Analysis of Class Separation and Combination of Class-Dependent Features for Handwriting Recognition

Il-Seok Oh, *Member*, *IEEE*, Jin-Seon Lee, and Ching Y. Suen, *Fellow*, *IEEE*

Abstract—In this paper, we propose a new approach to combine multiple features in handwriting recognition based on two ideas: feature selection-based combination and class-dependent features. A nonparametric method is used for feature evaluation, and the first part of this paper is devoted to the evaluation of features in terms of their class separation and recognition capabilities. In the second part, multiple feature vectors are combined to produce a new feature vector. Based on the fact that a feature has different discriminating powers for different classes, a new scheme of selecting and combining class-dependent features is proposed. In this scheme, a class is considered to have its own optimal feature vector for discriminating itself from the other classes. Using an architecture of modular neural networks as the classifier, a series of experiments were conducted on unconstrained handwritten numerals. The results indicate that the selected features are effective in separating pattern classes and the new feature vector derived from a combination of two types of such features further improves the recognition rate.

Index Terms—Handwriting recognition, class separation, nonparametric method, class-dependent feature combination, modular neural network.

1 Introduction

As a major factor influencing recognition performance, features play a very important role in handwriting recognition. This has led to the development of a variety of features for handwriting recognition and their recognition performances have been reported on standard databases. Readers can find surveys in [1], [2]. Some recent papers include those proposing directional distance features [2], gradient-based features [3], wavelet-based features [4], pixel distance features [5], and concavity features [6]. The features do not necessarily convey any intuitive meaning to a human and the dimensionality of the feature vectors is very high, in the hundreds, so it is difficult to understand their discriminative characteristics. A systematic evaluation of features in a specific feature vector is very important for designing a new feature vector by combining different feature vectors.

Using a single feature type has shown a certain limitation in achieving satisfactory recognition performance and this leads us to use multiple types of feature. This can be viewed as an analogy to the combination of multiple experts, which is now common practice [7]. Likewise, combination of multiple types of feature has been attempted. In [6], several combinations of features were tested. Using three types of feature representing local, intermediate, and global shapes, they reached the conclusion that the combination of three feature types improved recognition performance. A feature selection-based approach has also been tested [8], [9]. Those papers proved that the approach of multiple feature combination produced a promising improvement in recognition performance.

Some of the above-referenced papers used a simple scheme of combination which just cascades multiple feature vectors. This scheme results in a much larger dimensionality of the new feature vector and it is highly probable that many redundant features exist. Those redundant features may reduce the recognition performance. Also, only features common to all classes were used. This does not take into consideration that a feature has its own expertise (specialties) for discriminating different classes.

Our primary objectives in this study are twofold: 1) to analyze the class separation capability of features used in handwriting recognition and 2) to improve recognition performance by combining multiple features. In the first objective, we attempt to understand intuitively the discriminative characteristics of the features in a specific feature vector through a systematic experiment using an evaluation tool appropriate for handwriting. Regarding the evaluation tool, our research scope includes neither the presentation of a new feature selection algorithm nor the comparative study of the conventional algorithms, but the important point of choosing a proper tool and methodology for the domain of the handwriting. In the second objective, by utilizing knowledge of the features, we attempt to combine multiple feature vectors to produce a new compact feature vector with a higher discriminative power. A new approach called class-dependent features is proposed that uses knowledge of the features' different expertise in discriminating the different classes.

Regarding the tool for evaluating features, a good tutorial on the criteria and selection of a good subset of features can be found in [10], [11], [12]. Node pruning for neural network classifiers [13], entropy measurement [14], and class separation [15] are conventional methods. Our choice for feature evaluation is *class separation*, which is a measurement showing how well the class distributions of different classes are separated in feature space. A wider class separation implies a better discriminating power.

The first part of this paper explains class separation in conjunction with the actual recognition rate. A systematic and thorough analysis of features is used in designing a new feature vector by combining multiple feature vectors. In the second part, we combine multiple features based on class separation information. A simple fact is that a feature may have different merits to different classes in terms of discriminating power. For example, a feature may be especially superior in discriminating the numeral class 0 from the other nine classes while it is inferior to the other nine classes. Based on this fact, a new scheme of combining *class-dependent* features is proposed in this paper. In the proposed scheme, a class is considered to have its own optimal feature vector for discriminating itself from the other classes. These feature vectors are designed using the class separation information. Since a conventional neural network structure cannot accommodate this scheme because the classes have different feature vectors, the architecture of modular neural networks described in [16] is adopted as a classifier. The modular network is suitable for class-dependent features because it has a structure such that each class has its own subnetwork independent of other classes.

2 Measuring Class Separation

2.1 Preliminaries

We have g classes, each represented as ω_i. A *feature vector* $\underline{X}$ is a d-dimensional vector composed of a set of *feature cells* identified by x_i, that is, $\underline{X} = (x_0, x_1, \ldots, x_{d-1})$. A class ω_i has N_i samples in the training database. A set of samples from the class ω_i is denoted by $Z_{\omega i} = \{\underline{z}_i^1, \underline{z}_i^2, \ldots, \underline{z}_i^{Ni}\}$, where $\underline{z}_i^k$ means a sample located at the kth position in $Z_{\omega i}$.

- *I.-S. Oh is with the Department of Computer Science, Chonbuk National University, Chonju, Chonbuk 561-756, Korea. E-mail: isoh@moak.chonbuk.ac.kr.*
- *J.-S. Lee is with the Department of Computer Engineering, Woosuk University, Wanju-kun, Chonbuk 565-701, Korea. E-mail: jslee@core.woosuk.ac.kr.*
- *C.Y. Suen is with the Centre for Pattern Recognition and Machine Intelligence, Concordia University, Montreal, Quebec H3G 1M8, Canada. E-mail: suen@cenparmi.concordia.ca.*

Manuscript received 2 June 1997; revised 14 July 1999. Recommended for acceptance by J. Hull. For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 107721.

TABLE 1
Class Separation and Recognition Rate for the Top 16 Feature Cells in the Ordered List

	CGD			DDD		
order	sequence number	class separation	recognition rate	sequence number	class separation	recognition rate
1	142	1.433	86.00	156	1.736	93.00
2	150	1.239	83.50	136	1.732	91.25
3	246	1.210	75.75	149	1.605	90.50
4	90	1.179	75.00	155	1.566	92.50
5	237	1.165	79.25	143	1.520	86.50
6	184	1.162	79.50	146	1.505	88.25
7	185	1.060	74.75	150	1.489	87.75
8	91	1.030	68.75	152	1.446	82.50
9	159	1.017	71.75	145	1.425	87.50
10	214	0.976	77.00	172	1.418	83.25
11	149	0.962	68.00	72	1.395	87.25
12	166	0.945	70.50	251	1.391	83.00
13	245	0.942	68.75	154	1.384	86.75
14	151	0.935	78.00	153	1.331	84.00
15	89	0.920	71.00	148	1.330	85.25
16	228	0.905	67.75	79	1.328	86.00

First, let us introduce some definitions regarding *class separation*, which is a measurement of how well two classes are separated by a feature vector $\underline{X}$. It is labeled as $S^{cc}(\omega_i, \omega_j, \underline{X})$, where ω_i and ω_j are two classes and $\underline{X}$ is a feature vector composed of one or more feature cells. In case that we have the g classes, $\underline{X}$ must separate a group of those g classes instead of only two classes. So, for this case, we use the measurement, $S^g(\Omega, \underline{X})$, where $\Omega = \{\omega_1, \omega_2, \ldots, \omega_g\}$. Another measurement, $S^{cg}(\omega_i, \Omega, \underline{X})$ is related with class-dependent feature combination which will be proposed in Section 4. $S^{cg}(\omega_i, \Omega, \underline{X})$ is a separation measurement between a class ω_i and a group of g-1 classes, $\Omega = \{\omega_k | 1 \leq k \leq g, k \neq i\}$ by a feature vector $\underline{X}$. In other words, S^{cg} represents a degree of how well $\underline{X}$ can discriminate a class from the other g-1 classes. This measurement will be used to design a new feature vector for each of g classes.

A series of experiments will be performed for the task of evaluating features and improving recognition performance by combining multiple features. The database and features to be used are as follows: We have chosen the CENPARMI handwritten numeral database, which consists of 4,000 training samples (400 samples/class) and 2,000 test samples (200 samples/class). Since this database has been constructed from real-life postal mail pieces, it contains totally unconstrained samples produced by more than 1,000 anonymous writers and writing tools. So, we believe that it is well-suited to our experiments of estimating the class distributions. Actually, many researchers in handwritten numeral recognition have shown the effectiveness of their approaches by using this database from early 80s until now.

Two feature sets are used in the experiments. They are numerical features that have good recognition performance for handwritten numerals. An input pattern P_{m*n} is first size-normalized into a 16*16 mesh, R_{16*16}, and then converted into a 16*16 binary mesh. The first feature, called CGD (Contour-based Gradient Distribution), is computed by first applying the Sobel edge operator to the normalized mesh R and computing the gradient direction distribution map. The map has 256 real values and they constitute a 256-dimensional feature vector, CGD. The second feature called DDD (Directional Distance Distribution) is computed using distance information. Each pixel in the binary map R shoots rays in eight directions and each ray computes the distance to the pixel with opposite color (black or white). Using the directional distance information of the pixels in R, the directional distance distribution map is computed. The map has 256 real values and they constitute a 256-dimensional feature vector, DDD. Both CGD and DDD can be represented with a 256-dimensional feature vector $\underline{X} = (x_0, x_1, \ldots, x_{255})$. For their detailed algorithms, we refer the readers to [3], [2].

The CGD contains the local shape information about the input pattern because the edge operator can extract only the local gradient direction information. On the contrary, DDD has the global shape information since the eight directional distance information provides a rough sketch of the global pattern shape. Because of this, we have chosen the CGD and DDD as a pair of feature vectors having a good complementarity. This choice is similar to [6], where the authors used local, intermediate, and global shape features to exploit the complementarity.

2.2 Nonparametric Method

In this method, a probability density distribution for a class by a feature vector $\underline{X} = (x_0, x_1, \ldots, x_{d-1})$ is estimated as follows [15]: A unit step function is defined as follows:

$$\Phi(\underline{X}) = \begin{cases} 1, & \text{if } |x_i| < 1/2, i = 0, 1, \ldots, d-1 \\ 0, & \text{otherwise} \end{cases}$$

Note that $\Phi(\underline{X})$ is a function whose summation over the whole d-dimensional space R_d will be 1. $\Phi(\underline{X})$ is called a kernel function. A probability distribution for the class ω_i can be estimated by the formula,

$$P_n(\underline{X}) = \frac{1}{Ni} \sum_{k=1}^{N_i} \frac{\Phi((\underline{X} - \underline{z}i^k)/h_n)}{V_n},$$

where h_n acts as a smoothing factor and $V_n = (h_n)^d$. Note that $P_n(\underline{X})$ depends on the smoothing factor h_n. Large h_n means a large degree of smoothing and small h_n means a small degree of smoothing. Using the estimated class distributions, the separation between two classes ω_i and ω_j can be defined as,

$$S^{cc}(\omega i, \omega j, \underline{X}) = \int_{Rd} \left| Pn^{\omega i}(\underline{X}) - Pn^{\omega j}(\underline{X}) \right| dx,$$

where R_d is a d-dimensional real space and $P_n^{\omega i}(\underline{X})$ and $P_n^{\omega j}(\underline{X})$ are the estimated distributions for the classes ω_i and ω_j,

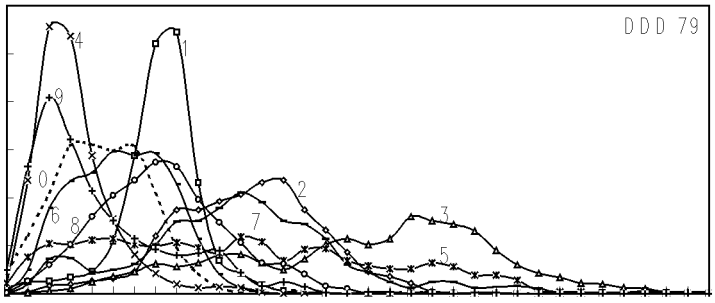

Fig. 1. Class distributions for 10 numeral classes.

respectively. This formula measures the degree of overlapping distance between two distributions. In an extreme case of nonoverlapping, S^{cc} will be 2.0 which is the maximum. In this case, the feature vector $\underline{X}$ can discriminate two classes ω_i and ω_j perfectly. In case of exact overlapping, S^{cc} will be 0 and $\underline{X}$ is useless in discriminating ω_i and ω_j.

The class separation for a group of classes is formulated using S^{cc} as follows:

$$S^g(\Omega, \underline{X}) = \sum_{\omega i \in \Omega} \sum_{\omega j \in \Omega, j \neq i} S^{cc}(\omega i, \omega j, \underline{X}),$$

where $\Omega = \{\omega_1, \omega_2, \ldots, \omega_g\}$. Also, the separation between a class ω_i and a class group $\Omega = \{\omega_k | 1 \leq k \leq g, k \neq i\}$ can be formulated using S^{cc} as

$$S^{cg}(\omega i, \Omega, \underline{X}) = \sum_{\omega k \in \Omega} S^{cc}(\omega i, \omega k, \underline{X}).$$

3 Class Separations and Recognition Capabilities

We evaluate features considering class separation in conjunction with the recognition rate obtained experimentally. In the experiments, individual feature cells are manipulated separately and they are ordered according to their values of class separation. The following describes the ordering algorithm for two classes ω_p and ω_q.

Algorithm-1D:
1. R = empty list and $P = \{x_i | 0 \leq i \leq d-1\}$.
2. If P is empty, stop.
3. Choose $x_k \subseteq P$ such that $S^{cc}(\omega_p, \omega_q, \underline{X} = (x_k)) \geq S^{cc}(\omega_p, \omega_q, \underline{X} = (x_{k'}))$ for all $x_{k'} \subseteq P$ and $k' \neq k$.
4. Insert x_k into R and $P = P - x_k$.
5. Goto step 2.

We use both a partial classifier and a full 10-classifier. A partial k-classifier where k is less than g is a classifier which takes into consideration only k classes. (In actual applications, 2-classifiers could be used as a confusing pair resolver.) A neural net classifier is trained and tested using individual feature cells one by one. The neural network architecture of a k-classifier is the same as a 10-classifier except that it has only k output nodes and we use only the samples belonging to the k classes in training the classifier.

To test a 2-classifier, the pair of numeral classes 3 and 8 were chosen for the experiment. We computed $S^{cc}(3, 8, \underline{X} = (x_i))$ by changing i from 0 to 255 for the CGD and DDD feature vectors. Table 1 lists the 16 feature cells in the top of the ordered list and their class separations. The DDD feature cells have much better class separation. (Actually, DDD has a better recognition rate than CGD, as can be seen in Table 4 in Section 4.) Also, the 2-classifier is trained and tested using one feature cell one by one taken from the CGD and DDD. Table 1 also shows the recognition rate of the individual feature cells. The general trend is that a feature cell with a higher class separation produces a better recognition rate. This means that class separation represents well the discriminating power of a feature cell.

For 10-classification, Fig. 1 depicts the class distributions estimated for a feature cell, x_{79} of DDD. In this case, we use the criterion function $S^g(\Omega, \underline{X})$ defined already in Section 2.2. Table 2

TABLE 2
Class Separations for a Feature Cell, x79 of DDD

	S^{cc}										S^{cg}
	0	1	2	3	4	5	6	7	8	9	
0	0.00	1.07	1.54	1.66	0.70	1.07	0.33	1.38	0.81	0.57	9.13
1	-	0.00	1.24	1.53	1.55	1.20	0.77	1.20	0.64	1.23	10.44
2	-	-	0.00	1.04	1.73	0.79	1.38	0.31	0.92	1.39	10.33
3	-	-	-	0.00	1.75	0.75	1.63	0.96	1.33	1.48	12.13
4	-	-	-	-	0.00	1.26	0.93	1.55	1.29	0.44	11.21
5	-	-	-	-	-	0.00	1.05	0.54	0.77	0.90	8.34
6	-	-	-	-	-	-	0.00	1.25	0.53	0.73	8.59
7	-	-	-	-	-	-	-	0.00	0.81	1.20	9.20
8	-	-	-	-	-	-	-	-	0.00	0.92	8.02
9	-	-	-	-	-	-	-	-	-	0.00	8.88
S^g											96.27

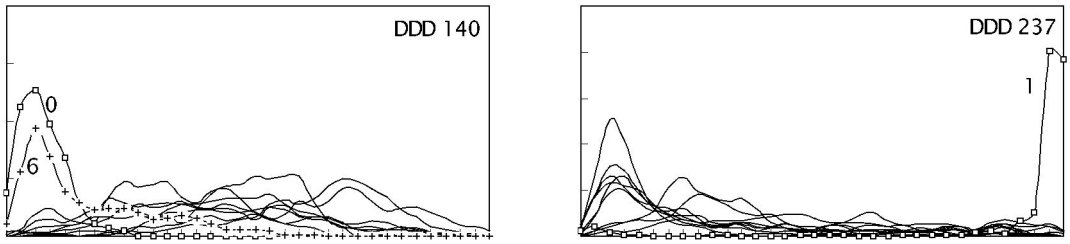

Fig. 2. Example class distributions advocating the class-dependent feature combination.

shows the class separations, S^{cc}, S^{cg}, and S^{g} for the distributions. The pair of classes 3 and 4 have the biggest class separation, 1.75. In the graphs of Fig. 1, we can observe the large overlapping distance between the classes 3 and 4. Contrary to this, the classes 2 and 7 have the smallest class separation, 0.31.

4 Feature Combinations

As can be observed from the last section, feature cells have different class separations, which means different power for discriminating the classes. The difference between the best group and the worst group is extreme. The set of feature cells in the best group can be used without loss of recognition performance. On the contrary, most of the cells in the worst group have no discriminatory power or are redundant. Based on this fact, this section proposes methods of combining multiple features using the class separation information with the purpose of increasing recognition performance. The task of feature combination is to compose a new feature vector from two or more original feature vectors. The dimensionality of the new feature vector must be less than the sum of dimensions of the original feature vectors. The purpose of feature combination is to construct a new feature vector which will produce a better recognition rate than any of the original feature vectors.

We can construct one feature vector which will be used by all the classes. We call this a class-common approach. In another approach, a class has its own specific feature vector different from those of the other classes. It will be called a *class-dependent* approach. Its rationale, algorithm, and modular classifier architecture suitable for this approach are described in Section 4.2. To the best of our knowledge, the class-dependent approach is a novel technique.

4.1 Class-Common Feature Combination

The original feature vectors are denoted by $\underline{\mathrm{Xi}}$, $1 \le i \le n$, with a dimension of d_i. The total number of feature cells in them is N computed by summing all the d_i, $1 \le i \le n$. These N feature cells form the input to the algorithm. Using the ordering algorithm described in Section 3, we obtain the ordered list of N feature cells. The best k feature cells are used as a new feature vector $\underline{\mathrm{F}}$. It is sure that $\underline{\mathrm{F}}$ will provide a better class separation than the CGD or DDD.

The setting of k is not our main concern in this paper. It may be determined empirically. In the experiment described in Section 4.3, we used two feature vectors, CGD and DDD, each having 256-dimensions. So, N is 512. And, we take 256 feature cells from the ordered list of 512, so k is 256. The new feature vector $\underline{\mathrm{F}}$ will be used by all the classes.

4.2 Class-Dependent Feature Combination

The basic idea behind the combination of class-dependent features comes from the simple fact that a feature cell has different merits to different classes in terms of its discriminatory power. Fig. 2 clarifies the idea. Two charts in the figure illustrate the class distributions for 10 numeral classes for feature cells, x_{140} and x_{237} in the DDD feature vector. The feature cell x_{140} in the first chart is very powerful in discriminating the specific numeral classes 0 and 6 from the other eight classes. However, discrimination between the classes 0 and 6 is poor. The second chart shows another case where the class 1 can be expected to be discriminated very well from the other nine classes.

In the class-dependent approach, we must have a modular concept to manipulate a class ω_i independently from the other g-1 classes. Note that we have already defined a criterion function, $S^{cg}(\omega_i, \Omega, \underline{\mathrm{X}})$ for this concept in Section 2.2. The function S^{cg} is computed for two feature cells in Fig. 2. Table 3 summarizes the results. As we have already seen in Fig. 2, x_{140} gives good class separation for the classes 0 and 6 and x_{237} results in an excellent separation for class 1.

In the class-dependent combination scheme, each of g classes is processed separately and a new feature vector $\underline{\mathrm{F_i}}$ will be designed for the class ω_i from the original feature vectors $\underline{\mathrm{X}}_i$, $1 \le i \le n$. Like the class-common approach, we can use the ordering algorithm in Section 3. Since the class-dependent scheme uses different criterion functions, the algorithms are modified accordingly. The following algorithm shows the modified Algorithm-1D.

Algorithm-1D-class-dependent:
for each ω_c from c = 1 to g do begin
1. $R_c =$ empty list and $P = \{x_i | 0 \le i \le N-1\}$.
2. If P is empty, stop.
3. Choose $x_k \subseteq P$ such that $S^{cg}(\omega_c, \Omega, \underline{\mathrm{X}} = (x_k)) \ge S^{cg}(\omega_c, \Omega, \underline{\mathrm{X}} = (x_{k'}))$ for all $x_{k'} \subseteq P$ and $k' \ne k$ where $\Omega = \{\omega_j | 1 \le j \le g, j \ne c\}$.
4. insert x_k into R_c and $P = P - x_k$.

TABLE 3
Class Separations of 10 Numeral Classes for Two Cases in Fig. 2

Classes / cells	0	1	2	3	4	5	6	7	8	9
x_{140}	14.94	9.20	6.95	8.92	6.80	6.90	10.66	10.32	6.96	9.81
x_{237}	6.19	15.31	6.53	8.98	9.03	6.01	5.99	8.97	6.03	7.96

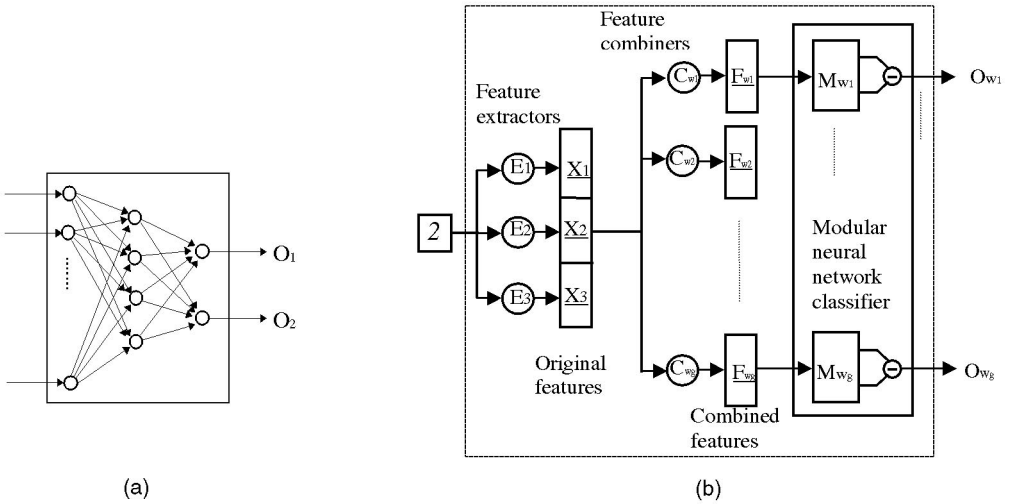

Fig. 3. Classifier architecture for the class-dependent features. (a) A subnetwork. (b) Whole network.

5. Goto step 2.
end.

The execution of Algorithm-1D-class-dependent provides g ordered lists, R_i for a class ω_i, $1 \leq i \leq g$. For the class ω_i, we can take the best k_i feature cells from R_i as a new feature vector which will provide a large class separation between the class ω_i and the class group $\Omega = \{\omega_j | 1 \leq j \leq g, j \neq i\}$. The value of k_i may differ among classes. Some classes with a poorer class separation or a lower recognition performance can have a bigger dimension than the other classes. We do not concern ourselves with deciding k_i in this paper. In our experiment, CGD and DDD are used as the original feature vectors, so N is 512. All the $\underline{F}_i$ have the same dimension, 256.

Now, we have the g different feature vectors for the g classes. One problem regarding this feature structure must be considered. The conventional classifiers have a structure which can accommodate only one feature vector common to all the g classes. So, a classifier with a new structure is required to accommodate our feature structure. Since a class ω_i has its own feature vector $\underline{F}_i$, it must also have its own subclassifier. And, as $\underline{F}_i$ has been designed to discriminate the class ω_i from the other g-1 classes, the subclassifier should have a structure which can classify the patterns coming from the class ω_i and those from the other g-1 classes.

The modular neural network classifier which has been proposed in [16] is well suited to this application. It consists of g subnetworks, each responsible for one of g classes. A subnetwork is shown in Fig. 3a. The function of this subnetwork is to classify two groups of classes, $\Omega_1 = \{\omega_i\}$ and $\Omega_2 = \{\omega_k | 1 \leq k \leq g, k \neq i\}$. So, it has two nodes at the output layer, one for Ω_1 and the other for Ω_2, called O_1 and O_2, respectively.

We put one input layer, one hidden layer, and one output layer on the subnetwork. The three layers are fully connected. The input layer has n nodes to accept the feature vector $\underline{F}_i = (f_0, f_1, \ldots, f_{n-1})$ for the class ω_i. Each of the g subnetworks is trained independently by using the error backpropagation algorithm. To train the subnetwork for the class ω_i, we reorganize the training samples into two groups, Z_{positive} and Z_{negative}. Z_{positive} will have the samples from the class in Ω_1 and Z_{negative} the samples from the classes in Ω_2. The samples in Z_{positive} are fed with the expected output, $(O_1, O_2) = (1.0, 0.0)$, and the samples in Z_{negative} with the expected output, $(O_1, O_2) = (0.0, 1.0)$. In the recognition process, the subnetwork produces a single output by subtracting the value of O_1 by the one of O_2.

The architecture of the whole classifier is depicted in Fig. 3b. From an input pattern, the original feature vectors are extracted. Each class has its own feature combiner which has been determined by the process of the class-dependent feature combination described in the above. The combined feature vectors are fed into the corresponding subnetworks. A class ω_i has its own subnetwork $M_{\omega i}$ which produces a single output, $O_{\omega i}$. The input pattern is finally classified into the class ω_i with maximum output.

4.3 Experimental Results

We used two features, CGD and DDD as the original feature vectors. Both have 256 dimensions. As already stated, each subnetwork is trained using two sample groups, Z_{positive} and Z_{negative}. The training and testing are performed using CENPARMI numeral database.

TABLE 4
Comparison of Recognition Rates (%) for CENPARMI Database (No Rejection)

	train set (4,000 samples)	test set (2,000 samples)
CGD	98.92	95.10
DDD	98.97	97.30
class-common	99.42	97.60
class-dependent	99.47	97.85

TABLE 5
Comparison of Recognition Rates (%) for CEDAR Database (No Rejection)

	train set (18,468 samples)	BS test set (2,711 samples)	good BS test set (2,213 samples)
CGD	99.52	96.50	98.15
DDD	99.49	96.55	98.42
class-common	99.68	96.98	98.59
class-dependent	99.71	97.27	98.73

Table 4 compares the recognition rate of the original feature vectors, CGD and DDD, and their class-common and class-dependent combinations. The DDD has a much better recognition performance than CGD. This can be explained using their class separation measurements shown in Table 1. The better class separation of DDD means a better discriminating power.Two combinations were tested. The first combination is class-common. The best 256 feature cells from the ordered list of 512 feature cells from the CGD and DDD feature vectors are taken as a new feature vector which is commonly used by all 10 classes. In this case, 42 CGD features and 214 DDD features have been selected. The second combination is class-dependent. From two feature vectors, CGD and DDD, we extract 10 new feature vectors with 256 dimensions for 10 numeral classes. Between 47 and 64 CGD features have been selected for each class. The others were DDD features. The class-dependent features produced the highest recognition rate of 97.85 percent on the CENPARMI test set, while the class-common features showed 97.6 percent.

We think that the improvement of 0.55 percent from the single best feature vector DDD by using the class-dependent combination is meaningful in view of the state of the art in the handwritten numeral recognition. A survey of recent papers like [2], [4], [14] revealed that the state of the art performance for the CENPARMI database is between 97 percent and 98 percent. We believe that the improvement by 0.55 percent is quite meaningful. The improvement is also promising in the sense that there still exists some room for further improvement by methods that choose more complementary feature types and take into consideration the mutual dependency of features.

To confirm the test results and our conclusion, we performed another experiment using CEDAR handwritten numeral database. The database consists of 18,468 training samples (BR dataset), 2,711 test samples (BS dataset), and 2,213 test samples (good BS dataset). The good BS has been constructed by choosing the well-segmented samples from the BS dataset. From Table 5, we can also conclude that the class-dependent features produce a significant improvement over the original and class-common features.

5 Concluding Remarks

A nonparametric method for feature selection was shown to work well on handwritten numeral data. Significant class separation was observed, as well as an improvement in recognition performance. Using the class separation information, feature combination was applied. A new scheme for class-dependent feature combination was proposed which exploits the simple fact that a feature has different merits to different classes in terms of discriminating power. By letting a class have its own feature vector specifically suitable for that class, an improvement of recognition performance was obtained.

Future work is to develop a more sophisticated formulation of the search space for composing new feature vectors and an efficient searching algorithm which takes into consideration the mutual dependencies of features. Another future task is to conduct an analysis of the complementarity among various features. This paper used only two types of features without information about their complementarity. We believe that using more complementary features will further improve recognition performance.

Acknowledgments

The authors of this paper wish to acknowledge the constructive comments made by the referees and the associate editor handling this paper.

References

[1] O.D. Trier, A.K. Jain, and T. Taxt, "Feature Extraction Methods for Character Recognition—A Survey," *Pattern Recognition,* vol. 29, no. 4, pp. 641-662, 1996.
[2] I.-S. Oh and C.Y. Suen, "Distance Features for Neural Network-Based Recognition of Handwritten Characters," *Int'l J. Document Analysis and Recognition,* vol. 1, no. 2, pp. 73-88, 1998.
[3] G. Srikantan, S.W. Lam, and S.N. Srihari, "Gradient-Based Contour Encoding for Character Recognition," *Pattern Recognition,* vol. 29, no. 7, pp. 1,147-1,160, 1996.
[4] S.W. Lee, C.H. Kim, H. Ma, and Y.Y. Tang, "Multiresolution Recognition of Unconstrained Handwritten Numerals with Wavelet Transform and Multilayer Cluster Neural Network," *Pattern Recognition,* vol. 29, pp. 1,953-1,961, 1996.
[5] N.W. Strathy and C.Y. Suen, "A New System for Reading Handwritten ZIP Codes," *Proc. ICDAR,* pp. 74-77, 1995.
[6] J.T. Favata, G. Srikantan, and S.N. Srihari, "Handprinted Character/Digit Recognition Using a Multiple Feature/Resolution Philosophy," *Proc. IWFHR '94,* pp. 57-66, 1994.
[7] J. Kittler and M. Hatef, "Improving Recognition Rates by Classifier Combination," *Proc. IWFHR '96,* pp. 81-101, 1996.
[8] A.K. Chhabra et al., "High-Order Statistically Derived Combinations of Geometric Features for Handprinted Character Recognition," *Proc. ICDAR,* pp. 397-401, 1993.
[9] L. Heutte et al., "Handwritten Numeral Recognition Based on Multiple Feature Extractors," *Proc. ICDAR,* pp. 167-170, 1993.
[10] J. Kittler, "Feature Selection and Extraction," *Handbook of Pattern Recognition and Image Processing,* T.Y. Young and K.-S. Fu, eds. Academic Press, 1986.
[11] J. Schurmann, *Pattern Classification: A Unified View of Statistical and Neural Approaches.* John Wiley and Sons, 1996.
[12] B.D. Ripley, *Pattern Recognition and Neural Networks.* Cambridge Univ. Press, 1996.
[13] R. Reed, "Pruning Algorithms—A Survey," *IEEE Trans. Neural Networks,* vol. 4, no. 5, pp. 740-747, 1993.
[14] P.D. Gader and M.A. Khabou, "Automatic Feature Generation for Handwritten Digit Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 12, pp. 1,256-1,261, Dec. 1996.
[15] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis.* Wiley-Interscience, 1973.
[16] I.-S. Oh, J.-S. Lee, K.-C. Hong, and S.-M. Choi, "Class-Expert Approach to Handwritten Numeral Recognition," *Proc. IWFHR '96,* pp. 35-40, 1996.