Combination of Decisions by Multiple Classifiers

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

A technique for combining the results of classifier decisions in a multi-classifier 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 40 classifiers are applied to degraded machine-printed word images and where a typical lexicon contains 235 words. A 96.6 percent correct rate is achieved within the 10 best decisions for 817 test images.

1. Introduction

In practical pattern recognition problems, a large number of features can often contribute to recognition. It is common to represent all such measurements in a single descriptor, such as a feature vector. A classifier decides the class identity by evaluating a discriminant function or distance function defined for such a descriptor [7].

An effective single classifier over a large set of features may be difficult to design because of the following reasons:

1. Different *classifier designs* are appropriate for different feature subsets. The alternatives include syntactic or various statistical classifier designs such as nearest-neighbor or Bayesian classifiers. Each classifier can itself contain valuable knowledge that is difficult to represent in a feature descriptor.

2. Different *types of feature measurements*. Parameter values that describe features may be measured on a nominal, ordinal, interval or ratio scale. The physical meaning of the parameters can be so different that they cannot be easily normalized to one single scale. For instance, we can be interested in higher level features such as the location of a perceptual entity (e.g. an edge, an end point etc.), and also a lower level feature like a count of all the black pixels in the image. Though a descriptor can always be developed to represent such a collection of features, it is nontrivial to define a meaningful distance function for such a descriptor. Arbitrary combination of such a mixture of scales may result in meaningless operations [22].

3. Different *recognition approaches* may be appropriate for the same recognition problem, and there may be no meaningful distance function defined on the feature values computed by all the approaches. Each method may contribute information not given by another one. For instance, one approach may recognize the object as a whole, whereas another approach may recognize the components of the object and then derive consistent decisions.

Moreover, variables of different types can be sensitive to different input conditions, such as various patterns of image degradation, and the characteristics of the classes if a variable input class set is given. It is not easy to optimize the performance of a single classifier using a large collection of feature variables under all circumstances.

Recently, it has been observed that a multiple classifier system can be advantageous in pattern recognition problems involving diverse features ([11], [13], [14], [19], [23]). A multiple classifier system applies a number of independent classifiers in parallel to each image and combines their results to generate a single decision. A multiple classifier system overcomes some of the disadvantages of a single classifier approach listed earlier in the following ways:

1. Different similarity measures and classification procedures can be defined for various feature subsets such that the information contained in each subset is best utilized.

2. Feature measurements on different scales can be matched separately in their corresponding scales, such that the evaluation operations will be appropriate for all the feature types.

3. Different approaches for recognition can be applied by using different classifier designs that are appropriate for the individual methods.

Moreover, a multiple classifier system has an additional advantage that

4. Features sensitive to different input conditions can be separated and *dynamically selected* when knowledge of the input conditions is available. Post classification selection is also possible if a confidence measure can be associated with the classification decisions.

A critical question for a multiple classifier 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 the individual classifiers.

We assume that the decisions of each classifier are given as rank orders of a given set of classes. The proposed combination methods are primarily based on these rank orders. In levels of strength in measurement, rank orders are weaker than the distance measures used by the individual classifiers but stronger than categorical decisions (either accepted to

be a certain class or rejected, such as the decisions produced by many syntactic classifiers) [15][24]. By using rank orders, we avoid the problem of combining a mixture of scales used by different classifiers. Nevertheless, more information can be utilized than the methods that rely on only categorical decisions. In fact, it has been well known in the theory of multidimensional scaling, that rank orders provide sufficient information to recover most of the distance relationship of a set of objects [6] [21]. Categorical decisions can be regarded as degenerate rank orders.

The decision combination is performed in two stages. A candidate set of classes is first derived. A group consensus function is then applied to re-rank the classes in the candidate set by combining the rankings assigned by the classifiers.

2. The Intersection Approach and the Union Approach

Assume that 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. The objective of recognition is to produce a neighborhood that contains the true class. Perfect recognition will be achieved when the true class is the only member of the neighborhood.

To produce a candidate set from 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 an 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 neighborhood 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 candidate set can be small, while the true class is not 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.

Besides these two approaches, an intermediate approach is also possible that relies on a voting scheme to determine which class should be included in the candidate set, though the design of the voting scheme is more difficult than either the intersection or the union approach. Practical requirements such as the maximum error rate allowed need to be considered.

The following discussions in this paper are mostly based 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 describe how the set of output candidates can be re-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 [18]. The topology of the tree is determined according to an optimization criterion, like minimizing the probability of misclassification and expected test cost.

Haralick [10] gives the error bounds when the intersection approach is applied to combine results from multiple Bayesian classifiers. An example problem is given where the use of multiple classifiers is motivated by the need to use multiple resolutions in various feature dimensions. Selection of different feature subsets to discriminate certain classes from others is also discussed.

Mandler and Schuermann [16] describe a method for combining independent classifier decisions based on the Dempster-Shafer theory of evidence. This method involves a series of transformations of computed pattern distances into confidence values, which are then combined by the rules of evidence theory.

The application of committee methods to two-class problems and an algorithm for the construction of a minimal committee is discussed in [17], where the final decision is made by majority vote. This method cannot be easily generalized to a many-class problem.

Hull et al. in [13] and [14] demonstrate the usefulness of multiple algorithms for character recognition, with the decisions combined by a decision tree method and an earlier version of the approach described in this paper. Suen et al. in [23] propose the use of several experts for handwritten character recognition, with the decisions combined by majority vote. The voting makes use of of a single decision from each classifier. In case that a classifier does not discriminate among a set of decisions, its vote is split evenly among this decision set. Nadal et al. in [19] describe several heuristic rules for combining decisions by a similar set of classifiers.

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 referred to as a *group consensus function* or a *social welfare function* [20]. Conditions on such functions and possibility theorems on various assumptions have been discussed in [1],[2] and [9]. The logic of committee decisions and of elections, as well as a history of mathematical theory of committees and elections are presented in [4]. A formal analysis of social choice functions, with much emphasis on the study of majority functions is presented in [8]. A theory of distances between partial orders is given in [5].

4. The Recognition System

The proposed algorithm for classifier combination anticipates a recognition system like the one shown in Figure 1.

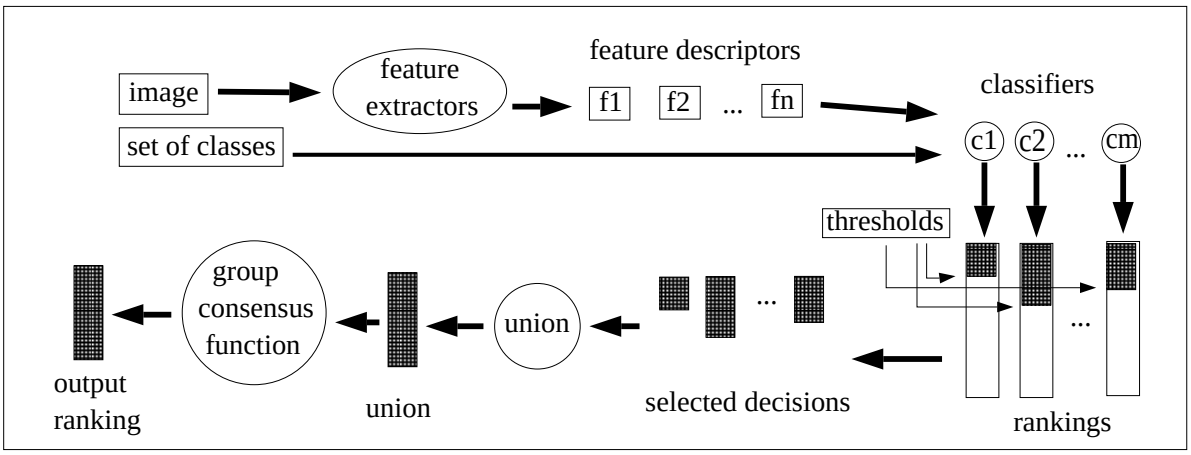


Figure 1: A Multiple-Classifier Recognition System.

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) [12].

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 $colmax_j$ 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 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.

One way to approximate such an oracle is to compute some mutually exclusive conditions from the images. A training set can be partitioned according to the computed conditions. Classifier performance can be measured separately on each partition. The best classifier for each partition can hence be determined. For the test set, similar conditions can be computed and the best classifier will be selected.

Note that this kind of selection is an intermediate method between two extremes, one being a static single classifier method and the other is with one classifier for each class that responds well for patterns of only that class. In the latter case selection of the best classifier is equivalent to the original classification problem in difficulty.

9. Combination of Rankings

The candidate classes contained in the set derived by either the union or the intersection approach 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* [20]. 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. If a candidate subset is selected from the set of allowable classes, in computing this count, we will consider the classes included in this subset only. Our definition is given as follows:

For any class c in the candidate subset S, let $B_j(c)$ be the number of classes in S which are ranked below the class c by classifier C_j. $B_j(c)$ is zero if $c \notin S$. 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 is dependent on the agreement among the classifiers. Intuitively, if the class c is ranked near the top by more classifiers, 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 holistic 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 is a range of possible numbers of characters in the word. 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. Exact estimate of the case reduces the number of word prototypes to be compared by half. Estimates of word length further reduce the size of the lexicon. 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, some topological features like dots, holes, and bridges between strokes [11], as well as a set of template defined features described in [3]. 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*
Template defined features	Convolution
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). Local features are then extracted. A fixed area partition grid of 4×10 is used to describe the location of the local features. A set of 36

different descriptors are used to represent the local features and their locations. They are in different forms including two feature vectors, 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, which is either a city-block distance [7] or a string edit distance [25]. Four composite distance functions are also used. A nearest-neighbor classifier is implemented using each of these distances. This results in 40 classifiers in total, each produces a ranking of the filtered lexicon. The level of resolution of the rankings varies among the different classifiers. Classifiers 1 and 2 use a refined distance measure so that the rankings are mostly total orders. The other classifiers that use string edit distances produce partial orders most of the time. The partial orders are converted into total orders by the alphabetical order of the words. The rankings are then combined using our proposed union approach and re-ranking 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 40 classifiers.

Table 3: Summary of Feature Descriptors and Corresponding Distance Functions

Features	*Descriptor*	*Example*	*Distance Function*
template features	1280 dimensional vector	[5 0 4 2 ....]	City-block distance
stroke distribution	160 dimensional vector	[10 26 0 0 ....]	City-block 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
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, where edit costs are differences of digit values

Table 4: Summary of Features Used by the Classifiers

Classifier No.	*Features*	*Distances*
1	convolution generated feature vector	city-block distance
2	stroke distribution vector	city-block distance
3	symbol string for edge features	string edit distance
4-11	location strings for edge features	string edit distance
12		sum of distances for 4-11
13	symbol string for endpoint features	string edit distance
14-18	location strings for edge features	string edit distance
19		sum of distances for 14-18
20-22	symbol strings for letter shape features	string edit distance
23		sum of distances for 20-22
24-39	location strings for letter shape features	string edit distance
40		sum of distances for 24-39

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 2 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.

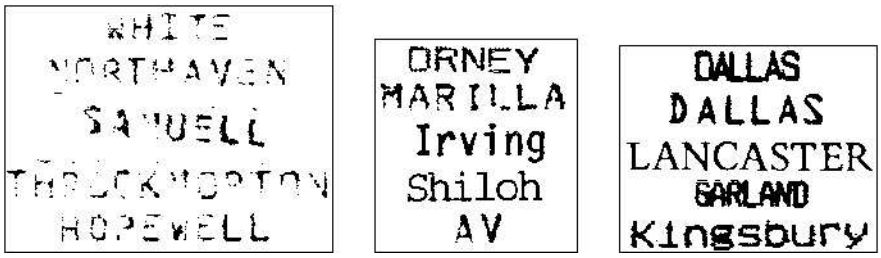


Figure 2: Example images with low, medium and high density levels.

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. Feature prototypes for characters were obtained by running the feature extractors on isolated characters from 179 font samples. Feature prototypes for the words in the lexicon were obtained by synthesizing feature descriptors computed for the characters.

A set of 1675 images, different from those we used in developing the feature extractors, were used to test the classifier combination algorithm. Input lexicons contained postal words including city names and street names. A different lexicon was associated with each input image, depending on other information in the address block from which the input image was extracted. The average size of the input lexicon was 235 words.

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 1634 images (97.55% 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 68 (words). This set was divided into two halves randomly as a training set and a testing set, each containing 817 images. The 40 classifiers were applied to both sets.

To derive the neighborhood thresholds, we constructed the ranking table for the training set. Table 5 gives the performance of the 40 classifiers on these 817 images, the maximum rank needed to get all the true words if each classifier was used in isolation, as well as the computed neighborhood thresholds. As observed from the computed thresholds, only classifiers 1, 2, 3, 4, 6, 8, 9, 10, 11, 12, 13, 16, 18, 19, 21, 22, 26, 37, and 40 were contributing, and the others were redundant. The maximum union size was determined to be 53, much smaller than the maximum rank required for each classifier. The average union size was determined to be 28 after the unions were actually computed, which was 53% of the average size of the filtered lexicons. That is, on the average, 47% of the candidate classes could be eliminated from further consideration after the union was computed. If an oracle selecting the best classifier for each image was available, the top choice correct rate would be 96.0%,

and the correct rates at top 2,3,4,5,10 choices would be 98.2%, 99.1%, 99.4%, 99.8%, and 100% respectively.

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

Classifier	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
% corr. top 1	89	90	67	9	5	21	12	14	21	18	19	42	39	13	5	6	21	24	35	30
% corr. top 2	92	93	77	14	7	30	17	21	27	35	27	54	51	22	8	12	29	34	45	38
% corr. top 3	92	95	82	17	9	36	22	25	34	41	34	58	57	27	11	16	36	42	51	44
% corr. top 4	93	96	84	20	11	40	24	28	37	45	39	63	62	30	12	23	42	46	55	47
% corr. top 5	94	97	85	23	13	43	27	31	42	47	45	67	66	33	14	35	44	49	58	50
% corr. top 10	96	98	90	44	41	55	39	47	51	57	60	77	78	55	42	47	56	59	69	60
thresholds	3	8	10	1	0	5	0	2	1	2	3	1	2	0	0	1	0	2	1	0
Classifier	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
% corr. top 1	12	11	28	22	4	26	6	5	6	3	5	6	5	7	8	6	8	5	8	27
% corr. top 2	18	19	38	29	7	33	14	7	9	6	8	12	8	10	13	9	12	15	12	36
% corr. top 3	22	26	45	33	10	37	17	9	11	9	12	15	10	12	17	12	16	18	17	43
% corr. top 4	26	32	50	37	12	40	22	11	16	11	14	18	13	14	18	14	26	22	19	46
% corr. top 5	30	36	52	40	14	43	25	13	34	13	18	20	15	17	21	17	29	27	22	51
% corr. top 10	46	53	65	56	41	52	47	41	48	40	33	28	38	35	33	34	45	44	41	64
thresholds	1	3	0	0	0	3	0	0	0	0	0	0	0	0	0	0	5	0	0	5

The same feature extractors and *contributing* classifiers were then applied to the set of 817 test images, which were different from all the images used in various training stages. Similar input lexicons were 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. The classes in the union were re-ranked using the Borda count which was computed on the rankings of classifier 1 and 2. Table 6 gives the performance of the 40 classifiers on these 817 images. As can be seen from the performance table, some of the 40 classifiers, when working in isolation, are very poor in (macroscopic) performance over the whole test set. However, the union combination was better than any of the individual classifiers. After the union computation and reranking, the correct rate at top choice was 94%. The correct rates at the top 2,3,4,5 and 10 choices were 96%, 97%, 97%, 98%, and 99% respectively. Degraded by the errors in the global feature computation stage, this was equivalent to 96.6% correct rate at top 10 choices for the input test set. The average union size was determined to be 28. The union included the true class in 99.3% of all the test images.

Table 6: Summary of Performance on Testing Set

classifier	1	2	3	4	6	8	9	10	11	12	13	16	18	19	21	22	26	37	40	*union*
% corr. top 1	92	92	67	9	23	16	21	17	22	43	39	7	26	34	12	10	32	7	29	94
% corr. top 2	94	96	76	14	32	23	28	33	31	55	53	12	33	45	19	18	36	10	38	96
% corr. top 3	95	96	80	17	38	28	35	40	36	61	60	15	39	51	23	24	43	14	44	97
% corr. top 4	96	97	83	19	42	31	38	44	40	66	64	22	42	55	27	28	46	24	51	97
% corr. top 5	96	97	84	21	45	33	43	47	44	69	68	37	45	58	31	31	47	28	54	98
% corr. top 10	97	98	90	46	57	48	53	57	58	78	81	47	56	73	48	47	57	42	68	99

This experiment showed that the combination performance was better than those of the individual classifiers. 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 40 classifiers performs better than what has been achieved.

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. The combination algorithm first determines a candidate set of classes and then derives a consensus ranking on the candidate set. A method was shown that 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 40 classifiers and involved an average of 235 classes. The data were binary 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 96.6% correct rate within the top 10 choices.

The use of Borda count to combine rankings is one among many other possible methods. It is believed that some descriptive statistics, such as measurement of correlation among the classifiers, should be taken into account in ranking combination. Follow up work will include studies of the possible applications of other group consensus functions to classifier ranking combination, consideration of correlation measures for the classifiers, as well as the functional dependence of the combined performance on the individual classifier performances. Our goal will be to identify some necessary conditions on the individual classifier performance such that the combined performance will be *always* better.

The multiple classifiers can be organized at multiple stages such that some subsets of the classifiers can be combined first, and the results are then combined at the next stage. Use of both the intersection approach and the union approach in the same system is also possible.

Future work will also include investigation of an oracle that selects the best feature sets dynamically. This is possible if some of the feature subsets are selectively ignored, based on top-down constraints such as the discriminative power of a particular feature with respect to the input class set, or bottom-up information such as a measurement of the degradation in the image and the confidence of feature detection.

Acknowledgements The support of the Office of Advanced Technology of the United States Postal Service is gratefully acknowledged. The authors appreciate the valuable suggestions by the reviewers of an earlier version of this paper. The discussions with Dr. David Sher of SUNY at Buffalo were especially helpful. Peter Cullen, Michal Prussak and Piotr Prussak assisted in the development of the database for the experiments.

References

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

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

[3] H.S. Baird, H.P. Graf, L.D. Jackel, W.E. Hubbard, A VLSI Architecture For Binary Image Classification, in *From Pixels to Features* , J.C. Simon (editor), North Holland, 1989, 275-286.

[4] D. Black, *The Theory of Committees and Elections*, Cambridge University Press, London, 1958, reprinted 1963.

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

[6] C.H. Coombs, *A Theory of Data*, John Wiley & Sons, 1964.

[7] R.O. Duda, P.E. Hart, *Pattern Classification and Scene Analysis*, Addison-Wesley, New York, 1973.

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

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

[10] R.M. Haralick, The Table Look-Up Rule, *Communications in Statistics - Theory and Methods*, **A5**, 12, 1976, 1163-1191.

[11] T.K. Ho, J.J. Hull, S.N. Srihari, A Word Shape Analysis Approach to Recognition of Degraded Word Images, *Proceedings of USPS Advanced Technology Conference*, Washington, D.C., November 1990, 217-231.

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

[13] J.J. Hull, A. Commike, T.K. Ho, Multiple Algorithms for Handwritten Character Recognition, *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, 117-124, Montreal, April 2-3, 1990.

[14] J.J. Hull, S.N. Srihari, E. Cohen, C.L. Kuan, P. Cullen and P. Palumbo, A blackboard-based approach to handwritten ZIP Code recognition, *Proceedings of the Third United States Postal Service Advanced Technology Conference*, Washington, D.C., May 3-5, 1988, 1018-1032.

[15] D.H. Krantz, R.D. Luce, P. Suppes, A. Tversky, *Foundations of Measurement, Volume I, Additive and Polynomial Representations*, Academic Press, 1971.

[16] E. Mandler, J. Schuermann, Combining the Classification Results of Independent Classifiers Based on the Dempster/Shafer Theory of Evidence, in *Pattern Recognition and Artificial Intelligence*, E.S. Gelsema and L.N. Kanal (editors), North Holland, 1988, 381-393.

[17] V.D. Mazurov, A.I. Krivonogov, V.L. Kazantsev, Solving of Optimization and Identification Problems by the Committee Methods, *Pattern Recognition*, **20**, 4, 1987, 371-378.

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

[19] C. Nadal, R. Legault, C.Y. Suen, Complementary Algorithms for the Recognition of Totally Unconstrained Handwritten Numerals, *Proceedings of 10th International Conference on Pattern Recognition*, Atlantic City, New Jersey, Vol.I, 1990, 443-449.

[20] F.S. Roberts, *Discrete Mathematical Models, with Applications to Social, Biological, and Environmental Problems*, Prentice-Hall, Inc., 1976.

[21] A.K. Romney, R.H. Shepard, and S.B. Nerlove, *Multidimensional Scaling, Theory and Applications in the Behavioral Sciences*, Vol. I and II, Seminar Press, 1972.

[22] S.S. Stevens, Measurement, Statistics, and the Schemapiric View, *Science*, **161**, 3844, 1968, 849-856.

[23] 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*, Montreal, April 2-3, 1990, 131-140.

[24] P. Suppes, D.M. Krantz, R.D. Luce, and A. Tversky, *Foundations of Measurement, Volume II, Geometrical, Threshold, and Probabilistic Representations*, Academic Press, 1989.

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