		
	P1	P0	P1	P0	P1	P0	
Average	19.8	2.2	16.3	5.7	19	3	T1
	2.6	15.4	2.1	15.9	1.9	16.1	T0
Best	22	0	18	4	20	2	T1
	2	16	2	16	2	16	T0
Worst	19	3	16	6	18	4	T1
	6	12	3	15	2	16	T0

In order to check the performance on noisy data, we randomly select 30% data points that is labeled as “Male” and change their labels to “Female”, also we randomly select 30% data points that is labeled as “Female” and change their labels to “Male”. The resulting simulation is exhibited in Table 8. From Table 8, we calculate the average accuracy rates for SNN, RF and MNN as 67.50%, 56.25% and 80.00%, respectively. The best accuracy rates are 82.50%, 62.50% and 82.50%, respectively. The worst accuracy rates are 47.50%, 52.50% and 75.00%, respectively. From the accuracy results, we find that averagely the method of MNN gives the best results when the data set is added with 30% noise. RF still produces poorest accuracy rates. Although the best accuracy rate of SNN is the same with that of MNN, SNN gives a much lower accuracy rate on worst performance than MNN.

Table 8 Classification for crabs classification data (30% noise level).

	SNN		RF		MNN		
	P1	P0	P1	P0	P1	P0	
Average	14.5	7.5	12.7	9.3	18.7	3.3	T1
	5.5	12.5	8.2	9.8	4.7	13.3	T0
Best	19	3	15	7	19	3	T1
	4	14	8	10	4	14	T0
Worst	4	18	11	11	18	4	T1
	3	15	8	10	6	12	T0

In summary, we put the accuracy rates of two cases together in Table 9 for easy comparison. From the table, we can see that RF produces the worst results both with and without added noise. The performance of MNN is almost the same with the performance of SNN when the data set has no noise. When the 30% noise is added, the accuracy rates of SNN and RF decrease sharply, while the performance of MNN is still good (80.00%). What is more important, the worst accuracy rate of MNN remains high in the case of 30% noise added, which results in a low deviation of the performance.

The results show that MNN has strong power in classifying noisy data set. It also demonstrates that the proposed method is more robust than the other two methods.

Table 9 Classification accuracy relative to noise level (crabs classification data)

Noise level	Method	Average	Best	Worst
No noise	SNN	88.00%	95.00%	77.50%
	RF	80.50%	85.00%	77.50%
	MNN	87.75%	90.00%	85.00%
30% noise	SNN	67.50%	82.50%	47.50%
	RF	56.25%	62.50%	52.50%
	MNN	80.00%	82.50%	75.00%

7. PRACTICAL EXAMPLES

7.1 Stock data

The stock data is always difficult to classify. In this experiment, we use the data from China stock market. To be concrete, we screen all the stocks over ten years (01.01.2001-31.12.2009) to pick up the cases for classification, using the following rule: today return rate (based on close prices) is above or equal to the band about its center line of the last p-day moving average of return rates, for which for illustration p is set as 20 and the band at 1.8 times the standard deviation of the return rates.

For each of these screened cases, we form its input vector x_i with:

- today's return rate
- yesterday's return rate
- the day before yesterday's return rate
- today's volume/(average volume of last 3 days)
- yesterday's volume/(average volume of last 3 days)
- the day before yesterday's volume/(average volume of last 3 days)

After input vectors are obtained, we design a trading rule in order to see its outcomes and define the output label y_i. We trade each screened case as follows:

Buy: today's close price;

Exit: 1.044 times the entry price in any of the following 3 days, or close price of the third day;

Stop: none.

From each round trade, there is an exit price. We then assign output label y_i to each case as follows:

y=1, if selling price >= 1.01 times buying price

y=0; if selling price < 1.01 times buying price

In the end, we have compiled a pair $[x_i \ y_i]$ for each case. Totally, there are 40,120 cases or data points. We randomly select 1600 points for classification experiment, of which 1200 points are assigned as the training data and the remaining 400 as the testing data.

With such a stock data set, the simulation results are shown in Tables 10 and 11. They indicate that;

- The performance of MNN is the best in all the methods,
- The accuracy rates of MNN are slightly better than the accuracy rates of SNN,
- RF gives the poorest performance.

Since the raw data are already very difficult to model, addition of artificial noise is not required. This experiment implies that our proposed method can work in such a real and very difficult application.

Table 10 Classification for stock data

	SNN		RF		MNN		
	P1	P0	P1	P0	P1	P0	
Average	10.7	141.3	24.7	127.3	8.1	143.9	T1
	14.4	233.6	42.2	205.8	9.6	238.4	T0
Best	13	139	25	127	10	142	T1
	12	236	36	212	9	239	T0
Worst	15	137	21	131	9	143	T1
	23	225	46	202	14	234	T0

Table 11 Accuracy rates of classification for stock data

Method	Average	Best	Worst
SNN	61.08%	62.25%	60.00%.
RF	57.63%	59.25%	55.75%
MNN	61.63%	62.25%	60.75%

7.2 MAGIC gamma telescope data 2004

We download the data from the website of Machine Learning Repository at http://archive.ics.uci.edu/ml/index.html. For completeness, we cite the data description from the website as follows:

"The data are MC generated (see below) to simulate registration of high energy gamma particles in a ground-based atmospheric Cherenkov gamma telescope using the imaging technique. Cherenkov gamma telescope observes high energy gamma rays, taking advantage of the radiation emitted by charged particles produced inside the electromagnetic showers initiated by the gammas, and developing in the atmosphere. This Cherenkov radiation (of visible to UV wavelengths) leaks through the atmosphere and gets recorded in the detector, allowing reconstruction of the shower parameters. The available information consists of pulses left by the incoming Cherenkov photons on the photomultiplier tubes, arranged in a plane, the camera. Depending on the energy of the primary gamma, a total of few hundreds to some 10000 Cherenkov photons get collected, in patterns (called the shower image), allowing to discriminate statistically those caused by primary gammas (signal) from the images of hadronic showers initiated by cosmic rays in the upper atmosphere (background).

Typically, the image of a shower after some pre-processing is an elongated cluster. Its long axis is oriented towards the camera center if the shower axis is parallel to the telescope's optical axis, i.e. if the telescope axis is directed towards a point source. A principal component analysis is performed in the camera plane, which results in a correlation axis and defines an ellipse. If the depositions were distributed as a bivariate Gaussian, this would be an equidensity ellipse. The *characteristic parameters of this ellipse (often called Hillas parameters) are among the image parameters that can be used for discrimination. The energy depositions are typically asymmetric along the major axis, and this asymmetry can also be used in discrimination. There are, in addition, further discriminating characteristics, like the extent of the cluster in the image plane, or the total sum of depositions."*

In this data set, we randomly select 1400 data points as a training set and other 400 data points as a testing set. The simulation results are shown in Tables 12 and 13.

Table 12 Classification for MAGIC gamma telescope data 2004.

	SNN		RF		MNN		
	P1	P0	P1	P0	P1	P0	
Average	80.2	119.8	102.6	97.4	77.8	122.2	T1
	20.4	179.6	40.5	159.5	2.9	197.1	T0
Best	79	121	112	88	87	113	T1
	2	198	40	160	3	197	T0
Worst	35	165	101	99	71	129	T1
	2	198	47	153	3	197	T0

Table 13 Accuracy rates of classification for MAGIC gamma telescope data 2004.

Method	Average	Best	Worst
SNN	64.95%	69.25%	58.25%
RF	65.53%	68.00%	63.50%
MNN	68.73%	71.00%	67.00%

From the tables, we can see

- The performance of MNN is the best in all the methods,
- SNN gives the poorest performance.

8. CONCLUSIONS

In this paper, we have proposed a new modeling tool – a group of multiple neural networks trained with randomized algorithms – for modeling complex data. It incorporates the randomization idea of random forests into a training algorithm of a neural network and the repeated running of such a revised training algorithm yields multiple independent neural networks. Such a group of models jointly may outperform individual models. Its effectiveness is demonstrated with a variety of examples including practical ones such as stock markets. It works particularly well for the data difficult to learn or model with the exiting methods.

Our simulation shows that the proposed method overall performs better than single neural networks and a random forest. When there is significant noise in the data set, the performance of the former drops relatively less fast than the latter. In particular, the former produces much lower deviation of the performance and high mean performance compared with the latter. Therefore, the proposed method has strong ability to classify the noisy data and perform robustly.

REFERENCES

[1]. Trevor Hastie, Robert Tibshirani and Jerome Friedman, The Elements of Statistical Learning, Springer New York, 2011.

[2]. Ben Krose, Patrick van der Smagt, An introduction to Neural Networks, 1996.

[3]. Hadzibeganovic, Tarik & Cannas, Sergio A., A Tsallis' statistics based neural network model for novel word learning, Physica A: Statistical Mechanics and its Applications 388 (5): 732–746. DOI:10.1016/j.physa.2008.10.042, 2009.

[4]. Van den Bergh, F. Engelbrecht, AP., Cooperative Learning in Neural Networks using Particle Swarm Optimizers, CIRG, 2000.

[5]. Zou Liping, Zou Tao, The application of neural network for Sneak Circuit Analysis on the aircraft electrical system, Prognostics and System Health Management Conference (PHM-Shenzhen), Page(s): 1-5, 2011.

[6]. Arsene. C.T.C., Lisboa. P.J., Bayesian Neural Network with and without compensation for competing risks, Neural Networks (IJCNN), The International Joint Conference, Page(s): 1-8, 2012.

[7]. Lou Haichuan, Su Hongye, Xie Lei, Gu Yong, Rong Gang, Multiple-prior-knowledge neural network for industrial processes, Automation and Logistics (ICAL), IEEE International Conference, Page(s): 385-390, 2010.

[8]. Kondo. T., Ueno. J., Nonlinear system identification by feedback GMDH-type neural network with architecture self-selecting function, Intelligent Control (ISIC), IEEE International Symposium, Page(s): 1521-1526, September 2010.

[9]. Shuang Han, Yongqian Liu, Jie Yan, Neural Network Ensemble Method Study for Wind Power Prediction, Power and Energy Engineering Conference (APPEEC), Asia-Pacific, Page(s): 1-4, March 2011.

[10]. McLeod. P., Verma. B., Clustered ensemble neural network for breast mass classification in digital mammography, Neural Networks (IJCNN), The International Joint Conference, Page(s): 1-6, June 2012.

[11]. E. Gelenbe, Random neural networks with negative and positive signals and product form solution, Neural Computation, vol. 1, no. 4, pp. 502–511, 1989.

[12]. Safavian S.R. and Landgrebe D., A survey of decision tree classifier methodology, Systems, Man and Cybernetics, IEEE Transactions on, volume: 21, pages: 660-674, 1991.

[13]. Anthony,J.M, Robert, N.F,Yang, L.,Nathaniel, A.W and Steven, D.B., An introduction to decision tree modeling, Journal of Chemometrics, 18: 275–285, 2004.

[14]. Yu. L. Pavlov., Random Forests, Utrecht, VSP, 2000.

[15]. Breiman, L., Random Forests. Machine Learning Journal 45, 532, 2001.

[16]. V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan and B. P. Feuston, Random forest: a classification and regression tool for compound classification and QSAR modeling, Journal of chemical information and computer sciences, volume: 43, pages: 1947--1958, 2003.

[17]. Maragoudakis, M and Serpanos, D., towards stock market data mining using enriched random forests from textual resources and technical indicators. AIAI, IFIP AICT 339, pp. 278–286, 2010.

[18]. Manish Kumar, Thenmozhi M., Forecasting Stock Index Movement: A Comparison of Support Vector Machines and Random Forest, Indian Institute of Capital Markets 9th Capital Markets Conference, January 2006.

[19]. Zhenping Yi, Jingchang Pan, Application of random forest to stellar spectral classification, Image and Signal Processing (CISP), 3rd International Congress, Volume: 7, Page(s): 3129-3232, 2010.

[20]. Dittman. D., Khoshgoftaar. T.M., Wald. R., Napolitano. A., Random forest: A reliable tool for patient response prediction, Bioinformatics and Biomedicine Workshops (BIBMW), IEEE International Conference, Page(s): 289-296, 2011.

[21]. Moschidis. E., Graham. J., Automatic differential segmentation of the prostate in 3-D MRI using Random Forest classification and graph-cuts optimization, Biomedical Imaging (ISBI), 9th IEEE International Symposium, Page(s): 1727-1730, 2012.

[22]. Breiman, L., Random Forests. Machine Learning Journal 45, 532, 2001.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TMLAI TRANSACTIONS ON
MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

VOLUME 2, ISSUE 1
ISSN: 2054 - 7390

Practical Virtual Use Method of CGI Program for User Creation Personal Portal Page Using DACS Web Service

Kazuya Odagiri[1], Shogo Shimizu[2] and Naohiro Ishii[3]
[1] Yamaguchi University, Ymaaguchi-shi, Ymaguchi, Japan
[2] Gakushuin Women's College, Software, Shinjuku, Tokyo, Japan
[3] Aichi Institute of Technology, Aichi, Japan
[1]odagiri@yamaguchi-u.ac.jp;[2]shogo.shimizu@gakushuin.ac.jp;[3]ishii@aitech.ac.jp

ABSTRACT

A personal portal, which is an entrance wherein each user can acquire the information that s/he is interested in on a network, is often used as an alternative means of communication. However, there were a number of problems in the existing personal portals. For example, because the Web page as a personal portal was generated by the program located on the specific Web server which is managed by a system administrator, it was not always ideal for all users. To solve this kind of problems, we developed two Web Service functions, which are realized on the network by introducing the Destination Addressing Control System (DACS) Scheme. These two Web Service functions are as next. The first is the function to extract the data for each user from a database and display it on the Web browser. The second is the function to retrieve the data for each user from a document medium and display it on the Web browser. Through these Web Service functions, each user can easily create a customized personal portal that displays personal information. Then, the above two functions were extended to manage information not only for each user but also for each group of users and for all users, and the extended two functions were integrated as a DACS Web Service. By using the DACS Web Service, each user can create and customize the Web page as a personal portal for practical usage in an individual organization. However, when the use of it in a real network is assumed, it is difficult to make the Web page as the personal portal with a complicated function by the present web page as the personal portal using DACS Web Service Therefore, in this study, functional extension is performed. To be concrete, we realize the method of using the Common Gateway Interface (CGI) program opened to Internet by a user unit, and improve the present Web page as the personal portal using DACS Web Service by the method

Keywords: Web Services; Common Gateway Interface; Destination Addressing Control System

DOI: 10.14738/tmlai.21.76
Publication Date:10th February 2014
URL: http://dx.doi.org/10.14738/tmlai.21.76

1. INTRODUCTION

Static Web pages are often used as an alternative means of communication in addition to e-mail by an unspecified number of users. However, they are unsuitable for communication among individual users for their individual purposes or interests. As an alternative means of communication, the personal portal is realized by a Web Service; it allows each user to change the contents of the Web page dynamically. The term “Web Service” refers to a service that is provided to users through a Web server, and the term “portal” often indicates a Web page for searching information [1][2] such as with Google or Yahoo. The meaning of the phrase “personal portal” used in this paper is different from that of the above “portal” as just described. Here, “personal portal” refers to an entrance where each user can acquire information that s/he is interested in on the network; it can display different information for individual users dynamically on a Web browser by using a program such as the Common Gateway Interface (CGI) [3]. As examples, Netvibes [4] and Google [5] are introduced. Therefore, personal portals are suitable for communication among individual users for their respective purposes or interests. To display the necessary information for an individual user on a Web browser, the information is searched and extracted from databases on the network, after which each user is notified by a program such as CGI. If each database is distributed on the network, the program used to retrieve the information is large and complicated. Given that the program is installed by a system administrator and cannot be freely changed by a user, using a Web page as a personal portal is not always ideal for all users. To solve this problem, a new form of personal portal, which each user can create and customize easily, is necessary. The Destination Addressing Control System (DACS) Web Service is proposed and examined to realize such a personal portal. This DACS Web Service is implemented by extending two types of Web Service functions, which are realized on the network introducing the DACS Scheme. Using this DACS Web Service, each user can easily create and customize a Web page as a personal portal for oneself.

The DACS Scheme is a network management scheme. The basic principle of the DACS Scheme is that the entire network system is managed through communication control for each user on the client machine [6]. Moreover, functional extension is achieved by allowing the communication control for each user to coexist with communication control for each client machine to use in a user group [7-8]. In addition, Secure DACS Scheme is proposed to solve a security problem [9]. Two types of Web Service functions are primitive functions as fundamental elements of DACS Web Service, which are also realized on the network introducing the DACS Scheme. However, they have following two functions. In the first function, data that is stored in the database and is dispersed on the network can be used efficiently [10]. In the second function, data that is stored in a document medium such as PDF or simple text files can be used efficiently. Using both functions, when different users input the same URL into the Web browser, the different information for each user is searched and extracted from the database or document medium and displayed on the Web browser. By incorporating various kinds of URLs into a static HTML, it has become possible file for each user to create and customize a Web page as a personal portal easily [11]. In addition, by extending and integrating two types of Web Service functions, we realized the DACS Web Service [12]. The DACS Web Service uses data stored in a database or document medium to enable individual users, groups of users, and for all users to send and receive information through the integrated user interface. However, when the use of it in a real network is assumed, it is difficult to make the Web page as the personal portal with a complicated function by the present web page using DACS Web Service Therefore, in this study, functional extension is performed. To be concrete, we realize the method of using the Common Gateway Interface (CGI) program opened to Internet by a user unit, and improve the present Web page using DACS Web Service by the method.

2. RELATED WORKS AND MOTIVATION

Existing personal portals are Web pages that serve as unified windows for data stored in the information systems of organizations. In the field of the personal portals, various studies have examined [13], [14], [15]. In addition, personal portals are developed in individual organizations for practical purposes, and commercial software packages are also used. The system known as “HInT” is an example developed in a university [16]. “Blackboard” [17] is an example of a software package often used in a university. In these cases, the basic mechanism of personal portals is described by the sequence of processes from (1) to (9) shown in Fig. 1. The arrows with the dotted lines show the actions that the user performs in the Web browser, whereas those with solid lines show the flow of processing. This mechanism is explained in sequence as follows. First, the user enters a URL as input into the Web browser (1). Next, the Web server corresponding to the URL is accessed (2), and the program corresponding to the URL is executed. At this point in time, the program on the Web server side does not have the user information (user name and password). Therefore, an input demand for user information is executed on the Web browser side (3). Subsequently, the user enters the user information as input (4), which is sent to the program on the Web server (5). In existing mechanisms, user authentication is performed at this point in time. If access is permitted, information related to the user is searched from data that is accumulated in advance (6). To accumulate the data, there are methods using relational databases or document medium. Following search, the data related to each user are extracted (7). Programs on the Web server side programs that receive the data, such as CGI, generate a Web page dynamically from the data. Then, the Web page is sent to the Web browser side (8). Thus, each user can view information which is related to him/herself and displayed in the form of a Web page (9).

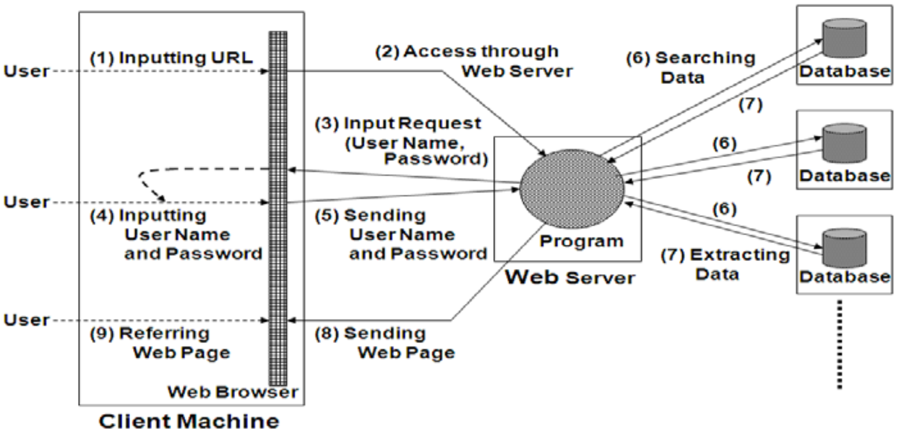


Fig. 1 Basic Mechanism of Existing Personal Portals

Moreover, as one of the methods to realize the personal portal, there is a mehtod by application frontend integration in Service Oriented Archtechture (SOA) [18][19]. As shown in Fig. 2, SOA, which is the study of information system integaration in organizations such as enterprises and universities, is realized by using WebAPIs. The mechanism is different from that of Fig. 1 in at least two points.

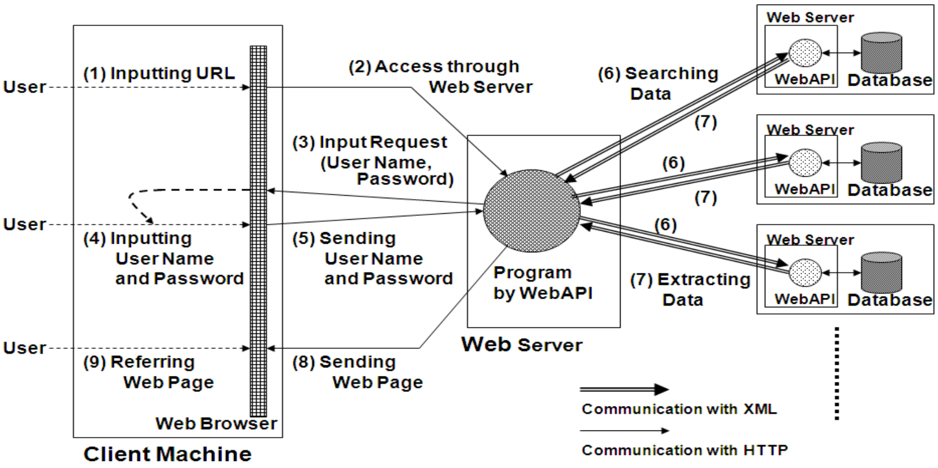


Fig. 2 Mechanism of Personal Portals by WebAPI

(1) The program on the Web server that the user accesses first is implemented with Web APIs such as Simple Access Object Protocol (SOAP) [20][21][22] and Representational State Transfer (REST) [23][24][25].

(2) Communication between (6) and (7) is performed with use of Extensible Markup Language (XML) [26][27] through each Web server distributed on the local area network.

The essential commonality of the mechanisms explained by here in Section 2 is that the program on the Web server extracts data from some database and generates the Web page as a personal portal. However, there were two problems with this general scheme.

(Problem 1) The program on the Web server is installed by a system administrator, and usually cannot be changed. Given that this program generates the Web page, it is not always easy for the user to customize the personal portal. Although this can be done in some cases within the specifications of the program, not all users can customize the Web page because the customization beyond these specifications is impossible.

(Problem 2) Although data from a database are used in existing personal portals, the data are often stored in the form of particular document medium in individual organizations. However, existing personal portals cannot use such data easily.

To solve these problems, two types of Web Service functions based on the DACS Scheme were proposed. By using the first function [10], problem 1 was solved. This enabled each user to create a Web page as a personal portal using data from the database. By using the second function [11], problem 2 was solved. This allowed the user to generate a Web page using data in the document medium. Using these functions, when different users enter same URL as input into Web browser, different information can be searched and extracted from the database or document medium and displayed on the Web browser. However, these functions allow one to deal only with information for individual users. In practice, these functions are insufficient in an organization.

Therefore, we proposed the DACS Web Service for improving the above problems. It has the functions of using information for groups of users and all users based on two functions of existing Web Service. In addition, the scheme integrated these two functions, and generated unified and personal user interfaces for each user. The personal portal for practical use in an individual organization was thus realized.

The following were improvements obtained over existing personal portal by incorporating the DACS Web Service into practical personal portals.

(1) Given that each user can create a Web page as a personal portal that fits his/her preference without the limitation of the program on the Web server, a user interface that is easy to use and suitable for practical use is realized

(2) Data stored not only in a database but also in a document medium, are used easily.

Next, we examined a new element for functional extension by access control technology for the data corresponding to position and the role of the user. Role-based Access Control (RBAC) [28], [29], [30] is available for this kind of access control. As RBAC enables access control corresponding to the role of individual user, access control for groups of users as well as all users is functionally possible.

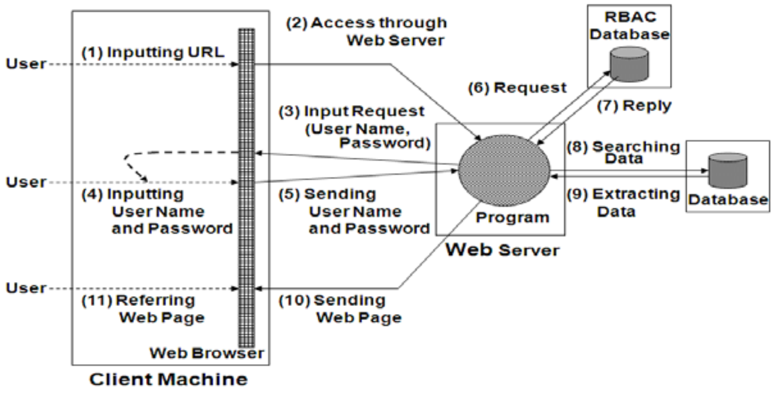


Fig. 3 Mechanism of Personal Portals with RBAC

The mechanism of personal portals when applying RBAC is shown in Fig. 3. The essential difference between Fig. 3 and Figs. 1, and 2 is that processes (6) and (7) are added in Fig. 3. These processes, which involve requesting and extracting access control information, must be performed between the gateway program on the Web server and the RBAC server that stores the access control information. When RBAC is applied to the two types of Web Service functions based on the DACS Scheme, the system configuration does not have the gateway program. Therefore, when an inquiry to a database is sent, the DACS Client must be extended to request and extract access control information from RBAC before the inquiry. In addition, because the DACS Scheme is the method used to maintain access control information in the DACS Server, but not in RBAC Server, access control information must be managed in two servers. This extension goes against the concept of the DACS Scheme, which is to control an entire network using a simple mechanism. Therefore, it is inappropriate to apply RBAC to two types of Web Service function based on the DACS Scheme. The extension was carried out in a form suitable to the DACS Scheme. As the result, we realized the Web page as a personal portal using the DACS Web Service, which was realized on the network introducing the DACS Scheme. *However, when the use of the Web page as the personal portal in a real network is assumed, it is difficult to make the Web page as the personal portal with complicated functions by the present web page using DACS Web Service. Therefore, in this study, functional extension is performed.*

In Section 3, a synopsis of the following is provided to explain the DACS Web Service: the DACS Scheme and the two types of functions based on the scheme, as well as the system configuration by these functions to realize the customized personal portal. In Section 4, the actual DACS Web Service is proposed. In particular, the features that are improved relative to the two types of the conventional Web Service functions are explained. In Section 5, after *the function of using the Common Gateway Interface (CGI) Program opened to Internet by a user unit, the virtual use method of the CGI using the function is shown.*

3. EXISTING DACS SCHEME

3.1 Functions of the DACS Scheme

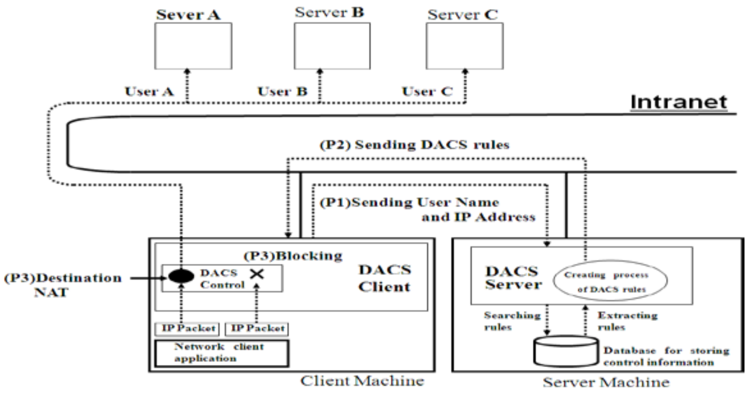


Fig. 4 Function of the DACS Scheme (1)

First, we provide a summary of the DACS Scheme. Figs. 4 and 5 show the functions of the network services according to the DACS Scheme. The DACS Server, which is located on a server machine, refers to the server software of the DACS Scheme. The DACS Client, which is located on each client machine, is the client software of the DACS Scheme. The DACS Control and DACS SControl in the DACS Client denote the control mechanisms for communications that are sent from the client software such as a Web browser. The DACS Control controls the normal communication from client software. On the other hand, the DACS SControl converts normal communication from the client software to Virtual Private Network (VPN) [31], [32], [33], [34] communications.

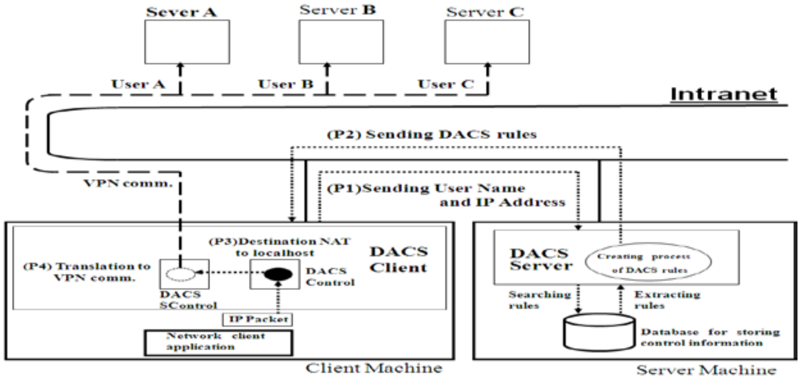


Fig. 5 Function of the DACS Scheme (2)

The DACS rules are the rules for controlling the DACS Control and DACS SControl. At the time of (a) or (b) below, the DACS rules (rules defined for each user) are distributed from the DACS Server to the DACS Client.

(a) When the user logs into the client machine

(b) When a notice is delivered from the system administrator

According to the distributed DACS rules, the DACS Client performs operation (1), (2) or (3) operations. Thereafter, communication control of the client machine is performed for each user who has logged in.

(1) The destination information on the IP Packet, which is sent from the client application, is changed by Destination Network Address Translation (NAT).

(2) The packet from the client machine, which is sent from the client application to out of the client machine, is blocked by a packet filtering mechanism.

(3) Communication between a client machine and a network server is supported by VPN with the port forward function of Secure SHell (SSH) [35], after the destination of the communication is changed to localhost (127.0.01) by function of (1).

An example of case (1) is shown in Fig. 4. Here, communication control by the system process (P3) is performed after the system proceeds from (P1) to (P2). Thereafter, the system administrator can distribute a communication of the user who has logged in to a specified server among servers A, B, or C. An example of case (2) is also shown. When the system administrator wishes to forbid a user to use Mail User Agent (MUA) as the client software of the control target, this is achieved by blocking the IP Packet with specific destination information.

Next, an example of case (3) is shown in Fig. 5. Here, communication control by the system proceeds (P3) and then, (P4) is performed after the system proceeds from (P1) to (P2). The communication is supported by VPN, and the system administrator can distribute the VPN communication of the user who has logged in to a specific server (A, B, or C). In the DACS Scheme, when a network service that handles personal and confidential information is controlled, the communication between a network server and client machine must be encrypted for protection. This function is used at that time.

In order to realize the DACS Scheme, communication controls on the client machine are performed by the DACS Protocol, as shown in Fig. 6. The DACS rules are distributed from the DACS Server to the DACS Client in (a) of Fig. 6, and applied to the DACS Control and the DACS S Control in (b) and (c) of Fig. 6. Normal communication control, such as modification of the destination information or communication blocking, is performed at the network layer in (d) of Fig. 6.

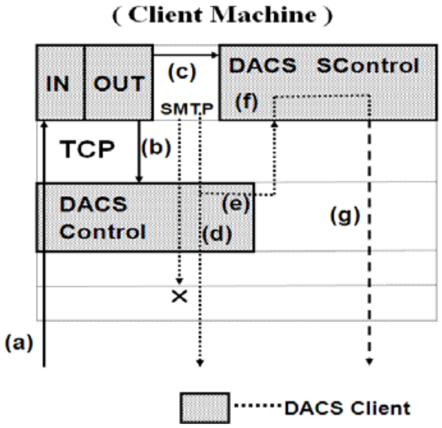


Fig. 6 Layer setting of the DACS Scheme

In case (1), as the function of destination change, the DACS rules are only applied to the DACS Control. The DACS rules consist of both destination-a (the destination IP address-a, destination port-a) before destination change and destination-b (destination IP address-b, destination port-b) after destination change. The communication sent from the client software is sent to destination-a. Thereafter, the destination of the communication is changed to destination-b by the control of destination NAT in the DACS Control, and communications are sent to the network server with this destination. In case (2), as the function of communication blocking, the DACS rules only are applied to the DACS Control similarly. The content of the DACS rules consists of destination-c (destination IP address-c, destination port-c) as the communication-blocking target. When the communication sent from the client software is destination-c, the communication is blocked by packet filtering in the DACS Control. In case (3), as the function of VPN communication, the DACS rules are applied to both the DACS Control and the DACS SControl. The content of the DACS rules consists of the following two rules.

(r1) Rules with both destination-d (destination IP address-d, destination port-d) before destination change and destination-e (127.0.0.1, port-e) after destination change

(r2) Rules with destination-f (destination IP address-f, destination port-f)

The DACS rules as (r1) are applied to the DACS Control and the DACS rules as (r2) are applied to the DACS SControl. Then, when a communication is supported by VPN, it is sent from (f) to (g) via (e). The VPN communication of (g) is sent by the DACS S Control. Using the port forwarding function of SSH, VPN communication which tunnels and encrypts the communication between a network server and a client machine with the DACS Client is realized. Normally, to communicate from the client software to a network server using port forwarding of SSH, it is necessary for the local host (127.0.0.1) to be indicated on that software as the communicating server. Using this function, transparent use of a client machine as a characteristic of the DACS Scheme is not failed. That the use of a client machine is transparent means that even if the configuration of the network servers is changed, the client machine can be used continuously without changing its setups. Communication control for this function is performed with the DACS SControl by SSH port forwarding. Through the use of these two functions, VPN or non-VPN communication for each network service can be selected for each user. In the case where non-VPN communication is selected, the communication control is performed by the DACS Control, as shown in (d) of Fig. 6. In the case where VPN communication is selected, the destination of the communication is changed by the DACS Control to the localhost. Then, the port number is changed to the number assigned for each communication. Subsequently, the communicating server is determined and the VPN communication is sent by the DACS SControl, as shown in (g) of Fig. 6. In the DACS rules applied to the DACS Control, the localhost is indicated as the destination of communication. In the DACS rules applied to the DACS SControl, the network server is indicated as the destination of communication. Thereafter, by changing the content of the DACS rules applied to the DACS Control and the DACS SControl, the control in the case of VPN communication or non-VPN communication is distinguished.

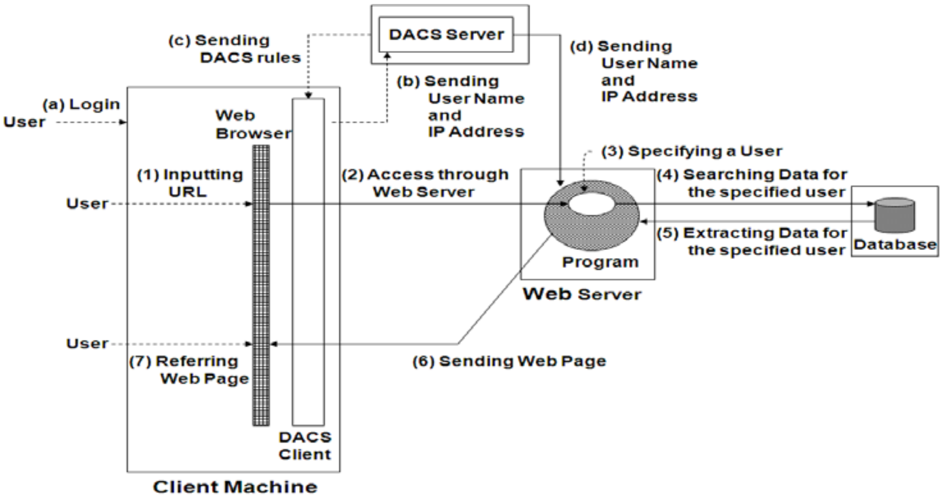


Fig. 7 Function Using Data from Database

3.2 Two Types of Web Service Functions based on DACS Scheme

In this Section, the network service corresponding to the DACS Scheme is explained. In the existing DACS Scheme, the communication control information for each user and each client machine has been maintained in the DACS rules on the DACS Server. By applying that information for communication control to the DACS Client (DACS Control) located on the client machine, communication from the client machine is controlled. As a result, the communication control mechanism on the network server is not required. However, on a network introducing the DACS Scheme, if a correspondence list of a client machine's IP address and user name is passed to the network server, it becomes possible to identify which user is sending the communication from a client machine. As a result, it becomes possible for a program on the network server to perform different processing for each user. A concrete example is the correspondence of the Web Service to the DACS Scheme. As the example, two types of Web Service functions based on the DACS Scheme are described as follows.

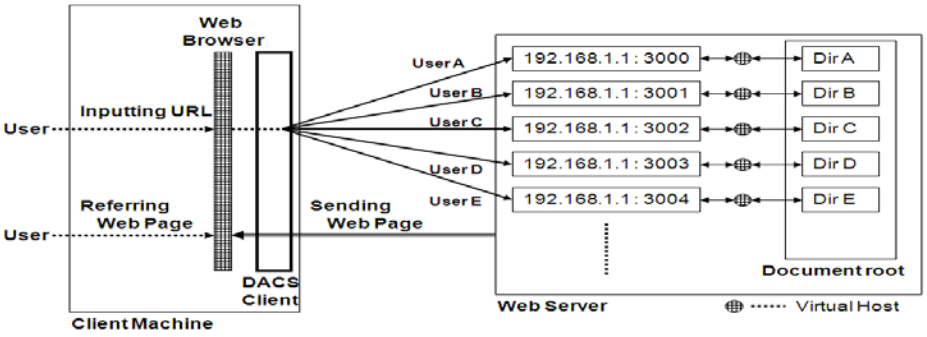


Fig. 8 Function Using Data from Document Medium

First, the function to retrieve data from the database is developed. To realize this function, the DACS Scheme must be extended, and the program on the Web server must be implemented in correspondence to the extended DACS Scheme, as shown in Fig. 7.In the existing DACS Scheme, after a user logs in to a client machine (a), the user name and IP address are sent to the DACS Server (b). Then, the DACS rules are sent back to the DACS Client (c). In the extended DACS Scheme, the user name and IP address are sent to the program on the Web server. A characteristic of the extended DACS Scheme is that the server side program on the Web server identifies the user by checking the login information and the source IP address from the client machine, and changes the processing of the program for each user. When different users access the program with the same URL, different information for each user can be searched and extracted from the database, and can be displayed on the Web browser. On the extended DACS Scheme, this new function is performed through the processing from (1) to (7).

Next, the function to retrieve data from the document medium for each user is developed. In the network with the DACS scheme, a different IP address and TCP port can be assigned for each user using the same host name. Therefore, a different document medium with the same file name on a different Web server can be referred for each user by entering the same URL into the Web browser as input. When this principle is combined with the function of a virtual host, which is equipped as a Web server, it is possible to use the Web server in the manner shown in Fig. 8. By the function of the virtual host, multiple sockets (IP address and TCP port) can be assigned to one Web server. The referred document can be changed for each socket.

First, in the document root of the Web server in Fig. 8, directories (i.e., Dir A, B, C, D….) are prepared for each user. By the function of the virtual host, each directory is connected to each socket as a pair. By changing the TCP port number (3000, 3001, 3002….) for an IP address (192.168.1.1), the sockets corresponding to each directory are prepared. Next, movement on this mechanism is described. First, a user enters a URL into a Web browser as input. When the URL is entered as input by User A, the file in Dir A that is connected to the socket (192.168.1.1:3000) is referred. When User B enters the URL as input, the file in Dir B that is connected to the socket (192.168.1.1:3001) is referred. For User C, the file in Dir C that is connected to the socket (192.168.1.1:3002) is referred. When a document medium with the same name exists in each directory (i.e., Dir A, B, C….), each user views different contents by entering the same URL into the Web browser as input. From the user's point of view, the same function as that of the Web Service based on the extended DACS Scheme is realized. For the information sender, because it is possible to deliver information to the specific user by uploading the document medium to the predetermined directory, the range of information usage broadens greatly. Because the information sender can easily describe the content of the document medium, it is possible to communicate information with much expressive power and impact.

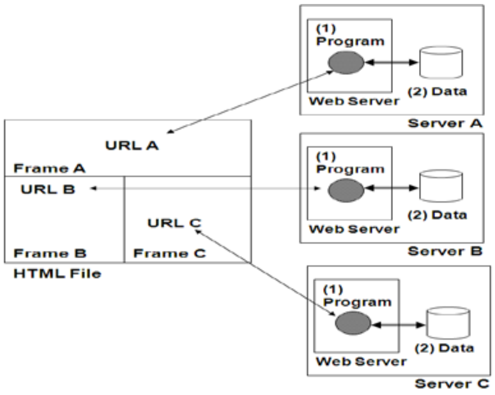


Fig. 9 Type1 System Configuration

3.3 System Configuration to Realize Personal Portals

Use of the function to retrieve data from the database enables the Type1 system configuration, which is shown in Fig. 9. First, server A-C, which have programs based on the extended DACS Scheme, as shown in (1) of Fig. 9, and data in the database as shown in (2) of Fig. 9, are distributed on the network. In the extended DACS Scheme, information related to each user is displayed on the Web browser by inputting the URL. One window of the Web browser is divided, for example, into three frames (Frames A-C). A static HTML file with each URL (URLs A-C) in each frame is created for displaying the Web page as a personal portal. The static HTML file is placed on the Web server or the client machine. When the static HTML file is opened through the Web browser, the information extracted from each server is distributed on the Web browser. In the extended DACS Scheme, the URL corresponding to each server is only incorporated in the static HTML file. Thus, when the static Web page is created, the user can easily create the customized Web page as a personal portal.

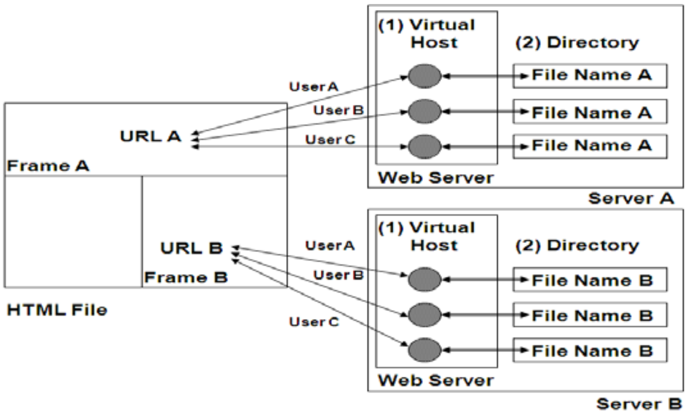


Fig. 10 Type2 System Configuration

As shown in Fig. 10, the Type 2 system configuration can be realized by using the function to retrieve data from the document medium. Each URL (URLs A, B) is prepared. The URL and the information that can be viewed after entering it in the Web browser as input are sent to the users. For example, the URLs for acquiring the homework in one classroom and for communicating from the office to each individual user are enumerated as different kinds of URL. By entering this URL into the Web browser as input, the file with the same name (File Name A or B) is referred. The file is stored in each directory for each user, and each user can view the stored file in the directory. Therefore, a customized personal portal with a static HTML File that is the same as Type 1 configuration can be created. As a result, by allowing both system configurations to coexist, a Web Service is realized in which a user can use information on the network regardless of the form of its storage. (Fig.11)

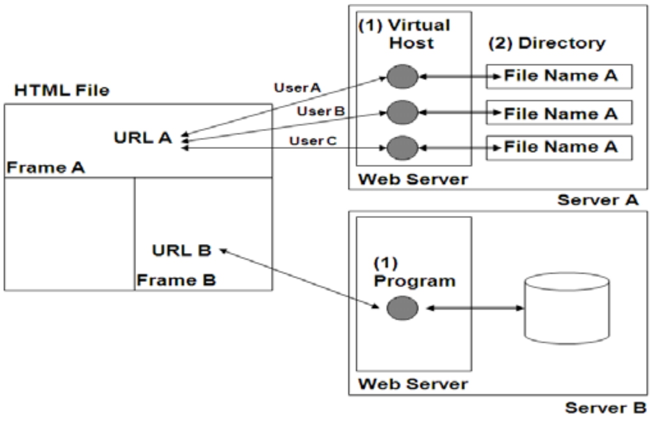


Fig. 11 Web Service by Two Functions

4. DACS WEB SERVICE

In Section 3, we provided a synopsis of the DACS Scheme. In addition, we described two types of Web Service functions that could be realized in the DACS Scheme or extended DACS Scheme, and the system configuration according to these two types of functions for realizing a customized personal portal. As these two functions are used to manage the information of each user, it is insufficient to create a customized personal portal for use in one group of users. Therefore, the following three functions are necessary.

(1) A function to manage the information of each user.

(2) A function to manage the information of user groups.

(3) A function to manage the information of all users.

Among these three functions, the latter two do not exist. Therefore, in this section, these two types of Web Service functions are integrated after extending the function to compensate for this insufficiency. The DACS Web Service, which is achieved by this integration, is proposed to realize the personal portal. Moreover, we explain the method to acquire the data they seek.

4.1 Operational Procedures for Personal Portal

In this section, operational procedures to create a web page for a personal portal are explained. To be concrete, a system administrator's procedure and a user's procedure are described.

(System Administrator's Procedure)

(Step1) After a system administrator assesses where data in information systems on a LAN exists, s/he make a list with contents of the data and the URLs to acquire it.

(Step2) The list is uploaded and released on a Web Server to let users know where the data they seek exists.

(Step3) The URL to view the lists notified to users through some methods such as the following.

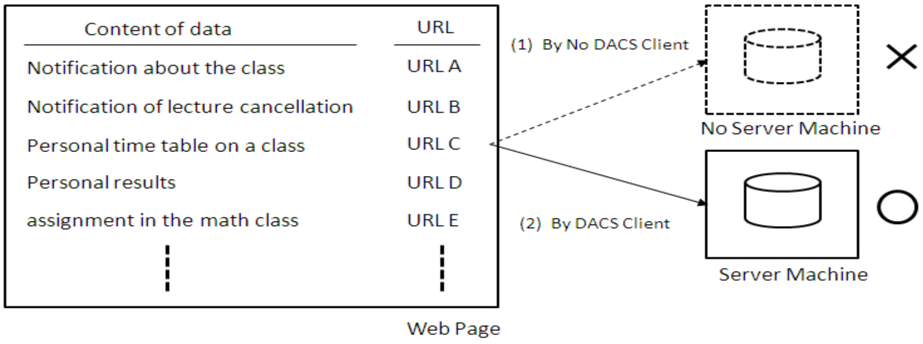
(1) Notice on a bulletin board

(2) E-mail

(3) Notice on a Web page that many users view

(4) Oral notification by telephone

In Fig.12, an example of such a Web page is described. In this page, content of the data and URLs for viewing the data are described side by side on one line. When the DACS Client is installed on the client machine, the user can view this page through the function of the DACS SContorl. That is, this page is accessed securely by VPN communication. When the DACS Client is not installed on the client machine, the user can not view this page. By clicking the URL of the page on a client machine with DACS Client, the user can easily acquire the data they seek. By user's procedure as follows, each user can create the web page for the personal portal.



Content of data	URL
Notification about the class	URL A
Notification of lecture cancellation	URL B
Personal time table on a class	URL C
Personal results	URL D
assignment in the math class	URL E
⋮	⋮

Fig. 12 Web Page to show Where Data and URLs Exist

(User's Procedure)

(Setp4) Each user inputs access the above list by inputting the specified URL with the DACS Client installed in a client.

(Step5) Each user views the list, and acquires the URL for getting the necessary information.

(Step6) Each user creates a static HTML file for the personal portal by implementing the above URL.

(Step7) Each user uploads the above HTML file.

After these steps, it is possible to view the web page for the personal portal.

4.2 Synopsis of DACS Web Service

In Fig.13, an overview of the DACS Web Service is shown. In this figure, the function to retrieve data from a database of an information system is shown as Function α, and the function to retrieve data from a document medium such as a simple text or PDF file is shown as Function β.

From here, a series of DACS Web Service's movement is explained. First, as URLs to use the DACS Web Service, three kinds of URLs as follows are prepared for Function α and Function β.

(1) URL for acquiring data of each user

(2) URL for acquiring data of each group

(3) URL for acquiring data of all users

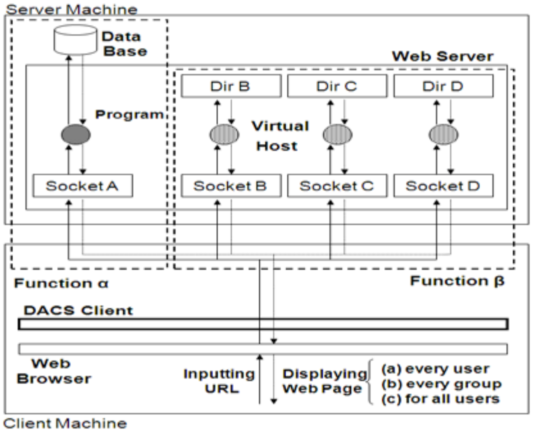
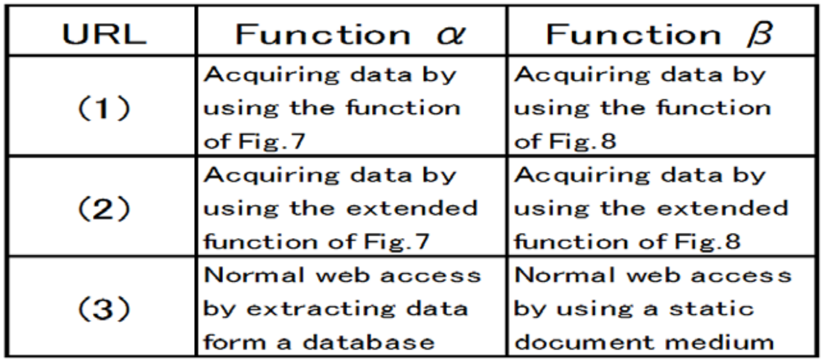


Fig. 13 DACS Web Service

By using these URLs, necessary information is acquired. The concrete functions by using these URL are described in Fig.14. By the URL (1), the existing two functions described in the Fig.7 and Fig.8 are used. By the URL (3), normal web access is done. In the case of Function α, the data is extracted from a database. For example, web access without three processes of (3),(4) and (5) in Fig.1 is given. In the case of Function β, the data is extracted from a static document medium. This is a normal web access that we do in accessing a home page on Internet. Because the above cases are realized by using the existing technology, it is not necessary to explain it especially.



URL	Function α	Function β
(1)	Acquiring data by using the function of Fig.7	Acquiring data by using the function of Fig.8
(2)	Acquiring data by using the extended function of Fig.7	Acquiring data by using the extended function of Fig.8
(3)	Normal web access by extracting data form a database	Normal web access by using a static document medium

Fig. 14 Details of Function α and Function β

However, in the case of using the URL (2), the extended functions from the functions of Fig.7 and Fig.8 are used. Therefore, technological explanation needs to be described. In Fig.15 the extended function from the function of Fig.7 is described. Processes from (4) to (7) in Fig.17 are additional processes from the function of Fig.7. First, a user name, which is sent from the DACS Sever by the process (d), is sent to an authentication server such as LDAP (5). The authentication server receives it, and returns the group name that the user belongs to by extracting it form an authentication database in process (5). The program on the Web server receives the group name, and sends it to the database in process (6). The database receives the group name, and extracts data for the group from the database. By process (7), the data is sent to the program on the Web server. The program receives the data, and sends a Web page to the Web browser on the client. In Fig.16, the extended function from the function of Fig.8 is described. This extended function is realized by changing the setups of the DACS rules. In Fig. 7, each user passes through the specified sockets and accesses the specified directory. Different multiple users do not access the same directory. However, by assigning the same DACS rules to each user belonging to the same group, the function described in Fig.16 is realized. To be concrete, when multiple users form UserA1 to UserA5 inputs same URL to the Web Browser, the same static document medium in the directory of GroupA is accessed through the specified sockets (192.168.1.1:4001). In the same way, When multiple users form UserB1 to UserB5 inputs same URL to the Web Browser, the same static document medium in the directory of GroupB is accessed through the specified sockets (192.168.1.1:4002). After these above access, each Web page for each user’s group is sent to the Web browser on the client.

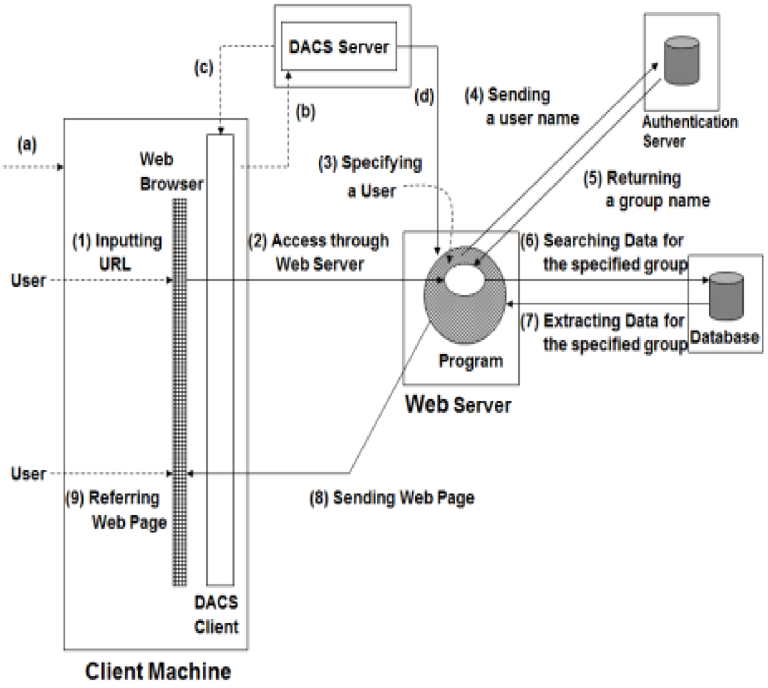


Fig. 15 Extended function of Functionα

In this way, a user can use data from an information system or document medium dispersed on the network, without being aware of which function is being used. In other words, a user can use information regardless of its form or place of storage, if a user knows the URL and the kind of information acquired by that URL. Regardless of whether Function α or Function β is used, data are displayed on the Web browser after entering the URL as input. Three kinds of data which are sent to each user (a), each group of users (b), and all users, are displayed.

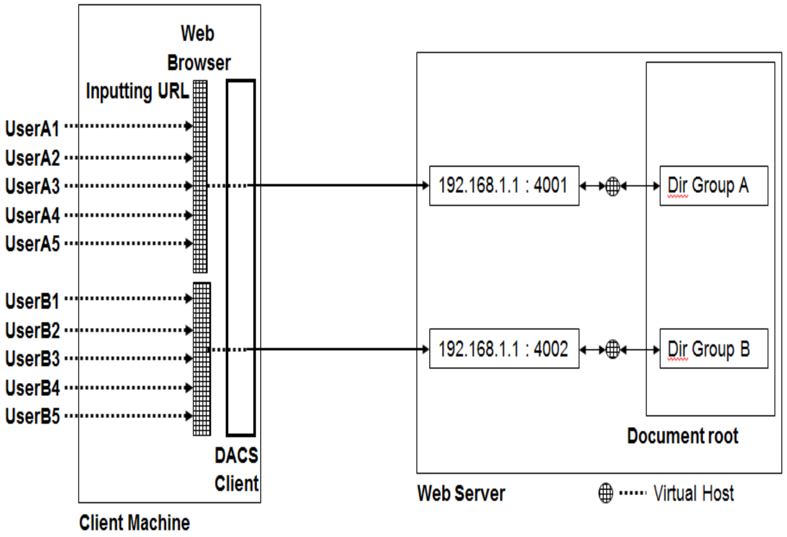


Fig. 16 Extended function of Function β

4.3 Characteristics of DACS Web Service

(1) Unifying Access Control

To manage access control information that is stored in one place, a network administrator can unify the access control for multiple Web servers distributed on the local area network. In a conventional network, access control for Web servers is performed by the mechanisms for access control that are distributed on the local area network, such as packet filtering of the fire-wall and router as well as access control of the Web server.

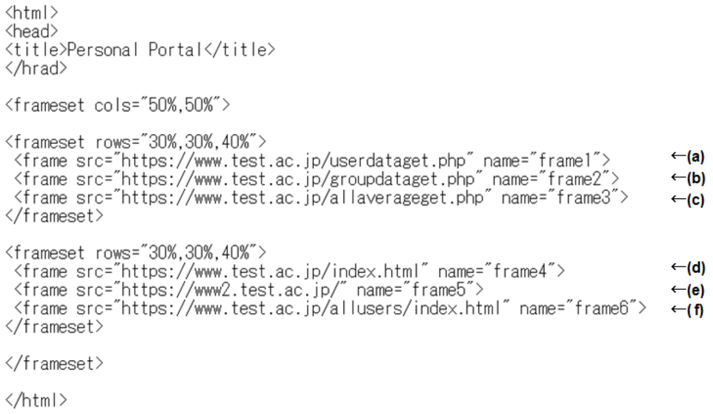
(2) Detailed Access Control

To change access control information that is stored in one place, a network administrator can manage the access control for the Web server according to the users and the kind of network (e.g., for office work, for students, public network). To be more specific, when a user moves multiple networks with a personal notebook computer and is permitted access to one Web server, the network administrator can refuse the user's access depending on the network configuration. In a conventional network, the same access control can be performed functionally by arranging a mechanism that combines access control by user authentication with that by use of the source IP address. However, this is a complicated and troublesome method because a network administrator needs to manage multiple Web servers individually.

(3) Change of the Referred Web Server

Using the same URL, the referred Web server can be changed according to the user and the type of network. When the Web Server, which is used by presenting the necessary information for each network (e.g., the guidance information of the network use), and the URL for information reference is decided as a promise, a user can refer to that information by automatically accessing the Web server located in that network.

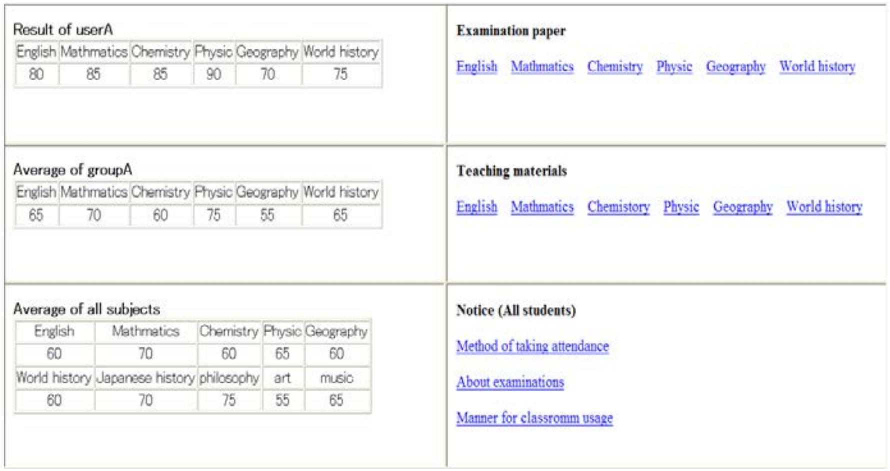
4.4 Personal Portal Creation by Using Prototype System



```
<html>
<head>
<title>Personal Portal</title>
</hrad>

<frameset cols="50%,50%">

<frameset rows="30%,30%,40%">
 <frame src="https://www.test.ac.jp/userdataget.php" name="frame1">          ←(a)
 <frame src="https://www.test.ac.jp/groupdataget.php" name="frame2">         ←(b)
 <frame src="https://www.test.ac.jp/allaverageget.php" name="frame3">        ←(c)
</frameset>

<frameset rows="30%,30%,40%">
 <frame src="https://www.test.ac.jp/index.html" name="frame4">               ←(d)
 <frame src="https://www2.test.ac.jp/" name="frame5">                        ←(e)
 <frame src="https://www.test.ac.jp/allusers/index.html" name="frame6">      ←(f)
</frameset>

</frameset>

</html>
```

Fig.17 HTML File as Personal Portal

Using this system, the experiments were performed. First, the content of the static HTML file as a personal portal (portal page) is described in Fig.17. This personal portal is divided into 6 frames. At each point from (a) to (f), each URLs for using 6 functions described in Fig.14 were set into each frame.



Result of userA

English	Mathmatics	Chemistry	Physic	Geography	World history
80	85	85	90	70	75

Average of groupA

English	Mathmatics	Chemistry	Physic	Geography	World history
65	70	60	75	55	65

Average of all subjects

English	Mathmatics	Chemistry	Physic	Geography
60	70	60	65	60
World history	Japanese history	philosophy	art	music
60	70	75	55	65

Examination paper

English Mathmatics Chemistry Physic Geography World history

Teaching materials

English Mathmatics Chemistry Physic Geography World history

Notice (All students)

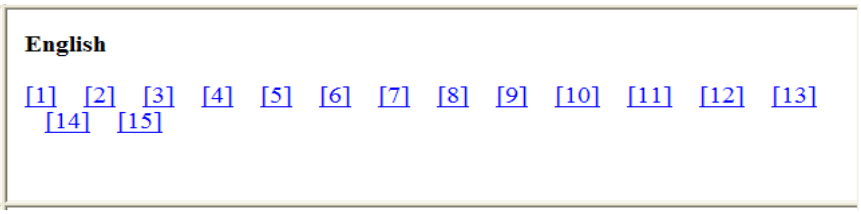
Method of taking attendance

About examinations

Manner for classromm usage

Fig.18 Portal Page for User A

Next, when the portal page was opened after user A logged in on a client machine, the page shown in Fig. 18 was displayed on the Web browser. The three frames on the left were as follows. Data stored in the database for user A, i.e., personal result extracted from the table which stored the results of all students were displayed in the top frame, which was realized with the Function α used by the URL (1). Data stored in the database for group A of which user A is a member, i.e., average results of the classes that user A attended which were extracted from the table with the average results for each class were displayed in the middle frame, which was realized with the Function α used by the URL (2). Data stored in the database for all users, i.e., average results of all classes were displayed in the bottom frame, which was realized with the Function α used by the URL (3). In addition, the right three frames on the right were as follows. The content of the static HTML file stored in the home directory for user A, i.e., marked examination papers of the attendance subjects were displayed in the top frame, which was realized with the Function β used by the URL (1). The content of the static HTML file stored in the home directory which is accessible by group A of which user A is a member, i.e., teaching materials of each attendance class were displayed in the middle frame. In this frame, URLs prepared for each subject are implemented, and each user is accessible to each home directory prepared for each class in each subject.

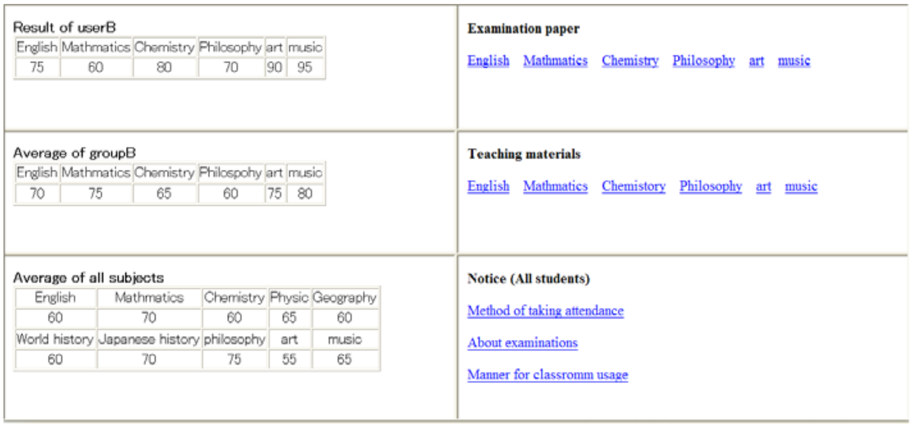


English

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15]

Fig. 19 Changed Frame

Therefore, user A became accessible to teaching materials of each attended class. In this case, when input the link of “English”, the content of the frame was changed to the page of the attended class with links of each materials in Fig.19. These were realized with the Function β used by the URL (2). The content of the static HTML file stored in home directory for all users, i.e., notices for all students were displayed in the bottom frame, which was realized with the Function β used by the URL (3).Then, when the HTML file serving as the personal portal was opened after user B logged in on a client machine, the data related to user B was displayed on each frame of the Web browser in the same way as the above-mentioned case (Fig. 20).



Result of userB

English	Mathmatics	Chemistry	Philosophy	art	music
75	60	80	70	90	95

Examination paper

English Mathmatics Chemistry Philosophy art music

Average of groupB

English	Mathmatics	Chemistry	Philospohy	art	music
70	75	65	60	75	80

Teaching materials

English Mathmatics Chemistory Philosophy art music

Average of all subjects

English	Mathmatics	Chemistry	Physic	Geography
60	70	60	65	60
World history	Japanese history	philosophy	art	music
60	70	75	55	65

Notice (All students)

Method of taking attendance

About examinations

Manner for classromm usage

Fig. 20 Portal Page for User B

5. VIRTUAL USE METHOD OF THE CGI PROGRAM

5.1 Principle of Virtual Use method of the CGI program

In this paper, the method that is realized by the Function α1 is proposed. By using this function, CGI program I is accessed virtually through same URL from users in each group. To be concrete, this method is realized by the following procedure.

(Step1) First setting of the CGI program

First, the CGI program is set by a normal procedure. For example, the program files as the CGI program are placed on the Web Server, and initial setting is performed. For example, the setting of initial parameter of the CGI program and permission of the program files. As the result, users can use the programs of the CGI program by inputting one URL into a Web Browser.

(Setp2) Setting for virtual use of the CGI program programs

After copying the directory that stores the programs, it is pasted as another directory with another name. By repeating a similar operation, multiple directories for each group are prepared. At the same time, the content of the DACS rules is changed. As the result, users that belong to same group become possible to access the programs of same directory by use of a URL. On the other hand, users that belong to other group become impossible to access the above programs by use of same URL.

By these procedures, in the form of using same URL, users in each group can access the programs of the directory in each group, and can not access the programs of other group. Virtual use of the CGI program is realized without a special mechanism.

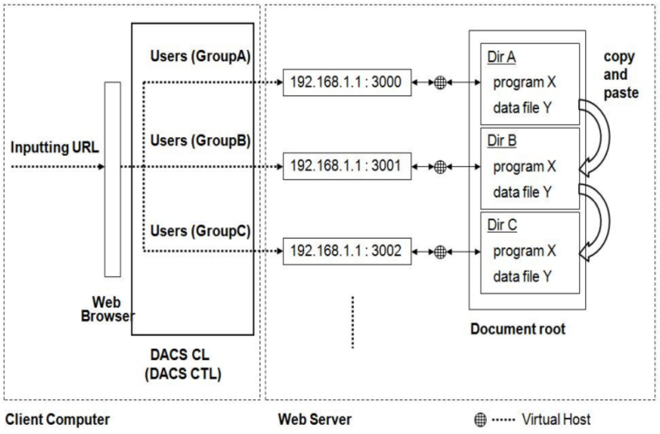


Fig. 21 Virtual Usage of the CGI Program

A concrete example of it is shown in Fig. 21. As first step, the program X as the CGI program and other files such as data file are placed in directory A (Dir A in Fig. 21), and initial setting of it is performed. As the result, users can access and use it. Next, second step is as follows. At first, Dir A is copied and pasted with another name. In Fig. 21, Dir B and Dir C are the pasted directories. Each directory is named with the regularity. Though each socket is connected to each directory through the virtual host by the system setting, each name is allocated to be easy to automate the setting. At the same time, by changing the DACS rules, the host name in URL and the communication port is converted to each socket every group. In Fig. 21, when users in Group A inputs one URL into a Web Browser, they access the program X in Dir A through by way of 192.168.1.1:3000. In the case of users in Group B, they access the program X in Dir B through by way of 192.168.1.1:3001. In the case of users in Group C, they access the program X in Dir C by way of 192.168.1.1:3002. Then, users in Group A con not access the program in Dir B and Dir C. Users in Group B con not access the program in Dir A and Dir C. Users in Group C con not access the program in Dir A and Dir B. In this way, virtual use of the CGI program is realized simply.

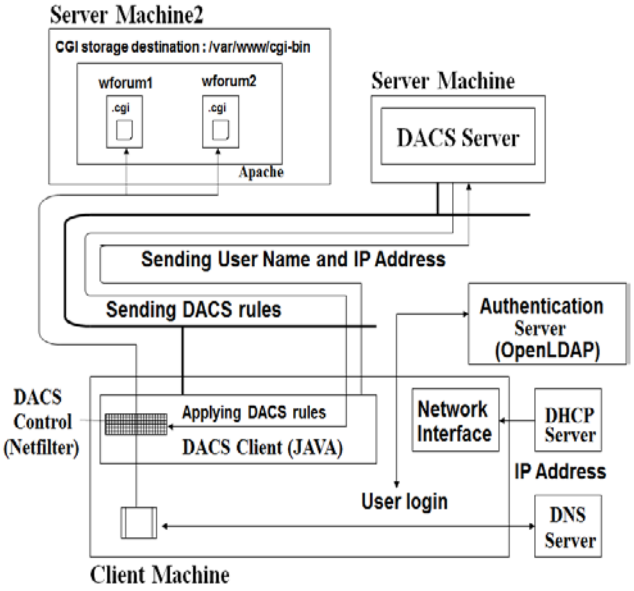


Fig. 22 System Configuration

5.2 Evaluations by experiments with the real CGI Program

Here, experimental results by using a bulletin board CGI program [17] open to Internet are show. The CGI programs were operated on the experimental network systems described in Fig. 22, which were built as follows.

(1)Server Machine

- CPU: Celeron M Processor340 (1.5GHz)
- OS: CentOS6
- DACS Server:

- Language: Visual C++ 7.1
- Database: postgressql-8.4.4-2

(2)Server Machine2

- CPU: Celeron M Processor340 (1.5GHz)
- OS: CentOS 6
- Apache: httpd-2.2.15-5
- Database: postgresql8.4.4-2
- CGI Program on Apache: bulletin board
- Language: php-5.3.2-6

(3)Client Machine

- CPU: Celeron M Processor340 (1.5GHz)
- OS: WindowsXP Professional
- DACS Client:
- Language: VisualC++ 7.1,WinsockSPI [18],Putty [19]

(4)Others

- AuthenticationServer: openldap-2.1.22-8(FedoraCore1)
- DHCP Server: dhcp-4.1.1-12
- DNS Server: bind-9.2.2.P3-9(FedoraCore1)

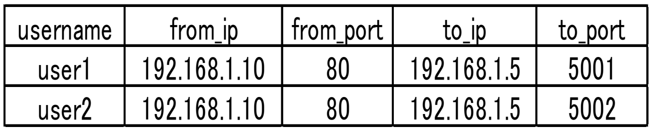
In this experimental network system, same kinds of two CGI programs were set in the directories of “wforum1” and “wforum2” on the Server Machine 2, which had same program files and data files and log files. To be concrete, after setting for programs in the directory of “wforum1” was performed, the directory “wforum1” was copied and pasted as the directory “wforum2”. That is, same identical files were stored. After these setting process, setting for the Web Server was performed and the following setting were added into the configuration file (httpd/conf/httpd.conf).

(Additional setting)

```
Listen 5001

Listen 5002

<VirtualHost 192.168.1.5:5001>

      DocumentRoot        “/var/www/cgi-bin/wforum1

</VirtualHost>

<VirtualHost 192.168.1.5:5002

      DocumentRoot        “/var/www/cgi-bin/wforum1
```

</VirtualHost>

Next, setting information as follows was set to DACS Server in Fig. 23. In this figure, “from_ip” shows the IP address before destination IP address change, and “from_port” shows the port number before destination port number change. Then, “to_ip” shows the IP address after destination IP address change, and “to_port” shows the port number after destination port number change.



username	from_ip	from_port	to_ip	to_port
user1	192.168.1.10	80	192.168.1.5	5001
user2	192.168.1.10	80	192.168.1.5	5002

Fig. 23 Setting information

After these settings were added, experiments were performed by using the experimental network system.

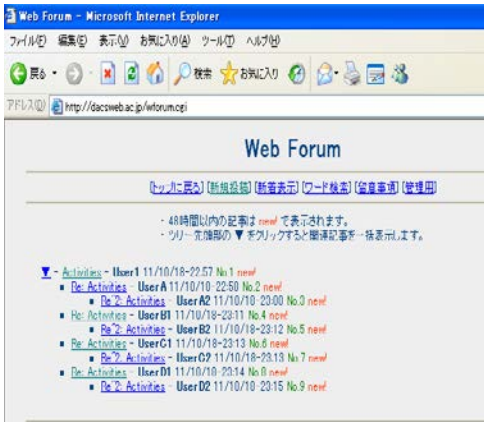
First, “user1” logged in to the client which was shown as “Client Machine” in Fig. 22, and input the URL（http://dacsweb.ac.jp/wforum.cgi）into a Web Browser.

At that time, the web page in Fig. 24 was displayed on the Web Browser. The system movements for displaying it were as follows.

(Step1) Inputting URL into a Web Browser

(Step2) Hostname was converted to IP Address (192. 168.1.10) by a name solution service of Domain Name Service (DNS).

(Step3) The Web Browser was trying to be communicated with the destination (192.168.1.10:80).



Web Forum - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
戻る 検索 お気に入り
アドレス(D) http://dacsweb.ac.jp/wforum.cgi

Web Forum

[トップに戻る] [新規投稿] [新着表示] [ワード検索] [留意事項] [管理用]

・48時間以内の記事は new で表示されます。
・ツリー先頭部の ▼ をクリックすると関連記事を一括表示します。

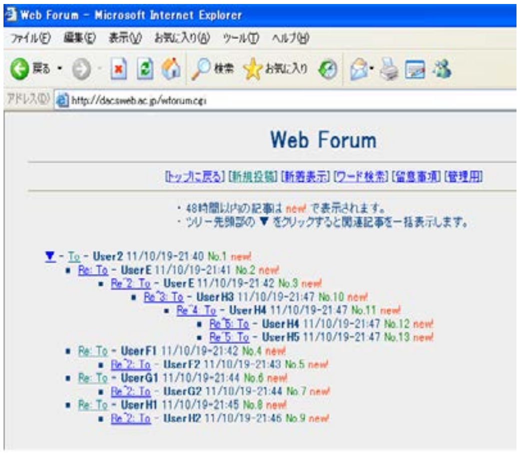
▼ - Activities - User1 11/10/18-22:57 No.1 new
- Re: Activities - UserA 11/10/10-22:50 No.2 new
  - Re^2: Activities - UserA2 11/10/10-23:00 No.3 new
- Re: Activities - UserB1 11/10/18-23:11 No.4 new
  - Re^2: Activities - UserB2 11/10/18-23:12 No.5 new
- Re: Activities - UserC1 11/10/18-23:13 No.6 new
  - Re^2: Activities - UserC2 11/10/18-23:13 No.7 new
- Re: Activities - UserD1 11/10/10-23:14 No.8 new
  - Re^2: Activities - UserD2 11/10/10-23:15 No.9 new

Fig. 24 Web page (1)

(Step4) The DACS Client changed the destination to 192.168.1.5:5001 form 192.168.1.10:80.

(Step5) The Web Server received the communication from the Web Browser, and the CGI program stored in /var/www/cgi-bin/wforum1 was accessed by the function of Virtual host function.

Next, “user2” logged in to the same client, and input the same URL into the Web Browser. The web page in Fig. 25 was displayed on the Web Browser. The system movements for displaying it were same in case of “user1”, and the CGI program stored in /var/www/cgi-bin/wforum2 was accessed by the function of Virtual host function with the same URL.


Web Forum - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
戻る
検索
お気に入り
アドレス(D)

http://dacsweb.ac.jp/wforum.cgi

Web Forum

[トップに戻る] [新規投稿] [新着表示] [ワード検索] [留意事項] [管理用]

・48時間以内の記事は new! で表示されます。
・ツリー先頭部の ▼ をクリックすると関連記事を一括表示します。

▼ - To - User2 11/10/19-21:40 No.1 new!
- Re: To - UserE 11/10/19-21:41 No.2 new!
    - Re^2: To - UserE 11/10/19-21:42 No.3 new!
        - Re^3: To - UserH3 11/10/19-21:47 No.10 new!
            - Re^4: To - UserH4 11/10/19-21:47 No.11 new!
                - Re^5: To - UserH4 11/10/19-21:47 No.12 new!
                - Re^5: To - UserH5 11/10/19-21:47 No.13 new!
- Re: To - UserF1 11/10/19-21:42 No.4 new!
    - Re^2: To - UserF2 11/10/19-21:43 No.5 new!
- Re: To - UserG1 11/10/19-21:44 No.6 new!
    - Re^2: To - UserG2 11/10/19-21:44 No.7 new!
- Re: To - UserH1 11/10/19-21:45 No.8 new!
    - Re^2: To - UserH2 11/10/19-21:46 No.9 new!

Fig. 25 Web page (2)

In the former experiments, by using the same URL, different users could access the same type of CGI programs that were set indifferent directories.

First, the additional function of user authentication mechanism is explained. Normally, when CGI programs with no user authentication mechanism are set on the Web Server, each CGI program is accessed from every user. However, on the experimental network system, only when each user could log in the client, the CGI program is accessed. Though every user can know the URL for accessing the CGI programs, the IP address which acquired from host name in the URL (IP address A) by DNS is not the IP address set as a virtual host(IP address B). By the function of the DACS Client, the destination IP address was changed from IP address A to IP address B. The rules for this destination change were acquired from the DACS Server at the timing of user authentication. In other words, the user authentication mechanism was used as the authentication mechanism for the CGI programs with no user authentication mechanism.

Next, the additional function of access control mechanism is explained. Normally, when CGI programs with no user authentication mechanism are set on the Web Server, each CGI program is accessed through each different URL. That is, each URL and each CGI program have one-to-one relationship. However, on the experimental network system, when each user logged in the client and input same URL, the different CGI program was accessed. In other words, each URL and each CGI program had one-to-many relationship. To be concrete, though “user1” could access the CGI programs in Fig. 24, “user2” could not the same CGI program by same URL. It is clear by “user2” accessing the CGI program in Fig. 25. In addition, though “user2” could access the CGI programs in Fig. 25, “user1” could not the same the CGI program by same URL. It is clear by “user1” accessing the CGI program in Fig. 24. Thus, the access control mechanism was added to the CGI programs with no access control mechanism.

6. CONCLUSION

In this paper, contents of the virtual use method of the CGI program were shown, and evaluations of the virtual use method of the CGI program were performed by experiments with the real CGI program opened to Internet. As the result, it became possible for the CGI program to be used in the Web page as the personal portal of the real network. In the future, we are going to study about the new implementation method of CGI on the network introducing the DACS Scheme.

REFERENCES

[1]. J.Hartmann, Y.Sure,”An infrastructure for scalable, reliable semantic portals,” IEEE Intelligent Systems,Vol.19,no.3,pp.58-65,2004.

[2]. N.Lowe, A.Datta,”A New Technique for Rendering Complex Portals,” IEEE Tran. on Visualization and Computer Graphics,Vol.11,no.1,2005.

[3]. D.Robinson,”The WWW Common Gateway Interface Version 1.1, Internet Draft,” 1995.

[4]. http://www.netvibes.com/ja-jp

[5]. http://www.google.co.jp/ig?hl=ja

[6]. K.Odagiri, R.Yaegashi, M.Tadauchi, N.Ishii,” Efficient Network Management System with DACS Scheme : Management with communication control,” International Journal of Computer Science and Network Security,Vol.6,no.1,pp.30-36,2006.

[7]. K.Odagiri, R.Yaegashi, M.Tadauchi, N.Ishii,” Efficient Network Management System with DACS Scheme,” in Proc. of International Conference on Networking and Services (ICNS), 16-19 July, Silicon Valley, USA, IEEE Computer Society, 2006.

[8]. K.Odagiri, R.Yaegashi, M.Tadauchi, N.Ishii,”New Network Management Scheme with Client 's Communication Control” Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Computer Science,Springer,Vol.4252,pp.379-386,2006.

[9]. K.Odagiri, R.Yaegashi, M. Tadauchi, N. Ishii,“Secure DACS Scheme,” Journal of Network and Computer Applications, Elsevier, Vol.31, Issue 4,pp.851-861, November, 2008

[10]. K.Odagiri, R.Yaegashi, M.Tadauchi, N.Ishii,”New Web Service Based on Extended DACS Scheme,” International Journal of Computer Science and Network Security, Vol.6, no.3, pp.8-13, March,2006.

[11]. K.Odagiri, R.Yaegashi, M.Tadauchi, N.Ishii,”Free Information Usage System on the Network Introducing DACS Scheme” in Proc of International Conference on Internet and Web Applications and Services (ICIW), Mauritius, IEEE Computer Society, May, 2007.

[12]. K.Odagiri, R.Yaegashi, M.Tadauchi, N.Ishii,”Practical DACS Web Service for User’s Free Portal Creation,” in Proc. of International Conference on Web Services (ICWS), Salt Lake City, UT, USA, pp.952-959, July, 2007.

[13]. C.Bouras, V.Kapoulas, I.Misedakis,“Web Page Fragmentation for Personalized Portal Construction,” in Proc. of the International Conference on Information Technology: Coding and Computing(ITCC'04), Las Vegas, Nevada, USA, IEEE Computer Society,2004.

[14]. Ge Kaikai, Zhao Wenbin, Zhang Xintong,"Service-oriented personalized potal platform with QoS guarantee," International Conference on Advanced Computer Theory and Engineering, Vol.6, pp.205-209, August,2010.

[15]. M.Vering, et al., “The E-Business Workplace: Discovering the Power of Enterprise Portals”, John Wiley & Sons, February,2001.

[16]. N.Hanakawa, Y.Akazawa, A.Mori, T.Maeda, S Inoue, S.Tsutsui,,"A Web-Based Integrated Education System for a Seamless Environment among Teachers, Students, and Administrators, "IEICE Transactions on Information and Systems, Pt.1 Vol.J88-D, no.2, pp.498-507,2005.

[17]. http://www.blackboard.com/

[18]. M. P. Papazoglou, D. Georgakopoulos， “Service Oriented Computing，” In Communications of the ACM, Vol.46, No.10, pp.25-28, October 2003.

[19]. M. Endrei: Service-Oriented Architecture and Web Service, IBM, SG24-6303-00(2004)

[20]. The World Wide Web Consortium (W3C), “Simple Object Access Protocol (SOAP) (version 1.1)”, W3C Note 08, May, 2000.

[21]. S.Woodman, G.Morgan, S.Parkin, “Portal Replication for Web Application Availability Via SOAP,” in Proc. of the 8th IEEE International Workshop on Object-Oriented Real-Time Dependable Systems, pp. 122-130, January, 2003.

[22]. SOAP Specifications，W3C Recommendation. http://www.w3.org/TR/soap/

[23]. Roy T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," University of California, 2000.

[24]. R.Fielding, R.Taylor,“Principled Design of the Modern Web Architecture” in ACM Transaction on Internet Technology, Vol.2, No.2, pp.115-150, May, 2002.

[25]. Roy T. Fielding and Richard N, "Taylor, Principled design of the modern Web architecture," ACM Transactions on Internet Technology, Vol.2, No.2, pp.115-150, May, 2002.

[26]. W3C Recommendation 16, “Extensible Markup Language (XML) 1.1 (Second Edition)”, http://www.w3.org/TR/xml11/, August, 2006.

[27]. Lawrence A. Cunningham. "Language, Deals and Standards: The Future of XML Contracts," Boston College Law School Faculty papers, Paper 138,2006.

[28]. D.F.Ferraiolo, D.R.Kuhn,“Role Based Access Control,” 15th National Computer Security Conference, pp.554-563, October, 1992.

[29]. R.S.Sandhu, E.J.Coyne, H.L.Feinstein and C.E.Youman,“Role-Based Access Control Models,” IEEE Computer, Vol.29, No.2, pp.38-47. IEEE Press, February, 1996.

[30]. R.Sandhu, D.F.Ferraiolo and D.R.Kuhn,“The NIST Model for Role Based Access Control: Toward a Unified Standard,” 5th ACM Workshop Role-Based Access Control, pp.47-63, July, 2000.

[31]. C.Metz, "The latest in virtual private networks: part I," IEEE Internet Computing, vol.7, No.1, pp.87–91, 2003.

[32]. C.Metz, "The latest in VPNs: part II," IEEE Internet Computing, Vol.8, No.3, pp.60–65, 2004.

[33]. D.Wood, V.Stoss, L.Chan-Lizardo, G.S.Papacostas, M.E.Stinson, "Virtual private networks," International Conference on Private Switching Systems and Networks, pp.132-136, Jun, 1998.

[34]. J.Mizusawa, N.Shigematsu, H.Itoh,"Virtual private network control system concept," International Conference on Private Switching Systems and Networks, pp.137-141, Jun, 1988.

[35]. SSH : The Secure Shell Connection Protocol, RFC 4254, 2006.

[36]. Winsock SPI: http://www.microsoft.com/msj/0599/layeredservice/layeredservice.aspx

[37]. Putty: http://www.chiark.greenend.org.uk/~sgtatham/putty/

Society for Science and Education
United Kingdom

TMLAI Transactions on
Machine Learning and Artificial Intelligence

Volume 2, Issue 1
ISSN: 2054 - 7390

Hierarchical sparse representation for object recognition

Toru Nakashika, Takeshi Okumura, Tetsuya Takiguchi, Yasuo Ariki
Graduate School of System Informatics, Kobe University, Japan;
nakashika@me.cs.scitec.kobe-u.ac.jp, {takigu,ariki}@kobe-u.ac.jp

ABSTRACT

Recently, generic object recognition that achieves human-like vision has being looked to for use in robot vision, automatic categorization of images, and image retrieval. In object recognition, semi-supervised learning, which incorporates a large amount of unsupervised training data (unlabeled data) along with a small amount of supervised data (labeled data), is regarded as an effective tool to reduce the burden of manual annotation. However, some unlabeled data in semi-supervised models contain outliers that negatively affect the parameter estimation during the training stage. Such outliers often cause an over-fitting problem especially when a small amount of training data is used. Furthermore, another problem that occurs when using the conventional methods is that when labeling an image based on super-pixel representation, the lack of discrimination of the image features and the scale variance of the objects decreases the recognition accuracy because the feature extraction is based on the mono-scale segmentation. In this paper, we propose an object recognition method for solving both problems. For the former problem, our method prevents the over-fitting associated with the semi-supervised based approach by using sparse representation to suppress existing outliers in the data. For the latter problem, we employ Tree Conditional Random Field to construct the hierarchical structure of an image. Experiment results using two datasets confirm the effectiveness of our method.

Keywords: Object recognition, Automatic image annotation, Sparse representation, Semi-supervised learning, Hierarchical representation, Tree Conditional Random Field.

1. INTRODUCTION

Generic object recognition (automatic image annotation), in which the system automatically assigns labels to an image, is one of the most significant tasks in computer vision. Most of the conventional methods are based on a supervised labeling approach in order to achieve an exact classification. However, it has been pointed out that with this approach the training cost is extremely high because an enormous amount of training data must be labeled manually. To reduce the amount of such a troublesome work, a semi-supervised approach has recently

DOI: 10.14738/tmlai.21.95
Publication Date:10th February 2014
URL: http://dx.doi.org/10.14738/tmlai.21.95

attracted considerable attention in machine learning [1][2][3][4][5]. The semi-supervised approach inputs a large amount of non-labeled data (unsupervised data) for the training as well as not so much labeled data. Hence, it helps to improve the training accuracy without using a lot of labeled data.

Descriptions of popular methods using semi-supervised learning in text classification can be found in [2], which introduces TSVM (transductive support vector machine) as a classification model, and in [3], which introduces SemiNB (semi-supervised naive Bayes classifier) as a generative model. TSVM extends the well-known SVM so that it can be trained not only with a few labeled data but also with a large volume of unlabeled data. During the training, labeled data first determine the margin, which classifies unlabeled data. The former SemiNB is a semi-supervised version of Naive Bayes (NB). Both methods, especially in SemiNB, are adversely affected by the influence of outliers in large amounts of unsupervised data, because they both take whole the unlabeled data as well as labeled data in the training process. The TSVM limits the influence of the outliers only to data around the margin. Therefore, the TSVM is not influenced as much by outliers as SemiNB is, though it is inevitable that the outliers negatively affect the margin estimation. Furthermore, the TSVM is a computationally expensive algorithm. Given a large number of training data, it needs to take an approximate approach that causes weak estimation of the margin.

In consideration of the drawbacks in a semi-supervised approach, we propose an automatic image annotation method where an effective semi-supervised tool, semi-supervised canonical correlation analysis (semi-CCA) [4], and sparse representation [6] collaboratively suppress the influence of outliers. First, subspaces that maximize the correlation between image features and label features are generated by semi-CCA, using a small amount of labeled data and much unlabeled data. Semi-CCA extends canonical correlation analysis (CCA), so as to avoid over-fitting when it has a few (labeled) training data. Given a large amount of unlabeled data as well as the labeled data, it grabs a global distribution. Since the trained distribution is affected by outliers somewhat, we adopt Regularized Orthogonal Matching Pursuit (ROMP) [7], one of the handy sparsing algorithms. Using sparse representation, it is possible to achieve the automatic annotation that utilizes an abundance of unlabeled data for the semi-supervised learning that is robust to the influence of outliers.

Our approach is based on super-pixel representation [8,9], where low-level features, such as the color feature and the texture feature, are extracted from the local region (super-pixel), and the class of each region is recognized based on such features. However, there is a problem in the super-pixel-based methods in that they are not robust to the scale variance due to their inability to discriminate the features extracted from the local regions. Therefore, we further employ hierarchical representation in this paper. This hierarchical structured model, which is called Tree Conditional Random Field (TCRF) [10], is robust to the scale variance since it accounts for multi-scale hierarchization.

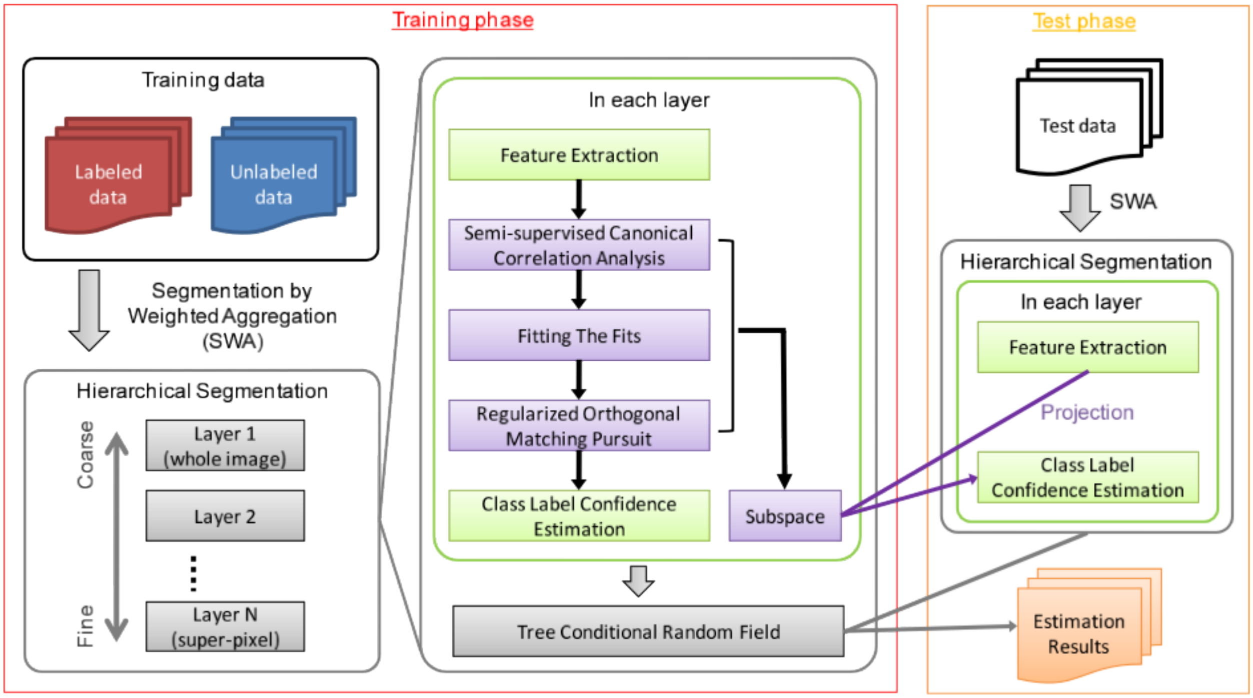

Training phase
Training data
Labeled data
Unlabeled data
Segmentation by Weighted Aggregation (SWA)
Hierarchical Segmentation
Coarse
Fine
Layer 1 (whole image)
Layer 2
Layer N (super-pixel)
In each layer
Feature Extraction
Semi-supervised Canonical Correlation Analysis
Fitting The Fits
Regularized Orthogonal Matching Pursuit
Class Label Confidence Estimation
Subspace
Tree Conditional Random Field
Test phase
Test data
SWA
Hierarchical Segmentation
In each layer
Feature Extraction
Projection
Class Label Confidence Estimation
Estimation Results

Figure 1: System flowchart of proposed method.

Figure 1 shows the flow of the proposed method. In the training stage, we first divide all training images including labeled and unlabeled data into hierarchical subregions using SWA (Segmentation by Weighted Aggregation) [11]. For each region in each layer, we extract features and apply sparse-representation in semi-supervised learning to estimate class label confidence. Then, we train TCRF using the confidence and the label data in all layers to estimate the probability of co-occurrences within classes in a hierarchical structure. In the test stage as well, we first apply SWA to obtain hierarchical subregions, project the regions into sub-space in each layer, and finally estimate the class label for each region (pixel) using TCRF.

The rest of the paper is organized as follows. In Section 2, we present the way of applying sparse representation in semi-supervised learning, and we explain the final annotation using TCRF in Section 3. The performance of the proposed method is evaluated in Section 4, and we conclude in Section 5.

2. SUBSPACE GENERATION AND SPARSE REPRESENTATION

Due to the high cost of preparing correct labels as training data in automatic image annotation, it is desirable to employ a semi-supervised approach which uses unlabeled data instead of some of labeled data. However, there exist outliers in the unlabeled data. In this section, we discuss a method that generates subspaces using the semi-supervised approach called Semi-supervised Canonical Correlation Analysis (semi-CCA) [4], and suppresses such outliers using Regularized Orthogonal Matching Pursuit (ROMP) [7] as shown in Figure 2.

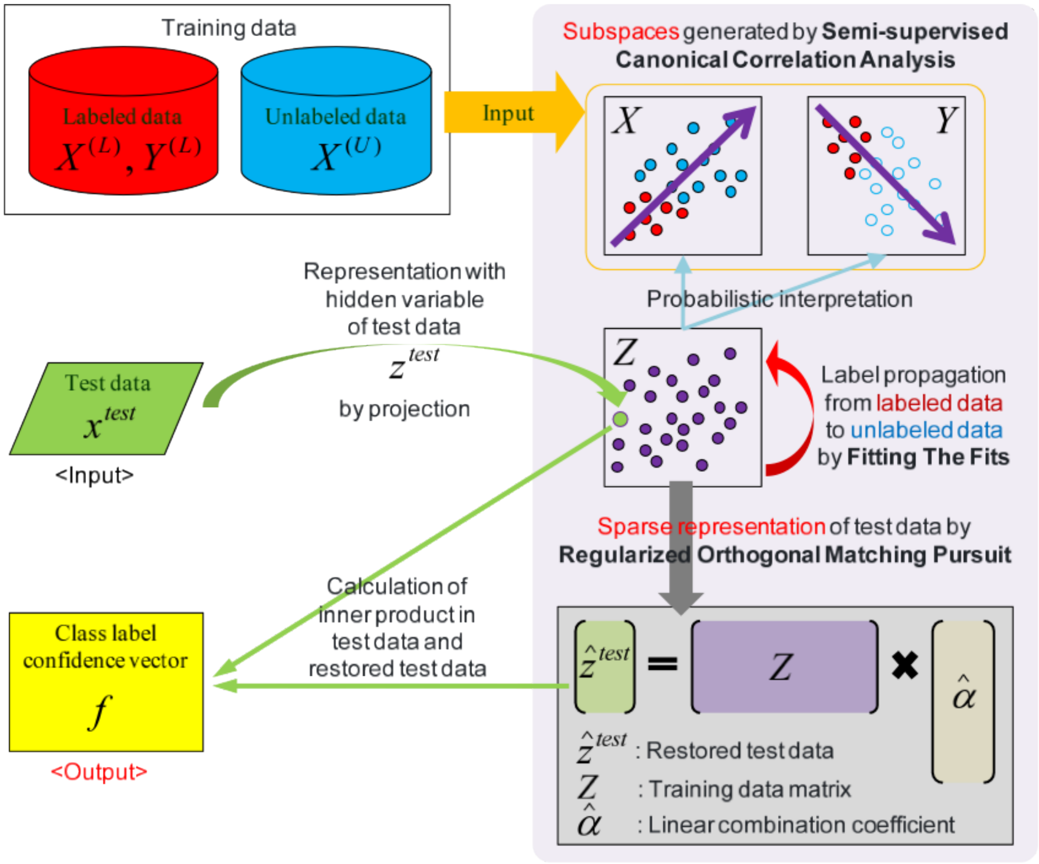

Training data
Labeled data
$X^{(L)}, Y^{(L)}$
Unlabeled data
$X^{(U)}$
Input
Subspaces generated by Semi-supervised Canonical Correlation Analysis
X
Y
Probabilistic interpretation
Representation with hidden variable of test data
z^{test}
by projection
Test data
x^{test}
<Input>
Z
Label propagation from labeled data to unlabeled data by Fitting The Fits
Sparse representation of test data by Regularized Orthogonal Matching Pursuit
Calculation of inner product in test data and restored test data
Class label confidence vector
f
<Output>
$\hat{z}^{test}$ = Z ✖ $\hat{\alpha}$
$\hat{z}^{test}$: Restored test data
Z : Training data matrix
$\hat{\alpha}$: Linear combination coefficient

Figure 2: Flow of outlier suppression in semi-supervised learning using sparse representation.

2.1 Semi-CCA

The semi-CCA is an extended version of Canonical Correlation Analysis (CCA) so that it substitutes unlabeled data for some labeled data. Both methods find the subspace that maximizes a correlation between two different types of features. In this paper, the relationship (correlation) between an image and the accompanying labels are obtained.

Let $\{X^{(L)}, Y^{(L)}, X^{(U)}\}$ be a training data set, where $X^{(L)} = \{x_n\}_{n=1}^{N}$ and $Y^{(L)} = \{y_n\}_{n=1}^{N}$ are N labeled data, and $X^{(U)} = \{x_m\}_{m=1}^{M}$ is M unlabeled data. x and y indicate image feature and label feature, respectively (See 2.2). The aim of Semi-CCA or CCA is to find the optimum subspace that maximizes a correlation between projected x and y:

$$r(w_x, w_y) = \frac{w_x^T S_{xy}^{(L)} w_y}{\sqrt{w_x^T S_{xx}^{(L)} w_x}\sqrt{w_y^T S_{yy}^{(L)} w_y}} \tag{1}$$

where w_x and w_y are projection vectors to the subspace from the original feature space x and y, respectively. $S_{\square\square}^{(L)}$ indicates each variance-covariance matrix within the labeled data. For example, $S_{xy}^{(L)} = N^{-1}\sum_{n=1}^{N} x_n y_n^T$.

If unlabeled data $\mathrm{X}^{(U)}$ is not given as the training (in other words, in the case of normal CCA), all that can be done is just to formulate the maximum problem in which the optimum $\mathbf{w}_x$ and $\mathbf{w}_y$ are found to maximize Eq. (1) using a Lagrange multiplier. In that case, the formulation boils down to an eigen-value problem.

When the amount of labeled data is not adequate, the obtained subspace is inefficiently overfitted to the training data. Hence, unlabeled data are added for correcting a global structure of data distribution in the subspace. In order to do that, PCA is employed using the concept of semi-CCA [4]. In a similar way to CCA, a projection matrix of the PCA can be calculated by solving an eigenvalue problem, in which a variance-covariance matrix of the data is maximized under a normalized orthogonal constraint.

As mentioned above, semi-CCA can be expressed as the combination of two factors: CCA with labeled data and PCA with all data including unlabeled data. Therefore, the semi-CCA formulation is also obtained by combination of the two eigenvalue problems, as in Eq. (2). A projection matrix can ultimately be obtained from the upper eigenvalues using semi-CCA.

$$\mathrm{B}\begin{bmatrix}\mathbf{w}_x\\ \mathbf{w}_y\end{bmatrix}=\lambda\mathrm{C}\begin{bmatrix}\mathbf{w}_x\\ \mathbf{w}_y\end{bmatrix} \tag{2}$$

where,

$$\mathrm{B}=\beta\begin{bmatrix}0 & \mathrm{S}_{xy}^{(L)}\\ \mathrm{S}_{yx}^{(L)} & 0\end{bmatrix}+(1-\beta)\begin{bmatrix}\mathrm{S}_{xx} & 0\\ 0 & \mathrm{S}_{yy}^{(L)}\end{bmatrix} \tag{3}$$

$$\mathrm{C}=\beta\begin{bmatrix}\mathrm{S}_{xx}^{(L)} & 0\\ 0 & \mathrm{S}_{yy}^{(L)}\end{bmatrix}+(1-\beta)\begin{bmatrix}\mathrm{I}_{D_x} & 0\\ 0 & \mathrm{I}_{D_y}\end{bmatrix} \tag{4}$$

and $\mathrm{S}_{xx}=(N+M)^{-1}\sum_{n=1}^{N+M}\mathrm{x}\mathrm{x}^T$ is a variance-covariance matrix of all image feature vectors including unlabeled images. I_{D_x} and I_{D_y} are identity matrices with the size $D_x\times D_x$ and $D_y\times D_y$, respectively. Note that the first term and the second term in Eq. (3) and (4) indicate the terms related to eigenvalue problems of CCA and PCA, respectively. β is a trade-off parameter which determines the effects of CCA and PCA.

The image feature and the label feature are connected via latent variables z in the subspace. These variables can be calculated by applying the conditional Gaussian model (For more details, see [4]). After this, we can rewrite the training and test data with the latent variables z for the sake of consideration in the subspace.

2.2 Image Feature and Label Feature

Each image is first divided into small subregions using Normalized Cuts [12]. In each subregion, the following features are extracted:

- Color : Statistics of RGB, HSV, Lab, and YCbCr
- Gabor : Gabor filter and Laplacian-of-Gaussian
- Position : Center position of a region
- Geometric : Area of a region

An image feature vector x is defined as a super-vector, where all these features are included. A label feature vector y is a binary vector, where each label is assigned in the subregion or not.

2.3 Annotation Using Sparse Representation

Recently, classification methods based on sparse representation, in which test data are represented as a linear combination of sparse bases, have been drawing attention in image processing [13][14]. It was reported in these papers that classification results showed favorable robustness of sparse representation against outliers. In this paper, we set out to suppress the effect of outliers that unintentionally appear when there is a large amount of unlabeled data, by employing sparse representation.

If a sufficient amount of training data is prepared, an input image $\mathbf{z}^{\text{test}}$ in the subspace can be represented as a linear combination of the training data. Our aim is to find sparse coefficients associated with each training data. Those entries are mostly zero, except for a few elements. This can be formulated as a minimizing problem with respect to a coefficient vector α in Eq. (5).

$$\min_{\boldsymbol{\alpha}} ||\boldsymbol{\alpha}||_\epsilon \quad s.t. \quad \mathbf{z}^{\text{test}} = \sum_{n=1}^{N+M} \alpha_n \mathbf{z}_n = \mathbf{Z}\boldsymbol{\alpha} \tag{5}$$

where $\mathbf{Z} \in \mathbb{R}^{D_z \times (N+M)}$ is a training data matrix (D_z is a dimension of subspace feature). $||\boldsymbol{\alpha}||_\epsilon$ indicates l_ϵ norm, which is the number of almost-zero elements in α, given by $||\boldsymbol{\alpha}||_\epsilon = (N+M)^{-1} \# \{n | \alpha_n \leq \epsilon\}$ with an experimentally-determined small value ϵ. However, it is computationally difficult to find the optimum vector in Eq. (5) because $||\boldsymbol{\alpha}||_\epsilon$ is indifferentiable. In this paper, we consequently adopt one of the popular greedy algorithms, Regularized Orthogonal Matching Pursuit (ROMP) [7] to solve this optimum problem.

At the end, the test data can be restored by multiplying a training data and the obtained vector $\hat{\boldsymbol{\alpha}}$ as $\hat{\mathbf{z}}^{test} = \mathbf{Z}\hat{\boldsymbol{\alpha}}$. By taking an inner production between the test data and the restored data, a restoration ratio f_c of the label class c can be calculated as

$$f_c = \frac{\mathbf{z}^{test^T}\hat{\mathbf{z}}_c^{test}}{||\mathbf{z}^{test}||_2||\hat{\mathbf{z}}^{test}||_2} = \frac{\mathbf{z}^{test^T}\mathbf{Z}_c\hat{\boldsymbol{\alpha}}_c}{||\mathbf{z}^{test}||_2||\hat{\mathbf{z}}^{test}||_2} \tag{6}$$

where Zc is a training data matrix that only contains the data given the label c, and $\ddot{\alpha}_c$ is a coefficient vector associated with the training data in Zc. The restoration ratio f_c implies a confidence of the class c. Therefore, multi-label classification can be realized by calculating all the confidences $f_c(c=1,\cdots,C)$ (C is the number of label classes). However, we use the best class label f_c in all classes by taking the maximum for the following procedure.

3. HIERARCHICAL REPRESENTATION

From Eq. (6), we obtain the class label for one sub-pixel in a certain layer. After obtaining all the class labels for each sub-pixel in each layer, we finally estimate the class label for each region (pixel), considering the hierarchical and spatial co-occurrence in adjacent subregions using Tree Conditional Random Field (TCRF) [10].

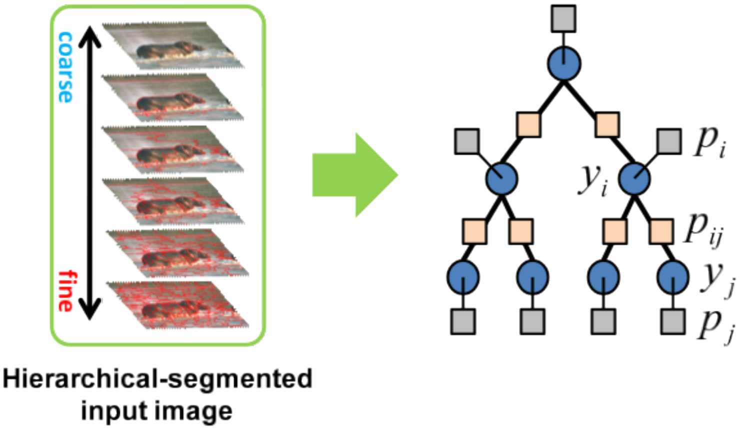


Figure 3: Graphical representation of the image using Tree Conditional Random Field.

Conditional Random Field (CRF) [15] was originally introduced in linguistic processing to represent a graphical and discriminative model. This model is used for estimating the class of the structured observation. When we apply the CRF to the hierarchical-segmented image as shown in Figure 3 *left*, each segment is represented as a node, and all segments that have the relation between layers are connected by an edge (Figure 3 *right*). Such a graphical model is called a TCRF.

Let $i \in T$ denote each segment in a hierarchical-segmented image, $\pi(i)$ be the set of the child nodes for the parent node i, $\mathbf{F} = \{f_i\}_{i \in T}$ describe the class reliability in each segment obtained from the previous section, and $\mathbf{Y} = \{y_i\}_{i \in T}$ indicate the class labels estimated in each node. Then, the model formula of the TCRF is written as the following conditional distribution $P(\mathbf{Y}|\mathbf{F};\theta)$.

$$P(\mathbf{Y}|\mathbf{F};\theta) = \frac{1}{Z}\exp\left\{\sum_{i\in T} p_i(y_i|\mathbf{f}_i;\alpha) + \sum_{i\in T}\sum_{j\in\pi(i)} p_{ij}(y_i,y_j;\beta)\right\} \tag{7}$$

where Z is called partition for regularization. $\theta = \{\alpha,\beta\}$ represents the model parameters of TCRF estimated based on the following Maximum A Posteriori (MAP) by using all the training images with ground truth.

$$\theta^* = \arg\max_{\theta}\left\{\sum_{t=1}^{T}\log P(\mathbf{y}^t|\mathbf{X}^t;\theta) - \frac{R}{2}||\theta||^2\right\} \tag{8}$$

where T is the number of the training images, and R is the parameter for preventing over-fitting. θ^* is computed analytically by an L-BFGS method [9]. The first term in Eq. (7) $p_i(y_i|\mathbf{x}_i;\alpha)$ is the class reliability distribution in each node, which is defined as follows.

$$p_i(y_i|\mathbf{f}_i;\alpha) = \sum_{k=1}^{C}\alpha_{ky_i}f_{ik} \qquad \left(\sum_{k=1}^{C} f_{ik} = 1\right) \tag{9}$$

The second term in Eq. (7) $p_{ij}(y_i,y_j;\beta)$ is the class co-occurrence between the adjacent nodes defined as:

$$p_{ij}(y_i,y_j;\beta) = \beta_{y_iy_j} \qquad (y_i \neq y_j) \tag{10}$$

For the final class estimation, we need to find the class of each node that maximizes the conditional distribution shown in Eq. (7), given the test image $I^{test} = \mathbf{F}^{test}$. For this purpose, we use Maximizer of Posterior Marginal (MPM) estimation.

$$y_i^* = \arg\max_{y_i\in\mathcal{C}} P(y_i|\mathbf{F}^{test};\theta^*) = \arg\max_{y_i\in\mathcal{C}}\sum_{\mathbf{Y}\backslash y_i} P(\mathbf{Y}|\mathbf{F}^{test};\theta^*), \quad i \in T \tag{9}$$

where y_i^* is the class maximizing the posterior marginal distribution, and $\mathcal{C}$ is a collection of the labels. Since the graph structure is one of the tree structures, the global optimal estimation can be done by Belief Propagation [16].

By decreasing the segmentation error, we regard the estimation result in the bottom layer as the final estimation result. Since this estimation considers all estimation results in all layers of the hierarchy and is a global optimum, the proposed method is robust to the scale variance of objects.

4. EXPERIMENTAL EVALUATION

In Section 4.1, we first see the effectiveness of the labeling method only using sparse representation in semi-supervised learning described in Section 2. Then, we evaluate our method that combines the sparse representation and hierarchical representation in Section 4.2.

4.1 Object Recognition Using Sparse Representation in Semi-supervised Learning

4.1.1 Experimental Conditions

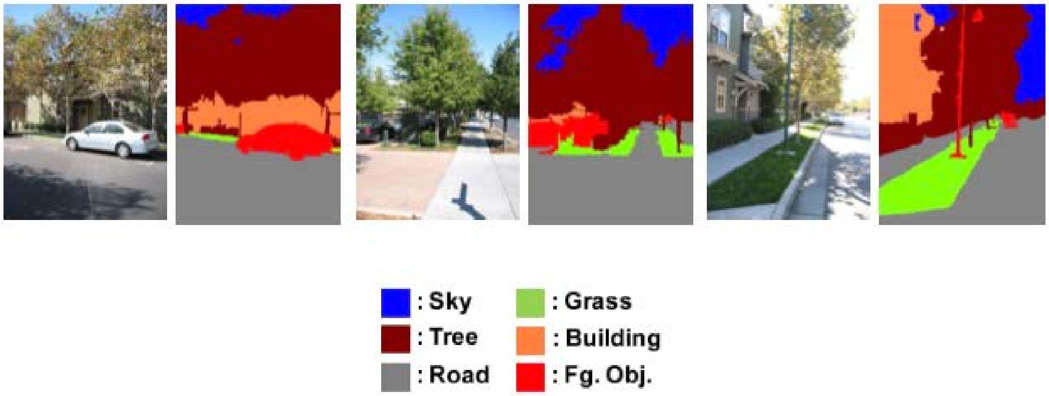


Figure 4: Examples from STAIR image dataset

For this image annotation experiments, we used a STAIR image data set [17], which contains 534 images along with pixel-wise 5 labels (“Sky”, “Tree”, “Road”, “Grass” and “Building”) as their examples are shown in Figure 4. In our experiments, the training and test are conducted using features in each subregion, which is divided by Normalized Cuts. Labeling accuracies for each class and their average are calculated by accumulating the subregion results. The accuracy was evaluated with 3-fold cross validation. Images for the training and the test were randomly selected (400 images for the training and 134 images for the test) three times for each validation.

We conducted two experiments in this section. In the first experiment, we compared with conventional semi-supervised methods: “SemiNB” and “TSVM”. Secondly, we examined these methods in a supervised manner; a supervised variation of our method (without hierarchical representation) was compared with “NB” and “SVM,” just to see the effectiveness of semi-supervised approach. Here we employed CCA instead of semi-CCA, given a full set of labeled data.

4.1.2 Results and Discussion

The results of semi-supervised and supervised approaches are shown in Table 1 and Table 2, respectively. As shown in these tables, the recognition accuracy of our method is higher than not only the other semi-supervised approaches but also the supervised approaches, such as SVM. The other methods, SVM and NB, suffer decreased accuracy in the semi-supervised case. This is, in general, because conventional approaches make extensive use of unsupervised data, and their classifiers were consequently affected by unsupervised factors, especially outliers. On the other hand, our approach increases the accuracy in the semi-supervised case. This is considered to be due to the benefit of semi-supervised learning, which helps the classifier to catch the global structure of data distribution in the case where there is a very small amount of labeled data for the training (due to the effective suppression of outliers using sparse representation).

Table 1: Recognition accuracies of non-hierarchical methods (semi-supervised approaches) [%]

Label	Sky	Tree	Road	Grass	Building	Average
SemiNB	25.3	30.5	70.6	47.5	31.3	41.0
TSVM	82.8	62.0	83.3	74.4	68.6	74.2
Our method	87.2	66.0	84.4	80.1	75.9	78.7

Table 2: Recognition accuracies for comparison of supervised methods [%]

Label	Sky	Tree	Road	Grass	Building	Average
NB	43.2	26.7	73.8	54.3	33.1	46.2
SVM	87.8	57.1	85.5	76.9	65.2	74.5
ROMP	54.5	39.2	51.5	44.5	39.9	45.9
Our method (sp)	85.0	63.7	89.6	75.4	73.2	77.4

4.2 Object Recognition Using Hierarchical Sparse Representation

4.2.1 Experimental Conditions

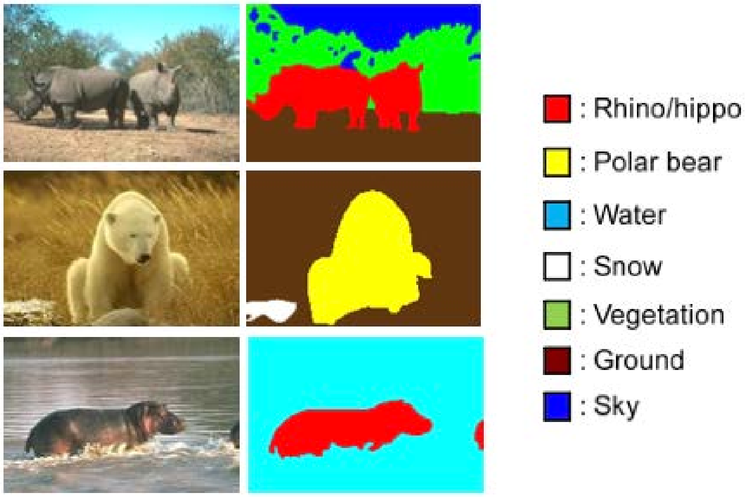


Figure 5: Examples from STAIR image dataset

In this section, we show the object recognition results of our proposed method, where we first estimate class labels for each of the subregions in each layer using the sparse representation method described in Section 2, and then adjust the labels using hierarchical representation as described in Section 3. For the experiments, we used a Corel image dataset

[18], in addition to a STAIR image dataset, which consists of 100 images with 7 labels (“Rhino/Hippo”, “Polar bear”, “Water”, “Snow”, “Vegetation”, “Ground”, and “Sky”) as shown in Figure 5. For this dataset, all images have the same size of 180x120. We divided all images into hierarchical subregions using SWA (Segmentation by Weighted Aggregation) [11], and extracted local features from the lowest layer where we set the number of subregions to 200 per image. Each method was evaluated using leave-one-out cross validation. The experimental conditions related to the STAIR image dataset are the same as in the previous section.

4.2.2 The Number of Layers

First, we investigated our hierarchical method to see how the accuracy changes as the number of layers increases. We changed the number of layers from 1 to 6, and set the number of subregions in each case as shown in Table 3. For example, we divide an image into 200 subregions, 100 subregions, and 50 subregions for the first layer, the second layer, and the third layer, respectively, when we use the four-layer structure.

The results are shown in Figure 6. In most of the cases, the recognition accuracy increases as the number of layers increases. We obtained the best performance when there were 5 layers. Therefore, we used the five-layer structure in the remaining experiments.

Table 3: The number of subregions in each layer

	Layer index					
Number of layers	1	2	3	4	5	6
1	200	-	-	-	-	-
2	200	1	-	-	-	-
3	200	100	1	-	-	-
4	200	100	50	1	-	-
5	200	100	50	25	1	-
6	200	100	50	25	12	1

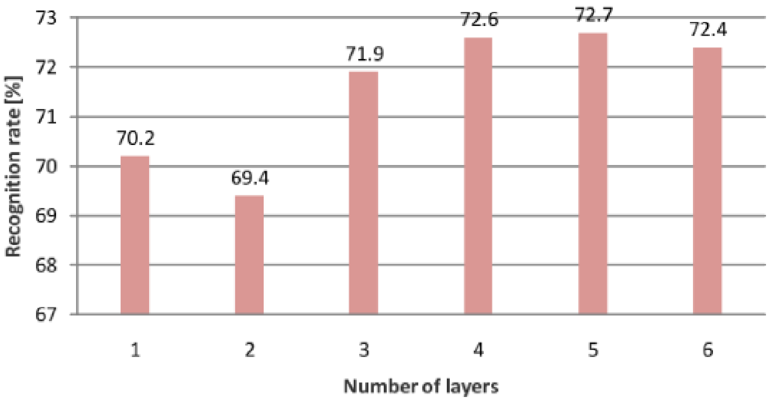


Figure 6: Change in accuracy due to increasing the number of layers.

4.2.3 Experimental Results and Discussion

The experimental results using a Corel image dataset are shown in Table 4. The results of the comparison method "CRF" were obtained from a non-hierarchical structure [19,20]; i.e. the same case as when the number of layers was 1 in the previous section. "LR" has the same conditions as "CRF", except for using logistic regression for the clustering method instead of CRF. As shown in Table 4, our proposed method improved the accuracy by 2.5 points due to the hierarchical structure.

Table 5 shows the results of our proposed method using the STAIR image dataset. The difference between "Our method(a)" and "Our method(b)" is that "Our method(a)" does not have a hierarchical structure but has sparse representation (in the same case as shown in Section 4.1.2). On the other hand, "Our method(b)" has a hierarchical structure. We obtained better performance with "Our method(b)".

Table 4: Recognition accuracies using Corel image dataset [%]

Label	Rhino	P. bear	Water	Snow	Vege.	Ground	Sky	Average
LR	73.5	65.1	70.3	68.2	75.3	71.0	56.6	68.6
CRF	71.8	71.0	82.6	70.6	78.9	74.7	41.7	70.2
Our method	76.2	74.1	80.4	73.0	80.9	74.1	50.4	72.7

Table 5: Recognition accuracies using STAIR image dataset [%]

Label	Sky	Tree	Road	Grass	Building	Average
Our method(a)	87.2	66.0	84.4	80.1	75.9	78.7
Our method(b)	92.1	63.2	85.1	84.9	74.9	80.0

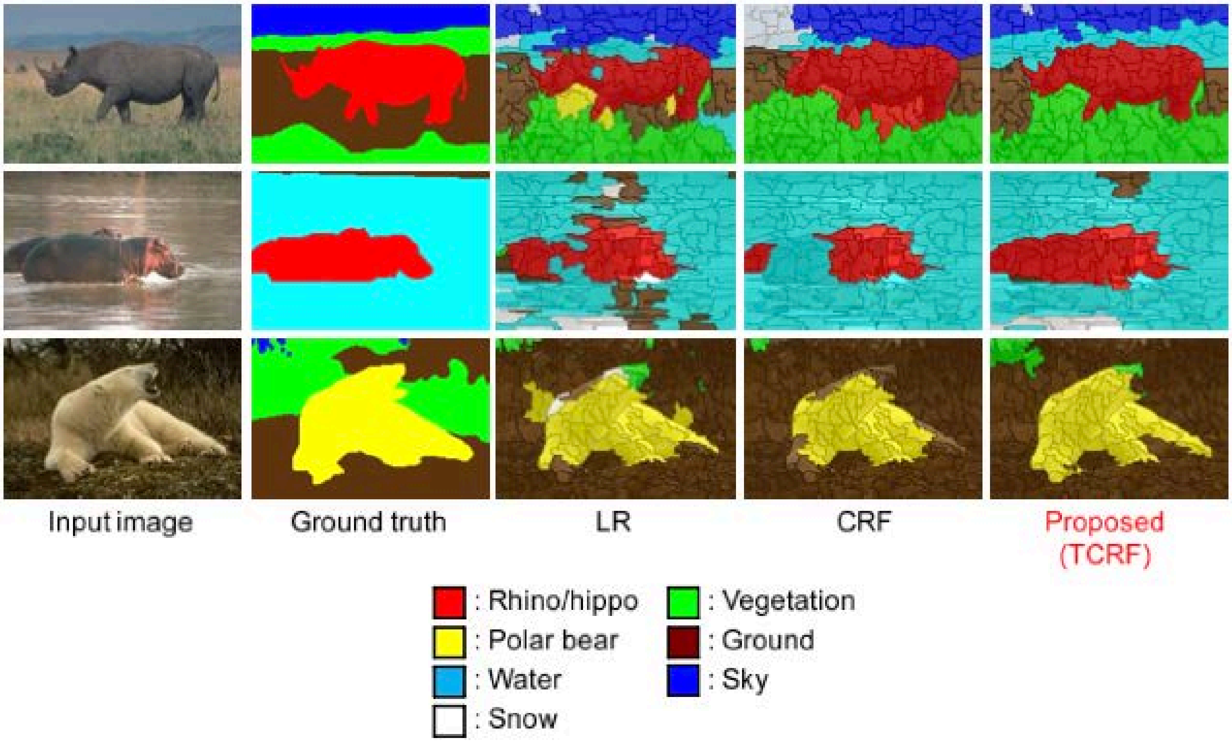


Figure 7: Examples of recognition results from each method

From Table 4 and Table 5, we see the effectiveness of the method that is based on sparse and hierarchical representation, regardless of the dataset. To show the details of the results, we provide some examples of the recognition results in Figure 7. First we compare the conventional methods "LR" and "CRF". The local errors are improved with "CRF" due to considering the co-occurrence of the labels in adjacent regions, resulting in better performance. Our method further considers the hierarchical structure, and comes up with labeling results that are more natural and closer to the ground truth.

5. CONCLUSION

In this paper, we proposed an effective object recognition method, which suitably combines a semi-supervised approach (Semi-supervised Canonical Correlation Analysis; semi-CCA), sparse representation (Regularized Orthogonal Matching Pursuit; ROMP), and hierarchical representation (Tree Conditional Random Field; TCRF). Semi-supervised learning has the advantage of being able to capture a global structure of the true data distribution even when given only a small amount of labeled training data. However, outliers included in unsupervised data often negatively affect the classifier construction. Our approach suppresses such outliers in terms of sparse representation in the subspace that is created using semi-CCA. Furthermore, we adopt TCRF in order to be robust to scale variance, which often causes errors in super-pixel-based object recognition.

The experimental results using two datasets showed the effectiveness of our proposed method satisfactorily. While conventional semi-supervised approaches decreased the labeling accuracy compared with their supervised methods, our sparse-representation-based approach, on the contrary, increased the accuracy, taking full advantage of semi-supervised learning. When we also applied hierarchical representation, we further obtained better results than when the non-hierarchical structured approach is used.

In our method, we used sparse representation and hierarchical representation in a tandem manner. We would like to investigate the integration of these methods into one framework in the future.

REFERENCES

[1]. X. Zhu, Semi-supervised learning literature survey. Proc. IEEE International Conference on Machine Learning (ICML), tutorial, 2007.

[2]. T. Joachims, Transductive inference for text classification using support vector machines. Proc. IEEE Inter6national Conference on Machine Learning (ICML), pp. 200–209, 1999.

[3]. K. Nigam, A. Mccallum, and T. Mitchell, Semi-supervised text classification using EM. In Semi- supervised Learning, pp. 33–56, 2006.

[4]. A. Kimura, et al., Semicca: Efficient semi-supervised learning of canonical correlations. Proc. IEEE International Conference on Pattern Recognition (ICPR), pp. 2933–2936, 2010.

[5]. M. Culp and G. Michailidis, An iterative algorithm for extending learners to a semi-supervised setting. Journal of Computational and Graphical Statistics, pp. 545– 571, 2008.

[6]. M. Elad, M. A. T. Figueiredo, and M. Yi, On the role of sparse and redundant representations in image processing. Proc. IEEE Special Issue on Applications of Sparse Representation and Compressive Sensing, pp. 972–982, 2010.

[7]. D. Needell and R. Vershynin, Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. Foundations of Computational Mathematic, pp. 317–334, 2009.

[8]. X. Ren, J. Malik, Learning a classification model for segmentation. Proc. IEEE International Conference on Computer Vision (ICCV), pp. 10-17, 2003.

[9]. P. F. Felzenszwalb, D. P. Hutenlocher, Efficient graph-based image segmentation. International Journal of Computer Vision (IJCV), vol. 59, no. 2, pp. 167-181, 2004.

[10]. Takeshi Okumura, Tetsuya Takiguchi, Yasuo Ariki, Generic Object Recognition by Tree Conditional Random Field Based on Hierarchical Segmentation. ICPR2010, pp. 3025-3028, 2010.

[11]. E. Sharon, A. Brandt, and R. Basri, Fast multiscale image segmentation. Proc. IEEE Computer Vision and Pattern Recognition, pp 70-77, 2000.

[12]. J. Shi and J. Malik, Normalized cuts and image segmentation. Proc. IEEE Computer Vision and Pattern Recognition, pp. 731–737, 1997.

[13]. J. Wright, A. Ganesh, S. Rao, and Y. Ma, Exact recovery of corrupted low-rank matrices by convex optimization. Proc. IEEE, 2009.

[14]. A. Y. Yang, R. Jafari, S. S. Sastry, and R. Bajcsy, Distributed recognition of human actions using wearable motion sensor networks. Journal of Ambient Intelli- gence and Smart Environments, pp. 103–115, 2009.

[15]. J. D. Lafferty, A. McCallum, and F. C. N. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proc. International Conference on Machine Learning, 2001.

[16]. C. M. Bishop, Pattern Recognition and Machine Learning. Springer, Chapter. 8, 2006.

[17]. Stanford artificial intelligence robot (stair) image dataset. http://cs.stanford.edu/group/stair/

[18]. P. Duygulu, K. Barnard, J. de Freitas, and D. Forsyth, Object recognition as machine translation: learning a lexicon for a fixed image vocabulary. ECCV, pp. 97-112, 2002.

[19]. J. Shotton, J. Winn, C. Rother, and A. Criminisi, Textonboost: joint appearance, shape and context modeling for multi-class object recognition and segmentation., Proc. IEEE European Conference on Computer Vision, pp. l-15, 2006.

[20]. S. Gould, J. Rodgers, D. Cohen, G. Elidan and D. Koller, Multi-Class segmentation with relative location prior. International Journal of Computer Vision, pp. 300-316, 2008.