Combination of Structural Classifiers[1]

Tin Kam Ho, Jonathan J. Hull, Sargur N. Srihari
Department of Computer Science
State University of New York at Buffalo, Amherst, NY 14260, USA

A technique for combining the results of structural classifiers in a multi-classifier image recognition system is presented. Each classifier produces a ranking of a set of classes. The combination technique uses these rankings to determine a small subset of the set of classes that contains the correct class. A group consensus function is then applied to re-rank the elements in the subset. This methodology is especially suited for recognition systems with large numbers of classes where it is valuable to reduce the decision problem to a manageable size before making a final determination about the identity of the image. Experimentation is discussed in which the proposed method is used with a word recognition problem where 41 classifiers are applied to degraded machine-printed word images and where a typical lexicon contains 500 words. A 92 percent correct rate is achieved within the 10 best decisions for 1710 test images.

1. Introduction

In practical feature-based multi-class pattern recognition problems, one often finds that a large number of features can contribute to recognition. In many existing systems, feature values are normalized and represented as a feature vector. A classifier decides class identity by evaluating discriminant functions or distance functions defined for these feature representations.

Parametric statistical classifiers using discriminant functions assume a certain underlying distribution model for the feature values of each class. These models are often idealistic and the estimation of parameters requires a large number of labeled sample patterns. The design of the classifier is generally difficult when the features are not statistically independent. For nonparametric statistical classifiers using distance functions, performance depends on the effectiveness of the feature descriptor and the distance function.

Useful features can be very different in nature, covering a range of strengths in measurement. Variables that describe the features could be nominal, ordinal, interval-scaled or ratio-scaled. For instance, we can be interested in the existence and location of a perceptual entity like an edge, and also a lower level feature like a count of all the black pixels in the image.

Though some descriptor can always be developed to represent such a collection of features, it is nontrivial to define a meaningful distance function. Variables of different types remain

[1]This work is supported by the Office of Advanced Technology of the United States Postal Service under Task Order No. 104230-88D-2576 and Contract No. 104230-84D-0962

invariant under different transformations, and can be sensitive to different patterns of degradation. Sometimes it is impossible to define a single distance function that is appropriate for all types of features that may contribute to recognition [16].

Recently, it has been observed that a multiple classifier system can be advantageous in pattern recognition problems involving diverse features [9], [10], [17]. A multiple classifier system has the following two advantages:

(i) different distance functions (metric or nonmetric) can be defined for different subsets of features, such that the evaluation operations will be appropriate for all the feature types, and

(ii) features sensitive to different degradation patterns can be separated, and *dynamically selected* when estimates on the degradation patterns are available.

Classifiers that use a single feature or a small set of features may result in very unstable performance, depending on the discrimination power of the feature set, accuracy of the detection algorithm, and the type of degradation in the image. As a result, each classifier may perform well on only a limited type of data. Figures 1(a) to (c) show images with some essential shape features present or missing. Classifiers using those features are expected to have unstable performance.

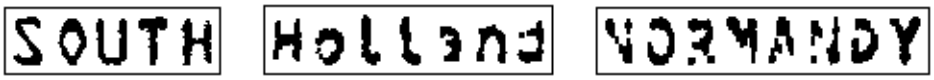


Figure 1: Word images illustrating presence/absence of some features: (a) complete hole and strokes, (b) broken holes and complete strokes, and (c) broken holes and strokes.

A critical question for this approach is whether one can combine the decisions of the individual classifiers, and obtain one classifier with performance better than any one working in isolation. In this paper, we demonstrate that this is possible, and propose a method that can be used to combine the decisions of individual classifiers to obtain a classification procedure which performs better than any of the individual classifiers.

2. The Intersection Approach and the Union Approach

In a multiple classifier system, each classifier produces a ranking of a set of allowable classes, by evaluating a measure of distance between the feature descriptors obtained from the image and those stored as prototypes. We will call the classes ranked near the top as the *neighborhood* of the true class for the image determined by a classifier. The objective of recognition is to produce a set of candidates that contains the possible true class. Perfect recognition will be achieved when the true class is the only output candidate.

To combine a set of rankings, two opposite approaches are possible. In the first approach, a large neighborhood is obtained from each classifier, and their intersection is then output as a set of candidates. Hence a class will be a output candidate if and only if it is in *all* the neighborhoods. In other words, a decision is made if it is confirmed by all classifiers. In the second approach, a small neighborhoods is obtained from each classifier. The *union* of these neighborhoods is output as a set of candidates. A class is output as a candidate if and only if it is in *at least* one of the neighborhoods.

Obviously the intersection approach is appropriate only when all the classifiers have moderate worst case performance, such that thresholds on neighborhood sizes can be selected in a way that the set of output candidates can be small, while the true class will not be missed. However, for a set of specialized classifiers working on small subsets of features, this is usually not achievable, since each classifier may perform poorly for certain types of inputs. We propose that *the union approach is preferred for combining a set of highly specialized classifiers*, because the strength of any individual classifier can be preserved, while its weakness can be compensated for by other classifiers. The following discussions in this paper are mostly on the union approach. We will first show how the worst case performance can be improved by using our method to determine the sizes of the neighborhoods, and then how the set of output candidates can be ranked by combining the rankings from the individual classifiers.

3. Related Studies

One way to organize a set of feature-based classifiers is as a decision tree. Decision trees have a sequential and hierarchical organization. They are well-studied; a survey of decision tree methods is given in [13]. The topology of the tree is determined according to an optimization criterion, like minimizing the probability of misclassification and expected test cost.

The application of committee methods to two-class problems and an algorithm for the construction of a minimal committee is discussed in [12], where the final decision is made by majority vote. In a problem with hundreds of classes and many classifiers, a rule is needed which will determine a consensus ranking from the rankings given by a group of individual decision makers. Such a rule is often referred to as a *group consensus function* or a *social welfare function* [14]. Conditions on such functions and possibility theorems on various assumptions have been discussed in [1] and [8]. The logic of committee decisions and of elections, as well as a history of the mathematical theory of committees and elections are presented in [3]. A formal treatment of social choice functions, with much emphasis on the study of majority functions is presented in [7]. A theory of distances between partial orders is given in [4].

4. The Recognition System

The proposed algorithm for classifier combination anticipates a recognition system like the one shown in Figure 2.

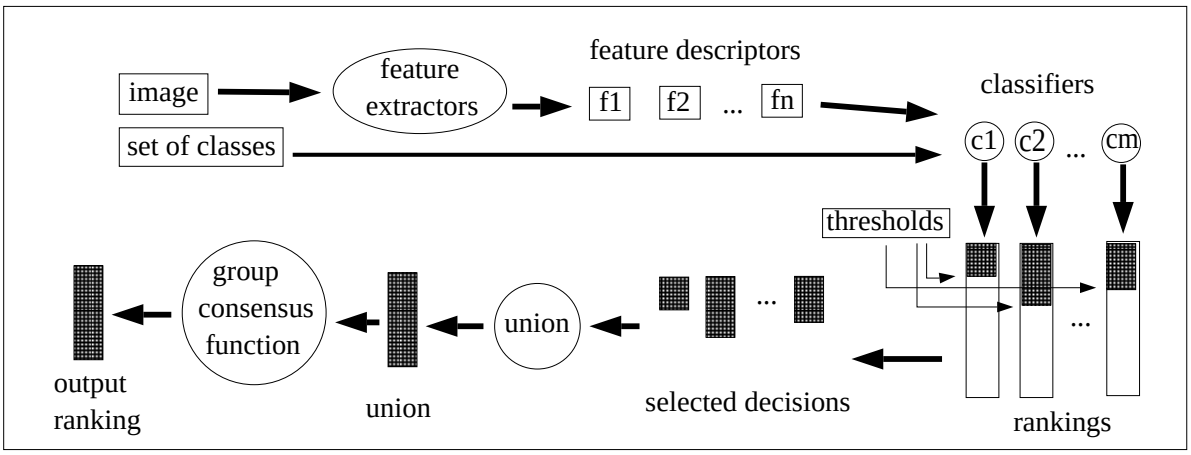


Figure 2: A Multiple-Classifier Recognition System.

A set of classifiers are applied to an input image. Each classifier produces a ranking of a set of allowable classes. A small number of classes are chosen from the top of each ranking by a set of thresholds. The union of these subsets of the set of classes is ranked by a group consensus function, and then output by the recognition system.

The proposed algorithm chooses the set of thresholds to minimize the size of the union and maximize the probability that it contains the class of the input image. We are concerned with cases where there are a large number of classes (500 or more) [11].

To choose the thresholds, the classifiers are first applied to a set of training data. A performance table for the training set like the one shown in Table 1 can then be obtained. The left half of the table shows the rankings of the *true class* in each image by each of the classifiers. We perform a transformation that takes the minimum ranking in each row and produces the right half of the table. The rankings can be partial orders, but, for convenience, we convert all of them into total orders by resolving ties using an arbitrary order in the set of classes. This data structure is the basis of the proposed technique for threshold selection.

5. Proposed Algorithm for Threshold Selection

This section presents a formal description of the algorithm for threshold selection in a multi-classifier recognition system.

Table 1: Example Performance Table for a Training Set

	$rank^i_j$				$rowmin^i_j$			
$I_i \backslash C_j$	C_1	C_2	C_3	C_4	C_1	C_2	C_3	C_4
I_1	3	12	1	24	0	0	1	0
I_2	1	5	29	12	1	0	0	0
I_3	34	3	4	6	0	3	0	0
I_4	9	7	6	7	0	0	6	0
I_5	4	36	5	5	4	0	0	0
I_6	16	2	3	4	0	2	0	0
$colmax_j$					4	3	6	0

Let $rank^i{}_j$ be the ranking of the *true class* in image i by classifier j, where $i = 1, 2, ..., n$ and $j = 1, 2, ..., m$.

Define
$$rowmin^i{}_j = \begin{cases} rank^i{}_j & \text{if } \forall l, 1 \le l \le m, rank^i{}_j \le rank^i{}_l \\ 0 & \text{otherwise} \end{cases}$$
$$colmax_j = Max_i\ (rowmin^i{}_j)$$

we can then claim that $\forall i, \exists j$ such that $rank^i{}_j \le colmax_j$.

Proof: (by contradiction) otherwise, $\exists k$, we have $\forall j \quad rank^k{}_j > colmax_j$.
Then $\exists l, \quad rowmin^k{}_l = Min_j\ (rank^k{}_j) > colmax_j \qquad (1)$,
and $colmax_j \overset{(\text{def})}{\ge} rowmin^k{}_l \overset{(1)}{>} colmax_j$ which is a contradiction.

This claim says that for any image I_i *in the training set*, if we take top $colmax_j$ decisions from classifier C_j for each j, and take the *union* of all these decisions, then the true class for image I_i *must be* contained in this union. The $colmax_j$ values are the thresholds determined by this method. The thresholds can then be used to obtain unions for other test sets.

Intuitively, $colmax_j$ for classifier j is the *worst case decision* by classifier j, *when no other classifiers can do better than it.* Consider an ideal system such that we know, by observing the performance on a training set, that for *any* input image, there exists a classifier which will rank its true class at the top. In this case all the $colmax_j$'s equal to one, that is, we only need to consider the top decisions from all the classifiers. Consider another system that there is only one classifier. In this case $colmax_1$ is the worst case rank given by that classifier.

6. Observations

For any j, if $colmax_j = 0$, then classifier j is redundant, in the sense that its decision is always inferior to some other classifiers. Classifier C_4 in Table 1 is redundant in this sense.

Let $\{C_{j_1}, C_{j_2}, ..., C_{j_k}\}$ be any subset of the original classifier set, and let the *maximum union size* be defined as

$$MUS = \sum_{j=j_1}^{j_k} colmax_j$$

For the example given in Table 1, $MUS = 4 + 3 + 6 = 13$. That means all the true classes for the images in this training set will be included in a union of size *at most* 13, which is smaller than the number of decisions needed from any individual classifier for the same correct rate.

Yet one may notice that for the example in Table 1, if we use only the classifier subset $\{C_1, C_2\}$, or $\{C_2, C_3\}$, and obtain the thresholds $colmax_j$'s following the aforementioned procedure, we will need a union of even smaller size (11 in both cases). In fact, the smallest union size (10) can be obtained if we use the subset $\{C_1, C_3\}$. This observation leads to a method for classifier subset selection.

7. Classifier Subset Selection

A method for classifier subset selection can be derived based on the previous observations.

If MUS is minimized over all possible subsets, then $\{C_{j_1}, C_{j_2}, ..., C_{j_k}\}$ is the theoretical minimum-union-size (optimal worst case performance) classifier subset for the training set.

A direct implementation of this selection technique would require evaluation of MUS over the power set of the classifiers and thus is not a polynomial algorithm. The complexity may preclude its application to larger classifier sets. One simplification is to omit the subset selection procedure, and use all the classifiers except the redundant ones. Another simplification is to approximate the theoretical optimum by a greedy algorithm, which aims at stepwise reduction of MUS by adding new classifiers.

Whereas MUS is the theoretical maximum size of the union, the true maximum or average union size can only be determined experimentally by actually obtaining all the decisions and forming the unions. Hence the minimum MUS does not necessarily give an optimal experimental union size. But this is a useful measure in the absence of more knowledge of the mutual dependence among the classifiers.

The performance of this method on test sets will depend on how well the training set represents the population. This includes both the quality of the images and the distribution of the classes in the training images.

8. Dynamic Feature Selection

Consider an *oracle* that will always select the best classifier C_j for each image I_i. If such an oracle is available, we can take the top $colmax_j$ decisions from classifier C_j only, and ignore the decisions by other classifiers. In this case MUS is the maximum $colmax_j$, which could be much smaller than the MUS given by the union.

Such an oracle will perform what we may call *dynamic feature selection*. It selects the best feature set for recognition for each image. The selection can be based on confidence of feature detection, or possible association of the classifier performance with other characteristics of the image like some global features, or some estimation of the degradation pattern in the image. Dynamic feature/classifier selection can be done on a set of features/classifiers that are *statically* determined to be useful when all possible cases are taken into consideration.

9. Combination of Rankings

The classes contained in the union can be further ranked by combining the rankings they received from the individual classifiers. One useful combination scheme is referred to as the *Borda Count* [14]. In the original definition, the rankings given by all the classifiers are considered. For any particular class c, the Borda count is the sum of the number of classes ranked below c by each classifier.

Since we are taking the union of only a small subset of decisions from each classifier, we have added $colmax_j$ to the original definition such that we count the number of classes below c *in the neighborhood only*. Our definition is given as follows:

For any class c in the union, let $B_j(c)$ be the number of classes ranked below the class c but above $colmax_j$ by classifier C_j. $B_j(c)$ is zero if the ranking of c by C_j is below $colmax_j$. The Borda count for class c is $B(c) = \sum_{j=1}^{m} B_j(c)$. The final ranking is given by arranging the classes in the union so that their Borda counts are in descending order.

This count, as in our definition, is dependent on the thresholds ($colmax_j$'s) as well as the agreement among the classifiers. Intuitively, if the class c is ranked near the top by more classifiers, it has better chance to be ranked above $colmax_j$ by more C_j's, and there are more classes below it after we apply the $colmax_j$ thresholds, hence its Borda count tends to be larger.

10. Example Application

The approach is illustrated with examples from a word recognition system. The input to the system includes an image of a machine-printed word, and a given lexicon which contains the word. We are developing a (wholistic) word shape based recognition system *without* applying

character segmentation. It is desired that a ranking of the lexicon be produced such that the target word should be as close to the top as possible. The rankings can then be combined with the decisions from an isolated-character-based recognizer.

A set of global and local features are identified and computed for each input image. Global features describe some overall characteristics of a word, and local features describe certain locally detectable characteristics.

The global features used include the case of the word, which says whether the word is in purely upper, purely lower or mixed case, and a word length estimate, which gives the number of characters in the word with some error allowance. In our application, purely lower case words are not considered. The shape of the same word is different when printed in upper or mixed case, and hence separate feature prototypes are needed for the different cases. Accurate estimate of the case helps to reduce the number of dictionary prototypes to be compared by half. Estimates of word length further reduce the size of the dictionary. However, the amount of the reduction depends on the accuracy of the estimate.

The local features that are used include geometrical characteristics such as endpoints, edges, vertical and horizontal strokes, diagonal strokes, curves, distribution of strokes, as well as topological features like dots, holes, and bridges between strokes [9]. Table 2 summarizes the detection methods for the features used in our design.

Table 2: Summary of Structural Features Used in Word Recognition

Global Features	*Extraction Method*
Case	Analysis on separation between reference lines, horizontal projection profile and connected components
Word length	Analysis on vertical projection profile and connected components
Local Features	*Extraction Method*
Stroke distribution	All direction run length analysis
Stroke edges	Run length analysis
End points of skeleton	Convolution
Holes and dots	Connected component analysis
Vertical and horizontal strokes	Run length analysis
Curves and diagonal strokes	Chain code analysis
Bridges between strokes	Connected component analysis

The recognition algorithm first filters the lexicon according to the computed global features (case and word length estimate). Then local features are extracted. A fixed area partition grid of 4×10 is used to describe the location of the local features. A set of 35 different descriptors are used to represent the local features and their locations. They are in different forms including a feature vector, 5 symbol strings for 5 different subsets of the features, as well as 29 digit strings, one for each feature. Each descriptor is compared to the counterpart in the prototypes by a distance function. Six composite distance functions are also used, two of which are identical to two independent distance functions in the set of 35. We introduced them to examine the effect of correlations between the classifiers. A nearest-neighbor classifier is implemented using each of these distances. This results in 41 classifiers in total, each produces a ranking of the filtered lexicon. The rankings are then

combined using our proposed method. Table 3 gives a summary of these descriptors and the corresponding distance functions. Table 4 gives a summary of the features used by the 41 classifiers.

Table 3: Summary of Feature Descriptors and Corresponding Distance Functions

Features	*Descriptor*	*Example*	*Distance Function*
stroke distribution	160 dimensional vector	[10 26 0 0 ....]	Euclidean distance
relative location of edges end points, ascenders, descenders, holes, dots, curves etc.	symbol strings, one for each feature subset. each symbol represents a specific feature	$AOOAD$	minimum edit distance, with edit costs derived from reliability of detection, and frequency of substitutions
horizontal position of edges, end points, ascenders, descenders, holes, dots, curves etc.	digit strings, one for each feature. digits are positions w.r.t. the 10 width partitions	2334567	minimum edit distance, with edit costs derived from reliability of detection, and differences of digit values

Table 4: Summary of Features Used by the Classifiers

Classifier No.	*Features / Distances*
1	stroke distribution vector
2	symbol string for edge features
3	sum of symbol string distance for edge features (identical to 2)
4-11	location strings for edge features
12	sum of distances in 4-11
13	symbol string for endpoint features
14	sum of symbol string distance for endpoint features (identical to 13)
15-19	location strings for edge features
20	sum of distances in 15-19
21-23	symbol strings for letter shape features
24	sum of symbol string distances for letter shape features
25-40	location strings for letter shape features
41	sum of distances for 25-40

11. Experimental Results

The recognition system has been developed using a collection of images of *machine-printed* postal words obtained from live mail. They were scanned at roughly 200 ppi and binarized. The font and quality of the images vary. We used a measure of density (number of black pixels divided by image area) to assess the image quality. Figure 3 shows some example images with low, medium and high density levels. In the database we have roughly the same number of images in each of these three categories.

The available images were divided into separate training sets and testing sets. The feature extractors were developed and modified by observing performance in a small initial training set of about 200 images. The distance functions (edit costs) were derived by running the

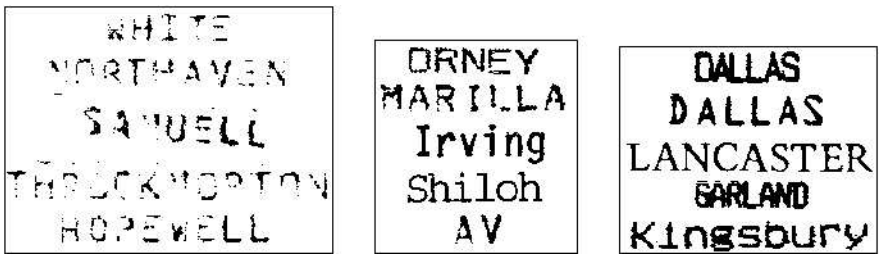


Figure 3: Example images with low, medium and high density levels.

feature extractors on a separate set of 2417 images. Feature prototypes for characters were obtained by running the feature extractors on isolated characters from 10 font samples. Feature prototypes for the words in the lexicon were obtained by synthesizing feature descriptors computed for the characters.

Another set of 481 training images, different from those we used in developing the feature extractors and distance functions, was used to obtain the neighborhood thresholds. The input lexicon contained 500 words from a city name database. The same lexicon was used for all the images. We allowed a rejection option in case assignment (which duplicated the size of the lexicon for the reject cases), and flexible ranges for word length estimate so that for 465 images (96.67% of total), the true word remained in the filtered lexicon, i.e. the computed global features were correct. The average size of the *filtered* lexicon was 172.92 (words). The 41 classifiers were applied to all the 481 images. To derive the neighborhood thresholds, we constructed the ranking table for only those words with correctly computed global features. Since the lexicons were different for different images after they were filtered by global features, we converted the rankings into percentages of the respective filtered lexicon. The thresholds computed were also in the unit of percentage of filtered lexicon size. Table 5 gives the performance of the 41 classifiers on these 465 images, as well as the computed neighborhood thresholds[2].

The same feature extractors and classifiers were then applied to another set of 1710 test images, different from all the images used in various training stages. The same input lexicon of 500 words was used. After the classifiers produced the rankings, the neighborhood thresholds were applied to select a number of top decisions from each classifier. The union of these selected decisions was formed and re-ranked using the Borda count. For 1638 images (95.79% of total) in this set, the global features were correctly computed. The average size of the filtered lexicons was 166.31 words. Table 6 gives the performance of the 41 classifiers on these 1638 images [3], The last column shows the performance of the combined decision.

As can be seen from the performance table, some of the 41 classifiers, when working in isolation, are very poor in (macroscopic) performance over the whole test set. However, the

[2]Classifiers 2 and 3, 13 and 14 are identical.

[3]We ran the 41 classifiers as well as the neighborhood combiner on all the 1710 images, without checking the correctness of the global features. However, to examine the performance on the local feature matching stage only, we compiled the performance table using 1638 as total.

Table 5: Summary of Performance on Training Set and Neighborhood Thresholds

Classifier No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
% correct at top choice	73	62	62	1	0.4	20	1	3	19	10	23	56	27	27
% correct in top 2 choices	82	71	71	7	1	28	4	6	28	32	30	67	35	35
% correct in top 3 choices	85	75	75	15	2	37	4	8	42	40	38	70	39	39
% correct in top 5 choices	89	81	81	17	2	45	8	14	49	46	43	75	48	48
% correct in top 10 choices	92	85	85	21	5	54	24	22	60	57	54	82	55	55
thresholds (in % lexicon size)	17	8	8	2	3	8	6	3	17	2	17	8	8	8
Classifier No.	15	16	17	18	19	20	21	22	23	24	25	26	27	28
% correct at top choice	2	0.4	2	18	23	40	11	5	5	23	20	0.4	28	2
% correct in top 2 choices	5	1	3	26	29	50	27	9	8	30	26	1	44	4
% correct in top 3 choices	13	2	4	29	36	55	28	14	11	36	34	2	49	5
% correct in top 5 choices	20	2	8	40	42	61	35	18	17	43	40	2	53	9
% correct in top 10 choices	25	6	12	51	52	68	45	27	27	52	46	4	57	14
thresholds (in % lexicon size)	3	3	17	8	8	8	8	3	8	8	8	1	3	3
Classifier No.	29	30	31	32	33	34	35	36	37	38	39	40	41	
% correct at top choice	0.2	1	0.2	1	2	5	2	1	3	3	3	5	28	
% correct at top 2 choices	1	2	1	6	5	6	4	4	6	7	6	6	37	
% correct at top 3 choices	2	3	2	8	6	7	9	6	7	8	9	9	42	
% correct at top 5 choices	3	5	3	13	8	8	13	9	10	9	10	12	50	
% correct at top 10 choices	5	9	5	22	15	12	25	17	17	16	16	25	61	
thresholds (in % lexicon size)	1	3	1	1	2	1	1	2	4	1	1	8	8	

Table 6: Summary of Performance on Test Set

Classifier No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
% correct at top choice	76	64	64	2	0.1	20	1	5	22	12	21	54	22	22
% correct in top 2 choices	84	74	74	9	1	28	3	6	29	32	28	64	29	29
% correct in top 3 choices	87	77	77	16	3	37	4	11	38	41	35	69	34	34
% correct in top 5 choices	90	81	81	18	3	43	8	17	46	48	43	74	42	42
% correct in top 10 choices	93	85	85	23	4	52	23	21	57	57	53	81	52	52
Classifier No.	15	16	17	18	19	20	21	22	23	24	25	26	27	28
% correct at top choice	5	0.2	1	18	20	32	13	6	6	21	18	0.4	31	2
% correct in top 2 choices	8	1	2	25	25	43	26	10	8	28	25	1	46	7
% correct in top 3 choices	19	3	3	29	31	49	28	13	10	33	30	3	51	9
% correct in top 5 choices	25	4	6	35	35	56	32	18	17	38	36	5	55	13
% correct in top 10 choices	29	5	9	45	46	67	40	27	28	48	42	6	60	18
Classifier No.	29	30	31	32	33	34	35	36	37	38	39	40	41	union
% correct at top choice	0.1	1	0.1	2	2	3	3	3	4	1	3	5	26	81
% correct at top 2 choices	1	2	1	6	4	6	5	7	5	5	7	6	35	88
% correct at top 3 choices	3	4	3	8	6	7	8	8	7	7	10	8	40	90
% correct at top 5 choices	4	6	3	13	9	8	14	11	9	10	12	12	47	92
% correct at top 10 choices	5	9	4	20	13	11	23	20	14	15	18	23	58	94

union combination works better than any of the individual classifiers.

This experiment shows only how good the combination performance is when compared to the individual classifiers. None of the computed thresholds is zero, meaning that none of the 41 classifiers is redundant (always inferior to another one). However, we did not perform a combinatorial subset selection procedure due to the complexity involved. Therefore it is possible that the combination of certain subsets of these 41 classifiers performs better than what has been achieved.

We tried using only those classifiers with correct rates larger than ten percent at the top choice (classifiers 1,2,6,9,10,11,12,13,18,19,20,21,24,25,27, and 41). Moreover, we removed from the training set 6 images with very small filtered lexicons (less than 20). Since we converted all the rankings into percentages of filtered lexicons in computing the neighborhood thresholds, for these images, even very good rankings will appear as very large ratios (e.g.

2/12 = 17%, 1/12 = 8%). With these 16 classifiers and thresholds computed in this way, the performance on the test set was 83%,90%,92%,93%,and 96% at the top and within the top 2,3,5,10 choices respectively.

We then carried out another experiment, where we performed a two-stage combination using the same algorithm twice. In particular, as we observed that classifier number 1 (stroke distribution vector matched with Euclidean distance) performed far better than the others, we decided to get the union of the other 40 classifiers first, and then combine this union with the result of classifier number 1 by another union operation. We used the same max-min operations to determine the thresholds for the 40 classifiers from the ranking table for the training set first. Then we reranked the union of the 40 classifiers for the *training set* using these thresholds. The rankings were saved to be listed together with those from classifier number 1, and a pair of thresholds were then derived (using the same max-min procedure). Later we applied the two-stage union procedure with these two sets of thresholds on the *test set*. The final rankings give the correct class at the top and within the top 2,3,5,10 choices for 82%, 89%, 91%, 92% and 94% of the 1638 images respectively, a slight improvement over the one-stage union of all 41 classifiers, but not as good as the 16-classifier subset performance. However, when we attempted a three-stage union by grouping the classifiers heuristically, no significant improvements were obtained.

Table 7 summarizes the performance of the several combination methods we attempted. Taking the correct rate of the global features into account, we have achieved a correct rate of 92% (96 × 95.79) within the top 10 choices, with a subset of the classifiers. The last column in Table 7 shows the performance with the assumption that we can select the best classifier for each image.

Table 7: Summary of Performance using Different Combination Strategies

Combination Method	*one-stage union of 41 classifiers*	*two-stage union of 41 classifiers*	*one-stage union of 16 classifiers*	*performance with an oracle*
% correct at top choice	81	82	83	92
% correct in top 2 choices	88	89	90	96
% correct in top 3 choices	90	91	92	97
% correct in top 5 choices	92	92	93	98
% correct in top 10 choices	94	94	96	99

12. Conclusions and Future Work

A technique was presented for combining the rankings of a large set of classes by the classifiers in a multi-classifier pattern recognition system. This method utilizes the results of applying the classifiers to training data to determine a set of thresholds on the rankings. The ranked subsets obtained by applying the thresholds are then combined with a union operation and re-ranked by a group consensus function. This technique discovers those classifiers that are redundant and removes them from the recognition system. A further methodology was discussed for selecting a subset of the non-redundant classifiers that minimizes the union size.

The proposed technique was applied to a word recognition problem that used 41 classifiers and involved 500 classes. The data were images of machine printed words that were scanned on a postal OCR at 200 pixels per inch. In many cases the image quality was very poor. Applying the strategy to a testing set resulted in a 92% correct rate within the top 10 choices.

Future work includes the investigation of an oracle that selects the best feature sets dynamically and eliminates the ambiguity introduced by less appropriate classifiers. This is possible if some of the feature subsets are selectively ignored, based on an estimation of the degradation in the image, or the confidence of feature detection.

The method we used to combine rankings is a preliminary one among many other possible methods. It is also believed that some descriptive statistics, such as measurement of correlation among the classifiers, should be taken into account in combining the rankings. Follow up work will include studies of the possible applications of other group consensus functions to classifier ranking combination, and consideration of correlation measures for the classifiers.

References

[1] Kenneth J. Arrow, *Social Choice and Individual Values*, Cowles Commission Monograph 12, John Wiley & Sons, Inc., New York, 1951, 2nd ed., 1963.

[2] Kenneth J. Arrow, Herve Raynaud, *Social Choice and Multicriterion decision-making*, MIT Press, Cambridge, Mass., 1986.

[3] Duncan Black, *The Theory of Committees and Elections*, Cambridge University Press, London, 1958.

[4] Kenneth P. Bogart, Preference Structures I: Distances between Transitive Preference Relations, *Journal of Mathematical Sociology*, **3**, 49-67, 1973.

[5] Wade D. Cook, Information and Preference in Partial Orders: A bimatrix representation. *Psychometrika*, **51**, 197-207, 1986.

[6] R.O. Duda, P.E. Hart, *Pattern classification and scene analysis*, Addison-Wesley, New York, 1973.

[7] Peter C. Fishburn, *The Theory of Social Choice*, Princeton University Press, Princeton, 1972.

[8] Leo A. Goodman, Harry Markowitz, Social Welfare Functions Based on Individual Rankings, *The American Journal of Sociology*, **58**, 257-262, 1952.

[9] Tin Kam Ho, Jonathan J. Hull, Sargur N. Srihari, Recognition of Degraded Text Images, *Computer Science Research Review*, Department of Computer Science, SUNY at Buffalo, 1989.

[10] Jonathan J. Hull, Alan Commike, Tin Kam Ho, Multiple Algorithms for Handwritten Character Recognition, *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, 117-124, Montreal, April 2-3, 1990.

[11] Jonathan J. Hull, Feature Selection and Language Syntax in Text Recognition, in *From Pixels to Features* , J.C. Simon (editor), North Holland, 249-260, 1989.

[12] V.D. Mazurov, A.I. Krivonogov, V.L. Kazantsev, Solving of optimization and identification problems by the committee methods, *Pattern Recognition*, **20**, 4, 371-378, 1987.

[13] Bernard M. E. Moret, Decision Trees and Diagrams, *Computing Surveys*, **14**, 593-623, December 1982.

[14] Fred S. Roberts, *Discrete Mathematical Models, with applications to social, biological, and environmental problems*, Prentice-Hall, Inc., 1976.

[15] Marc Roubens, Phillippe Vincke, *Preference Modelling*, Springer-Verlag, 1985.

[16] S. S. Stevens, Measurement, Statistics, and the Schemapiric View, *Science*, **161**, 3844, 849-856.

[17] C. Y. Suen, C. Nadal, T. A. Mai, R. Legault and L. Lam, Recognition of Totally Unconstrained Handwritten Numerals Based on the Concept of Multiple Experts, *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, 131-140, Montreal, April 2-3, 1990.

[18] Robert A. Wagner, Michael J. Fischer, The String-to-String Correction Problem, *Journal of the ACM*, **21**, 1, 168-173, January 1974.